Desenvolvimento Orientado por Comportamento (BDD) como Mecanismo Alternativo entre os Modelos de Desenvolvimento Tradicional e Ágil

Trabalho de Conclusão de Curso

Engenharia da Computação

Nome do Aluno: Eduarda Beatriz Cavalcanti Ottoni
Orientador: Prof. Joabe Bezerra Jesus Júnior

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

EDUARDA BEATRIZ CAVALCANTI OTTONI

DESENVOLVIMENTO ORIENTADO POR COMPORTAMENTO (BDD) COMO MECANISMO ALTERNATIVO ENTRE OS MODELOS DE DESENVOLVIMENTO TRADICIONAL E ÁGIL

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife,
Outubro de 2022.

Ottoni, Eduarda Beatriz Cavalcanti

Desenvolvimento Orientado por Comportamento (BDD) como Mecanismo Alternativo entre os Modelos de Desenvolvimento Tradicional e Ágil/ Eduarda Beatriz Cavalcanti Ottoni. - Recife - PE, 2022.

11, 46 f. : 27 il. ; 29 cm.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) Universidade de Pernambuco, Escola Politécnica de Pernambuco local, ano

Orientador: Prof. Joabe Bezerra de Jesus Júnior.

Inclui referências.

1. Manifesto Ágil. 2. BDD. 3. Desenvolvimento Orientado por Comportamento. 4. Gestão de Projetos. 5. Metodologia Híbrida
I. Desenvolvimento Orientado por Comportamento (BDD) como Mecanismo Alternativo entre os Modelos de Desenvolvimento Tradicional e Ágil - UPE. II. Júnior, Joabe Bezerra de Jesus. III. Universidade de Pernambuco.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 19/10/2022, às 17h30min, reuniu-se para deliberar sobre a defesa da monografia de conclusão de curso do(a) discente **EDUARDA BEATRIZ CAVALCANTI OTTONI**, orientado(a) pelo(a) professor(a) **JOABE BEZERRA DE JESUS JÚNIOR**, sob título BDD como Ferramenta de Migração entre Modelos de Desenvolvimento Tradicional e Ágil, a banca composta pelos professores:

LARISSA TENÓRIO FALCÃO ARRUDA (PRESIDENTE)
JOABE BEZERRA DE JESUS JÚNIOR (ORIENTADOR)

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

☒ Aprovada ☐ Aprovada com Restrições* ☐ Reprovada

e foi-lhe atribuída nota: 9,5 (nove e meio)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O(A) discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.

Larissa T. Falcão Arruda

AVALIADOR 1: Prof (a) **LARISSA TENÓRIO FALCÃO ARRUDA**

Joabe Bezerra de Jesus Júnior

AVALIADOR 2: Prof (a) **JOABE BEZERRA DE JESUS JÚNIOR**

AVALIADOR 3: Prof (a)

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

Acima de tudo, dedico este trabalho a Deus, que me manteve firme nos momentos de dificuldadee indecisão, me permitindo atravessar o ano de 2022.

Agradecimentos

Meus sinceros agradecimentos aos meus pais e à minha irmã, que me movem de forma incontestável e incomparável.

À toda minha família, que sempre foi um alicerce emocional para mim.

Aos meus amigos da faculdadee da época de colégio, que me acompanham e me incentivam de forma genuína ao longo de todas as minhas iniciativas.

Não poderia deixar de mencionar o meu profundo e significativo agradecimento aos colegas de empresa, símbolos de competência e conhecimento, que contribuem diretamente para o meu crescimento. Em especial, menciono o meu querido *Enablement Team*, que foi uma verdadeira encubadora no início da minha carreira, me dando amigos, vivência, oportunidade e um caminho profissional a seguir. A vocês, meu eterno carinho e agradecimento.

Ao professor Joabe, ao qual tenho grande admiração e que me auxiliou com maestria e sensibilidade, sendo fundamental para a conclusão deste trabalho.

Por fim, a todos que eventualmente me proporcionaram momentos inesquecíveis de descontração, aprendizado e companheirismo, fundamentais para a minha formação pessoal e profissional. Muito obrigada!

ESCOLA POLITÉCNICA
DE PERNAMBUCO

Autorização de publicação de PFC

Eu, **Eduarda Beatriz Cavalcanti Ottoni** autor(a) do projeto de final de curso intitulado: **BDD como Ferramenta de Migração entre Modelos de Desenvolvimento Tradicional e Ágil;** autorizo a publicação de seu conteúdo na internet nos portais da Escola Politécnica de Pernambuco e Universidade de Pernambuco.

O conteúdo do projeto de final de curso é de responsabilidade do autor.

Eduarda Beatriz Cavalcanti Ottoni

Orientador(a): **Joabe Bezerra de Jesus Júnior**

Coorientador(a):

Prof, de TCC: **Daniel Augusto Ribeiro Chaves**

Data: 19/10/2022

Resumo

O desenvolvimento de *software* pode ser gerenciado de diversas formas. A partir dos anos 2000, o Manifesto Ágil ganhou força e introduziu conceitos e valores derivados da indústriaautomobilística na indústria de *software*, trazendo mais responsividade e menos rigidez no processo de desenvolvimento. Porém, nem todas as organizações absorveram completamente essa nova tendência, sendo necessário adotar soluções alternativas para manter a competitividade dos seus produtos. Dessa forma, a gestão e o desenvolvimento híbridos de *software* emergiram como uma alternativa às exigências do mercado e às limitações dessas organizações. Essa hibridez se refere à adoção de práticas de duas ou maismetodologias de gestão de projetos – geralmente oriundas das práticas de metodologias tradicionais e ágeis, onde a organização modula um novo modelo para atender às necessidades do projeto. Com isso, este trabalho se propõe a levantar indícios sobre o uso do desenvolvimento orietado por comportamento (BDD) como um mecanismo alternativo e prática independente em contextos de metodologia híbrida ou transição entre modelos de desenvolvimento tradicional e ágil.

Palavras-chave: Manifesto Ágil, BDD, Desenvolvimento Orientado por Comportamento,Gestão de Projetos, Metodologia Híbrida.

Abstract

Software development can be managed in several ways. Starting in the 2000s, the Agile Manifesto gained strength and introduced concepts and values derived from the automobile industry into the software industry, bringing more responsiveness and less rigidity to the development process. However, not all organizations have completely absorbed this new trend, and it's necessary to adopt alternative solutions to maintain the competitiveness of their products. Thus, hybrid software management and development emerged as an alternative to market demands and the limitations of these organizations. This hybridity refers to the adoption of practices from two or more project management methodologies usually originating from the practices of traditional and agile methodologies, where the organization modulates a new model to meet the needs of the project. So, this work proposes to raise evidence about the use of behavior-driven development (BDD) as an alternative mechanism and independent practice in contexts of hybrid methodology or transition between classic and agile development models.

***Keywords**: Agile Manifesto, BDD, Behavior-Driven Development, Project Management, Hybrid Methodology.*

Lista de ilustrações

Lista de abreviaturas e siglas

BDD	*Behavior-Driven Development*
TDD	*Test-Driven Development*
XP	*Extreme Programming*
SM	*Scrum Master*
P.O.	*Product Owner*
QA	*Quality Assurance*
HTML	*HyperText Markup Language*

Sumário

Capítulo 1

Introdução

O desenvolvimento de *software* pode ser entendido como um trabalho processual que envolve tecnologia, requisitos técnicos e estratégicos com o objetivo de resolver algum problema de negócio. O *software* é tido como um bloco de instruções sequenciais, cuja execução está orientada a manipular ou modificar um dado ou um evento, por exemplo. Durantea história, a concepção de desenvolvimento de *software* surgiu bem antes de se entender e formalizá-lo propriamente na versão atual da sua definição.

É importante destacar que boa parte das metodologias que inspiraram a gestão e desenvolvimento de *software* vieram da indústria automobilística. Os anos de 1900 destacaramo avanço da indústria automobilística e os avanços na sua produção. Dentro desse universo, a Ford Motor Company esteve à frente de grande parte desses avanços. Henry Ford investiu massivamente em padronização de processos de trabalho, refletindo na eliminação de desperdícios, na produtividade e na escalabilidade do seu negócio. Isso fez a empresa se tornarespecialista e líder de mercado com o produto que vendia.

Figura 1: ciclo de desenvolvimento do Fordismo

Fonte: Redes de Saúde. Disponível em: https://www.redesdesaude.com.br/como-surgiram-e-para-que-servem-as-metodologias-ageis/. Acesso em: 23 setembro 2020

Anos depois, a empresa perdeu força no mercado graças a uma nova tendência entre os consumidores e que foi ignorada por Henry Ford: a busca por variedade e produtos diversificados. É nesse momento que surge outra grande inspiração para os futuros modelos de gestão e desenvolvimento de *software*: a Toyota. A empresa estava atenta ao que vinha acontecendo no mercado nos últimos anos, sobretudo aos erros e acertos da concorrência.

Após a Segunda Guerra Mundial, os líderes da Toyota utilizaram as lições aprendidas da concorrência para levantar uma série de melhorias que fariam a empresa driblar a crise dos últimos anos. Nesse momento, pode-se dizer que surgiu o Sistema Toyota de Produção. Esse sistema pivotou o fluxo de produção da época, saindo de processos rígidos e sequenciais para processos mais flexíveis e enxutos, baseado nos conceitos "jidoka"[1] e “Just-in-Time” [1]. Esse novo fluxo de produção fundamentou metodologias como o Lean [2] e o Kanban [3], largamente utilizadas na indústria de *software*.

