

GIOVANNI VICENTE GENTILE DA SILVA

Utilização de *Behaviour-Driven Development* para apoio à modelagem
de requisitos e Gestão das Comunicações

São Paulo
2021

GIOVANNI VICENTE GENTILE DA SILVA

Utilização de *Behaviour-Driven Development* para apoio à modelagem
de requisitos e Gestão das Comunicações

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof^a. Dr^a. Jussara Pimenta Matos

São Paulo
2021

[verso da folha de rosto]

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

FICHA CATALOGRÁFICA

[Colocar na versão final do trabalho. Obter em:

<https://www.poli.usp.br/bibliotecas/servicos/catalogacao-na-publicacao>]

Nome: [SOBRENOME, Nome do aluno]

Título: [Título da monografia]

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Aprovado em:

Banca Examinadora

Prof(a). Dr(a). _____

Instituição: _____

Julgamento: _____

Prof(a). Dr(a). _____

Instituição: _____

Julgamento: _____

Prof(a). Dr(a). _____

Instituição: _____

Julgamento: _____

DEDICATÓRIA

*Em memória de meus avós Giovanni
e Isabella.*

AGRADECIMENTOS

À minha esposa Ana Caroline, por todo o amor, companheirismo, compreensão e incentivo dados ao longo deste trabalho, transformando até mesmo os momentos mais difíceis em algo leve.

À meus familiares, em especial minha mãe Antonia pelo carinho, apoio incondicional e sacrifícios feitos por mim e meu irmão de consideração Fernando Parolini.

À minha orientadora Jussara Pimenta Matos, por toda a generosidade, paciência e direcionamentos que contribuíram muito para o esclarecimento de questões complexas e o andamento do trabalho.

Ao coach de metodologias ágeis e amigo Magno Mendonça, por ter me apresentado o BDD e me apoiado com materiais e conversas de alto nível.

Aos meus amigos da squad de Comprovação, que gentilmente aceitaram participar de todos os experimentos realizados com o BDD que eu propus, além de me ajudarem muito no dia a dia e serem minha referência pessoal de uma equipe de alta performance.

Aos professores e colegas do PECE por todos os ensinamentos e experiências compartilhadas ao longo destes 2 anos.

RESUMO

[SOBRENOME, Nome abreviado]. [**Título da monografia**]. [Ano]. [Número de folhas]. Monografia (MBA em Tecnologia de Software). Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo. São Paulo. [Ano].

[Deve conter motivações, objetivos, métodos empregados, resultados e conclusões em 1 parágrafo]

Palavras-chave: [bla, bla, bla, bla, bla]

ABSTRACT

[SOBRENOME, Nome abreviado]. [**Título da monografia**]. [Ano]. [Número de folhas]. Monografia (MBA em Tecnologia de Software). Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo. São Paulo. [Ano].

[Should have motivations, objectives, applied methods, results and conclusions in 1 paragraph]

Keywords: [bla, bla, bla, bla, bla]

LISTA DE ILUSTRAÇÕES

Figura 1 – Elementos presentes no processo de comunicação.....	21
Figura 2 – Fluxo do Planejamento da Gestão das Comunicações.....	22
Figura 3 – Fluxo do Gerenciamento das Comunicações.....	23
Figura 4 – Fluxo de Monitoração do Gerenciamento das Comunicações.....	24
Figura 5 – Fases de um projeto utilizando XP.....	28
Figura 6 – Ciclo de Propriedade Coletiva de Código no XP.....	29
Figura 7 - Ciclo do Test-Driven Development.....	30
Figura 8 – Fluxo de processos para validar realização das práticas sugeridas.....	41
Figura 9 – Cenário BDD de listagem.....	44
Figura 10 – Cenário BDD de transferência intrabancária.....	45
Figura 11 – Cenário BDD de transferência interbancária.....	45
Figura 12 – Cenário BDD de listagem após reunião de alinhamento com envolvidos.....	46
Figura 13 – Cenário BDD de transferência intrabancária após reunião de alinhamento com envolvidos.....	46
Figura 14 – Cenário BDD de transferência interbancária após reunião de alinhamento com envolvidos.....	47

LISTA DE TABELAS

Tabela 1 – Práticas propostas para refinamento dos requisitos e Gestão das Comunicações	39
Tabela 2 – Questionário de validação	42
Tabela 3 – Lista de verificação do cenário de lista	54
Tabela 4 – Lista de verificação do cenário de transferência intrabancária	55
Tabela 5 – Lista de verificação do cenário de transferência interbancária	56

LISTA DE ABREVIATURAS E SIGLAS

PMBOK® - *Project Management Body of Knowledge*

BDD - *Behaviour-Driven Development*

CI – *Continuous Integration*

CD – *Continuous Deploy*

TDD - *Test-Driven Development*

DDD - *Domain-Driven Design*

PMI - *Project Management Institute*

XP - *Extreme Programming*

PO – *Product Owner*

QA – *Quality Assurance*

BPMN - *Business Process Model and Notation*

REST - *Representational State Transfer*

SUMÁRIO

1. INTRODUÇÃO	13
1.1 Motivações	13
1.2 Objetivo	15
1.3 Justificativas	15
1.4 Método de Pesquisa	17
1.5 Estrutura do Trabalho	17
2. REVISÃO BIBLIOGRÁFICA	19
2.1 Considerações Iniciais	19
2.2 Engenharia de Requisitos	19
2.3 Gestão das Comunicações	21
2.3.1 O processo de comunicação	21
2.3.2 Fluxos de Gestão das Comunicações	21
2.3.3 Gestão das Comunicações em ambientes ágeis ou adaptativos	25
2.4 Metodologias Ágeis	25
2.4.1 Refinamento de Requisitos em Metodologias Ágeis	26
2.5 Extreme Programming	27
2.6 Test-Driven Development	29
2.7 Behaviour-Driven Development	31
2.8 Considerações do capítulo	35
3. PROPOSTA DE PRÁTICAS COM BDD	37
3.1 Proposta de práticas	37
3.2 Definição das questões para avaliar a influência das práticas	42
4 ESTUDO DE CASO E ANÁLISE DE RESULTADOS	43
4.1 Aplicação das práticas em um projeto de software	43
4.2 Avaliação da influência das práticas na finalização do projeto	47
5. CONSIDERAÇÕES FINAIS	48
REFERÊNCIAS	49
APÊNDICE A – Listas de Verificação utilizadas no estudo de caso	53

1. INTRODUÇÃO

Este capítulo apresenta as motivações, objetivo, justificativas, método de pesquisa e a estrutura do trabalho.

1.1 Motivações

No contexto de transformação digital que o mercado de tecnologia está vivenciando, a busca pela excelência na entrega dos *softwares* com qualidade, ou seja, realizando-a de acordo com o prazo e o custo, além de gerar valor dentro das expectativas do negócio, com centralidade no cliente e adaptabilidade a mudanças se faz cada vez mais necessária, e é preciso adotar técnicas e práticas para atingimento destes objetivos.

Dentre os fatores que podem influenciar de forma desfavorável os resultados de um projeto e o atingimento dos objetivos, estão os problemas relacionados ao entendimento dos requisitos e dificuldades nas comunicações (VERNER; SAMPSON; CERPA, 2008), principalmente em projetos onde a atuação está distribuída em várias equipes (HINDS; BAILEY, 2003).

Diante deste cenário, há a necessidade de proposição e aplicação de medidas para minimizar os problemas causados por conta destes fatores, validando-as em um projeto real de desenvolvimento de *software*, em um contexto no qual as equipes de desenvolvimento são impactadas por estes problemas.

Duas práticas podem ser aplicadas para mitigar estes impactos: o BDD (*Behaviour-Driven Development*; em português, Desenvolvimento Dirigido a Comportamentos) para especificação de requisitos e a Gestão das Comunicações conforme proposto pelo PMBOK®.

Behaviour-Driven Development consiste em um conjunto de práticas, tipicamente aplicadas em um contexto de desenvolvimento ágil, pois foi oriundo do *Test-Driven Development* que é um conceito introduzido pelos métodos ágeis (OLIVEIRA, 2015).

O BDD procura criar uma linguagem ubíqua (ou seja, comum) através de uma sintaxe conhecida como *Gherkin* (WYNNE; HELLESØY, 2017), de modo que todos os envolvidos tenham o mesmo entendimento sobre os conceitos tratados, além facilitar a compreensão de quais critérios precisam ser satisfeitos para atingimento dos objetivos do projeto, visto que busca descrever o comportamento esperado do *software* (NORTH & ASSOCIATES, 2006) aumentando a eficiência das comunicações, principalmente entre as equipes técnicas e equipes que representam o negócio, bem como a probabilidade de sucesso dos projetos (HÄSER *et al.*, 2016).

O uso do BDD como apoio às integrações e entregas contínuas (CI e CD) já foi abordado em outro trabalho (OLIVEIRA, 2015), de modo que o presente trabalho tem como uma das motivações complementar as abordagens compartilhadas e também apresentá-lo como técnica para apoiar a modelagem de requisitos e comunicações.

O PMBOK® é um guia de boas práticas para gerenciamento de projetos, dentre as disciplinas contidas no guia está a Gestão de Comunicações, que visa garantir a geração, coleta, armazenamento, recuperação e destinação das informações geradas ao longo do projeto, de modo a procurar minimizar problemas relacionados à quaisquer ruídos nestas comunicações.

Comunicação não abrange somente a transmissão de uma mensagem, mas também a compreensão do significado desta mensagem entre os envolvidos, portanto o BDD pode se tornar um importante aliado, principalmente para melhorar a comunicação entre as equipes durante a identificação e definição de requisitos e de seus critérios de aceite, de modo a contribuir com o andamento e entrega do projeto e diminuir os problemas relacionados ao entendimento do sistema.

A utilização do BDD, aliado à Gestão das Comunicações, busca o atingimento de um entendimento mútuo sobre os comportamentos (ou seja, requisitos) que o *software* a ser entregue precisa apresentar bem como a correta disseminação deste entendimento entre as partes envolvidas, de modo a mitigar os riscos de falhas na entrega dos projetos.

1.2 Objetivo

Propor a utilização das práticas de BDD para definir o comportamento e critérios de aceite do *software* (de acordo com a abordagem de Engenharia de Requisitos), em conjunto com as definições de Gestão das Comunicações do PMBOK®, para avaliar a influência desta abordagem na diminuição de ruídos nas comunicações e compreensão dos requisitos bem como o seu auxílio na finalização bem-sucedida de um projeto com envolvimento de múltiplas equipes.

A partir desta proposta, é definida uma lista de verificação de práticas, que busca alinhar as definições dos comportamentos do sistema em uma linguagem ubíqua com o controle das comunicações do projeto, de modo a assegurar que as informações e definições sejam transmitidas de forma clara e atualizada para as partes interessadas.

Cada item desta lista deve ser cumprido e verificado ao longo da realização do trabalho, para posterior avaliação da relevância na utilização destas práticas no resultado final.

1.3 Justificativas

De acordo com o estudo de (VERNER; SAMPSON; CERPA, 2008), que avaliou o andamento de oito (8) projetos de desenvolvimento de *software* que não obtiveram sucesso, decisões tomadas sem o entendimento pleno dos requisitos do sistema estiveram entre os fatores responsáveis pela falha de 73% dos projetos avaliados.

Um artigo publicado pelo PMI (PMI, 2013) relata que mais da metade do orçamento de um projeto entra em risco logo em seu início devido à comunicações ineficientes, o que mostra que estes ruídos acarretam em maiores custos e podem fazer com que um projeto falhe ou seja entregue fora do orçamento e até mesmo prazo.

Estes problemas não são novos em desenvolvimento de software, inclusive foram citados por Frederick Brooks em seu livro *The Mythical Man-Month*, que possui um capítulo destacando que adicionar mais pessoas a um projeto atrasado tende a atrasá-lo ainda mais, pois dentre outros fatores, adicionam novos canais de

comunicação que se não forem gerenciados podem implicar em mais problemas do que soluções (BROOKS JR, 1995).

Com o advento das metodologias ágeis, a importância de realizar as comunicações de forma efetiva foi muito enfatizada e gerou uma vasta literatura (MAZUCA, 2018), reforçando a relevância deste tema.

Outro fator que reforça a necessidade de explorarmos estes assuntos, é o de que em projetos de grande porte, há o envolvimento de diferentes equipes, que por vezes estão distribuídos, e consequentemente existem problemas de comunicação por conta desta homogeneidade que faz com que existam muitos canais de comunicação para gerenciar (HINDS; BAILEY, 2003). Devemos levar em consideração que não adianta mapear todos os canais de comunicação e atualizá-los com frequência se as documentações ou mensagens compartilhadas não estão corretas e alinhadas com os reais objetivos e necessidades do projeto, portanto a importância de especificar os requisitos corretamente e reforçar o entendimento sobre eles entre os *stakeholders* é muito grande.

