



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

RUBENS BATISTA ARAÚJO DE SOUSA

TDD:

**UM ESTUDO DE CASO SOBRE SUA UTILIZAÇÃO NOS
PROJETOS DE DESENVOLVIMENTO DE SOFTWARE**

CAMPINA GRANDE - PB

2021

RUBENS BATISTA ARAÚJO DE SOUSA

TDD:

**UM ESTUDO DE CASO SOBRE SUA UTILIZAÇÃO NOS
PROJETOS DE DESENVOLVIMENTO DE SOFTWARE**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientadora: Professora Dra. Melina Mongiovi Cunha Lima Sabino

CAMPINA GRANDE - PB

2021



S725t Sousa, Rubens Batista Araújo de.
TDD: um estudo de caso sobre sua utilização nos projetos de desenvolvimento de software. / Rubens Batista Araújo de Sousa. - 2021.

10 f.

Orientadora: Profa. Dra. Melina Mongiovi Cunha Lima Sabino.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Test Driven Development - TDD. 2. Desenvolvimento de software. 3. Projetos de desenvolvimento de software. 4. Survey. 4. Enquete. 5. Estudo de caso - desenvolvedores de software. I. Sabino, Melina Mongiovi Cunha Lima. II. Título.

CDU:004.415.2(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

RUBENS BATISTA ARAÚJO DE SOUSA

TDD:

**UM ESTUDO DE CASO SOBRE SUA UTILIZAÇÃO NOS
PROJETOS DE DESENVOLVIMENTO DE SOFTWARE**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

Professora Dr.(a.) Melina Mongiovi Cunha Lima Sabino

Orientador(a) – UASC/CEEI/UFCG

Professor Dr. Jorge Cesar Abrantes de Figueiredo

Examinador – UASC/CEEI/UFCG

Professor Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 24 de Maio de 2021.

CAMPINA GRANDE - PB

ABSTRACT

Test Driven Development (TDD) is a practice considered by many to be of great importance in software development, making business rules clear even before starting the development and helping to build clearer, high quality code. However, even with all its benefits, many developers prefer not to use this methodology in their projects. In this study I will analyze the current scenario of the use of TDD in software projects and what are the main reasons that lead developers to not use the practice in their projects. The study was carried out through a questionnaire with 101 programmers from 43 companies, who use or have already used TDD in practice or who have at least theoretical knowledge on the subject. The results showed a low acceptance of TDD by the participants and it was observed that the increase in development time is the main motivator that leads developers to give up the practice.

TDD: um estudo de caso sobre sua utilização nos projetos de desenvolvimento de software

Rubens Batista Araújo de Sousa
Universidade Federal de Campina Grande
Campina Grande, Brasil
rubens.sousa@ccc.ufcg.edu.br

Melina Mongiovi
Universidade Federal de Campina Grande
Campina Grande, Brasil
melina@computacao.ufcg.edu.br

RESUMO

O Test Driven Development (TDD) é uma prática considerada por muitos de grande importância no desenvolvimento de um software, deixando claro as regras de negócio antes mesmo de se iniciar o desenvolvimento e ajudando na construção de um código limpo e de melhor qualidade. Porém, mesmo com todos os seus benefícios, muitos desenvolvedores preferem não utilizar esta metodologia nos seus projetos. Neste estudo irei analisar o atual cenário da utilização do TDD nos projetos de software e quais são os principais motivos que levam os desenvolvedores a não utilizarem a prática nos seus projetos. O estudo foi realizado por meio de um questionário com 101 programadores de 43 empresas, que utilizam ou já utilizaram o TDD na prática ou que possuem pelo menos conhecimento teórico sobre o assunto. Os resultados mostraram uma baixa aceitação do TDD por parte dos participantes e foi observado que o acréscimo de tempo no desenvolvimento é o principal motivador que leva os desenvolvedores a desistirem de empregar a prática.

PALAVRAS-CHAVE

TDD, test-driven development, survey.