Figura 2. ciclo de desenvolvimento do Toyotismo

Fonte: Redes de Saúde. Disponível em: https://www.redesdesaude.com.br/como-surgiram-e-para-que-servem-as-metodologias-ageis/. Acesso em: 23 setembro 2020

Esses dois métodos se relacionam bastante com os Processos Tradicionais de Desenvolvimento de *software*. Estes processos pavimentaram a evolução das metodologias de gerenciamento e desenvolvimento dentro da Engenharia de *software*, permitindo que esse segmento passasse a ser um ramo forte e promissor dentro da indústria.

Ao final da década de 1990 e início dos anos 2000 surge um marco temporal equivalente ao Fordismo e ao Toyotismo em suas épocas: o Manifesto Ágil, que foi resultado de uma série de insatisfações tidas por um grupo de desenvolvedores da

época. O Manifesto foi de encontro às práticas da época e propôs uma série de práticas e valores voltados aos processos de desenvolvimento de *software*. Daí em diante, várias camadas desta indústria foram exploradas e especializadas pelo mercado, como a gestão, a codificação, a arquitetura, a qualidade e os testes de *software*.

Capítulo 2

Processos de Desenvolvimento de *Software*

Este capítulo apresenta o referencial teórico necessário para a compreensão deste trabalhoe envolve os processos de desenvolvimento de software tradicional, ágil e híbrido.

2.1 O Processo Tradicional de Desenvolvimento de *Software*

Nos diferentes tipos de indústria, a gestão dos projetos se faz necessária para planejar, estruturar e executar as atividades. O controle e o monitoramento das etapas detêm a maior parte do sucesso de qualquer produção. Na área de Tecnologia não seria diferente. A gestão dos projetos evoluiu à medida que a complexidade dos produtos, das demandas e dos desafios aumentou. A evolução tecnológica da sociedade também reflete diretamente no aumento dessa complexidade, pois o consumo voltado aos produtos e serviços que acompanham as inovaçõesdo mercado são cada vez mais procurados.

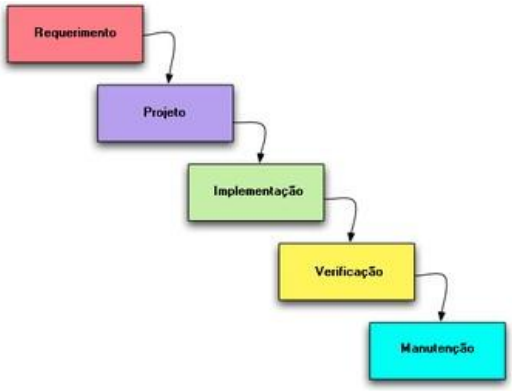
Todavia, tudo tem um ponto de partida. A gestão de *softwares* tem suas raízes fincadas nos modelos prescritivos. Um modelo de processo prescritivo para o desenvolvimento de *software* foi originalmente proposto para minimizar a desordem dos projetos e a insatisfação doconsumidor final. Desde que esse tipo de modelo foi adotado, é possível observar claramente que houve um salto de entendimento sobre a estrutura mínima de um projeto de *software*. O modelo prescritivo serviu – e ainda serve – como um roteiro razoavelmente eficaz para as equipes de desenvolvimento.

Dentre os tipos de modelo prescritivo, está o modelo Cascata, também conhecido como *Waterfall*. O Cascata é um processo tradicional de desenvolvimento composto por etapas mais rígidas, lineares e sequenciais, sendo necessário finalizar todas as tarefas prescritas para uma etapa para que seja possível evoluir para a etapa seguinte. Assim, o *software* é entregue quandotodas as etapas propostas pelo modelo

são finalizadas.

A abordagem em cascata foi proposta por Winston W. Royce em 1970, sendo rápida e largamente adotada por várias indústrias. O sequenciamento lógico das tarefas e a relativa facilidade de implementação chamaram a atenção dos mais diversos setores do mercado. O modelo cascata é composto por cinco etapas: levantamento de requisitos, projeto, implementação, realização de testes e manutenção do sistema. Os tópicos a seguir detalharão cada uma destas etapas.

Figura 3. ciclo de desenvolvimento no modelo cascata



Fonte: Wikipédia. Disponível em: https://upload.wikimedia.org/wikipedia/commons/thumb/0/08/Modelo_em_cascata.png/350px-Modelo_em_cascata.png. Acesso em: 23 setembro de 2022

2.1.1 Fases do Desenvolvimento Tradicional

O levantamento dos requisitos é a primeira fase do projeto regido pelo modelo cascata. Nela, as partes interessadas se reúnem e uma parte da equipe do projeto levanta os requisitos eas necessidades do cliente. Também é nesta etapa que as expectativas sobre o produto são acordadas para possibilitar a definição das funcionalidades que serão posteriormente implementadas. Esta etapa é crucial para um bom desenvolvimento de todo o projeto, pois define aspectos primários que serão norteadores de todas as outras fases. Em caso de erro no levantamento de requisitos, é muito provável que parte significativa do projeto seja descartada por falta de correspondência entre os recursos estimados e as funcionalidades preteridas, inviabilizando muitas vezes a continuação do projeto.

Na fase de Projeto, documentos, estimativas e alocações fundamentais são formalizados,como o cronograma do projeto, a especificação das tarefas de acordo com os requisitos passados, a estimativa da finalização de cada etapa, alocação do

time de desenvolvimento; a modelagem da interface e arquitetura do sistema, as estruturas de dados, dentre outros aspectoscruciais da implantação, que corresponde à fase de codificação do *software* e é imediatamente seguinte a fase de Projeto.

Na fase de implementação, o *software* é codificado na linguagem de programação definida. É definitivamente a etapa tecnicamente mais complexa e central no projeto, pois é nela que o *software* sai do papel e começa a tomar forma de produto. A codificação é regida pelos requisitos e especificações levantados na primeira fase do projeto e a sua duração deve estar em conformidade com o cronograma estabelecido na fase de Projeto.

A fase de testes é a fase mais crítica do modelo Cascata. Como o modelo propõe fases sequenciais, somente neste momento as funcionalidades entram em teste para se verificar a conformidade com os requisitos e as especificações passadas. Isso faz com que o papel do analista de testes seja mais reativo do que proativo, pois a função dele no projeto se concentra nessa fase, não sendo possível estabelecer um processo colaborativo entre a equipe em todas asetapas. Também é neste momento que a preocupação em entregar um *software* funcional e acessível fica latente, pois a codificação precisa permitir uma manutenção eficiente ao longo do tempo. Os erros encontrados nesta etapa são imediatamente considerados e tratados até que as funcionalidades atendam aos acordos iniciais do projeto. Ao final de todas as validações, o cliente volta a ser envolvido para ver o produto completo e em funcionamento.

Ao final da fase de teste, é iniciada a quinta e última fase do modelo cascata. Na manutenção, o *software* está implantado, em uso e eventuais erros e intercorrências do sistemasão reparados antes que comprometam a sua versão de produção. Para realizar a manutenção, existem algumas técnicas disponíveis, como a separação estática. Essa técnica permite que sejam observados todos os blocos de código do programa que apresentaram alguma anomalia,com a finalidade de corrigi-los de forma mais assertiva. Caso seja necessária uma mudança drástica no *software*, o modelo cascata pode ser reaplicado regido pelas especificações das mudanças requeridas. A fase de manutenção tem a finalidade de aumentar a vida útil do produto,mantendo a confiabilidade, as atualizações e usabilidade do *software* em dia.

2.2 O Manifesto Ágil e a Evolução no Desenvolvimento de *Software*

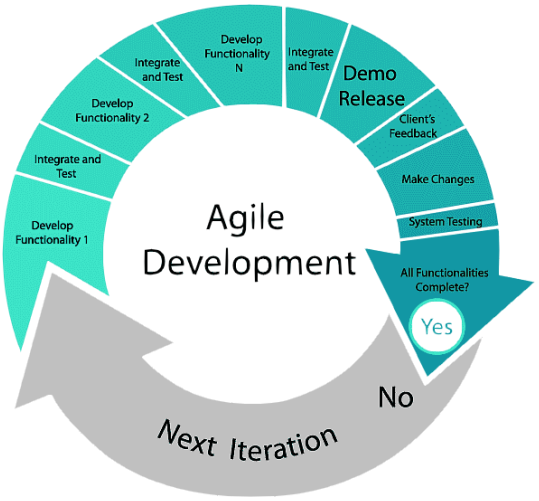
Até o início dos anos de 1990, o processo tradicional de desenvolvimento supria o problemade produtividade e escalabilidade dos anos anteriores com seus processos sequenciais. Porém, com a expansão da internet e dos processos concomitantes, os métodos orientados a objeto e acriticidade do *software* ganharam força como aspectos de competitividade do mercado. Time- to-market, interatividade com o usuário, o controle da concomitância, o desenvolvimento do open-source, a usabilidade e a interação home-computador também se tornaram aspectos importantes no desenvolvimento e na gestão de *software*.

Estes aspectos impulsionaram a adoção de metodologias mais flexíveis, como o XP [4] eo *Scrum* [5], que já existiam antes da formalização do Manifesto Ágil. O Manifesto Ágil surgiuda reunião de 17 desenvolvedores que estavam sofrendo com a rigidez dos contratos, dos processos e da cultura de desenvolvimento de *software*. Embora esses desenvolvedores já seguissem práticas mais enxutas e flexíveis, ainda existia a necessidade de emergir uma nova visão para este desenvolvimento.