Estes pontos também contribuem para uma maior incidência de conflitos, tipicamente causados por expectativas desalinhadas entre as equipes envolvidas em projetos nestes contextos (WILLIAMS; O'REILLY, 1998), evidenciando também oportunidades de melhorias na modelagem dos requisitos e no alinhamento sobre eles (HINDS; BAILEY, 2003).

As principais causas que resultam em problemas na modelagem de requisitos são: falta de clareza sobre os objetivos a serem atingidos ou qual problema é necessário resolver; gastos com funcionalidades nunca utilizadas ou desnecessárias; complexidade desnecessária da especificação (BITTNER, 2008). Estes fatores vão a encontro aos impactos causados e apresentados no estudo referenciado no primeiro parágrafo (VERNER; SAMPSON; CERPA, 2008).

Os estudos e artigos citados reforçam os impactos que ocorrem quando não há a preocupação em modelar os requisitos e critérios de aceitação do *software* de um modo que o entendimento sobre o que precisa ser feito é comum aos *stakeholders*, e

também quando não há uma gestão efetiva das comunicações. Isso corrobora com a importância do tema e da proposta de práticas para minimizar o risco destes impactos acontecerem.

1.4 Metodologia

A metodologia utilizada neste trabalho é uma pesquisa de natureza qualitativa, realizada após a verificação de uma lista de práticas em um projeto real de desenvolvimento de *software*.

Esta pesquisa visa avaliar se, através dos processos e práticas propostas, de fato houve menor incidência de problemas de entendimento sobre o que precisava ser feitos e conflitos causados por conta desse fator, bem como qual a influência destes processos e práticas nos resultados do projeto.

Para o desenvolvimento deste trabalho, as seguintes etapas foram realizadas:

1. Definição do escopo do trabalho e motivações para sua realização.
2. Levantamento bibliográfico e estudo de material relacionado à prática de definição de comportamentos (requisitos) esperados utilizando BDD e notação *Gherkin*, verificando como isso auxilia na comunicação entre as partes envolvidas no projeto, bem como estudo do tópico de Gestão das Comunicações do Projeto presente no PMBOK®.
3. Criação de uma lista com práticas de Gestão das Comunicações, considerando documentos de requisitos e plano de testes no formato proposto pelo BDD.
4. Aplicação dos itens desta lista em um projeto real de desenvolvimento de *software*.
5. Avaliação da influência das práticas na finalização do projeto mediante um questionário.

1.5 Estrutura do Trabalho

O Capítulo 1 INTRODUÇÃO apresenta as motivações, o objetivo, as justificativas, método de pesquisa e a estrutura do trabalho.

O Capítulo 2 REVISÃO BIBLIOGRÁFICA apresenta os principais conceitos relacionados ao trabalho, que são Metodologias Ágeis, XP, TDD, *Behaviour-Driven Development* e Gestão das Comunicações dentro do proposto pelo PMBOK®.

O Capítulo 3 PROPOSTA PRÁTICAS COM BDD apresenta uma lista de práticas, focadas em refinamento de requisitos mediante definição de comportamentos com BDD e gestão das comunicações.

O Capítulo 4 ESTUDO DE CASO E ANÁLISE DE RESULTADOS irá verificar, através de um estudo de caso em um projeto real de *software* e questionário aplicado aos participantes deste estudo, se as práticas propostas contribuíram para uma especificação de requisitos e gestão das comunicações eficientes. O *software* entregue também será parâmetro para avaliar os resultados do trabalho, sendo que o principal critério a ser considerado é a funcionalidade entregue, ou seja, se o *software* faz aquilo que o usuário realmente precisa.

O Capítulo 5 CONSIDERAÇÕES FINAIS descreve a conclusão, contribuições e propostas para trabalhos futuros.

REFERÊNCIAS contém os artigos, livros, monografias e documentos utilizados na pesquisa realizada para elaboração deste trabalho.

2. REVISÃO BIBLIOGRÁFICA

2.1 Considerações Iniciais

Este capítulo fornece as principais referenciais teóricos aplicáveis ao desenvolvimento deste trabalho.

Inicialmente, são apresentados os conceitos de Engenharia de Requisitos, principalmente no que diz respeito à modelagem de requisitos e processos de comunicação, de modo a relacionar este item com os demais pontos abordados no trabalho.

Em seguida, serão tratados os principais pontos da área de conhecimento de Gestão das Comunicações do PMBOK®, para posteriormente relacionar suas práticas com as técnicas de modelagem de requisitos utilizando BDD, e verificar como essas duas abordagens podem se complementar para facilitar as comunicações e consequentemente o sucesso de um projeto.

Posteriormente, são tratados os processos e técnicas que deram origem ao BDD, destacando-se: Metodologias Ágeis, *Extreme Programming*, que será abordado devido à sua relação com as práticas de teste unitário que posteriormente foram evoluídas para o BDD, além do próprio *Test-Driven Development*.

Por fim, serão compartilhadas as definições e conceitos do BDD, com foco na definição de requisitos e critérios de aceite com utilização de uma linguagem ubíqua e apoio nas comunicações.

2.2 Engenharia de Requisitos

A medida primária de sucesso de um sistema de *software* é o atingimento dos propósitos e objetivos de seus usuários (NUSEIBEH; EASTERBROOK, 2000), que são expressados através de seus requisitos.

A Engenharia de Requisitos é um processo da Engenharia de *Software* que diz respeito à objetivos, funções e restrições de sistemas de software, aliado à definição de comportamentos (funcionais e não-funcionais) do sistema e sua evolução ao longo do tempo (ZAVE, 1997).

As principais atividades da Engenharia de Requisitos são (NUSEIBEH; EASTERBROOK, 2000):

- Elicitação, ou seja, o processo de coleta dos requisitos
- Análise e Modelagem
- Comunicação dos requisitos
- Acordo sobre os requisitos gerados
- Manutenção e Evolução dos Requisitos

Dentre essas atividades, iremos enfatizar as que têm maior relação com o tema, isto é: Modelagem e Comunicação dos Requisitos.

Modelagem de requisitos implica na construção de descrições, modelos e abstrações que facilitem a interpretação sobre quais requisitos precisam ser atendidos. Dentre as categorias de modelagem de requisitos, existe a comportamental, que implica em modelar os comportamentos atuais (se houverem) e futuros do sistema e de seus *stakeholders* (NUSEIBEH; EASTERBROOK, 2000). Este tipo de modelagem tem forte relação com o que foi proposta posteriormente na concepção do BDD, de que requisitos também são comportamentos (NORTH & ASSOCIATES, 2006).

Além de descobrir e especificar as necessidades que o sistema precisa atender, a Engenharia de Requisitos precisa ser um meio para facilitar as comunicações do projeto (NUSEIBEH; EASTERBROOK, 2000), principalmente através de sua rastreabilidade, de modo que o ciclo de vida do requisito seja facilmente identificado e disseminado (GOTEL; FINKELSTEIN, 1994). Essa rastreabilidade pode ser potencializada com técnicas da área conhecimento de Gestão das Comunicações proposta pelo PMBOK®.

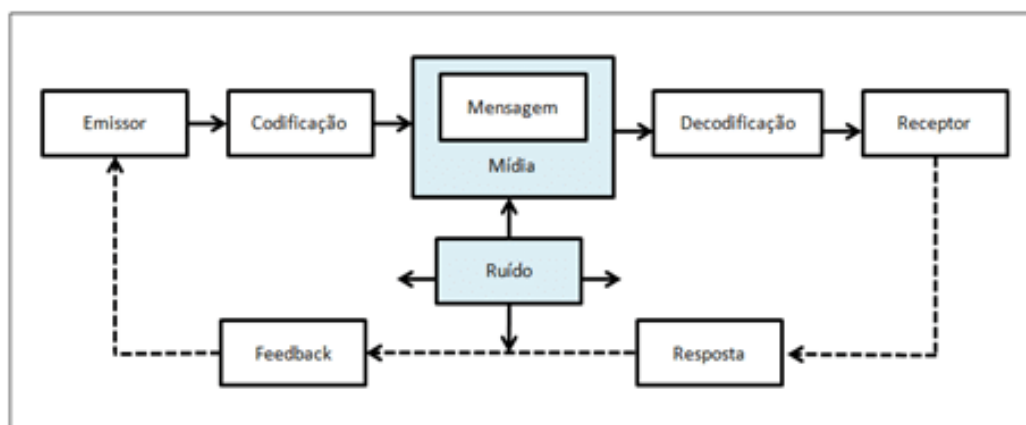
2.3 Gestão das Comunicações

2.3.1 O processo de comunicação

De forma genérica, o conceito de comunicação surgiu do termo em latim *communicare*, que significa “tornar comum” (MAZUCA *apud* HARPER, 2018), cujo objetivo é transmitir um propósito expresso em alguma mensagem ou informação e torná-lo comum entre as partes envolvidas no processo de comunicação. Para tanto, é necessário no mínimo um emissor e um receptor desta mensagem.

A mensagem é codificada por seu emissor para algum formato textual ou simbólico, e é transmitida através de alguma mídia (meio de comunicação) até o receptor, que realiza o entendimento (decodificação) da mensagem iniciada pelo emissor e gera um feedback deste entendimento ao receptor. Neste processo, também podem surgir os ruídos, que são quaisquer interferências que comprometam o entendimento do receptor da mensagem ou mesmo a produção da mesma por seu emissor (SONDERMANN *et. al.*, 2013). A figura 1 apresenta estes elementos do processo de comunicação em um diagrama de fluxo de dados.

Figura 1 – Elementos presentes no processo de comunicação



Fonte: SONDERMANN *et. al.* (2013)

2.3.2 Fluxos de Gestão das Comunicações

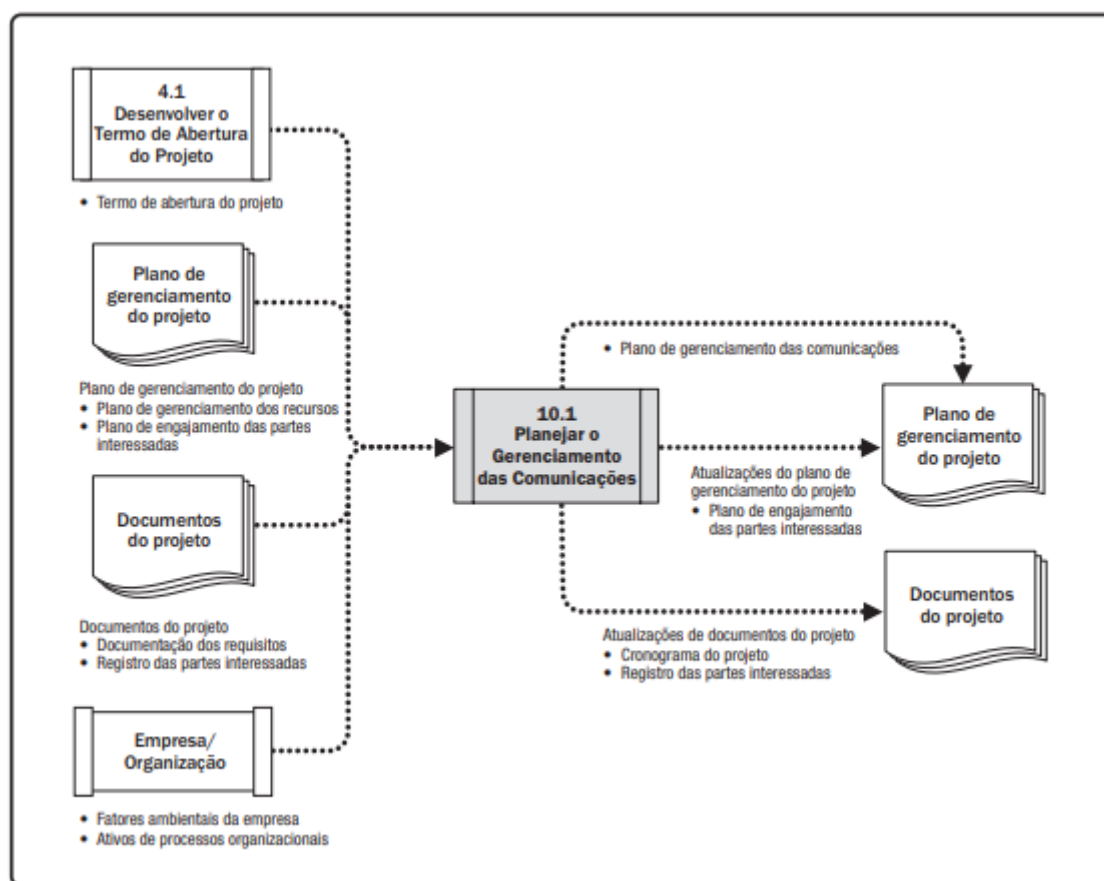
A Gestão das Comunicações é uma das áreas de conhecimento do PMBOK®, que inclui os processos necessários para assegurar que as informações do projeto sejam

planejadas, coletadas, criadas, distribuídas, armazenadas, recuperadas, gerenciadas, controladas, monitoradas e finalmente organizadas de maneira oportuna e apropriada.