1. INTRODUÇÃO

Todos sabemos que testes são de extrema importância no desenvolvimento de um software, com o objetivo de prover a detecção de possíveis bugs e aumentar a confiança que o programa irá se comportar da forma que foi planejado. Visto isso, precisamos de uma garantia de que nosso código vai ser testado, sabendo que em processos ágeis geralmente temos prazos curtos, isso poderia ser um pretexto para que não sejam feitos testes que cubram a maior parte código. Para suprir este problema, existe um método que permite desenvolver seu projeto pensando primeiro nos testes: TDD (*Test-Driven Development*). Utilizando de pequenos passos para a construção do projeto, no TDD os testes são escritos antes da codificação propriamente dita, fazendo com que se tenha uma cobertura de testes em quase todo o projeto. O TDD visa melhorar a qualidade do código e reduzir a quantidade de bugs, visto que a refatoração do código é um dos passos envolvidos no processo e também pelo fato da garantia que todas as unidades do código estão sendo testadas. Porém, mesmo o método apresentando vários benefícios [5,6], boa parte dos desenvolvedores preferem não utilizá-lo.

Uma survey realizada em 2010 e intitulada “How Agile Are You? 2010 Survey Results” revelou que 53% dos 293 entrevistados utilizam o TDD para validar os seus trabalhos [1], porém, como está a situação atual do TDD? As pessoas ainda utilizam a prática nos seus projetos? Pretendo trazer neste estudo, por meio de uma survey, uma estimativa da utilização do Test-driven-development nos projetos de desenvolvimento de

software, e tentar entender quais os principais motivadores que levam os desenvolvedores a não utilizarem a prática.

O presente documento está disposto da seguinte maneira. A seção 2 apresenta estudos com objetivos semelhantes a este. A seção 3 descreve de forma resumida sobre o TDD e os passos que são realizados na prática. Mais adiante, a seção 4 discorre sobre a estrutura da survey, seu objetivo, como as respostas foram analisadas e o período que o questionário ficou disponível. A seção 5 expõe as respostas colhidas na survey. Logo após, a seção 6 faz um resumo do artigo e apresenta trabalhos futuros. Por fim, são feitos os agradecimentos e indicadas as referências.¹

2. TRABALHOS RELACIONADOS

Um artigo intitulado “Why developers don’t use TDD” elencou, por meio de entrevistas técnicas, as diferentes razões que os desenvolvedores alegam para evitar o TDD e utilizou argumentos para tentar refutar se as razões apontadas são reais. As alegações apontadas pelos entrevistados foram [9]:

- A gestão não nos permite;
- Não há tempo suficiente para escrever testes;
- Minha equipe discorda se devemos usar TDD;
- O retorno do investimento em TDD não é comprovado;
- Tentamos, mas não funcionou;
- Processo de construção lenta;
- Não sei todos os requisitos antecipadamente para escrever todos os testes.

Este artigo se mostrou interessante por apresentar argumentos que possam servir como justificativa para invalidar as alegações feitas pelos entrevistados. No entanto, o trabalho não deixou claro quais foram as perguntas feitas aos entrevistados, quantas pessoas participaram da pesquisa ou porcentagens das respostas. O trabalho vigente utilizou algumas das razões descritas anteriormente para a elaboração de parte das questões da survey, que vai ser detalhada posteriormente.

Salo, O. e Abrahamsson, P. [3] propuseram em seu estudo identificar a adoção e experiência dos métodos ágeis Extreme Programming e Scrum em uma série de organizações europeias de software integrado. Para isso foi enviada uma survey para um total de 18 organizações de software industrial

¹ Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

embarcado do projeto Agile-ITEA². Nos resultados, o TDD apareceu como uma das práticas do XP que mais teve respostas nas categorias “Raramente” e “Nunca Aplicado” (56%). Apesar do estudo ter chegado a estes resultados acerca da prática, o seu objetivo era avaliar a utilização do XP (todas as suas práticas) e Scrum, e não unicamente o TDD.