O grupo de desenvolvedores se dedicou a escrever um documento que desencadearia o *Agile Manifesto*, ou Manifesto Ágil, que continha crenças e valores do grupo sobre o desenvolvimento fluido de *software*, tendo como centro das considerações a Agilidade e o Valor.

Um processo ágil de desenvolvimento considera fundamental pivotar algumas perspectivas para fomentar uma cultura de fluxo de valor na organização. Indivíduos e interações devem se sobressair em relação aos processos e ferramentas; *software* em funcionamento têm mais valor do que apenas documentações abrangentes; colaboração com ocliente traz mais retorno do que somente negociar contratos; e responder às mudanças dá maiscredibilidade do que simplesmente seguir um plano. O ciclo de vida de um projeto regido por metodologias ágeis essencialmente segue as fases abaixo:

Figura 4. ciclo de desenvolvimento no modelo



Fonte: Product School. Disponível em: https://productschool.com/product-glossary/agile-definition/. Acesso em: 23 setembro 2022

Cada ciclo do projeto é chamado de iteração. Uma iteração é uma entrega de valor completa feita ao cliente. O conjunto das iterações constrói o produto de forma incremental e garante que o cliente seja envolvido durante todo o desenvolvimento, podendo usufruir do valordo projeto antes mesmo do produto estar totalmente finalizado.

Na figura 4, é possível perceber que os testes integrados são uma das práticas do desenvolvimento ágil. Os testes integrados têm um papel importante na iteração, pois têm o objetivo de aperfeiçoar continuamente as funcionalidades implementadas, se opondo à prática do desenvolvimento tradicional, onde os testes são feitos ao final da implementação total das funcionalidades, se concentrando no fim do ciclo de vida do projeto.

A Demo Release, outra prática importante do ciclo de desenvolvimento ágil, é o momento em que a equipe de desenvolvimento demonstra as funcionalidades ao cliente para aperfeiçoar o produto. Caso alguma funcionalidade não possa ser implementada na iteração atual, esta entrano *backlog* do produto da próxima iteração.

O que é fundamentalmente importante de ser considerado é a forma como o valor é percebido nos dois processos de desenvolvimento. No processo tradicional, a percepção de valor está no sequenciamento das atividades para maior controle do projeto. No processo ágil, a percepção de valor está na melhoria contínua da construção incremental do produto para que ocliente possa usufruir.

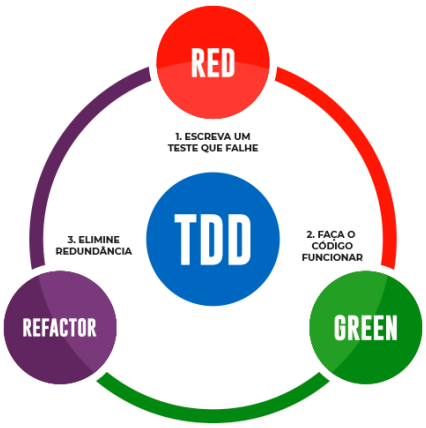
2.2.1 O TDD e a Perspectiva de Testes no Desenvolvimento de *Software*

Um dos diferenciais mais notórios das metodologias ágeis é ter os testes integrados a todas as etapas do desenvolvimento. Dentro dessa perspectiva, existem *frameworks* e metodologias especialmente orientados e centralizados à etapa de testes. Ainda nos anos 1990 e em paralelo com o surgimento das primeiras metodologias ágeis, surgia o TDD, ou Desenvolvimento Orientado a Testes. O TDD é uma técnica de desenvolvimento de *software* que se relaciona com conceitos de verificação e validação, através de ciclos curtos de repetições.Essa técnica preza pelo desenvolvimento de códigos simples para que sejam geradas validaçõesigualmente simples.

O TDD propõe a criação dos cenários para os testes automatizados antes mesmo do código da aplicação ser desenvolvido. Os testes contêm asserções que podem ser verdadeiras ou falsas. Durante a execução, caso o cenário de teste seja considerado verdadeiro, os desenvolvedores podem prosseguir com o desenvolvimento e a refatoração do código. É importante destacar que esses testes acontecem de forma contínua, de modo que a cada funcionalidade criada, um cenário de teste seja bem descrito e projetado.

A técnica propõe três fases para a sua implementação. A primeira delas é a fase Red, queinicia o ciclo. Nela, o desenvolvedor escreve o código do teste automatizado da nova funcionalidade antes dessa funcionalidade estar em produção. Nessa fase, é esperado que esse teste não tenha êxito. Na segunda fase, a Green, um código simples é elaborado para ser aprovado no teste recém-criado. Na terceira e última fase, a Refactor, o código é refatorado emprol da melhoria contínua, com o objetivo de torná-lo mais funcional e sanitizado.

Figura 5. ciclo de desenvolvimento do TDD



Fonte: Treina Web. Disponível em: https://www.treinaweb.com.br/blog/afinal-o-que-e-tdd. Acesso em: 26 setembro 2022.

Além de ser uma técnica de desenvolvimento de *software*, o TDD traz consigo uma filosofia de desenvolvimento que se preocupa em aumentar a capacidade de resposta da equipede desenvolvimento diante das inúmeras alterações sofridas pelo produto durante a sua implementação.

2.2.2 O BDD como Evolução do TDD

Embora TDD seja uma técnica essencialmente prática, o isolamento dos desenvolvedoresem relação aos outros stakeholders traz dificuldades práticas para o uso de uma técnica orientada a testes. Estes podem não ter clareza suficiente nos objetivos do negócio para discernirpor onde começar, o que testar e o que não testar. Também ocorre que em muitos casos, esse isolamento faz com que os testes não abarquem requisitos importantes de outras áreas do projeto, como qualidade e segurança, por exemplo. A comunicação não é eficiente entre as equipes do projeto no nível de código. Esse problema tende a piorar a medida que a complexidade do projeto aumenta.

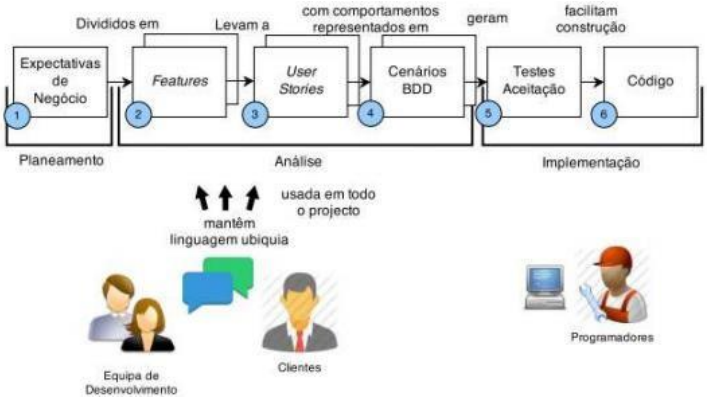
É exatamente nesse contexto que surge o BDD. Behavior Driven Development, ou Desenvolvimento Orientado a Comportamento, surgiu em 2004 como uma resposta às lacunasentre os requisitos do negócio e a solução técnica. Dan North conceituou o BDD como uma metodologia de desenvolvimento ágil orientada aos testes e que visa integrar regras de negócioscom linguagem de programação. Por manter a prática do refinamento dos requisitos ainda na concepção de ideia do

software por meio dos testes, o BDD ancora suas raízes no TDD e adiciona aspectos comportamentais ao processo de desenvolvimento. É possível aproveitar o conteúdo das histórias de usuário como cenário de teste em alguns *framework*s do BDD.

O estímulo à visão do todo também se mostra um diferencial nessa metodologia. Há umapreocupação clara com o entendimento das regras de negócio por todas as partes envolvidas noprojeto, explicitando o porquê de o código ser criado, ao invés de apenas detalhá-lo tecnicamente. O incentivo à comunicação e à colaboração reforçam entregas eficientes e resultam num código de boa qualidade, com alta coesão e com número reduzido de bugs, aumentando a vida útil do *software* e reduzindo o custo de manutenção. É observado também que o BDD estimula a multifuncionalidade das equipes, pois o trabalho próximo e colaborativodas pessoas envolvidas promove o repasse natural de conhecimento dentro do projeto. Além disso, as especificações executáveis do *software* também auxiliam na documentação dinâmica,pois alguns dos *frameworks* do BDD geram relatórios destes artefatos em formato HTML.

Utilizando a escrita das histórias de usuário como cenário de teste, juntamente com critérios de aceitação e Linguagem Ubíqua, o BDD busca fazer sua curva de aprendizagem ser menor, possibilitando que equipes com novos integrantes entreguem resultados muito mais rápidos e com maior qualidade. A Linguagem Ubíqua é uma linguagem estruturada em torno do modelo de domínio usado por todos os membros de uma equipe para conectar todas as suasatividades com o *software*. Numa equipe de projeto, significa dizer que os jargões técnicos e asterminologias das discussões, por exemplo, são normalizadas para que todas as pessoasenvolvidas no projeto consigam compreender e se comunicar de forma efetiva.

Figura 6. fluxo padrão de escrita, validação e codificação dos cenários com BDD



Fonte: Madium. Disponível em: https://medium.com/desenvolvimento-orientado-por-comportamento/desenvolvimento-conduzido-de-comportamento-bdd-b2c98daea331. Acesso em: 26 setembro 2022

De forma objetiva, o BDD se concentra em três pilares:

- Tudo é comportamento: a área de negócios e de Tecnologia devem se referir para o sistema da mesma forma;

- Onde está o valor do negócio: todo sistema deve ter comportamentos que sejam um verificador do valor para o negócio;

- Faça o suficiente: analisar, projetar e planejar tudo de cima para baixo, evitando o detalhamento prematuro.