Esta área de conhecimento engloba 3 processos:

- **Planejar o Gerenciamento das Comunicações:** desenvolve uma abordagem e um plano adequado para execução das atividades relacionadas à Gestão das Comunicações, com base nas necessidades de informações dos *stakeholders* e do projeto em si. A figura 2 apresenta fluxo do planejamento da Gestão das Comunicações.

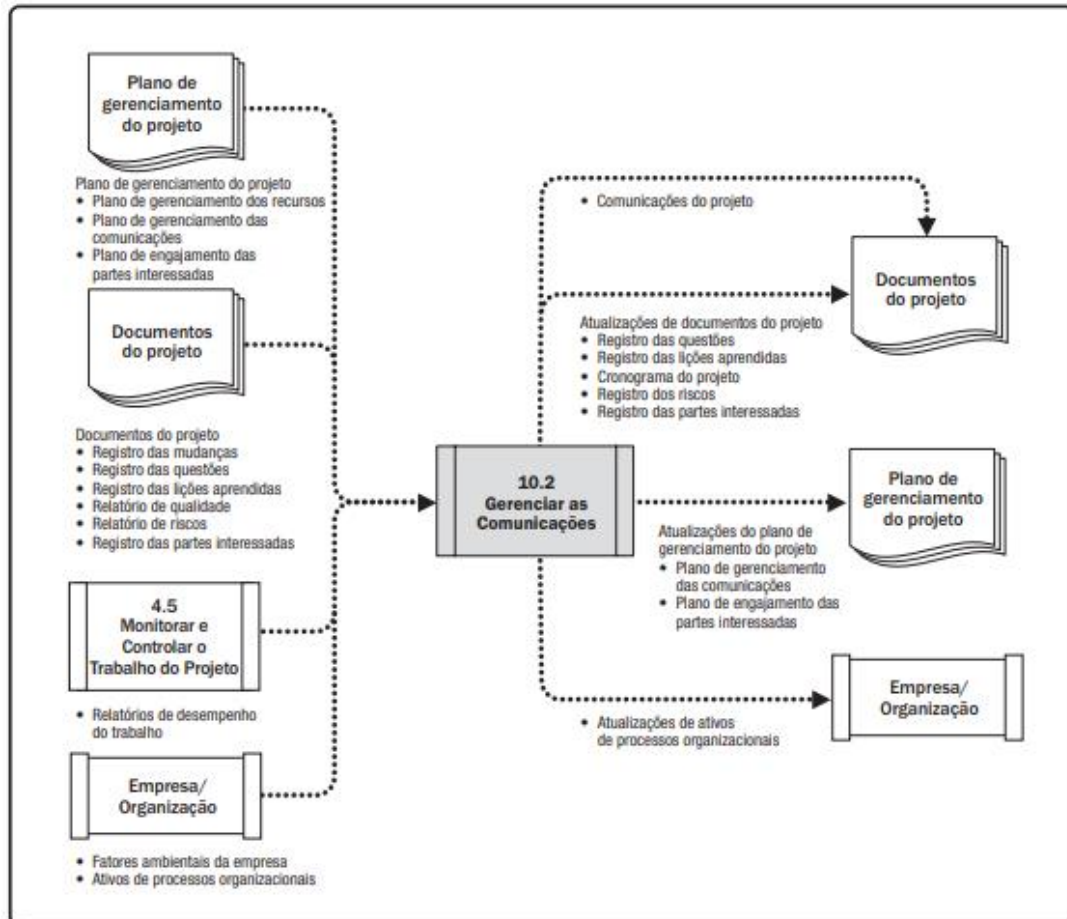
Figura 2 – Fluxo do Planejamento da Gestão das Comunicações



Fonte: PMBOK® (2017)

- **Gerenciar as Comunicações:** assegura a coleta, criação, distribuição, armazenamento, recuperação, gerenciamento e disposição final das informações do projeto. A figura 3 apresenta fluxo do Gerenciamento das Comunicações.

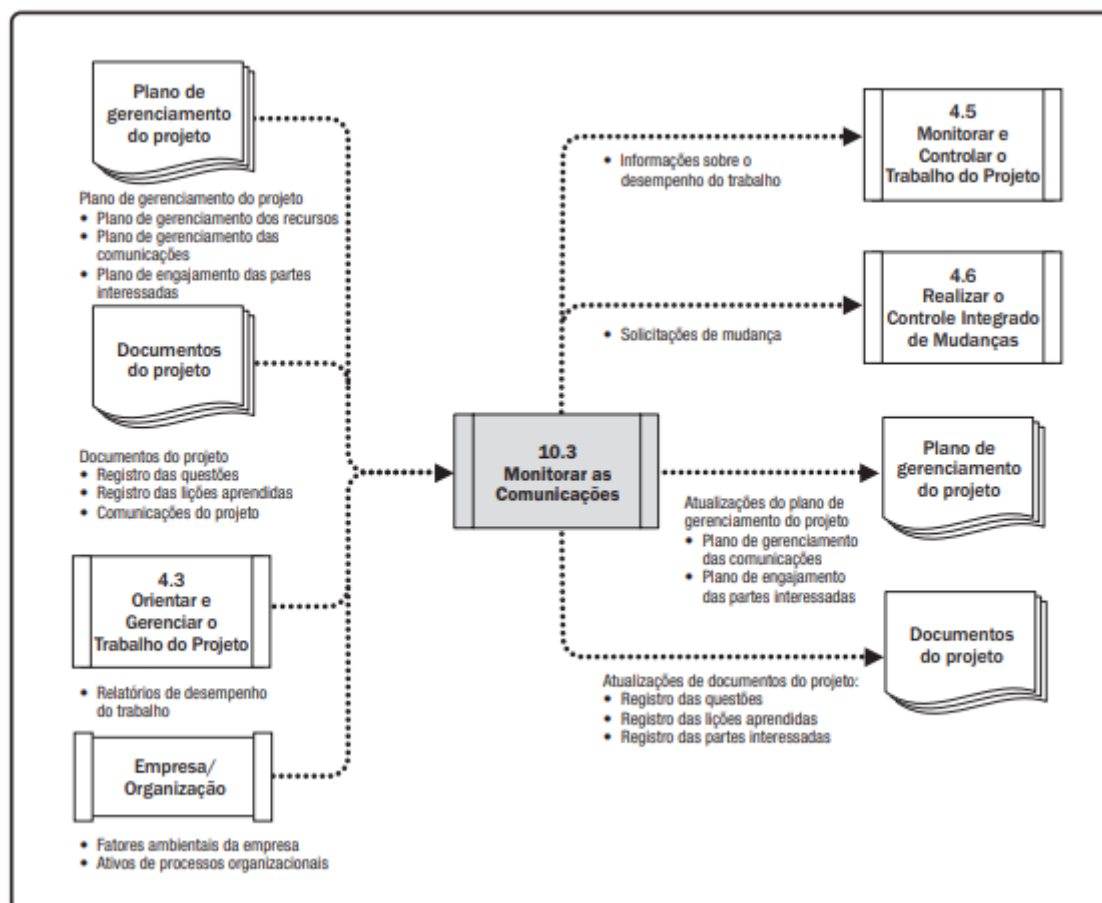
Figura 3 – Fluxo do Gerenciamento das Comunicações



Fonte: PMBOK® (2017)

- **Monitorar as Comunicações:** garante que as necessidades de informação do projeto sejam atendidas. A figura 4 apresenta fluxo de Monitoração do Gerenciamento das Comunicações.

Figura 4 - Fluxo de Monitoração do Gerenciamento das Comunicações



Fonte: PMBOK® (2017)

Ainda segundo o PMBOK®, a informação pode ser transmitida de forma escrita, falada, de maneira formal (documentos formais da organização) ou informal (mídias sociais ou aplicativos de troca de mensagens) e através de mídias (imagens e vídeos, por exemplo). Para a comunicação de forma escrita, uma das mais típicas, o guia recomenda o uso dos 5 C's:

- **Correta:** evitar o uso deficiente da ortografia e gramática, pois isso pode causar distrações e diminuir a credibilidade da mensagem.
- **Concisa:** redigir a mensagem de forma eficiente, para evitar ambiguidades que possam comprometer o entendimento.

- **Clara:** garantir um propósito claro e direcionado às necessidades do receptor.
- **Coerente:** utilizar um fluxo lógico, coerente e fluído de ideias.
- **Controlada:** controlar o fluxo de palavras, se utilizar de imagens e resumos, quando aplicável.

Outras habilidades em comunicação podem potencializar o efeito de mensagens escritas considerando os 5 C's, como: escuta ativa; consciência de diferenças sociais e pessoais entre os envolvidos e identificação, definição e gerenciamento de expectativas das partes interessadas. Também podemos nos valer de práticas consideradas tendências na Gestão das Comunicações para minimizar os ruídos ao longo do projeto, como a inclusão das partes interessadas em revisões e reuniões de projetos.

2.3.3 Gestão das Comunicações em ambientes ágeis ou adaptativos

Em ambientes dinâmicos e fortemente sujeitos a mudanças (o que é uma tendência nos projetos de *software*), há o reforço de que existe uma grande necessidade de se realizar as comunicações com eficiência e rapidez. Com isso, a recomendação do PMBOK® é a dinamização do acesso às informações, pontos de verificação frequentes e, se possível, agrupamento dos membros da equipe / projeto no mesmo local. Também há a necessidade de publicar as documentações do projeto de forma transparente e realizar constantes revisões com as partes interessadas.

2.4 Metodologias Ágeis

Os fundamentos das metodologias ágeis surgiram no Manifesto Ágil (BECK *et al.*, 2001), que compartilha os valores e princípios destas metodologias, sendo uma alternativa às abordagens tradicionais utilizadas na época.

Apesar de representarem uma resposta à burocracia e baixa adaptabilidade à mudanças de modelos como o Cascata, estas metodologias envolvem processos disciplinados e o sucesso na adoção das mesmas estão relacionados aos fatores abaixo (PAULK, 2002):

1. Aplicar as metodologias com fidelidade às filosofias e princípios do desenvolvimento ágil.
2. Implementar a metodologia de forma adequada ao ambiente da organização e características da aplicação a ser desenvolvida.

Através da identificação e compilação de melhores práticas de desenvolvimento que utilizavam, e a fim de colaborar com outros profissionais na tarefa de desenvolver *software* (OLIVEIRA *apud* BECK *et al.*, 2015), os autores apresentaram os seguintes valores (BECK *et al.*, 2001):

1. Indivíduos e interações, mais que processos e ferramentas.
2. *Software* em funcionamento, mais que documentação abrangente.
3. Colaboração com o cliente, mais que negociação de contratos.
4. Responder a mudanças, mais que seguir um plano.

De acordo com os autores, os itens tratados no início de cada frase devem ser mais considerados e tratados com maior prioridade do que os itens do final de cada frase. Isso não implica que os valores à direita não tenham importância, mas direciona as prioridades sobre o que deve ser mais valorizado.

Os autores também apresentaram os princípios que estão por trás do Manifesto Ágil e que são seguidos por eles (BECK *et al.*, 2001), que estão presentes em *frameworks* consolidados como o Scrum (SCHWABER; SUTHERLAND, 2017) e metodologias como o XP (*Extreme Programming*).

2.4.1 Refinamento de Requisitos em Metodologias Ágeis

Dentre as práticas e rituais propostos em Metodologias Ágeis, estão a reunião de refinamento de requisitos, que justamente tem o propósito de especificar e modelar os requisitos elicitados.

O termo “refinamento” pode ser interpretado como a fase de análise e projeto (design) em abordagens mais tradicionais, e equivale ao *Refinement Meeting* (antigo Backlog Grooming) no Scrum, podendo ser incorporado à outras metodologias ágeis (BALDINI, 2019), que consiste em uma reunião que tem como objetivo reforçar o

entendimento dos requisitos priorizados e documentá-los de forma que os mesmos estejam prontos para serem desenvolvidos (SEDANO; RAPLH; PÉRARE, 2019).