Scanniello *et al.* [11] realizou uma investigação qualitativa para identificar as opiniões dos desenvolvedores que usam TDD em relação ao processo em si. O estudo foi realizado com 13 alunos de mestrado e cinco desenvolvedores profissionais de software. No trabalho, os grupos focais discutiram a sua experiência utilizando o TDD. Dentre os resultados, os participantes alegaram que aplicar o TDD sem o conhecimento de técnicas avançadas de teste de unidade pode ser complicado e que a refatoração não é utilizada na frequência que deveria.

Os trabalhos indicados acima apresentam a opinião de desenvolvedores acerca do TDD ou a utilização da prática pelos mesmos. Os resultados que vão ser discutidos neste artigo irão servir para somar a estes dados, apresentando novas apurações e ajudando a validar os dados que venham a ser semelhantes com os obtidos nos artigos citados.

3. TEST-DRIVEN DEVELOPMENT

O Test-Driven Development (TDD) é uma prática de desenvolvimento onde, seguindo uma série de passos, desenvolvemos o código baseado nos testes, que são escritos antes do código de produção. A técnica utiliza a abordagem “Test-first”, e foi idealizada por Kent Beck. Segundo seu criador, o TDD pode ser resumido com os seguintes passos [2]:

- Adiciona-se um teste rapidamente;
- Executa todos os testes e ver o mais novo falhando;
- Faz uma pequena mudança no código;
- Executa todos os testes e ver todos funcionando;
- Refatorar para remover possíveis *bad smells*.

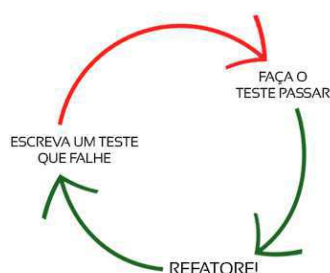


Figura 1: Ciclo do TDD [10].

A Figura 1 resume os passos do TDD. A ideia aqui é utilizar pequenos passos para chegar em uma solução final. Primeiro escrevemos um teste que nem sequer compila. Agora rodamos a bateria de testes e vemos este primeiro falhar. O próximo passo é escrever o código mínimo necessário para fazer o teste passar. Neste momento rodamos todos os testes novamente, agora finalizando com sucesso. Após estes passos refatora-se o código para melhorar sua qualidade, de forma que os testes continuem com sucesso.

4. METODOLOGIA

Para atender o objetivo desta pesquisa, foi realizada uma survey com o intuito de coletar opiniões dos participantes acerca do TDD, podendo estes participantes terem ou não utilizado a técnica na prática. A survey foi criada utilizando o Google Forms e enviada em vários canais de comunicação. Ela teve como público alvo desenvolvedores que estão envolvidos com projetos de software reais, ou seja, programadores que estão ligados a alguma empresa ou que desenvolvem software para algum cliente.

O questionário ficou disponível por 2 semanas, entre os períodos de 06/04/2021 a 19/04/2021. A survey foi pensada de forma que fosse possível entender as motivações que levam os programadores a fazer ou não o emprego do TDD nos seus projetos, visto isso, o questionário segue dois caminhos, um deles caso o entrevistado responda que utiliza ou já utilizou o TDD na prática, e outro para aqueles que tenham apenas conhecimento teórico sobre o assunto.

A survey contém 16 perguntas, das quais 3 delas ficam disponíveis apenas para quem respondeu que conhece o TDD apenas na teoria e 4 para os que responderam que utilizam ou já utilizaram a metodologia na prática. O questionário obteve 101 respostas, que posteriormente foram exportadas para um CSV, assim sendo possível realizar uma análise destes dados utilizando a linguagem R.

5. RESULTADOS OBTIDOS

Esta seção juntamente com suas subseções tem o intuito de apresentar e discutir sobre os resultados obtidos na survey. As subseções foram divididas da seguinte forma: Primeiramente é mostrado o perfil dos entrevistados (5.1), em seguida, a seção 5.2 dá uma visão geral da utilização do TDD pelos respondentes. As subseções 5.3 e 5.4 discutem os resultados obtidos fazendo a divisão entre experiência teórica ou prática pelos entrevistados. A subseção 5.5 discorre sobre os motivos que levaram os participantes a desistirem do TDD, seguindo os mesmos.