2.3 O Processo de Desenvolvimento Híbrido de *Software*

Desde a formalização dos processos de desenvolvimento ágeis na indústria de *software*, a concepção de desenvolvimento híbrido se estabeleceu, mesmo que informalmente. Desenvolvimento híbrido se refere às práticas de gestão e desenvolvimento de *software* que vêm de mais de uma vertente, mesclando processos ágil com processos tradicionais - ou tradicionais. Também é possível falar de hibridismo quando se tem apenas práticas da vertente ágil, mas oriundas de mais de uma metodologia ágil. Neste trabalho, será enfatizado o hibridismo que se refere às práticas oriundas de vertentes distintas.

O desenvolvimento híbrido de *software* reflete alguns indícios que possivelmente justificam sua adoção. O tipo de produto, a maturidade e a cultura da organização e dos times do projeto são alguns fatores relevantes nesta análise. É possível observar que o hibridismo pode ser adotado como uma resposta da organização às mudanças e aos avanços impostos pelomercado. Isso nos diz que mesmo que uma organização não tenha maturidade suficiente para absorver novos modelos de gestão e desenvolvimento, suas práticas sofrem algum tipo de adaptação para acompanhar minimamente as novas tendências de mercado. Em qualquer organização, aderir novas práticas de trabalho é um processo delicado que requer atenção. Se tratando de organizações com grau de adaptabilidade baixo, essa adesão vem acompanhada de resistência por parte dos times, dificultando a compreensão dos motivos e benefícios das mudanças.

A polarização em relação às práticas adotadas num projeto pode colaborar diretamente para a rejeição dos times às novas metodologias. Em meio a essas adversidades, práticas e metodologias alternativas são bem vistas por organizações que procuram evitar choques culturais muito fortes para não desestimular suas equipes.

Capítulo 3

Materiais e Métodos

Este capítulo apresenta os materiais e métodos usados na construção deste trabalho, incluindo o objeto de estudo, o tipo de pesquisa, a coleta e a análise dos dados.

3.1 Objeto de Estudo

Este trabalho tem como objeto de estudo os processos de desenvolvimento de *software* e, em particular, o uso de BDD como prática que auxilia a transição de processos tradicionals de desenvolvimento de *software* para processos ágeis de desenvolvimento de *software*.

3.2 Classificação da Pesquisa

Este trabalho está classificado como uma pesquisa de levantamento que visa obter dados ou informações que ajudem a descrever como o mercado de desenvolvimento de *software* está em relação ao uso de BDD como prática que auxilia a transição de processos tradicionals de desenvolvimento de *software* para processos ágeis de desenvolvimento de *software*. Assim, como uma pesquisa de campo que visa observar o uso de BDD, realiza-se uma pesquisa quantitativa com análise estatística dos dados. Entretanto, também foi decidido que a pesquisa seguiria uma linha qualitativa, com o intuito de identificar e analisar dados que não podem ser mensurados estatisticamente.

3.3 Coleta de Dados

De forma prévia, foi preciso fundamentar vários conceitos que circundam o BDD para compreender o que está envolvido nessa prática e de que forma ela se relaciona

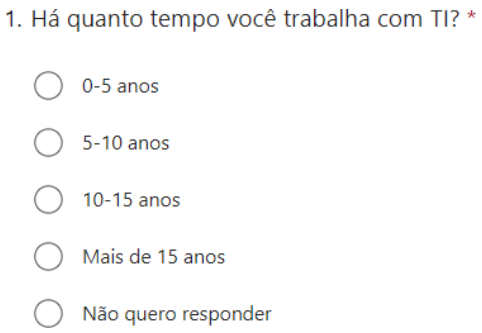
com as práticasde metodologias ágeis e com o desenvolvimento de *software*. Após isso, uma pesquisa do tipo*Survey* foi elaborada com 10 perguntas baseadas nessa fundamentação.

A pesquisa foi dimensionada na escala Likert, comumente usada em pesquisas de opinião. A escala Likert é uma espécie de tabela de classificação e originalmente dimensiona as respostas em 5 pontos, sendo o primeiro ponto a discordância total sobre o questionamento; o segundo ponto sendo a discordância parcial; o terceiro ponto sendo neutro ou indiferente; o quarto sendo concordância parcial; e finalmente o cinco sendo a concordância total em relação ao questionamento. Cada pergunta elaborada atende a um aspecto importante dentro do objetivodeste trabalho e durante a análise das respostas, os dados serão descritos estatisticamente para contribuir com a interpretação dos resultados.

Visando levantar a percepção dos entrevistados sobre a prática do BDD nas equipes de desenvolvimento de *software*, foi realizada uma pesquisa *Survey* com 32 pessoas, que estão inseridas na área de tecnologia atuando em diversos cargos, como desenvolvedores, testers, SM,P.O, analistas e funções relacionadas. O conteúdo da pesquisa é voltado para questionamentossobre a aplicabilidade e os fundamentos do BDD. Assim, este trabalho selecionou uma amostraintencional (não-probabilística) na coleta de dados, isto é, o formulário foi encaminhado para entrevistados que possuem alguma experiência tanto com o processo tradicional quanto com processos ágeis.

Para iniciar a pesquisa, foi perguntado aos entrevistados há quanto tempo estão trabalhando no mercado de Tecnologia da Informação. Acredita-se que conhecimento sobre metodologias mais específicas, como o BDD, está diretamente relacionado ao nível de experiência do entrevistado. Foi feito um agrupamento em blocos de anos para facilitar a análise. O primeiro bloco se refere aos entrevistados com experiência entre 0-5 anos, o segundobloco com experiência entre 5-10 anos, o terceiro entre 10-15 anos e o quarto bloco aos entrevistados com mais de 15 anos de experiência. Durante a análise dos resultados, esta questão será levada em consideração na interpretação das respostas.

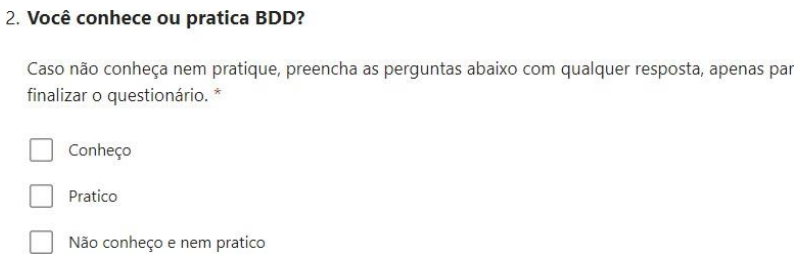
Figura 7. pergunta nº 1 do questionário



Fonte: gerado pela autora (2022).

Logo em seguida, o entrevistado deixa claro se conhece, prática ou se nunca ouviu falar sobre BDD. Esta pergunta é fundamental para fazer a triagem das respostas, uma vez que as pessoas que não têm familiaridade com a metodologia poderiam interferir nos resultados estatísticos da pesquisa e gerar dados ruidosos.

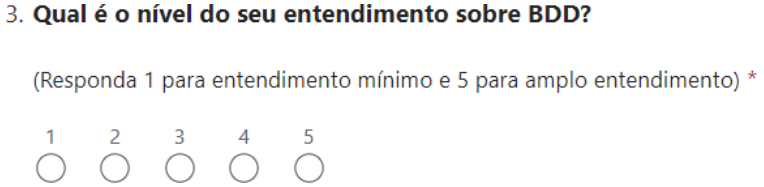
Figura 8. pergunta nº 2 do questionário



Fonte: gerado pela autora (2022).

A partir da resposta anterior, a pesquisa segue para a terceira pergunta, onde quer saber sobre o conhecimento do BDD por parte dos entrevistados. Esta pergunta tem como objetivo fazer relação entre o entendimento sobre o assunto com o nível de experiência do entrevistado.

Figura 9. pergunta nº 3 do questionário

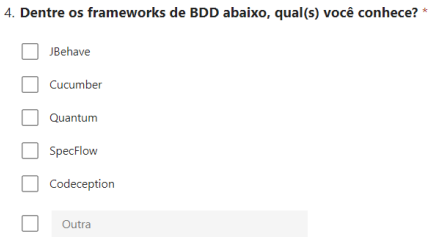


Fonte: gerado pela autora (2022).

A quarta pergunta se propõe a levantar quais são os *frameworks* mais populares dentro doespaço amostral de entrevistados. Dentre as alternativas, estão Jbehave, Cucumber, Quantum, SpecFlow, Codeception e um campo adicional “Outro” para o

entrevistado que não utiliza nenhuma das opções anteriores. A pergunta tem como objetivo analisar se motivação para a escolha do *framework* se relaciona com o uso de linguagens de programação comumente maisusadas, por exemplo.

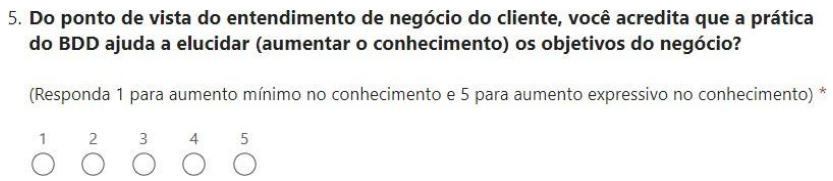
Figura 10. pergunta nº 4 do questionário aplicado



Fonte: gerado pela autora (2022).