2.5 *Extreme Programming*

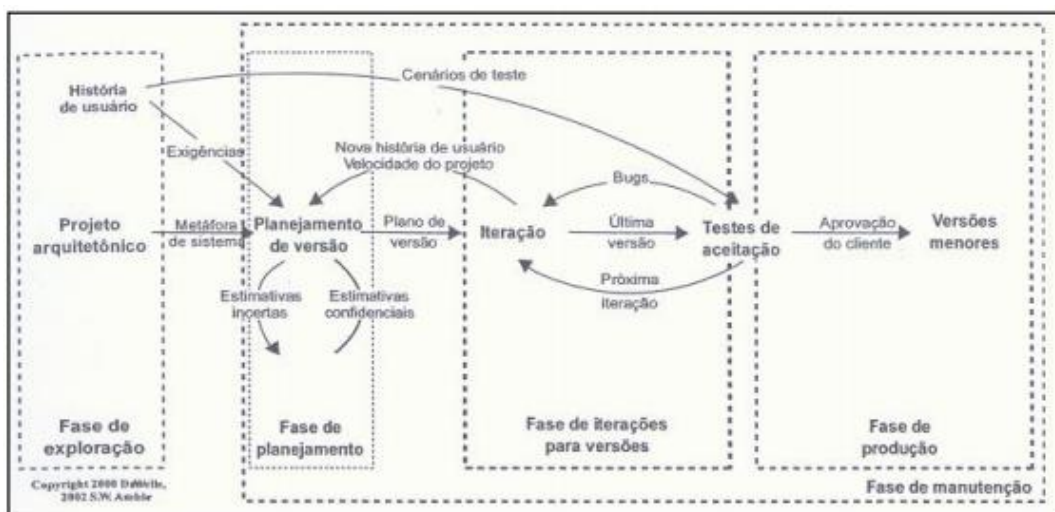
O *Extreme Programming* (XP), introduzido por Kent Beck, é um conjunto de práticas de desenvolvimento ágil de *software* que enfatiza o trabalho colaborativo (WELLS, 1999) com o objetivo de se obter um produto de maior qualidade e aderente à necessidade dos clientes (OLIVEIRA *apud* BECK, 2015).

Os valores do XP (comunicação, simplicidade, *feedback*, respeito e coragem), incentivam a colaboração entre desenvolvedores, equipes de negócio e demais envolvidos, promovendo um ambiente de entregas contínuas, transparência e rápida adaptabilidade a mudanças (WELLS, 2000), abaixo uma breve descrição destes valores (BECK, 2000):

- **Comunicação:** uma comunicação precisa, objetiva e oportuna ajuda os envolvidos a entenderem quais são os objetivos do projeto e como alcançá-los.
- **Simplicidade:** almejar sempre a solução mais simples busca minimizar o impacto causado pelas mudanças e o esforço em realizá-las.
- **Feedback:** obter constantes retornos sobre o funcionamento do *software*, mediante testes unitários, programação em pares e alinhamento com equipes de negócio.
- **Respeito:** é necessário ter escuta ativa e levar em consideração as ideias que são mais simples e que levam ao atingimento dos objetivos traçados.
- **Coragem:** capacidade de aceitar e se adaptar a mudanças.

O ciclo de vida de um projeto utilizando XP passa pelas seguintes fases: Exploração; Planejamento; Iterações para versões; Produção e Manutenção, conforme apresentado na figura 5.

Figura 5 – Fases de um projeto utilizando XP

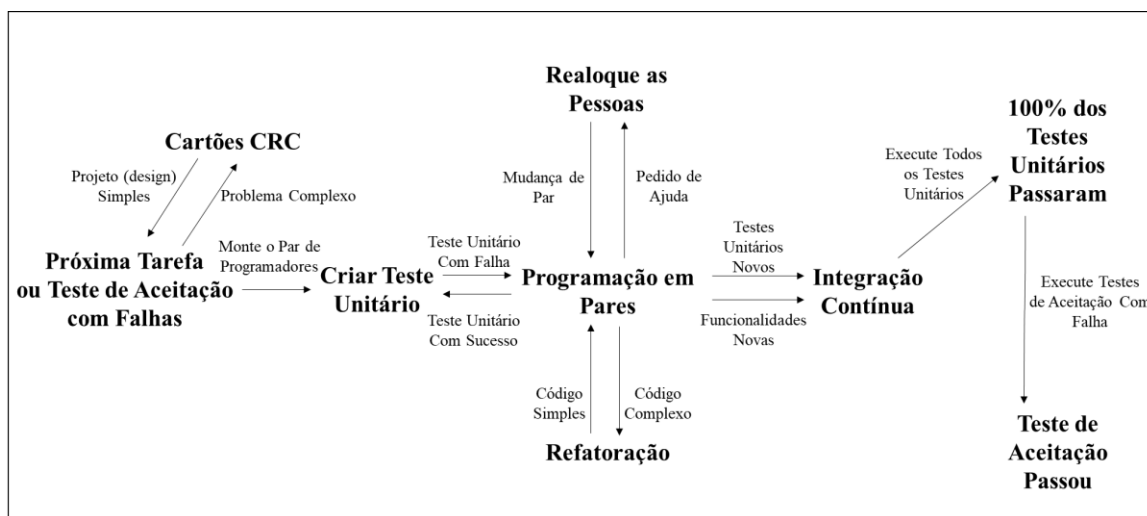


Fonte: HENRAJANI (2007)

Na fase de exploração, há o levantamento das primeiras histórias de usuário e elaboração da versão inicial do projeto (*design*) de arquitetura. Na fase de planejamento, as estimativas são realizadas e as histórias de usuário que serão implementadas na próxima iteração são definidas. Na iteração, uma pequena versão do produto é codificada e testada (a nível de teste unitário e posteriormente teste de aceitação). As funcionalidades testadas e aprovadas pelo cliente são liberadas para a fase de produção, e as rejeitadas geram novas histórias que serão reconsideradas na fase de planejamento e adicionadas em versões futuras do produto. A fase manutenção implica em responder às mudanças que surgirem ao longo do ciclo de vida (HENRAJANI, 2007).

Dentre as práticas introduzidas, estão aquelas relacionadas à qualidade e propriedade coletiva de código (realizadas na fase de iteração), como programação em pares (promovendo revisão do código e aplicação de melhores práticas de desenvolvimento), desenvolvimento dirigido a testes (TDD, no qual os testes unitários são escritos antes da programação da funcionalidade), refatoração de código, integração contínua e testes de aceitação (WELLS, 2000). A figura 6 apresenta um fluxo com estas práticas de propriedade coletiva de código.

Figura 6 – Ciclo de Propriedade Coletiva de Código no XP



Fonte: WELLS (2000), traduzido pelo autor

Das práticas de propriedade coletiva de código promovidas pelo XP, o trabalho irá explorar o TDD, que é o desenvolvimento dirigido por testes unitários.

2.6 Test-Driven Development

O *Test-Driven Development* (TDD; em português, Desenvolvimento Dirigido à Testes) é uma abordagem cíclica de desenvolvimento apresentada por (BECK, 2003) no XP, na qual codificação deve se iniciar pela escrita dos testes, e posteriormente a implementação da funcionalidade que irá fazer aquele teste passar é realizada.

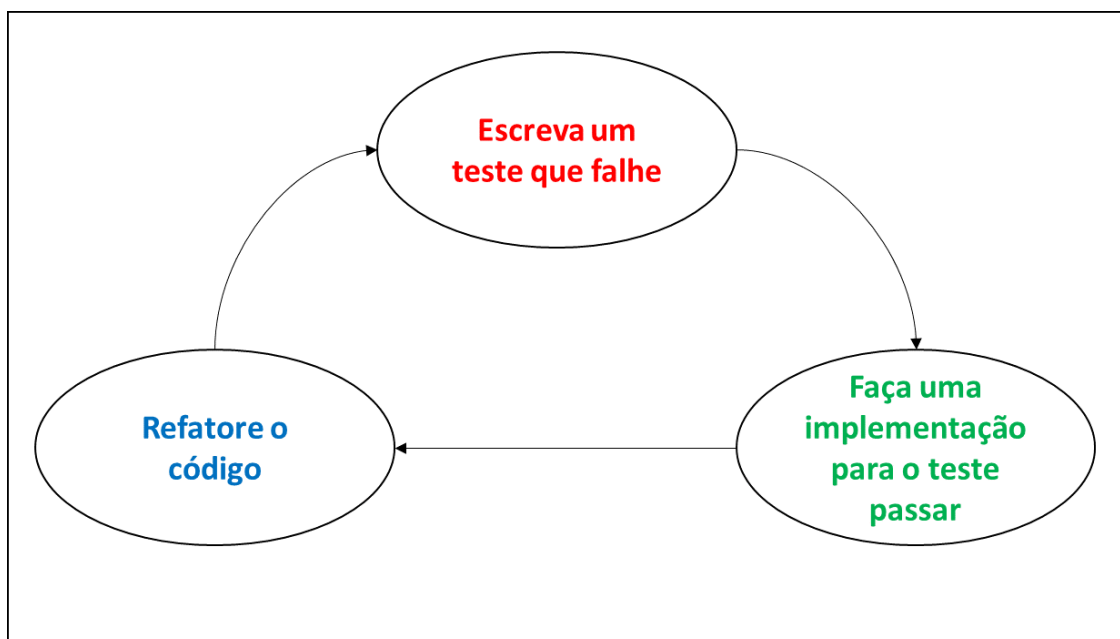
Essa dinâmica visa a obtenção de maior agilidade na entrega das funcionalidades, maior facilidade de manutenção do código e aumento da confiabilidade na solução de modo que os custos com correção de defeitos em produção sejam reduzidos, pois uma base de testes automatizados é gerada e executada a cada nova versão do *software*, antes da publicação para produção.

Apesar destas vantagens, a percepção geral é que o TDD é algo difícil de aprender e aplicar, além da questão contra-intuitiva de que a velocidade na entrega do *software*, ao invés de aumentar, irá diminuir com o surgimento das novas tarefas de escrita de testes no processo. Além disso, a exigência de mais dedicação e disciplina do que nas práticas de desenvolvimento *test-last* mais tradicionais, reforçam a necessidade

de capacitar os profissionais no entendimento e aplicação destas técnicas *test-first* (LATORRE, 2013) para que os benefícios de as utilizar sejam obtidos.

O ciclo do TDD ocorre conforme representado na figura 7.

Figura 7 – Ciclo do Test-Driven Development



Fonte: OLIVEIRA (2015), adaptado pelo Autor

A classificação das tarefas ocorre da seguinte maneira (BECK, 2003):

- **Vermelhas:** escrita e execução de um teste que vai falhar pois o código não está implementado.
- **Verdes:** implementação do código de maneira pragmática para o teste que falhou anteriormente passar.
- **Refatoração:** melhoria do código e evoluções na implementação, com o cuidado de eliminar possíveis duplicidades e más práticas de implementação.

A fim de colaborar com a limpeza dos testes e do código, e para consolidar a aplicação do TDD entre os profissionais de desenvolvimento, Robert C. Martin propôs que a criação dos testes antes da implementação é apenas o primeiro passo para realização desta prática com excelência, e que as três leis do desenvolvimento dirigido à testes devem ser consideradas (MARTIN, 2008):

1. Não se deve escrever o código de produção até criar um teste de unidade de falhas.
2. Não se deve escrever mais de um teste de unidade do que o necessário para falhar, e não compilar é falhar.
3. Não se deve escrever mais códigos de produção do que o necessário para aplicar o teste de falha atual.

Entretanto, existe a percepção de que a aplicação do TDD é complexa, principalmente em seus estágios iniciais, nos quais os desenvolvedores sentem dificuldades na compreensão do que seria a escrita de um teste que falhe.

Uma das respostas à essas dificuldades foi o BDD (NORTH & ASSOCIATES, 2006), que além das abordagens de testes de aceitação, busca apoiar a compreensão sobre os comportamentos esperados do sistema, de modo que seja possível tirar proveito dessa compreensão não somente nas atividades de testes, mas também de refinamento de requisitos.

2.7 Behaviour-Driven Development

O *Behaviour-Driven Development* (BDD; em português Desenvolvimento Dirigido a Comportamentos), é um conjunto de práticas apresentado em um artigo de Dan North publicado em 2006 (NORTH & ASSOCIATES, 2006), no qual o autor propõe uma evolução das práticas do TDD, com o objetivo de minimizar as dificuldades de entendimento dos desenvolvedores sobre o quê deve ser testado, de que forma aquilo deve ser testado ou mesmo porquê um teste falhou.