5.1. Perfil dos entrevistados

A ideia deste trabalho é avaliar a atual utilização do TDD nos projetos de software, ou seja, qualquer desenvolvedor que estivesse participando de um projeto, seja para alguma empresa ou até mesmo de forma autônoma estaria apto a responder a pesquisa. Visto isso, foram obtidas respostas de profissionais de 43 empresas distintas e também de desenvolvedores autônomos. A faixa etária ficou entre 18 a 45 anos e a maioria dos programadores trabalha com desenvolvimento de software há 1 ano ou mais. Em relação ao nível acadêmico, as respostas mais comuns foram entre os desenvolvedores que estão na graduação ou possuem a graduação completa como nível acadêmico atual. As Figuras 2 e 3 apresentam, respectivamente, a relação do nível acadêmico dos participantes e sua experiência com desenvolvimento de software.

² Foi uma iniciativa europeia de pesquisa sobre métodos ágeis.

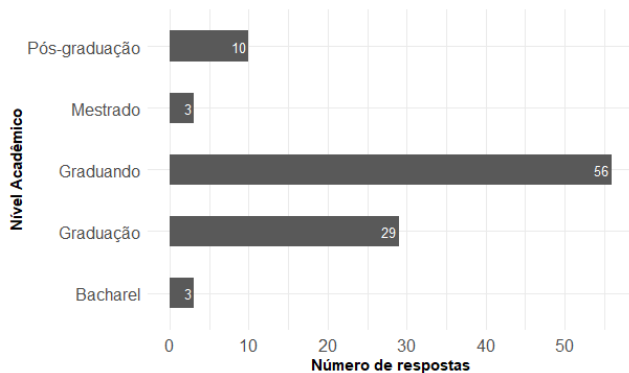


Figura 2: Nível Acadêmico dos Participantes.

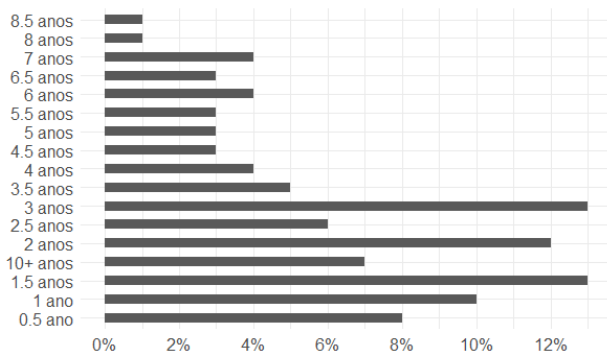


Figura 3: Experiência dos Participantes com Desenvolvimento de Software.

5.2. Utilização do TDD pelos respondentes

Apenas 30% dos respondentes afirmaram estar utilizando o TDD atualmente, contudo, 38% responderam que já utilizaram a prática em algum momento. Aqueles que conhecem a técnica somente na teoria foram 26% das respostas e os que não conhecem a metodologia apenas 6%. Se utilizarmos como base o conjunto dos entrevistados que usam ou já utilizaram o TDD em algum momento, observamos uma boa taxa de utilização da prática, mas esse número é bem baixo se olharmos apenas para aqueles que responderam utilizar atualmente. Uma pesquisa realizada entre 2004 e 2006, que contou com 35 projetos de 13 empresas de software de tamanhos variados localizadas em 8 países da Europa teve o intuito de avaliar o nível de adoção de duas metodologias ágeis: Extreme Programming and Scrum. Entre as práticas que são adotadas no XP, a que mais recebeu respostas “Nunca aplicado” foi o TDD, com 41% [3].

Falando de nível acadêmico, o maior índice de utilização do TDD partiu dos mestrandos, com 100% das respostas indicando estarem utilizando atualmente, seguido dos respondentes com nível de pós-graduação, com 50%. Em terceiro ficaram os respondentes Bacharéis (33%), seguido dos que têm a graduação completa (31%), e por último os alunos de graduação, com 21%. No geral, um fator que pode explicar a baixa aceitação do TDD observada neste estudo pode estar relacionada com o nível acadêmico dos participantes, que foi em maioria alunos de graduação. Resultados de uma survey realizada em janeiro de 2010 com 218 programadores voluntários mostraram que apenas 20% dos participantes utilizam o TDD no meio acadêmico, enquanto 90% utilizam na indústria [8].