Já a quinta pergunta objetiva mensurar o quanto a prática do BDD melhora o entendimento sobre os objetivos e regras do negócio. Esta pergunta vai de encontro a uma das problemáticas que impulsionou o surgimento do BDD. Ela é fundamental para apurar se a prática das equipes que utilizam o BDD reflete nas suas fundamentações. Comumente os *frameworks* e as metodologias sofrem adaptações por parte das equipes para que possam ser melhor associadas a cada realidade. Todavia, se estas adaptações não preservarem a essência da prática, sua utilização perde o sentido e compromete a percepção de valor por parte da equipe.

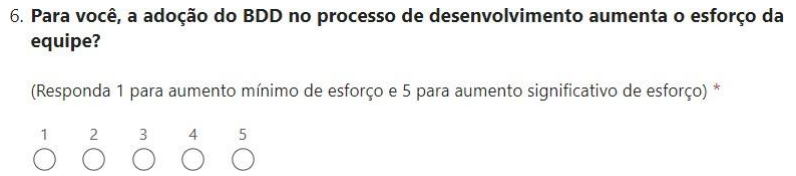
Figura 11. pergunta nº 5 do questionário



Fonte: gerado pela autora (2022).

A sexta pergunta deseja entender como a equipe de desenvolvimento percebe o entendimento do negócio com a prática do BDD. É uma questão que reflete bastante nas premissas do projeto e traz resultados extremamente prejudiciais a longo prazo.

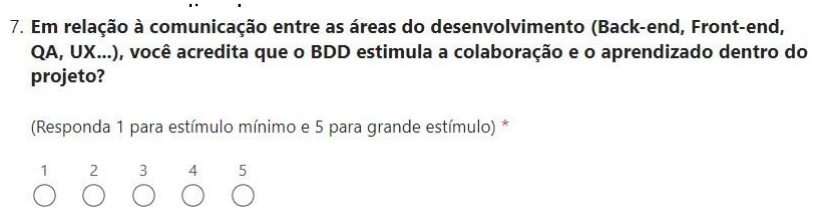
Figura 12. pergunta nº 6 do questionário aplicado



Fonte: gerado pela autora (2022).

A sétima pergunta se refere à colaboração e ao aprendizado dentro da prática do BDD. Esta pergunta é complementar à sexta questão e deseja analisar outra problemática que motivou e fundamentou o surgimento do BDD.

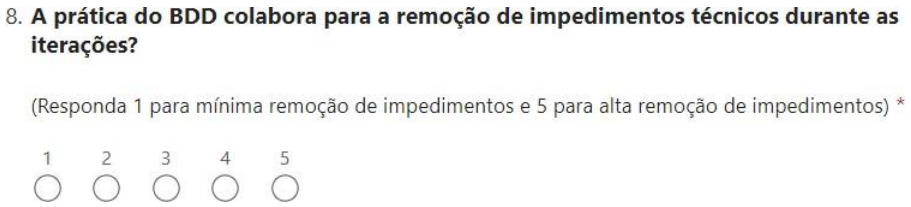
Figura 13. pergunta nº 7 do questionário



Fonte: gerado pela autora (2022).

A oitava pergunta se propõe a mensurar o quanto o BDD colabora para a remoção de impedimentos técnicos no projeto. Uma das proposições da metodologia é que a colaboração ea comunicação fluida entre as partes interessadas promova um código mais eficiente, coeso, com vida útil maior e com valor de manutenção mais baixo. É importante levantar se esse pilaré percebido de forma prática por quem o utiliza.

Figura 14. pergunta nº 8 do questionário aplicado

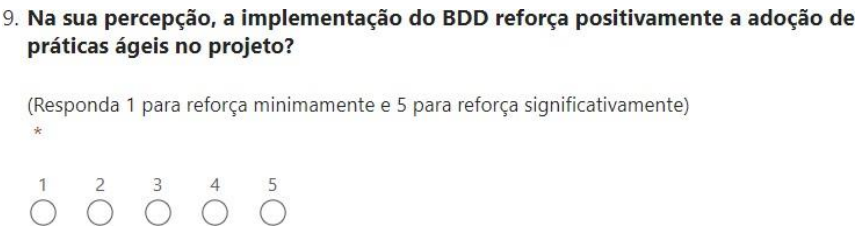


Fonte: gerado pela autora (2022).

Finalmente, na nona pergunta, o questionário deseja saber o quanto os entrevistados relacionam o BDD com as práticas ágeis no projeto. O intuito é perceber se os entrevistados têm clareza de que o BDD é tido como uma metodologia ágil e que este preza por práticas ágeis em todas as suas fases. Esta pergunta também se relaciona diretamente com o objetivo geral deste trabalho, que é avaliar o BDD como

mecanismo e prática independentes no desenvolvimento de *software*.

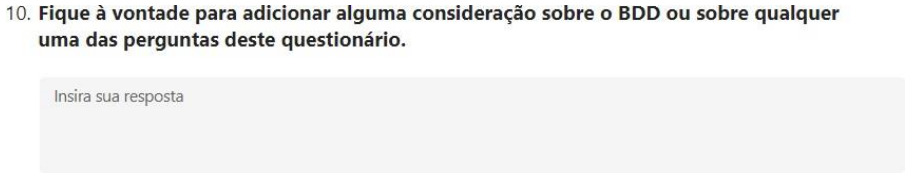
Figura 15. pergunta nº 9 do questionário aplicado



Fonte: gerado pela autora (2022).

Ao final das perguntas, foi aberto um espaço para respostas subjetivas, para que, eventualmente, os entrevistados fizessem considerações sobre as respostas anteriores ou sobre o tema do questionário. Esta seção da pesquisa será usada para complementar a interpretação sobre seu resultado.

Figura 16. pergunta nº 10 do questionário



Fonte: gerado pela autora (2022).

Todas as perguntas foram adicionadas a um formulário digital criado com o Microsoft Forms e disponível online[1].

3.4 Análise de Dados

Para a análise dos dados coletados, foram utilizados recursos estatísticos como a moda ea visualização gráfica e percentual dos mesmos.

[1] Formulário online disponível em: https://forms.office.com/pages/responsepage.aspx?id=DQSIkWdsW0yxEjajBLZtrQAAAAAAAAAAAAZAAJ8 BtqVUNFE4VEI1WFNNRERGWEs3SFQ0RjI5Q0pBTS4u

Capítulo 4

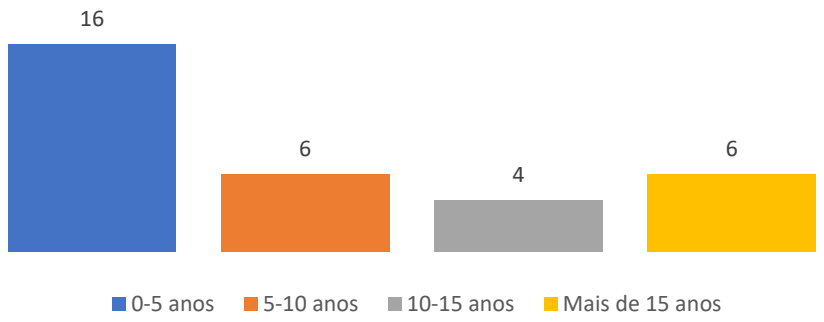
Resultados

4.1 Análise Estatística dos Dados

Para cada questão do formulário foi feita uma análise estatística das respostas obtidas. Aquestão 1 obteve a moda 0-5 anos com 50% das respostas (16 respostas das 32 respostas ao questionário). Assim, nota-se que metade dos respondentes possui mais de 5 anos de experiência com desenvolvimento de *software*.

1. Há quanto tempo você trabalha com TI?

Figura 17. resultado quantitativo da pergunta nº1 do questionário aplicado



Fonte: gerado pela autora (2022)

Para a questão 2 sobre a prática de BDD pelo respondente, 23 respondentes afirmaram conhecer ou praticar a técnica, sendo “Conheço” a moda das respostas, com 13 ocorrências. Assim, houve um índice de concordância de 71,87%.

2. Você conhece ou pratica BDD?

- Índice de Concordância = 71,87%

Figura 18. resultado quantitativo da pergunta nº2 do questionário aplicado

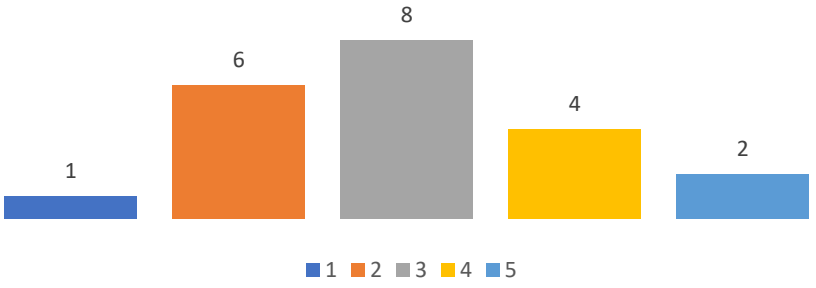
Fonte: gerado pela autora (2022)

Quanto ao nível de entendimento, questão 3, considerando a escala Likert onde 1 significa entendimento mínimo, 2 significa entendimento fraco, 3 significa entendimento razoável, 4 significa bom entendimento e 5 significa amplo entendimento, observou-se a moda 3 com 8 respondentes afirmando ter entendimento razoável.