Por focar na especificação e testes dos comportamentos que o sistema deve apresentar, essas práticas aproximam os desenvolvedores das equipes de negócios e testes, visto que a linguagem utilizada é comum para as partes (ubíqua) e o entendimento do quê deve ser feito e de como o sistema deve se comportar é equalizado. Isso também serve como apoio sobre quais funcionalidades (ou seja, comportamentos) devem ser priorizadas nas entregas de valor do projeto (NORTH & ASSOCIATES, 2006).

Além das boas práticas para escrita de testes, como nomear a assinatura do método de teste com uma frase com o comportamento esperado, no BDD há a proposta de

que requisitos também são comportamentos, com inspiração no conceito de linguagem ubíqua proposta nas abordagens do *Domain-Driven Design* – DDD (EVANS, 2004).

Os princípios do BDD afirmam que as equipes de negócio e tecnologia devem se referir ao sistema, no que diz respeito às suas terminologias e comportamentos, da mesma forma, e que esse entendimento é permitido justamente pela definição de uma linguagem comum entre as partes (LAZĂR; MONTAGNA; PÂRV, 2010). Esta definição permite que:

1. Os requisitos sejam especificados através de uma perspectiva de negócio.
2. Os analistas de negócio definem exemplos concretos a fim de esclarecer o comportamento do sistema.
3. Os desenvolvedores implementem as funcionalidades através de uma abordagem dirigida à testes (ou seja, escrevem os testes primeiro).

A fim de criar uma sintaxe que colabore com o conceito e aplicação de uma linguagem ubíqua, o autor buscou inspiração em um formato de história de usuário utilizado na época (NORTH & ASSOCIATES, 2006), no qual os critérios de aceite estavam incorporados à escrita da história, conforme:

```
Como <ator>  
Quero <alguma funcionalidade>  
Para <atingir um objetivo>
```

Com estas bases, uma estrutura de especificação de funcionalidades foi proposta (NORTH & ASSOCIATES, 2006) em uma sintaxe que depois veio a ser conhecida como *Gherkin* (OLIVEIRA; MARCZAK, 2017).

Uma especificação de um cenário utilizando *Gherkin* tem a seguinte sintaxe (OLIVÉRIO, 2016):

Dado algum contexto inicial (em inglês, “Given”)
 Quando um evento ocorre (em inglês, “When”)
 Então, garanta alguns resultados (em inglês, “Then”)

Além das palavras reservadas “Dado”, “Quando” e “Então”, também está presente na sintaxe o comando de sentenças adicionais “E” (em inglês, “And”), caso seja necessário incluir mais pré-condições, eventos ou pós-condições (ZEN, 2013). Para identificação da funcionalidade e cenários, também são utilizadas palavras reservadas.

Um cenário que pode ser utilizado como exemplo para demonstrar esta sintaxe é o de saque de dinheiro em um caixa eletrônico, conforme proposto por Dan North no artigo que apresentou o BDD (NORTH & ASSOCIATES, 2006). Um fator relevante sobre a escrita de cenários é a de que eles devem contemplar os casos de sucesso e de falhas, dentro de uma mesma funcionalidade. O exemplo compartilhado neste trabalho é o de um caso de sucesso, os exemplos na íntegra estão disponíveis no artigo de North.

Funcionalidade: Cliente saca dinheiro
 Cenário: A conta está com saldo positivo

Dado que a conta que está com saldo positivo,
 E o cartão seja válido,
 E o dispensador contenha dinheiro,

Quando o cliente requisita dinheiro

Então, garanta que a conta seja debitada,
 E garanta que o dinheiro seja dispensado,
 E assegure-se que o cartão seja devolvido.

Além disso, indo ao encontro à definição de que os analistas de negócios devem fornecer exemplos concretos para esclarecimento dos comportamentos do sistema (LAZAR; MONTAGNA; PÂRV, 2010), é possível utilizar tabelas de dados

apresentando possíveis valores e demais informações que possam ser relevantes para o entendimento dos comportamentos esperados (WYNNE; HELLESØY, 2017), em qualquer uma das três expressões principais (Dado, Quando e Então).

Para a escrita dos cenários especificando os comportamentos esperados e posterior automação dos testes de aceitação, recomenda-se o uso de ferramentas como o *Cucumber* (WYNNE; HELLESØY, 2017), que tem suporte à sintaxe *Gherkin*.

Da documentação oficial do *Cucumber* (SMARTBEAR SOFTWARE©, 2019), foi extraído um exemplo do uso de tabela de dados para explicitar valores e comportamentos esperados:

Dado que os seguintes usuários estejam cadastrados no sistema:

Nome	Email	Twitter
Aslak	aslak@cucumber.io	@aslak_hellesoy
Julien	julien@cucumber.io	@jbpros
Matt	matt@cucumber.io	@mattwynne

As próprias recomendações sobre como utilizar o *Cucumber* e seu propósito maior de validar o entendimento mútuo dos envolvidos sobre como o *software* deve se comportar reforçam que as práticas do BDD, antes de qualquer coisa, se referem à alcançar este entendimento, de modo que a automação dos testes seja consequência (HELLESØY, 2015), conforme citação direta de Aslak Hellesøy, criador do *Cucumber*:

Cucumber não é uma ferramenta de testar *software*. É uma ferramenta para testar o entendimento das pessoas em como um *software* (ainda a ser escrito) deve se comportar.

Este entendimento pode ser utilizado para modelar os requisitos funcionais de um *software*, e aliados à alguns conceitos de escrita de histórias de usuário através da técnica INVEST (COHN, 2004), podemos obter uma padronização ainda maior sobre a escrita dos cenários bem como uma avaliação qualitativa dos envolvidos à respeito das especificações.

Inspirado pelo INVEST, um estudo definiu 14 fatores para escrita de cenários em BDD com qualidade, que são: atômico; completo; consistente; conciso; estimável; viável;

independente; negociável; priorizável; pequeno; testável; intelegível; não-ambíguo e valioso (OLIVEIRA; MARCZAK, 2017).

Diante dos conceitos apresentados nesta seção, o enfoque será na utilização do BDD como mecanismo de modelagem dos requisitos para equalizar o entendimento mútuo entre as equipes de desenvolvimento e negócios dos comportamentos desejados do *software*, criando uma base de conhecimento cujo acesso e utilização podem ser amplificados utilizando técnicas de Gestão das Comunicações, que é o próximo tópico abordado neste trabalho.

2.8 Considerações do Capítulo

O BDD tem como seu principal propósito proporcionar um entendimento mútuo entre as equipes de desenvolvimento e negócios sobre quais são os comportamentos esperados de um sistema, enquanto abordagem de Engenharia de Requisitos, visto que as técnicas de modelagem através de comportamentos estão presentes no processo (NUSEIBEH; EASTERBROOK, 2000), definindo uma linguagem comum entre as partes (LAZĂR; MONTAGNA; PĂRV, 2010).

Este entendimento é particularmente útil para minimizar problemas causados por tomadas de decisão sem entendimento pleno dos requisitos, um dos ofensores típicos relacionados à finalização mal-sucedida de um projeto (VERNER; SAMPSON; CERPA, 2008).

A busca pela linguagem comum, ou ubíqua, inspirada pelos conceitos de Desenvolvimento Dirigido a Domínios (EVANS, 2004), é representada no BDD através da sintaxe *Gherkin*, tipicamente utilizada em ferramentas como o *Cucumber*, que permite a escrita e automação de cenários nesta sintaxe (WYNNE; HELLESØY, 2017).

Mesmo com a utilização destas ferramentas de automação, é importante ressaltar que a ênfase em modelar cenários através do BDD é validar o entendimento dos envolvidos sobre quais comportamentos o sistema deve ter antes mesmo de sua implementação, a automação dos testes de aceitação é uma consequência deste

entendimento, conforme enfatizado pelo próprio criador do *Cucumber* (HELLESØY, 2015).

Dado que é possível e recomendável que as abordagens do BDD sejam propostas como práticas de Engenharia de Requisitos, podemos avaliar se a escrita e entendimento dos cenários está adequada, de uma forma inspirada pelo INVEST (COHN, 2004), mas que consiste em 14 fatores que, se satisfeitos, implicam no entendimento mútuo dos cenários modelados entre os analistas de negócio, desenvolvedores e testadores de *software* (OLIVEIRA; MARCZAK, 2017). Os 5 C's para uma comunicação escrita de qualidade (PMI, 2017) também é um aliado para modelagem destes cenários.

Também é importante reforçar que, por mais que a especificação esteja correta, é preciso assegurar que ela seja conhecida, compreendida e esteja disseminada à todos os envolvidos, ainda mais em um contexto no qual equipes diferentes e distribuídas estão atuando no mesmo projeto (HINDS; BAILEY, 2003), pois isso pode causar uma série de problemas ou mesmo conflitos devido à expectativas divergentes (WILLIAMS; O'REILLY, 1998). Desta forma, também podemos reconhecer que a comunicação está entre as atividades da Engenharia de Requisitos (NUSEIBEH; EASTERBROOK, 2000).

Aliado à isso, um estudo demonstrou que quanto piores as comunicações, maiores os custos de um projeto e sua propensão a ser concluído com falhas (PMI, 2013). Estes riscos podem ser diminuídos se utilizarmos as abordagens de Gestão das Comunicações propostas no PMBOK®, que podem ser utilizadas como apoio para disseminação das documentações produzidas, atacando dois dos pontos que levam projetos à serem finalizados com falha: problemas na especificação de requisitos e nas comunicações.

3. PROPOSTA DE PRÁTICAS COM BDD

Este capítulo consiste na proposta de um conjunto de práticas de refinamento de requisitos utilizando BDD e Gestão das Comunicações, em um projeto real de desenvolvimento de *software*, com atuação de mais de uma equipe, em uma empresa cujo nome será preservado.

A especificação dos requisitos do sistema é escrita utilizando *Gherkin* (ZEN, 2013), e vai passar por uma verificação de aderência à uma série de práticas propostas em uma lista de verificação. O objetivo é validar a influência destas práticas no apoio do entendimento sobre o quê deve ser feito, diminuição de conflitos entre equipes e consequentemente na finalização bem-sucedida do projeto. Esta influência é mensurada através de uma avaliação qualitativa, mediante aplicação de questionário aos participantes do projeto.

3.1 Proposta de práticas

A fim de realizar as validações propostas neste trabalho, um conjunto de práticas de especificação de requisitos com BDD e Gestão das Comunicações será proposto e posteriormente aplicado em um projeto de desenvolvimento de *software*.

Esta proposta pode ser adotada tanto em projetos nos quais há atuação de uma única equipe quanto em projetos com múltiplas equipes. Para os casos de atuação de mais de uma equipe, a separação entre equipe responsável e equipe(s) envolvida(s) deve ser realizada. A determinação de equipe responsável e envolvido é realizada através da análise sobre qual domínio / contexto é o maior impactado na entrega do projeto e qual(is) domínio(s) são menos afetados ou não dizem respeito à funcionalidade principal, mas que fazem parte da solução como um todo.

A verificação da aplicação correta destas práticas é realizada pelo papel denominado Responsável do Projeto, de acordo com o que foi desempenhado pelos desenvolvedores, PO's (ou algum representante de negócio equivalente) e engenheiros de qualidade (QA), das equipes responsáveis e envolvidas.

Além da responsabilidade de verificar se o conjunto de práticas está sendo aplicado corretamente, o Responsável pelo Projeto precisa atuar como facilitador para realização das mesmas e assegurar que todas as cerimônias e processos que envolvam seu cumprimento sejam realizados. Essa verificação é importante pois para que a avaliação das contribuições das práticas propostas para a entrega do projeto seja legítima, é necessário que todos os itens sejam contemplados e aplicados de forma aderente ao que cada técnica e processo se propõe.

A própria lista de práticas fornece uma base para a Gestão das Comunicações no projeto, visto que os requisitos / funcionalidades que estão sendo tratados, bem como os responsáveis e envolvidos do projeto, são identificados através da mesma. Além disso, ela busca assegurar que a especificação dos requisitos foi realizada de acordo com as técnicas propostas pelo BDD e garante práticas para facilitar a comunicação como alinhamentos entre as partes para reforçar o entendimento e plano de comunicações.

A tabela 1 representa o conjunto de práticas propostas para apoiar o entendimento dos requisitos e melhorar a comunicação entre as diferentes equipes do projeto em desenvolvimento, desta forma , contribuindo na finalização do projeto como um todo.