5.3. Conhecimento teórico

Nesta seção foram feitas perguntas relacionadas ao conhecimento teórico por parte daqueles que indicaram nunca terem utilizado o TDD na prática, a maior parte dos respondentes afirmaram que pretendem utilizar a técnica futuramente. A afirmativa mais comum quando se foi perguntado “Por quais motivos você não chegou a utilizar o TDD na prática” aponta que os programadores avaliam que seus projetos iriam levar mais tempo caso optassem por utilizar a prática. Segue abaixo a sumarização das respostas obtidas para esta pergunta, que obteve 27 respostas, considerando que cada respondente tinha a opção de marcar mais de uma alternativa.

Tabela 1: Respostas referentes a pergunta “Por quais motivos você não chegou a utilizar o TDD na prática”

Resposta	Porcentagem
Com TDD meu projeto iria levar mais tempo para ser terminado	37%
Eu prefiro criar a funcionalidade antes dos testes	33%
A gerência do meu projeto não permite que utilizemos o TDD	29%
Outros	15%
O TDD não me pareceu uma boa prática	Não obteve respostas

Aqui notou-se uma certa aversão ao TDD pelos desenvolvedores motivado pelo provável aumento no tempo de desenvolvimento do projeto. Pelo fato de utilizar pequenos passos, que podem até parecer chatos em algum momento, é normal que os programadores que nunca tenham aplicado o TDD na teoria esperem um aumento de tempo de desenvolvimento, e veremos até o final deste artigo que existem estudos que realmente mostraram o aumento de tempo no processo, porém, também há um acréscimo na qualidade do produto [5,6]. A segunda resposta mais destacada diz respeito à preferência do programador, e faz total sentido os desenvolvedores que optam por esta opção não terem praticado ainda o TDD, visto que a técnica utiliza a abordagem “test-first”, ou seja, criamos os testes antes de funcionalidade.

Em relação a permissão ou não da gerência, deve-se ser feita a seguinte pergunta: Seria interessante que o time de desenvolvimento também participasse dessa decisão? O programador deveria saber o que é melhor para a qualidade do código escrito, e também, participar dessas decisões. É provável também que a estimativa do projeto não tenha deixado uma abertura para aplicação do TDD, no entanto, caso realmente deseje-se utilizar a prática no seu projeto o planejamento já deve ser feito contando com este passo. Na opção “Outros”, os desenvolvedores alegaram que nos seus locais de trabalho não era utilizado a prática ou seu conhecimento e também o conhecimento da equipe sobre o TDD era pouco.

A segunda pergunta, que era aberta, dependia da opção “O TDD não me pareceu uma boa prática”, disponível na primeira questão, porém não obteve respostas. A terceira questão visava avaliar se os entrevistados pensam em utilizar o TDD futuramente, 70% responderam que pretendem utilizar a prática no futuro, enquanto 30% não pretendem usar.

5.4. Conhecimento prático

Esta seção obteve respostas dos desenvolvedores que indicaram utilizar o TDD atualmente ou já utilizaram em algum momento. As perguntas visavam avaliar o nível de satisfação dos entrevistados com o TDD e levantar os pontos positivos e negativos que os desenvolvedores apontaram na sua experiência com a prática.

A primeira pergunta, que consistia em uma escala likert, mostrou que 28% dos programadores tiveram uma experiência muito boa com o TDD, 41% tiveram uma experiência boa, 30% regular e 1% ruim. Não foram obtidas respostas que indicassem uma experiência muito ruim com a técnica. Abaixo, seguem as tabelas com a sumarização das 68 respostas obtidas referente às perguntas para os pontos positivos e negativos na relação dos programadores com o TDD. Vale salientar que os entrevistados podiam marcar mais de uma opção.

Tabela 2: Respostas referentes à pergunta “Quais foram os principais pontos positivos na sua relação com o TDD?”