3. Qual é o seu nível de entendimento sobre BDD?

- Moda de respostas = 3 (entendimento razoável)

Figura 19. resultado quantitativo da pergunta nº3 do questionário aplicado

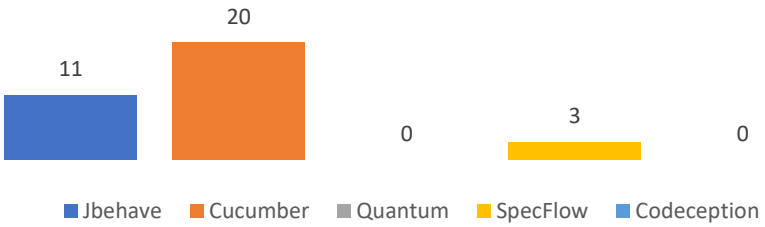


Fonte: gerado pela autora (2022)

Sobre o conhecimento em relação aos *frameworks* do BDD, questão 5, 60,86% (14 respondentes) afirma conhecer apenas um framework (Cucumber ou JBehave). Já 31,13% (9 respondentes) afirmam conhecer mais de um *framework*. Não foram registradas respostas para*frameworks* Quantum e Codecepction.

4. Dentre os *frameworks* de BDD abaixo, qual(s) você conhece?

Figura 20. resultado quantitativo da pergunta nº4 do questionário aplicado



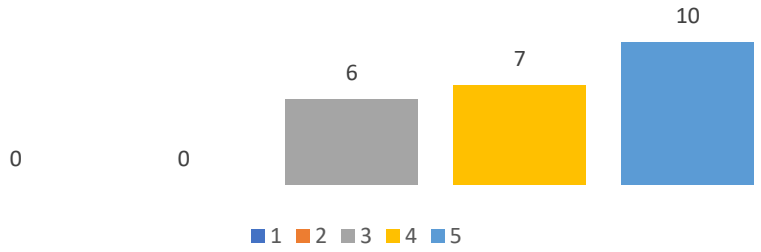
Fonte: gerado pela autora (2022)

Em relação à ajuda do BDD no aumento de conhecimento sobre os objetivos do negócio,questão 5, os resultados mostraram que a moda foi 5 (aumento expressivo) e não houve respostas indicando aumento mínimo ou fraco.

5. Do ponto de vista do entendimento de negócio do cliente, você acredita que a prática doBDD ajuda a elucidar (aumentar o conhecimento) os objetivos do negócio?

• Moda das respostas = 5 (aumento expressivo)

Figura 21. resultado quantitativo da pergunta nº5 do questionário aplicado



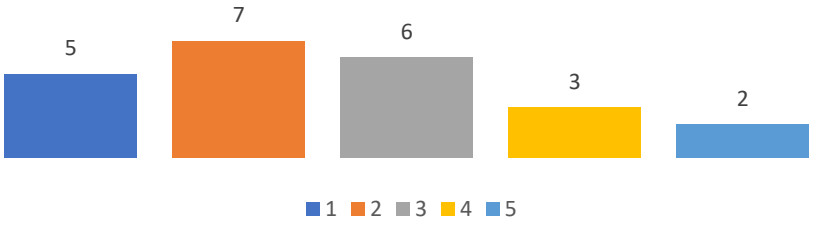
Fonte: gerado pela autora (2022)

Quanto ao aumento do esforço da equipe ao utilizar BDD, questão 6, a moda das respostas foi 2 (aumento pequeno do esforço), enquanto apenas 21,73% (5 respondentes) consideraram um aumento grande ou expressivo do esforço.

6. Para você, a adoção do BDD no processo de desenvolvimento aumenta o esforço da equipe?

• Moda das respostas = 2 (aumento pequeno do esforço)

Figura 22. resultado quantitativo da pergunta nº6 do questionário aplicado



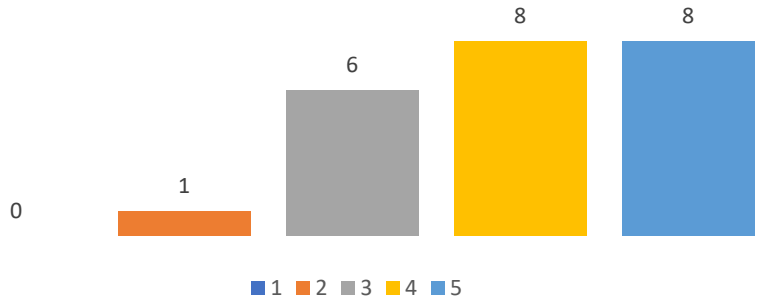
Fonte: gerado pela autora (2022)

Sobre o estímulo à colaboração e o aprendizado dentro do projeto, questão 7, o resultado foi bimodal, com valores 4 e 5, indicando que 69,57% dos entrevistados afirmam que há grande ou expressiva colaboração e aprendizado com o uso do BDD. Já os que afirmam haver colaboração e aprendizado fracos ou razoáveis, 30,43% (7 respondentes) têm essa percepção. Não foram registradas respostas para colaboração e aprendizados mínimos.

7. Em relação à comunicação entre as áreas do desenvolvimento (Back-end, Front-end, QA, UX...), você acredita que o BDD estimula a colaboração e o aprendizado dentro do projeto?

- Moda das respostas = 4

Figura 23. resultado quantitativo da pergunta nº7 do questionário aplicado



Fonte: gerado pela autora (2022)

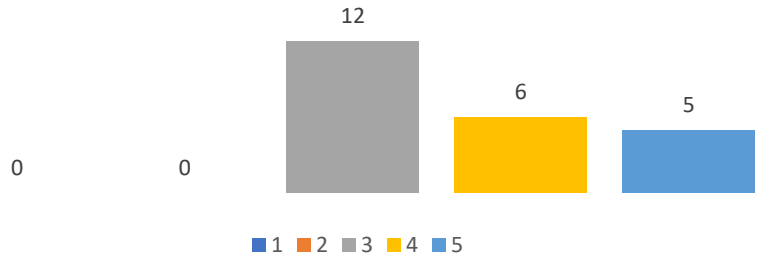
Quanto à remoção dos impedimentos técnicos durante as interações, questão 8, a moda foi 3, indicando que 52,17% dos entrevistados afirmam que há razoável remoção dos impedimentos técnicos com o uso do BDD. Já para 47,83% (11 respondentes), há grande ou expressiva remoção de impedimentos, não sendo registradas respostas para remoção mínima ou fraca.

8. A prática do BDD colabora para a remoção de impedimentos técnicos

durante asiterações?

• Moda das respostas = 3 (colaboração moderada)

Figura 24. resultado quantitativo da pergunta nº8 do questionário aplicado



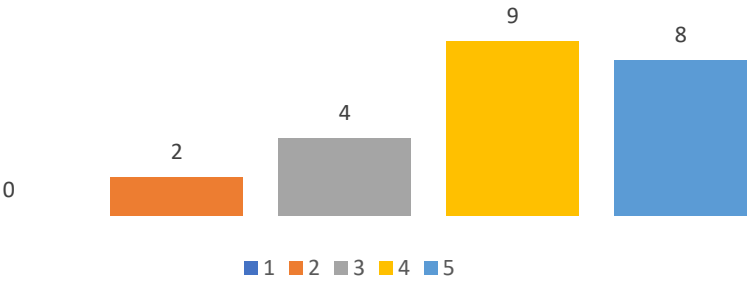
Fonte: gerado pela autora (2022)

Sobre o reforço das práticas ágeis com o uso do BDD, questão 9, a moda foi 4, onde 73,91% dos entrevistados afirmam que há reforço grande ou expressivo. Já 26,08% (6 respondentes) consideram que há reforço fraco ou moderado nas práticas ágeis.

9. Na sua percepção, a implementação do BDD reforça positivamente a adoção de práticaságeis no projeto?

• Moda das Respostas = 4 (reforço grande das práticas ágeis)

Figura 25. resultado quantitativo da pergunta nº9 do questionário aplicado



Fonte: gerado pela autora (2022)

Na décima pergunta, o estrevistados tinham a opção de opinar subjetivamente e de forma livre sobre o BDD. Abaixo, estão descritas as respostas obtidas nessa questão.

10. Fique à vontade para adicionar alguma consideração sobre o BDD ou sobre qualquer uma das perguntas deste questionário.

- *“Acredito que o BDD também é uma forma de aproximar os times voltados para*

o produto e os times de desenvolvimento."

- *"A implementação do BDD contribui bastante para o entendimento de todas as partes dos projetos, diferentemente com o que ocorre com as metodologias comumente utilizadas nas empresas, que deixa boa parte da noção do software isolado com os times técnicos, o que muitas vezes gera ruído e um entendimento incorreto do produto que está sendo desenvolvido."*

- *"O BDD é uma excelente prática para melhorar o entendimento do negócio e do comportamento do sistema, além de ser uma prática de automação de testes complementar ao TDD. Ela também serve como uma documentação viva, de fácil manutenção e extremamente útil para desenvolvedores."*

Capítulo 5

Discussão

Após observar os resultados estatísticos descritos no capítulo anterior, se faz necessário interpretar e fazer algumas considerações. O desenvolvimento deste trabalho se voltou para o levantamento de benefícios e *trade-offs* da prática do BDD em equipes de desenvolvimento de *software*. É fundamental que este levantamento seja interpretado de acordo com alguns aspectos qualitativos.

Antes de iniciar a análise, é preciso destacar em detalhes o espaço amostral que compõe a pesquisa. A pesquisa contou com 32 pessoas entrevistadas anonimamente, que trabalham em empresas de consultoria, tecnologia da informação e em fábricas de *software* espalhadas pelo Brasil. Todavia, estas organizações não serão expostas neste trabalho, pois o intuito é analisar e interpretar a percepção dos entrevistados sobre o tema sem esse tipo de enviesamento. Vale salientar que os entrevistados têm níveis de experiência e funções corporativas distintas para garantir a diversificação dos pontos de vista sobre o tema.