Tabela 1 – Práticas propostas para refinamento dos requisitos e Gestão das Comunicações

1. Práticas para Especificação de Requisitos	
1.1 Identificador do Requisito / História	[id único]
1.2 Descrição do Requisito / História	[breve descrição, a especificação em si estará no formato BDD]
2. Identificação Responsáveis	
2.1 PO ou Representante de Negócio	[nome e e-mail]
2.2 Engenheiro de Qualidade (QA)	[nome e e-mail]
2.3 Desenvolvedor	[nome e e-mail]
3. Identificação Envolvidos	
3.1 Domínio do sistema mantido pela equipe envolvida	[descrição do domínio do sistema, exemplo: Transferências Bancárias]
3.2 PO ou Representante de Negócio	[nome e e-mail]
3.3 Engenheiro de Qualidade (QA)	[nome e e-mail]
3.4 Desenvolvedor	[nome e e-mail]
4. Refinamento dos requisitos entre os responsáveis	
4.1 Houve reunião de refinamento e especificação de requisitos entre os responsáveis?	[marcar com "x" para sim, deixar em branco para não]
4.2 A especificação de requisitos está em formato BDD?	[marcar com "x" para sim, deixar em branco para não]
4.3 As boas práticas de comunicação (5 C's) foram utilizadas na especificação?	[marcar com "x" para sim, deixar em branco para não]
4.4 Os cenários escritos estão de acordo com os 14 fatores?	[marcar com "x" para sim, deixar em branco para não]
5. Refinamento dos requisitos com os envolvidos	
5.1 Houve reunião de refinamento e especificação de requisitos com os envolvidos?	[marcar com "x" para sim, deixar em branco para não]
5.2 O entendimento dos requisitos foi normalizado junto aos envolvidos?	[marcar com "x" para sim, deixar em branco para não]
5.3 A especificação de requisitos está armazenada em um repositório centralizado, com fácil acesso pelos responsáveis e envolvidos?	[marcar com "x" para sim, deixar em branco para não]
6. Plano de comunicação	
6.1 Há a necessidade de realizar reuniões de checkpoint do projeto?	[marcar com "x" para sim, deixar em branco para não]
6.2 Existe um plano de comunicação / alinhamento a ser aplicado para o caso de mudanças?	[marcar com "x" para sim, deixar em branco para não]
6.3 Tanto os responsáveis quanto os envolvidos possuem conhecimento deste plano?	[marcar com "x" para sim, deixar em branco para não]

Fonte: O autor

As práticas apresentadas na tabela 1 se referem à verificações acerca da qualidade da especificação de requisitos em formato BDD, se fazendo valer de técnicas de Gestão das Comunicações. Diante desse cenário, ele é aplicado no momento em que se realiza o refinamento dos requisitos.

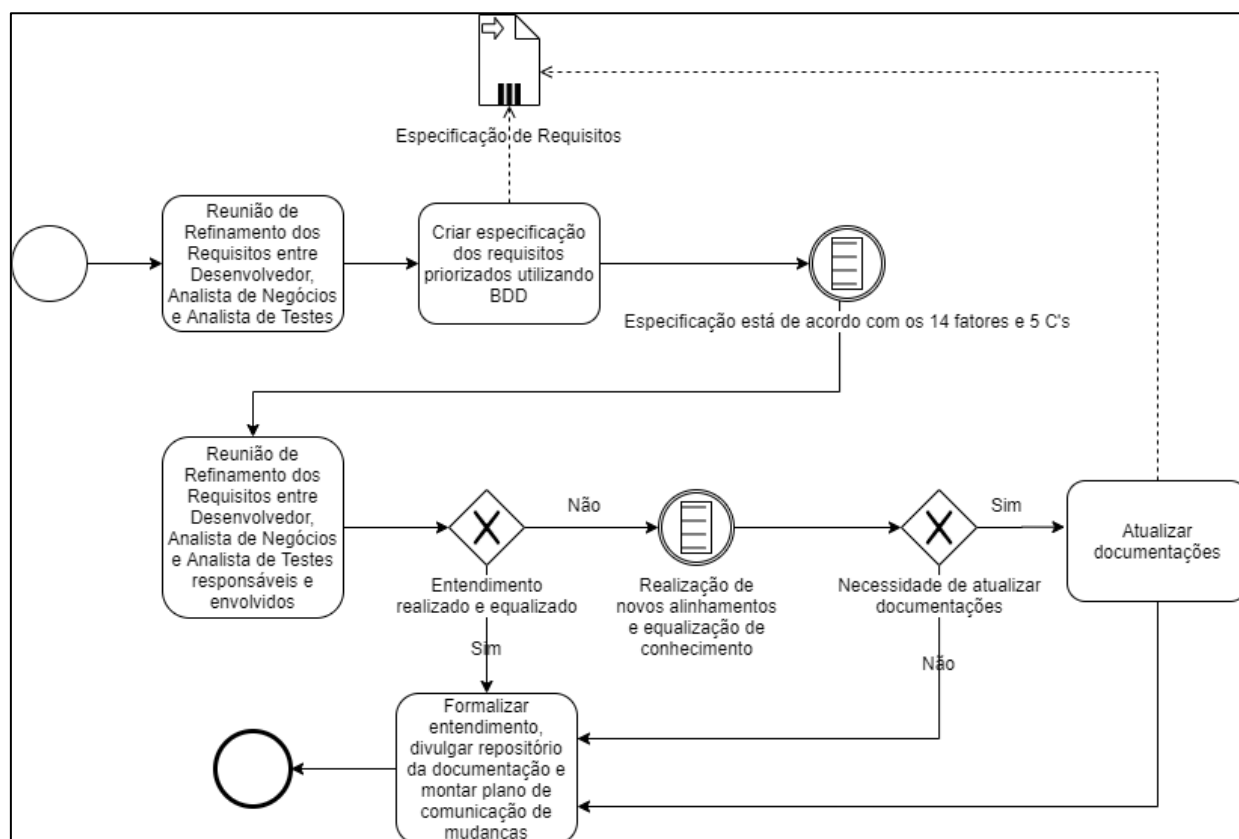
No modelo proposto, PO, Desenvolvedor e QA realizam esta reunião de refinamento de requisitos em conjunto, com a atuação do Responsável pelo Projeto como facilitador, e o entregável da mesma é a especificação das funcionalidades priorizadas em formato BDD. Este alinhamento e a especificação gerada a partir dele permitem identificar os membros das equipes, tanto responsável quanto envolvida, bem como verificar a aderência das especificações produzidas ao BDD, 14 fatores e 5 C's (com o objetivo de validar a estrutura e entendimento dos comportamentos especificados), estas verificações, aliada à necessidade de identificação dos envolvidos, foram a base para elaborar as seções 1 a 4 da lista apresentada na tabela 1.

Após essa reunião, as documentações produzidas são apresentadas ao membros da equipe envolvida em uma cerimônia de entendimento e refinamento conjunto. Caso o entendimento não tenha sido equalizado nesta conversa, as dúvidas ou pendências serão anotadas, e um novo alinhamento é marcado com o retorno destes pontos. Após a equalização do entendimento, é possível verificar os itens acerca dos alinhamentos e, principalmente, entendimento mútuo dos requisitos apresentados, bem como a rastreabilidade destes requisitos, obtida mediante armazenamento em repositório centralizado e conhecido por todos os envolvidos no projeto. Este ponto foi a base para elaboração da seção 5 da lista de práticas apresentada na tabela 1.

Após a conclusão deste entendimento entre as partes, os itens referentes ao plano de comunicações e resposta à mudanças também podem ser satisfeitos, de modo que todas as equipes envolvidas tenham conhecimento de como funcionarão as comunicações do projeto, bem como quais são os profissionais de cada equipe que atuarão no mesmo, que foi o critério utilizado para elaboração seção 6 da lista de práticas da tabela 1.

A figura 8 ilustra, através de um diagrama BPMN, os processos e cerimônias que permitirão a aplicação e verificação das práticas propostas.

Figura 8 – Fluxo de processos para validar realização das práticas sugeridas



Fonte: O autor

A importância de assegurar a conclusão correta de cada processo presente no fluxo apresentado na figura 8 (bem como a verificação de cada item da lista de práticas da tabela 1), é a de que, além de garantir aderência às técnicas de especificação de requisitos com BDD e de Gestão das Comunicações, a validação e critérios para avaliar a importância da utilização destas práticas em um projeto poderão ser realizados de forma assertiva, visto que não houveram inconsistências na aplicação das técnicas e o processo proposto foi concluído de modo satisfatório. Desta forma, a equipe terá os insumos necessários para responder o questionário apresentado na próxima seção deste trabalho.

3.2 Definição das questões para avaliar a influência das práticas

Conforme abordado ao longo do trabalho, é aplicado um questionário após a aplicação das práticas propostas e finalização do desenvolvimento do projeto, a fim de validar as contribuições da especificação de requisitos e comunicações para a entrega como um todo, esta validação ocorrerá de forma qualitativa.

A fim de qualificar a influência destas práticas na finalização do projeto, os principais pontos a serem validados são: compreensão dos envolvidos sobre o quê deve ser feito; velocidade e assertividade da entrega (ou seja, a quantidade de retrabalho foi pouca ou não existiu e o produto entregue estava dentro das expectativas dos clientes); incidência de conflitos; velocidade e assertividade das comunicações. As questões serão respondidas pelos participantes do projeto, tanto da equipe envolvida quanto da equipe responsável.

Ao fim da aplicação do questionário, os resultados serão consolidados e avaliados, conforme os critérios compartilhados, alinhado com os objetivos do trabalho.

A tabela 2 apresenta as questões aplicadas aos participantes do projeto.

Tabela 2 – Questionário de validação

#	Questão
1	Qual papel você representou no projeto?
2	Em qual equipe você atuou?
3	A escrita das histórias BDD contribuiu com o entendimento do quê precisa ser feito?
4	Como você classifica a dificuldade de aprendizado do BDD e sintaxe Gherkin?
5	As reuniões e validações de escopo contribuíram para impedir a ocorrência de conflitos durante a execução do projeto?
6	A finalização do projeto foi bem-sucedida (considerando prazo e cumprimento de escopo)?
7	Em uma escala de 0 a 10, quanto o BDD influenciou nos resultados do projeto?
8	A utilização do BDD, sem as práticas de Gestão das Comunicações, seriam suficientes para a finalização bem sucedida do projeto? Justifique.
9	Com relação aos projetos entregues sem esta técnica, quais foram as principais diferenças notadas com relação ao entendimento do escopo e comunicações do projeto?
10	De 0 a 10, qual a chance de você utilizar estas técnicas em outros projetos?

Fonte: O autor

4. ESTUDO DE CASO E ANÁLISE DE RESULTADOS

4.1 Aplicação das práticas em um projeto de software

Após a definição da lista de práticas, processos de Gestão das Comunicações e método de avaliação, foi dado início no projeto que iria se utilizar de todas estas técnicas.

O projeto consiste na realização de transferências bancárias, no qual serão enfatizados os requisitos referentes à comprovação destas transferências (consultas internas dos comprovantes das transferências efetuadas), com a participação de duas equipes: a equipe do sistema de comprovantes, classificada como responsável, e a equipe que irá implementar o *front-end* de consultas em uma estação administrativa corporativa, classificada como envolvida.

Para possibilitar a realização das consultas das comprovações de transferências para o sistema envolvido, o sistema de comprovantes irá fornecer uma API, utilizando o padrão arquitetural REST (FIELDING; TAYLOR, 2000), na qual as dúvidas serão sanadas e eventuais ajustes nas especificações serão realizados ao longo dos alinhamentos propostos e realizados neste estudo de caso.

Na condução deste estudo, o autor exerceu a função de Responsável pelo Projeto, e iniciou a aplicação das práticas com uma reunião de alinhamento inicial e refinamento dos requisitos, da qual participaram o desenvolvedor, PO e QA da equipe responsável. Nesta reunião, o facilitador passou uma breve contextualização sobre o BDD e sintaxe *Gherkin* para todos, além de se colocar à disposição como facilitador da aplicação da técnica e para tirar as dúvidas. Os requisitos foram definidos e modelados, se utilizando das técnicas apresentadas, com algumas o apoio do facilitador à respeito da correta aplicação da sintaxe. Foi acordado que para as consultas às comprovações realizadas seriam tratados 3 cenários: a listagem dos comprovantes atrelados à uma conta em um determinado período e o detalhamento dos comprovantes das transferências intra (mesmo banco) e inter (bancos diferentes).

Após a escrita dos cenários, foi realizada em conjunto uma avaliação qualitativa para verificar a aderência do que foi escrito aos 5 C's da comunicações e 14 critérios para escrita de BDD. Após alguns ajustes, os envolvidos entraram em um consenso e os cenários foram classificados como aderentes aos critérios estabelecidos. Um ponto relevante e que foi destacado pela equipe é como o uso de tabelas de dados nas especificações contribuiu para que o item "Controlada" dos 5 C's das comunicações fosse atendido, visto que fornece uma referência explícita e visual de como as informações devem ser tratadas tanto a nível de front-end quanto de back-end. Como a equipe tinha experiência em atuar com engenharia de software, todos ficaram confortáveis com relação ao entendimento e avaliação dos cenários de acordo com os 14 fatores.