Resposta	Porcentagem
Utilizando TDD o código se torna mais confiável	76%
TDD melhora o processo de testes	75%
O TDD melhora o entendimento dos requisitos	73%
É mais fácil localizar e corrigir bugs utilizando TDD	72%
O processo de refatoração se torna mais fácil	68%
TDD favorece a documentação	52%
O código implementado com TDD é mais legível	38%
Outros	3%

Tabela 3: Respostas referentes à pergunta “Quais foram os principais pontos negativos na sua relação com o TDD?”

Resposta	Porcentagem
TDD tornou o processo de desenvolvimento do software mais custoso	54%
TDD não garante a criação de bons testes	36%
Tenho dificuldade de criar testes antes de implementar a funcionalidade	34%
Não gosto do fato de escrever pequenas porções do código que se adaptam ao meu teste	23%

Nenhum	6%
Outros	4%

O intuito deste estudo não é provar que o TDD é uma boa prática e nem apontar seus pontos positivos ou negativos, e sim apresentar a percepção de uma parcela de desenvolvedores em relação à prática, explanando suas opiniões e as analisando, por meio da literatura e experimentos práticos encontrados no meio acadêmico. Porém, é fato que TDD melhora a qualidade do código [4]. Como o TDD garante uma maior cobertura de testes, visto que todo código escrito parte sempre de um teste, é esperado que a aplicação se torne mais confiável, e futuros bugs sejam reduzidos. Também é esperado que o TDD melhore o entendimento de requisitos, visto que os testes irão servir como uma especificação da funcionalidade [7].

Assim como na maioria das perguntas feitas na survey, o tempo também aparece aqui como o principal ponto negativo do TDD nas respostas. Estudos mostram que o TDD pode aumentar o tempo de desenvolvimento, no entanto existe um trade-off em relação a este fato, pois também há um aumento na qualidade do código escrito [5,6].

5.5. Principais motivadores que levam os desenvolvedores a optarem por não utilizar o TDD

A última pergunta do questionário foi direcionada para os desenvolvedores que pararam de utilizar o TDD por algum motivo. A ideia desta pergunta era obter as principais motivações que levaram esses desenvolvedores a deixarem a prática de lado. Era uma questão não obrigatória, e nela foram disponibilizadas motivadores pré-definidos mais comuns quando se é questionado sobre a não adoção do TDD. Os respondentes tinham a opção de escolher mais de uma alternativa e também foi deixado um campo aberto para os mesmos explanarem algum outro motivador que possa surgir, abaixo segue uma tabela que sumariza a porcentagem das 47 respostas obtidas para esta questão:

Tabela 4: Respostas referentes à pergunta “Quais foram os motivos que fizeram com que você parasse de utilizar o TDD?”

Motivos	Porcentagem
Nos projetos em que atuo não possuímos tempo suficiente para implementar o TDD	66%
O TDD tornou o desenvolvimento mais lento	38%
Tentamos utilizar, porém não funcionou	12%
A gerência não permite que utilizemos o TDD	10%
Outros	10%
A minha equipe discorda em utilizar o TDD	8%
Não há provas de que o TDD tenha um bom retorno	2%

Os dois motivadores mais pontuados tem uma relação, ambos falam sobre tempo. O fato deste motivo (tempo) aparecer com

uma alta porcentagem de respostas pode decorrer dos prazos apertados que temos hoje em dia nos projetos ágeis, isso junto do fato de que a maioria dos desenvolvedores focam pouco tempo e esforço na criação de testes, criando a impressão que o tempo que é gasto criando testes poderia estar sendo aplicado na implementação em si. Contudo, esse provável aumento no tempo de desenvolvimento pode também trazer vantagens.