Em relação ao nível de experiência, 50% dos entrevistados têm entre 0 e 5 anos de experiência, 18,75% têm entre 5-10 anos, 12,5% têm entre 10-15 anos e 18,75% têm mais de 15 anos de experiência. É interessante observar que dentre os 9 entrevistados que não conhecem nem praticam BDD, 6 deles estão iniciando a carreira na indústria da Tecnologia, tendo entre 0 e 5 anos de experiência. Isso indica que, possivelmente, o BDD ainda seja uma prática familiar apenas para pessoas com certo nível de vivência empresarial, sendo parcial ou completamente desconhecida por quem está ingressando nesse tipo de indústria.

É bem verdade que a prática do BDD independe do nível de experiência da equipe de desenvolvimento, assim como não se restringe aos profissionais da área de Testes, visto que um dos seus objetivos principais é justamente extrapolar as barreiras da área de Testes, aumentar a colaboração entre as partes envolvidas no projeto e o entendimento dos objetivos de negócio por parte da equipe. Inclusive, é visto em CEVERINO, Aparecida e NASCIMENTO, Fernando Paes [6] que o BDD é uma

excelente técnica alternativa aos métodos tradicionais para levantaros requisitos do projeto. Contudo, é perceptível que a prática ainda está longe de ser tida comouma metodologia conhecida nessa etapa inicial de formação e desenvolvimento profissional.

Em contrapartida, o nível de entendimento sobre o BDD por parte da parcela que conheceou pratica a metodologia se manteve razoável ou mediano, indicando que o nível de conhecimento sobre o assunto não é diretamente proporcional ao nível de experiência do entrevistado. É estimado que alguns fatores justifiquemesse resultado, como por exemplo praticar o BDD de forma experimental, sem avançar no conhecimento teórico sobre ele. Praticar uma metodologia de forma experimental, adaptando- a de acordo com a demanda e as necessidades do contexto é perfeitamente normal dentro de uma rotina de projeto ou de uma organização. Porém, eventualmente pode indicar que a metodologia está sendo subutilizada pelos profissionais, fazendo com que muitas vezes a equipedo projeto ou o próprio cliente não sintam integralmente os benefícios propostos pela prática.

É essencial que qualquer tipo de metodologia usada no desenvolvimento do produto sejaadaptada e praticada de forma consciente e eficiente, preservando seu propósito e seus fundamentos para que qualquer adaptação prática que venha a ser feita mantenha ligação diretacom o propósito e a essência da metodologia.

Dos 32 entrevistados, 9 deles não conhecem nem praticam o tema, representando 28,12% da amostra. Foi um quantitativo inferior, mas próximo ao estimado na etapa de idealização do estudo, tendo em vista que a pesquisa foi veiculada em organizações que estão dentro do ecossistema da Tecnologia, mas que não necessariamente produzem *softwares*. Para evitar ruído nos dados e distorção dos resultados, estes 9 entrevistados não familiarizados com o tema foram descartados da amostra, reduzindo o espaço amostral das questões seguintes para 23 respondentes válidos.

Em relação aos *framework*s mais utilizados e conhecidos por quem é familiarizado com BDD, o entrevistado poderia escolher uma ou mais de uma opção como resposta. Foram listados os *frameworks* Cucumber, JBehave, SpecFlow, Quantum e Codeception como opções, além deuma alternativa nomeada como Outro para que o entrevistado pudesse adicionar outro *framework* relacionado ao BDD e que não foi listado.

O Cucumber foi destacado por 86,95% dos entrevistados como sendo o *framework* mais conhecido. Em paralelo, o JBehave também foi citado por 43,47% dos entrevistados, seguido pelo SpecFlow com 13,04% das menções. De acordo com o Perfecto (2021) [7], o Cucumber foi originalmente criado por membros da comunidade de desenvolvedores Ruby para apoiar o desenvolvimento de testes de aceitação automatizados utilizando o BDD. O Cucumber foi traduzido para outras linguagens de programação, inclusive o Java e JavaScript, popularizando ainda mais o *framework*.

O curioso é que o Cucumber não é o primeiro *framework* criado para o BDD. O JBehave foi o primeiro *framework* desenvolvido para o BDD e criado pelo seu próprio idealizador, Dan North. O JBehave suporta linguagens como Java e outras linguagens JVM, como Groovy, Kotlin e Scala. Contudo, é importante considerar que o JBehave não oferece alguns recursos comuns do Gherkin, fazendo com que, possivelmente, equipes que procuram estruturas compatíveis com a linguagem migrem para o Cucumber. Gherkin, que é uma Linguagem de Domínio Específico que permite descrever o comportamento do *software* sem o detalhamento da sua implementação. O Gherkin é tido como uma ferramenta eficiente para documentação e automação de testes, sendo igualmente eficiente dentro da metodologia e dos *frameworks* do BDD.

Ainda de acordo com o Perfecto (2021) [7], há a possibilidade de utilizar o Cucumber dentro de uma estrutura que utiliza o *framework* Selenium, que é uma estrutura de teste muito popular para testar aplicativos web. Com isso, entende-se que a popularidade deste *framework* na pesquisa deste trabalho se justifica em cima desses aspectos.

Do ponto de vista de negócio, os entrevistados são enfáticos em pontuar que há aumento de razoável a expressivo no entendimento e nos objetivos do negócio. De acordo com [8], o canal de comunicação entre os stakeholders do projeto pode ter algum “ruído” que leva à perda de informação, produzindo o que ele chama de lacuna semântica entre o que se deseja e o que será desenvolvido. A lacuna semântica seria a inconsistência nos requisitos representados pelos cenários – que são as histórias de usuários em um contexto de desenvolvimento orientado por comportamento - e pelo modelo conceitual.

Existem diversas abordagens que trabalham em cima dessa lacuna semântica para estreitar a distância entre os cenários descritos e o modelo conceitual dos

requisitos, implicandoconsequentemente na aproximação entre a área de negócios e a fábrica de *software*. Vale enfatizar que a abordagem *low-code* com BDD, como visto em [9], é uma possibilidade de abordagem que visa diminuir o ruído da comunicação entre as partes envolvidas, ampliando a discussão fluida e harmoniosa entre as partes interessadas. Dito isso, se observa porque, para os entrevistados, o BDD, seus fundamentos e suas ferramentas favorecem a melhoria do entendimento e da comunicação entre estas áreas, colaborando para projetos mais coesos e eficientes.

Todavia, não é possível ter um apontamento claro sobre o resultado dessa questão, pois os respondentes tiveram percepções equilibradas sobre esse aspecto, que refletiram nos dados coletados. Não há uma percepção marjoritária sobre a questão. Estima-se que isso se deva ao fato do esforço ser uma questão relativamente subjetiva, não sendo possível, da forma que a presente pesquisa se propôs a fazer, levantar aspectos mais profundos sobre a percepção de esforço na rotina de um projeto. Reforçando esse ponto de vista, foi realizado um estudo voltado para desmistificar a adoção do BDD em projetos *open source* [10], que analisa o BDD além da simples e objetiva adoção da metodologia, destacando a versatilidade de aplicações dessa prática. O estudo concluiu que "*embora o uso de estruturas de BDD seja difundido entre os projetos open source, (...) os desenvolvedores sentiram que o BDD continua bastante propensoa esforços, e sua aplicação vai além da simples adoção de uma estrutura BDD.*", indicando que a percepção sobre esforço é relativa e difícil de ser mensurada. Assim, a conclusão sobre o esforço empregado por parte da equipe na adoção do BDDainda é uma questão em aberto, que merece um estudo e detalhamentos à parte.

No que diz respeito à colaboração e ao aprendizado em projetos que utilizamo BDD, os entrevistados entendem que há grande aumento nesses dois quesitos. Esse resultado está bastante relacionado ao resultado da pergunta sobre aumento do entendimento e dos objetivos do negócio. É esperado que exista uma correlação alta entre esses dois resultados, porque uma vezexistindo uma comunicação eficiente entre as partes relacionadas, de forma que ruídos sejam mitigados e os objetivos do negócio sejam clarificados, é esperado que ao menos a percepção da equipe em relação à colaboração seja favorecida, de modo que, consequentemente, a visão do todo estimulada pela metodologia e praticada dentro dessa comunicação fluía contribua paraum aumento de conhecimento natural dentro do projeto.

Já em relação a remoção de impedimentos técnicos, a pesquisa aponta que há uma razoável percepção de remoção, indicando que, na prática, os entrevistados ainda não sentem uma remoção de impedimentos significativa em seus projetos. Um ponto interessante nesse resultado é que, na visão dosentrevistados com mais de 15 anos de experiência – assumindo que estes ocupem cargos mais executivos e menos operacionais, essa percepção passa de razoável para expressiva. Isso pode indicar que existe uma discrepância entre o que é percebido no dia a dia do projeto pelo time operacional e o que é percebido pela governança do projeto. Porventura, isso pode resultar, de forma cumulativa, num ruído e fortalecer os silos no projeto, indo de encontro aos princípios da metodologia do BDD.

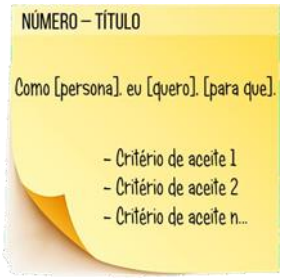
Por fim, o questionário busca saber se, na visão dos entrevistados, a prática de BDD reforça positivamente a adoção de práticas ágeis no projeto. Os entrevistados entendem que há um grande reforço positivo. É interessante observar que um arquivo Gherkin, usualmente presente em projetos com BDD, contém a descrição dos cenários com as seguintes informações:

- Título da funcionalidade;
- Descrição da funcionalidade;
- Cenários compostos por uma sequência de passos que descrevem uma interação entreum usuário e o sistema.