As figuras 9, 10 e 11 representam, respectivamente, a versão inicial das modelagens dos cenários de lista, detalhamento transferência intrabancária e transferência interbancária.

Figura 9 – Cenário BDD de listagem

Cenário: Consultar lista de comprovantes

Dado o funcionário da área de operações vai realizar uma consulta no portal corporativo

Quando o menu "Consultar Comprovantes" for acessado

E os dados da consulta são informados no formato abaixo

Nome do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)
Agência	agencia_origem	Alfanumerico	4
Conta	conta_origem	Alfanumérico	12
Período Inicial	data_inicio_consulta	Alfanumérico	10
Período Final	data_fim_consulta	Alfanumérico	10

Então a listagem dos comprovantes é apresentada de acordo com o filtro:

Nome do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)
Data	data_operacao	Alfanumerico	10
Tipo Operação	tipo_operacao	Alfanumerico	60
Canal Origem	canal_origem	Alfanumerico	30
Valor	simbolo_moeda + valor_operacao	Alfanumérico + Decimal	4 + 13
Link para visualizar	identificador_comprovante	Alfanumerico	36

Fonte: O autor

Figura 10 – Cenário BDD de transferência intrabancária

```

Cenário: Consultar comprovante de transferência intrabancária (conta origem e destino pertencem à mesma instituição)

Dado que o funcionário da área de operações realizou a listagem de comprovantes (cenário "Consultar lista de comprovantes")
E clicar em "Link para visualizar" em um comprovante de transferência intrabancária

Então o comprovante da operação é armazenado e apresentado ao cliente conforme regras especificadas abaixo (todos os campos são obrigatórios e a ordem de campos deve ser respeitada)

| Dados da Conta Origem |
| Label do campo no canal | Nome do campo no sistema | Tipo Campo | Tamanho (em caracteres) |
| Nome | nome_cliente | Alfanumérico | 256 |
| Agência | agencia_origem | Alfanumérico | 4 |
| Conta | conta_origem | Alfanumérico | 12 |

| Dados da Conta Destino |
| Label do campo no canal | Nome do campo no sistema | Tipo Campo | Tamanho (em caracteres) |
| Nome Favorecido | nome_favorecido | Alfanumérico | 256 |
| Agência | agencia_destino | Alfanumérico | 4 |
| Conta | conta_destino | Alfanumérico | 12 |
| Valor | simbolo_moeda + valor_operacao | Alfanumérico + Decimal | 4 + 13 |

| Dados de Rodapé |
| Label do campo no canal | Nome do campo no sistema | Tipo Campo | Tamanho (em caracteres) |
| Operação efetuada em | data_operacao | Alfanumérico | 10 |
| às | hora_operacao | Alfanumérico | 8 |

```

Fonte: O autor

Figura 11 – Cenário BDD de transferência interbancária

```

Cenário: Consultar comprovante de transferência intrabancária (conta origem e destino pertencem à mesma instituição)

Dado que o funcionário da área de operações realizou a listagem de comprovantes (cenário "Consultar lista de comprovantes")
E clicar em "Link para visualizar" em um comprovante de transferência intrabancária

Então o comprovante da operação é armazenado e apresentado ao cliente conforme regras especificadas abaixo (todos os campos são obrigatórios e a ordem de campos deve ser respeitada)

| Dados da Conta Origem |
| Label do campo no canal | Nome do campo no sistema | Tipo Campo | Tamanho (em caracteres) |
| Nome | nome_cliente | Alfanumérico | 256 |
| Agência | agencia_origem | Alfanumérico | 4 |
| Conta | conta_origem | Alfanumérico | 12 |

| Dados da Conta Destino |
| Label do campo no canal | Nome do campo no sistema | Tipo Campo | Tamanho (em caracteres) |
| Nome Favorecido | nome_favorecido | Alfanumérico | 256 |
| Agência | agencia_destino | Alfanumérico | 4 |
| Conta | conta_destino | Alfanumérico | 12 |
| Valor | simbolo_moeda + valor_operacao | Alfanumérico + Decimal | 4 + 13 |

| Dados de Rodapé |
| Label do campo no canal | Nome do campo no sistema | Tipo Campo | Tamanho (em caracteres) |
| Operação efetuada em | data_operacao | Alfanumérico | 10 |
| às | hora_operacao | Alfanumérico | 8 |

```

Fonte: O autor

Após o alinhamento inicial, modelagem e avaliação dos cenários, foi realizado um alinhamento com a equipe envolvida, para apresentação dos requisitos e validação do entendimento.

Apesar de o entendimento ter ficado praticamente concluído nessa reunião, os representantes da equipe envolvida, que irão realizar as consultas em um portal corporativo, enfatizaram a importância de se incluir os tipos de formatação e um exemplo de valor válido a ser retornado pela API, na tabela de dados utilizada na modelagem. Diante deste cenário, a documentação foi atualizada, o entendimento entre as partes foi validado e equalizado e um plano de comunicação foi criado pelo Responsável pelo Projeto, utilizando a própria lista de práticas como base para

identificação dos contatos, aliado à um repositório centralizado das documentações e um alinhamento realizado dois dias por semana para atualizar o andamento do projeto e eliminar eventuais impedimentos.

As figuras 12, 13 e 14 representam a modelagem dos cenários de listagem, transferência intrabancária e transferência interbancária após as considerações dos envolvidos.

Figura 12 – Cenário BDD de listagem após reunião de alinhamento com envolvidos

Cenário: Consultar lista de comprovantes

Dado o funcionário da área de operações vai realizar uma consulta no portal corporativo

Quando o menu "Consultar Comprovantes" for acessado
E os dados da consulta são informados no formato abaixo

Nome do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo
Agência	agencia_origem	Alfanumerico	4	Não se Aplica	9999
Conta	conta_origem	Alfanumérico	12	999999999-9	000099999-9
Período Inicial	data_inicio_consulta	Alfanumérico	10	DD.MM.AAAA	10.10.2020
Período Final	data_fim_consulta	Alfanumérico	10	DD.MM.AAAA	10.10.2020

Então a listagem dos comprovantes é apresentada de acordo com o filtro:

Nome do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo
Data	data_operacao	Alfanumerico	10	DD.MM.AAAA	10.10.2020
Tipo Operação	tipo_operacao	Alfanumerico	60	N/A	Transferência Interbancária
Canal Origem	canal_origem	Alfanumerico	30	N/A	Internet
Valor	simbolo_moeda + valor_operacao	Alfanumérico + Decimal	4 + 13	XXXX99999999,99 (XXXX = moeda)	R\$420,00
Link para visualizar	identificador_comprovante	Alfanumerico	36	UUID	897e7fb1-1b38-4381-996c-fbdbb3c23298

Fonte: O autor

Figura 13 – Cenário BDD de transferência intrabancária após reunião de alinhamento com envolvidos

Cenário: Consultar comprovante de transferência intrabancária (conta origem e destino pertencem à mesma instituição)

Dado que o funcionário da área de operações realizou a listagem de comprovantes (cenário "Consultar lista de comprovantes")
E clicar em "Link para visualizar" em um comprovante de transferência intrabancária

Então o comprovante da operação é armazenado e apresentado ao cliente conforme regras especificadas abaixo (todos os campos são obrigatórios e a ordem de campos deve ser respeitada)

Dados da Conta Origem					
Label do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo
Nome	nome_cliente	Alfanumérico	256	Não se Aplica	Giovanni Vicente
Agência	agencia_origem	Alfanumerico	4	Não se Aplica	9999
Conta	conta_origem	Alfanumérico	12	999999999-9	000099999-9
Dados da Conta Destino					
Label do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo
Nome Favorecido	nome_favorecido	Alfanumérico	256	Não se Aplica	Giovanni Vicente
Agência	agencia_destino	Alfanumerico	4	Não se Aplica	9999
Conta	conta_destino	Alfanumérico	12	999999999-9	000099999-9
Valor	simbolo_moeda + valor_operacao	Alfanumérico + Decimal	4 + 13	XXXX99999999,99 (XXXX = moeda)	R\$150,00
Dados de Rodapé					
Label do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo
Operação efetuada em	data_operacao	Alfanumérico	10	DD/MM/AAAA	10/09/2020
às	hora_operacao	Alfanumérico	8	HH:MM:SS	08:00:00

Fonte: O autor

Figura 14 – Cenário BDD de transferência interbancária após reunião de alinhamento com envolvidos

Cenário: Consultar comprovante de transferência interbancária (conta origem e destino pertencem à instituições diferentes)

Dado que o funcionário da área de operações realizou a listagem de comprovantes (cenário "Consultar lista de comprovantes")
E clicar em "Link para visualizar" em um comprovante de transferência interbancária

Então o comprovante da operação é armazenado e apresentado ao cliente conforme regras especificadas abaixo (todos os campos são obrigatórios e a ordem de campos deve ser respeitada)

Dados da Conta Origem						
Label do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo	
Nome	nome_cliente	Alfanumérico	256	Não se Aplica	Giovanni Vicente	
Agência	agencia_origem	Alfanumerico	4	Não se Aplica	9999	
Conta	conta_origem	Alfanumérico	12	9999999999-9	0000999999-9	
Dados da Conta Destino						
Label do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo	
Nome Favorecido	nome_favorecido	Alfanumérico	256	Não se Aplica	Giovanni Vicente	
* Não se Aplica *	tipo_pessoa	Alfanumérico	1	"F" para física ou "J" para jurídica	F	
CPF / CNPJ	numero_documento	Alfanumérico	18	XXX.XXX.XXX-XX ou XX.XXX.XXX/XXXX-XX	999-999-999-99	
Instituição	nome_instituicao_financeira	Alfanumérico	256	Não se Aplica	Banco Fictício SA	
Agência	agencia_destino	Alfanumérico	4	Não se Aplica	9999	
Conta	conta_destino	Alfanumérico	12	9999999999-9	0000999999-9	
Valor	simbolo_moeda + valor_operacao	Alfanumérico + Decimal	4 + 13	XXXX9999999999,99 (XXXX = moeda)	R\$150,00	
Finalidade	finalidade_transferencia	Alfanumérico	256	Não se Aplica	Crédito em Conta	
Dados de Rodapé						
Label do campo no canal	Nome do campo no sistema	Tipo Campo	Tamanho (em caracteres)	Máscara de Formatação	Exemplo	
Operação efetuada em	data_operacao	Alfanumérico	10	DD/MM/AAAA	10/09/2020	
às	hora_operacao	Alfanumérico	8	HH:MM:SS	08:00:00	

Fonte: O autor

Após a conclusão das modelagens e verificações das práticas, os desenvolvimentos foram efetivamente iniciados, e o projeto foi progredindo até ser concluído. Durante os alinhamentos acordados no plano de comunicações, não foram relatadas ocorrências de conflitos ou impedimentos que poderiam comprometer a entrega, no entanto, é a resposta do questionário por parte dos participantes, a ser apresentada na próxima seção, que irá estabelecer a avaliação e percepções oficiais da influência das práticas e resultados deste estudo de caso.

4.2 Avaliação da influência das práticas na finalização do projeto

Concluídos os desenvolvimentos, bem como a verificação do cumprimento de cada item da lista de práticas, disponíveis no apêndice A, o questionário apresentado na seção 3.2 foi aplicado aos participantes do projeto, a fim de obter suas percepções e avaliações à respeito das contribuições nas técnicas propostas para a finalização do projeto.

No total, tivemos 8 (oito) respondentes, que foi o número de participantes dessa entrega: da equipe responsável, participaram o PO, o QA e 1 (um) desenvolvedor; da equipe envolvida, o PO, o QA e 3 (três) desenvolvedores.

5. CONSIDERAÇÕES FINAIS

[Este capítulo descreve as conclusões e as contribuições do trabalho e sugestões de trabalhos futuros que poderão ser desenvolvidos a partir deste trabalho]

5.1 Conclusões

5.2 Contribuições do Trabalho

5.3 Trabalhos Futuros

REFERÊNCIAS

BALDINI, Daniel. Refinement Meeting: Por que e como fazer? Disponível em: <<http://www.productmanagement.com.br/por-que-e-como-fazer-refinement-meeting-anteriormente-backlog-grooming/>>. Outubro de 2019. Acesso em 9 de setembro de 2020.

BECK, Kent. Extreme programming explained: embrace change. Addison-Wesley professional, 2000.

BECK, Kent et al.. Manifesto for Agile Software Development. Disponível em: <<http://agilemanifesto.org/>>. 2001. Acesso em 4 de setembro de 2020.