Um estudo realizado em unidades diferentes da Microsoft mostrou um aumento na qualidade do desenvolvimento em relação às equipes que não adotaram a abordagem, porém, também houve um acréscimo no tempo no início do desenvolvimento [5]. Um experimento publicado em 2004 colocou programadores para desenvolverem um pequeno projeto em Java utilizando TDD enquanto outro grupo utilizava uma abordagem cascata. Os resultados mostraram que os programadores do projeto em Java desenvolveram um código com maior qualidade pois eles passaram em 18% mais testes funcionais caixa preta, no entanto, levaram 16% de tempo a mais para a conclusão [6]. Isso indica que o motivo apontado realmente acontece, porém deve-se avaliar se, em seu projeto, vale a pena (ou se é possível) adicionar um pouco mais de tempo para que se ganhe em qualidade, incluindo a redução na quantidade de futuros bugs.

A terceira resposta mais marcada fala que ao aplicar o TDD, o mesmo não funcionou. Vários fatores podem ter influenciado para isso acontecer, como por exemplo, o nível de conhecimento em TDD por parte dos desenvolvedores, ou, a correta aplicação da técnica. Também é possível que o escopo do projeto seja apertado ao ponto de não deixar tempo suficiente para a aplicação do método. É importante que o TDD seja aplicado desde o início do projeto, pois o mesmo vai servir como uma interface e especificação do comportamento para a funcionalidade que está sendo desenvolvida [7], no resumo das contas, tentar aplicar o TDD em projetos legados não parece uma boa ideia.

Sobre a afirmativa “A gerência não permite que utilizemos o TDD”, esse mesmo motivador já foi comentado na seção em que foram debatidas as respostas dos programadores que não utilizaram o TDD na prática, e lá foi questionado se essa decisão deveria realmente vir por parte da gerência. Porém, 8% afirmaram que sua equipe discorda em utilizar o TDD, o que é justamente o oposto do motivador anterior, pois agora a decisão está vindo da equipe como um todo. O ideal é que o TDD seja empregado por todos da equipe, pois, a adoção parcial é uma das armadilhas típicas da prática, quando utilizada em equipe [4].

A motivação pré-definida menos marcada fala que não há provas de um bom retorno vindo do TDD. Nesse estudo já foram mostradas pesquisas que comprovaram um aumento na qualidade do código e redução de bugs ao implementar o TDD [5,6], porém junto com esse benefícios também foi acrescentado um tempo no desenvolvimento do projeto.

Em “Outros”, as principais respostas foram: “Não há cultura de utilizar TDD, nem existem muitos testes nos projetos” e “Fui abandonando a prática com o tempo”. Em relação a primeira motivação, a quantidade de testes vai naturalmente crescer quando for empregado o TDD, pois o código de produção irá depender destes testes, porém, deve haver um consenso entre a equipe para a utilização da prática, e, caso parte da equipe não tenha experiência com a técnica, podem ser feitos treinamentos partindo dos mais experientes. Sobre o segundo, seria interessante ir mais a fundo para entender o que motivou esse abandono da prática.

6. CONCLUSÃO E TRABALHOS FUTUROS

O presente estudo teve como objetivo entender qual a atual situação do TDD nos projetos de software, no sentido de analisar as opiniões dos desenvolvedores em relação à prática e quais os principais argumentos que levam esses programadores a deixarem de utilizar a técnica. Para isso, foi enviada uma survey para desenvolvedores envolvidos em algum projeto de software. Com o questionário foi possível obter os dados necessários para a análise proposta neste estudo.

A porcentagem de participantes que usa o TDD atualmente se mostrou baixa (30%), porém, de forma geral, uma boa parte dos entrevistados já utilizaram o TDD na prática em algum momento (68%, se incluirmos também os que estão fazendo uso da técnica atualmente). Os que responderam conhecer o TDD somente na teoria também foram poucos (26%), e os que não conhecem a prática foram os de menor número na pesquisa (6%). Os desenvolvedores que estão na graduação foram a maioria em todas as categorias, não sabe-se ao certo se esse fator influenciou para a baixa utilização do TDD pelos participantes no geral, seria interessante fazer um estudo mais aprofundado neste aspecto, para entender possíveis causadores do baixo emprego da técnica no meio acadêmico [8].