Estes arquivos usam a extensão .feature e são salvos em texto plano para serem lidos e editados por ferramentas simples. A estrutura de um arquivo Gherkin é estabelecida por palavras-chave utilizadas de acordo com o idioma desejado. Abaixo, algumas das palavras- chaves que compõe a estrutura do arquivo: Feature – Funcionalidade; Background – Contexto;Scenario – Cenário; Given – Dado; When – Quando; Then – Então; And – E; But – Mas; Comentários;Dessa forma, é possível fazer uma comparação direta entre o formato das histórias de usuário e os cenários descritos no BDD. A história de usuário é a menor unidade de trabalho deuma estrutura ágil. Em linhas gerais, elas descrevem de forma genérica e informal uma funcionalidade do *software*, a partir da perspectiva do usuário final ou do cliente. Os cenários descritos pelo BDD são histórias de usuário razoavelmente adaptadas para serem executadas pelo *framework* escolhido. A percepção dos entrevistados de que a adoção do BDD reforça práticas ágeis é esperada, uma vez que o BDD é uma

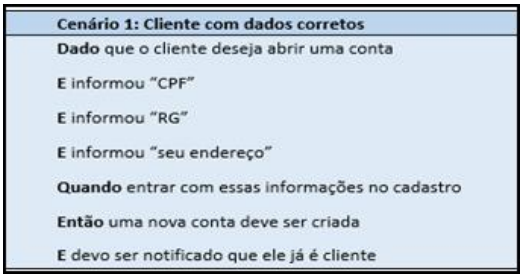
metodologia ágil e mesmo que o entrevistado não saiba disso, a própria prática da metodologia referencia termos, hábitos e conceitos de outras metodologias ágeis, fazendo com que naturalmente a equipe sinta essa relação.

Figura 26.
exemplo de escrita de história de usuário



Fonte: O Dono do Produto. Disponível em: https://odonodoproduto.com/user-story-historia-de-usuario/. Acesso em: 12 Outubro 2022

Figura 27.
exemplo de escrita de cenário no BDD



Fonte: AMCOM. Disponível em: https://odonodoproduto.com/user-story-historia-de-usuario/. Acesso em: 12 Outubro 2022

Por fim, alguns entrevistados utilizaram o espaço disponibilizado no questionário para expressar suas opiniões sobre o tema. Foi interessante observar que os comentários feitos reforçam a ideia de que o BDD é mais do que uma metodologia dentro do projeto, assumindo também um papel cultural nos times. Um entrevistado mencionou "*(...) que o BDD também é uma forma de aproximar os times voltados para o produto e os times de desenvolvimento.*", sereferindo ao fato de que o BDD promove a quebra dos silos entre as equipes do projeto, estimulando a visão do todo e a orientação de todos ao objetivo em comum.

Outro entrevistado menciona que *"O BDD é uma excelente prática para melhorar o entendimento do negócio e do comportamento do sistema (...). Ela também serve como uma documentação viva, de fácil manutenção e extremamente útil para desenvolvedores*.", demonstrando que os artefatos do BDD facilitam o dia a dia do projeto pela versatilidade do seu uso. Nesse ponto, é importante destacar que os comentários feitos pelos entrevistados não referenciam diretamente uma ferramenta ou uma habilidade técnica relacionada ao BDD. A percepção deles é mais holística em relação aos benefícios da metodologia, envolvendo a visãodo produto e do projeto como um todo, não apenas uma das fases do seu desenvolvimento.

Nenhum dos entrevistados mencionou diretamente o BDD como uma prática ágil. A metodologia é percebida como uma prática independente, sendo associada às metodologias ágeis apenas quando as perguntas aos entrevistados direcionam suas

visões para esse ponto. Isso pode se dever ao fato de que muitos o praticam de forma empírica, sem conhecer sua fundamentação teórica. Por outro lado, isso alimenta a hipótese de usá-lo como uma prática independente e neutra na gestão e no desenvolvimento do produto. O BDD tem um bom desempenho em fases do projeto que são comuns ao desenvolvimento tradicional e ao desenvolvimento ágil, podendo ser ponte entre as vertentes.

Capítulo 6

Conclusão

Este trabalho analisou aspectos fundamentais do desenvolvimento orientado por comportamento, desde o cenário antes da sua concepção até a opinião de profissionais que atualmente estão envolvidos direta ou indiretamente com a sua prática. Foi observado que o BDD ainda tem bastante espaço e potencial para crescer no mercado de Tecnologia da Informação, tanto em popularidade quanto em aplicabilidade. Por outro lado, não foi possível apontar com alto grau de precisão se sua adoção no projeto demanda um esforço extra significativo por parte da equipe, o que fatalmente interferiria de forma significativa na sua adoção.

Em relação ao questionamento central deste trabalho, a pesquisa apontou indícios de queo BDD pode ser tido como metodologia neutra alternativa em contextos de gestão e desenvolvimento híbridos, pois além de ser percebido como uma boa prática cultural no projeto, têm significativa eficiência nolevantamento de requisitos, no refinamento e na percepção do produto. O desenvolvimento orientado por comportamento ultrapassa as barreiras caricatas das metodologias ágeis e pode ser absorvido pelas equipes como uma prática independente e não polarizada entre os modelosde desenvolvimento tradicional e ágil.

Ainda, pode-se estimar que o BDD seja igualmente interessante em contextos de transiçãoentre modelos tradicionais e modelos ágeis, já que a sua adoção abarca práticas de melhoria contínua interessantes e nativas do modelo ágil, mas as introduz de forma flexível e suave, sem conflitar diretamente com as práticas do modelo tradicional. Dessa forma, o BDD se estabelece como uma prática qualitativa dentro dos projetos, orientado as equipes para boas práticas de desenvolvimento de *software*.

Referências

[1] **JIDOKA**, Lean Institute Brasil, Disponível em: https://www.lean.org.br/artigos/102/jidoka.aspx#:~:text=O%20Just%20in%20time%20relaciona,ao%20aspecto%20qualitativo%20do%20sistema. Acesso em: 23 de Set. de 2022

[2] **O que é Lean - Definição e Aplicações**, Disponível em: https://www.lean.org.br/o-que-e-lean.aspx. Acesso em: 26 de Set. de 2022

[3] CAROLI, P. **O que é Kanban?**, Disponível em: https://caroli.org/sobre-o-kanban/. Acesso em: 26 de Set. de 2022

[4] CAROLI, P. **eXtreme Programming**, Disponível em: https://caroli.org/extreme-programming-xp/. Acesso em: 26 de Set. de 2022

[5] CAROLI, P. **O que é Scrum?**, Disponível em https://caroli.org/sobre-o-scrum/. Acesso em: 26 de Set. de 2022

[6] CEZERINO, A., NASCIMENTO, FERNANDO P., **Utilização da técnica de desenvolvimento orientado por comportamento (bdd) no levantamento de requisitos**. Disponível em: https://web.archive.org/web/20180412210832id_/http://rica.unibes.com.br/index.php/rica/article/viewFile/728/609. Acesso em: 9 de Out. de 2022

[7] **5 BDD Restinga Frameworks to Consider**, Perfecto. Disponível em https://www.perfecto.io/blog/bdd-testing-frameworks. Acesso em: 12 de Out. de 2022

[8] WANDERLEY, F., DA SILVERIA, SILVA D., **A Framework to Diminish the Gap between the Business Specialist and the software Designer**. Disponível em: https://ieeexplore.ieee.org/abstract/document/6511809. Acesso em: 12 de Out. de 2022

[9] PATKAR, N., CHIŞ, A., STULOVA, N., NIERSTRASZ, O., **Interactive Behavior-driven Development: a Low-code Perspective.** Disponível em: https://ieeexplore.ieee.org/abstract/document/9643783/authors#authors. Acesso em: 12 de Out. de 2022

[10] ZAMPETTI, F., DI SORBO, A., AARON VISAGGIO, C., CANFORA, G., DI PENTA, M., **Demystifying the adoption of behavior-driven development in open source projects**. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S095058492030063X. Acesso em: 12 de Out. de 2022

APÊNDICE A

Questionário aplicado para validação debenefícios e *trade-offs* da prática do BDD

1. **Há quanto tempo você trabalha com TI?**
2. **Você conhece ou pratica BDD?**
3. **Qual é o seu nível de entendimento sobre BDD?**
4. **Dentre os *frameworks* de BDD abaixo, qual(s) você conhece?**
5. **Do ponto de vista do entendimento de negócio do cliente, você acredita que aprática do BDD ajuda a elucidar (aumentar o conhecimento) os objetivos do negócio?**
6. **Para você, a adoção do BDD no processo de desenvolvimento aumenta o esforço da equipe?**
7. **Em relação à comunicação entre as áreas do desenvolvimento (Back-end, Front- end, QA, UX...), você acredita que o BDD estimula a colaboração e o aprendizadodentro do projeto?**
8. **A prática do BDD colabora para a remoção de impedimentos técnicos durante asiterações?**
9. **Na sua percepção, a implementação do BDD reforça positivamente a adoção de práticas ágeis no projeto?**
10. **Fique à vontade para adicionar alguma consideração sobre o BDD ou sobre qualquer uma das perguntas deste questionário.**