BECK, Kent et al.. Principles behind the Agile Manifesto. Disponível em: <<http://agilemanifesto.org/principles.html>>. 2001. Acesso em 4 de setembro de 2020.

BECK, Kent. Test-driven development: by example. Addison-Wesley Professional, 2003.

BITTNER, Kurt. When requirements go bad. Ivar Jacobson Consulting Whitepaper, 2008.

BROOKS JR, Frederick P. O Mítico Homem-Mês (Edição de 20º Aniversário). 1995.

COHN, Mike. User stories applied: For agile software development. Addison-Wesley Professional, 2004.

EVANS, Eric. Domain-driven design: tackling complexity in the heart of software. Addison-Wesley Professional, 2004.

FIELDING, Roy T.; TAYLOR, Richard N. Architectural styles and the design of network-based software architectures. Irvine: University of California, Irvine, 2000.

GOTEL, O. & FINKELSTEIN, A. An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering (ICRE'94), Colorado Springs, April 1994, pp. 94-101.

HARPER, D. Online Etymology Dictionary. Disponível em: <<https://www.etymonline.com/word/communication>>. 2017. Acesso em 28 de outubro de 2020.

HÄSER, Florian; FELDERER, Michael; BREU, Ruth. Is business domain language support beneficial for creating test case specifications: A controlled experiment. Information and software technology, v. 79, p. 52-62, 2016.

HELLESØY, Aslak. Disclaimer: I am the creator of Cucumber. Disponível em: <<https://news.ycombinator.com/item?id=10194242>>. Setembro de 2015. Acesso em 4 de setembro de 2020.

HENRAJANI, A. Desenvolvimento ágil em Java com Spring, Hibernate e Eclipse. São Paulo: Pearson Prentice Hall, 2007.

HINDS, Pamela J.; BAILEY, Diane E. Out of sight, out of sync: Understanding conflict in distributed teams. Organization science, v. 14, n. 6, p. 615-632, 2003.

LATORRE, Roberto. Effects of developer experience on learning and applying unit test-driven development. IEEE Transactions on Software Engineering, v. 40, n. 4, p. 381-395, 2013.

LAZĂR, Ioan; MOTOGNA, Simona; PÂRV, Bazil. Behaviour-driven development of foundational UML components. Electronic Notes in Theoretical Computer Science, v. 264, n. 1, p. 91-105, 2010.

MARTIN, Robert C. Código Limpo: Habilidades Práticas do Agile Software. Pearson Education, 2008.

MAZUCA, Vinícius Canhisares. O Papel da Comunicação no Desenvolvimento de Software Ágil - Uma Revisão Sistemática da Literatura. 64p. Monografia – Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo, Curso de Especialização em Tecnologia de Software. MBA-USP, 2018.

NORTH & ASSOCIATES, Dan. Introducing BDD. Disponível em: <<http://dannorth.net/introducing-bdd>>. Março de 2006. Acesso em 18 novembro de 2014.

NUSEIBEH, Bashar; EASTERBROOK, Steve. Requirements engineering: a roadmap. In: Proceedings of the Conference on the Future of Software Engineering. 2000. p. 35-46.

OLIVEIRA, Breno Henrique Duarte de. Práticas de Behavior Driven Development em Scrum para Entrega Contínua de Valor. 57p. Monografia – Programa de Educação Continuada da Escola Politécnica da Universidade de São Paulo, Curso de Especialização em Tecnologia de Software. MBA-USP, 2015.

OLIVEIRA, Gabriel; MARCZAK, Sabrina. On the empirical evaluation of BDD scenarios quality: preliminary findings of an empirical study. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, 2017. p. 299-302.

OLIVÉRIO, Gabriel. Introduzindo o BDD. Disponível em: <<http://broncodev.com/2016-10-11-introduzindo-o-bdd/>>. 2016. Acesso em 4 de setembro de 2020.

PAULK, Mark C. Agile Methodologies and Process Discipline. Carnegie Melon Software Engineering Institute, 2002.

PMI. Guia PMBOK® - 6ª Edição. Project Management Institute, Inc., 2017.

PMI. More Than Half of All Project Budget Risk is Due to Innefective Communications. Disponível em: <<https://www.businesswire.com/news/home/20130522006700/en/PMI-More-Than-Half-of-All-Project-Budget-Risk-is-Due-to-Ineffective-Communications#:~:text=Communications%20%7C%20Business%20Wire->>

,PMI%3A%20More%20Than%20Half%20of%20All%20Project%20Budget,is%20Due%20to%20Ineffective%20Communications&text=The%20new%20research%20indicates%20that,deficiencies%20at%20the%20enterprise%20level.>. Maio de 2013. Acesso em 4 de setembro de 2020.

SCHWABER, Ken & SUTHERLAND, Jeff. Um guia definitivo para o Scrum: As regras do jogo. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Outubro de 2017. Acesso em 01 agosto de 2020.

SEDANO, Todd; RALPH, Paul; PÉRAIRE, Cécile. The product backlog. In: 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). IEEE, 2019. p. 200-211.

SMARTBEAR SOFTWARE®. Gherkin Reference - Cucumber Documentation. Disponível em <<https://cucumber.io/docs/gherkin/reference/>>. 2019. Acesso em 14 de setembro de 2020.

SONDERMANN, Danielli Veiga Carneiro et al. O desafio do processo de comunicação eficaz frente a heterogeneidade dos alunos na modalidade a distância: perspectivas a partir do Universal Design for Learning-Design Universal para Aprendizagem. Em: ESUD 2013 - X CONGRESSO BRASILEIRO DE ENSINO SUPERIOR A DISTÂNCIA.

VERNER, June; SAMPSON, Jennifer; CERPA, Narciso. What factors lead to software project failure?. In: 2008 Second International Conference on Research Challenges in Information Science. IEEE, 2008. p. 71-80.

WELLS, Don. Extreme Programming: A gentle introduction. Disponível em: <<http://www.extremeprogramming.org/>>. 2000. Acesso em 13 de setembro de 2020.

WELLS, Don. XP Collective Code Ownership. Disponível em: <<http://www.extremeprogramming.org/map/code.html>>. 2000. Acesso em 13 de setembro de 2020.

WILLIAMS, K. Y.; O'REILLY III, C. A. Demography and Diversity in Organisations: A review of 40 years of research in BM Staw and LL Cummings (eds) Research in Organisational Behaviour Vol. 20. Jai Pres, Connecticut, 1998.

WYNNE, Matt; HELLESØY, Aslak; TOOKE, Steve. The cucumber book: behaviour-driven development for testers and developers. Pragmatic Bookshelf, 2017.

ZAVE, P. (1997). Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys, 29(4): 315-321

ZEN, Roberto. Gherkin and Cucumber. Tese de Doutorado. Ph. D. thesis, University of Trento, 2013.

APÊNDICE A – Listas de verificação utilizadas no estudo de caso

Tabela 3 – Lista de verificação do cenário de lista

1. Práticas para Especificação de Requisitos	
1.1 Identificador do Requisito / História	BDD-SC-001
1.2 Descrição do Requisito / História	Listar comprovantes de acordo com agência, conta e período.
2. Identificação Responsáveis	
2.1 PO ou Representante de Negócio	[nome e e-mail serão preservados]
2.2 Engenheiro de Qualidade (QA)	[nome e e-mail serão preservados]
2.3 Desenvolvedor	[nome e e-mail serão preservados]
3. Identificação Envolvidos	
3.1 Domínio do sistema mantido pela equipe envolvida	Portal Corporativo
3.2 PO ou Representante de Negócio	[nome e e-mail serão preservados]
3.3 Engenheiro de Qualidade (QA)	[nome e e-mail serão preservados]
3.4 Desenvolvedor	[nome e e-mail serão preservados]
4. Refinamento dos requisitos entre os responsáveis	
4.1 Houve reunião de refinamento e especificação de requisitos entre os responsáveis?	x
4.2 A especificação de requisitos está em formato BDD?	x
4.3 As boas práticas de comunicação (5 C's) foram utilizadas na especificação?	x
4.4 Os cenários escritos estão de acordo com os 14 fatores?	x
5. Refinamento dos requisitos com os envolvidos	
5.1 Houve reunião de refinamento e especificação de requisitos com os envolvidos?	x
5.2 O entendimento dos requisitos foi normalizado junto aos envolvidos?	x
5.3 A especificação de requisitos está armazenada em um repositório centralizado, com fácil acesso pelos responsáveis e envolvidos?	x
6. Plano de comunicação	
6.1 Há a necessidade de realizar reuniões de checkpoint do projeto?	x
6.2 Existe um plano de comunicação / alinhamento a ser aplicado para o caso de mudanças?	x
6.3 Tanto os responsáveis quanto os envolvidos possuem conhecimento deste plano?	x

Fonte: O autor

Tabela 4 – Lista de verificação do cenário de transferência intrabancária

1. Práticas para Especificação de Requisitos	
1.1 Identificador do Requisito / História	BDD-SC-002
1.2 Descrição do Requisito / História	Detalhar comprovante de Transferência Intrabancária.
2. Identificação Responsáveis	
2.1 PO ou Representante de Negócio	[nome e e-mail serão preservados]
2.2 Engenheiro de Qualidade (QA)	[nome e e-mail serão preservados]
2.3 Desenvolvedor	[nome e e-mail serão preservados]
3. Identificação Envolvidos	
3.1 Domínio do sistema mantido pela equipe envolvida	Portal Corporativo
3.2 PO ou Representante de Negócio	[nome e e-mail serão preservados]
3.3 Engenheiro de Qualidade (QA)	[nome e e-mail serão preservados]
3.4 Desenvolvedor	[nome e e-mail serão preservados]
4. Refinamento dos requisitos entre os responsáveis	
4.1 Houve reunião de refinamento e especificação de requisitos entre os responsáveis?	x
4.2 A especificação de requisitos está em formato BDD?	x
4.3 As boas práticas de comunicação (5 C's) foram utilizadas na especificação?	x
4.4 Os cenários escritos estão de acordo com os 14 fatores?	x
5. Refinamento dos requisitos com os envolvidos	
5.1 Houve reunião de refinamento e especificação de requisitos com os envolvidos?	x
5.2 O entendimento dos requisitos foi normalizado junto aos envolvidos?	x
5.3 A especificação de requisitos está armazenada em um repositório centralizado, com fácil acesso pelos responsáveis e envolvidos?	x
6. Plano de comunicação	
6.1 Há a necessidade de realizar reuniões de checkpoint do projeto?	x
6.2 Existe um plano de comunicação / alinhamento a ser aplicado para o caso de mudanças?	x
6.3 Tanto os responsáveis quanto os envolvidos possuem conhecimento deste plano?	x

Fonte: O autor

Tabela 5 – Lista de verificação do cenário de transferência interbancária

1. Práticas para Especificação de Requisitos	
1.1 Identificador do Requisito / História	BDD-SC-003
1.2 Descrição do Requisito / História	Detalhar comprovante de Transferência Interbancária.
2. Identificação Responsáveis	
2.1 PO ou Representante de Negócio	[nome e e-mail serão preservados]
2.2 Engenheiro de Qualidade (QA)	[nome e e-mail serão preservados]
2.3 Desenvolvedor	[nome e e-mail serão preservados]
3. Identificação Envolvidos	
3.1 Domínio do sistema mantido pela equipe envolvida	Portal Corporativo
3.2 PO ou Representante de Negócio	[nome e e-mail serão preservados]
3.3 Engenheiro de Qualidade (QA)	[nome e e-mail serão preservados]
3.4 Desenvolvedor	[nome e e-mail serão preservados]
4. Refinamento dos requisitos entre os responsáveis	
4.1 Houve reunião de refinamento e especificação de requisitos entre os responsáveis?	x
4.2 A especificação de requisitos está em formato BDD?	x
4.3 As boas práticas de comunicação (5 C's) foram utilizadas na especificação?	x
4.4 Os cenários escritos estão de acordo com os 14 fatores?	x
5. Refinamento dos requisitos com os envolvidos	
5.1 Houve reunião de refinamento e especificação de requisitos com os envolvidos?	x
5.2 O entendimento dos requisitos foi normalizado junto aos envolvidos?	x
5.3 A especificação de requisitos está armazenada em um repositório centralizado, com fácil acesso pelos responsáveis e envolvidos?	x
6. Plano de comunicação	
6.1 Há a necessidade de realizar reuniões de checkpoint do projeto?	x
6.2 Existe um plano de comunicação / alinhamento a ser aplicado para o caso de mudanças?	x
6.3 Tanto os responsáveis quanto os envolvidos possuem conhecimento deste plano?	x

Fonte: O autor