Os participantes que têm apenas conhecimento teórico acreditam que seus códigos iriam levar mais tempo para serem desenvolvidos e também preferem criar as funcionalidades antes dos testes. Estes também afirmaram, em maioria (70%), que pretendem utilizar o TDD futuramente. Os que utilizam ou já utilizaram o TDD na prática tiveram uma experiência boa ou muito boa com a técnica (69%), os demais apontaram uma experiência regular ou ruim (31%). Estes também indicaram como pontos positivos a confiabilidade do código, melhora no processo de testes e entendimento de requisitos e a facilidade de se localizar bugs. Como pontos negativos, estes alegaram que o TDD tornou o processo de desenvolvimento mais custoso, também relataram que a prática não garante a criação de bons testes, além de terem dificuldades de criar testes antes de implementar a funcionalidade.

O tempo foi o principal motivador para a desistência da aplicação do TDD pelos desenvolvedores. O aumento de tempo no desenvolvimento se mostrou aparente mesmo em estudos que realizaram experimentos práticos onde se era comparado os resultados de projetos com e sem a utilização do TDD, acompanhado também de acréscimos na qualidade do código [5,6].

Os resultados obtidos nesta pesquisa não podem ser generalizados, visto o tamanho da população para que foi direcionada, por isso, seria interessante replicar o estudo com diferentes conjuntos do público alvo, então, os próximos passos planejados são:

- Replicar o estudo com outras parcelas da população, para agregar os resultados obtidos nesta pesquisa e fazê-los mais precisos.
- Realizar uma pesquisa complementar com programadores que utilizam o TDD na indústria e aqueles que usam a prática no meio acadêmico. Neste trabalho poderiam ser elaboradas questões que levassem ao entendimento da baixa adoção da técnica no meio acadêmico [8].

AGRADECIMENTOS

Agradeço primeiramente a Prof^ª. Dra. Melina Mongiovi por ter sido de grande ajuda como orientadora neste trabalho. A minha

família que me apoiou nos estudos desde sempre. Aos amigos que fiz durante todos esses anos de curso, por terem me ajudado a passar pelas fases mais difíceis e também a todos os professores do curso que ajudaram na minha formação.

REFERÊNCIAS

- [1] W. Ambler, S. How Agile Are You? 2010 Survey Results. Disponível em <http://www.ambysoft.com/surveys/howAgileAreYou2010.html#Downloads>> Acesso em 10 abr 2021.
- [2] Beck, K. Test-Driven Development: By Example. Addison-Wesley Professional, 2003.
- [3] Salo, O. & Abrahamsson, P. Agile methods in European embedded software development organisations: A survey on the actual use and usefulness of Extreme Programming and Scrum. Software, IET, 2, 2008, 58 - 64.
- [4] Buchan, J., Li, L., & MacDonell, S.G. Causal Factors, Benefits and Challenges of Test-Driven Development: Practitioner Perceptions. 18th Asia-Pacific Software Engineering Conference, 2011, 405-413.
- [5] Bhat, T. & Nagappan, N. Evaluating the efficacy of test-driven development: Industrial case studies. ISESE'06 - Proceedings of the 5th ACM-IEEE International Symposium on Empirical Software Engineering, 2006, 356-363.
- [6] George, B. & Williams, L. A structured experiment of test-driven development. Information and Software Technology. 46, 2004, 337-342.
- [7] SOMMERVILLE, I. Engenharia de software. São Paulo:Person, 2010.
- [8] Aniche, M. & Gerosa, M. Aurelio. Most Common Mistakes in Test-Driven Development Practice: Results from an Online Survey with Developers. ICSTW 2010 - 3rd International Conference on Software Testing, Verification, and Validation Workshops, 2010, 469-478.
- [9] Olszewski, D. Why Developers Don't Use TDD. Disponível em <http://dolszewski.com/testing/why-developers-dont-use-tdd/>> Acesso em 14 mai 2021.
- [10] Aniche, M. Test-Driven Development. Disponível em <https://tdd.caelum.com.br/>> Acesso em 14 mai 2021.
- [11] Scanniello, G. & Romano, S. & Fucci, D. & Turhan, B. & Juristo, N. Students' and professionals' perceptions of test-driven development: a focus group study. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16). Association for Computing Machinery, New York, NY, USA, 2016, 1422-1427.