



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO**

**LUCAS SILVA CAMELO**

**APLICAÇÃO DE UM PROCESSO DE V&V EM UMA EMPRESA DE  
DESENVOLVIMENTO DE SOFTWARE ÁGIL: UM ESTUDO DE CASO**

**FORTALEZA-CEARÁ**

**2023**

**LUCAS SILVA CAMELO**

APLICAÇÃO DE UM PROCESSO DE V&V EM UMA EMPRESA DE  
DESENVOLVIMENTO DE SOFTWARE ÁGIL: UM ESTUDO DE CASO

Monografia apresentada no curso de Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia da Computação. Área de concentração: Computação.

Orientadora: Prof<sup>a</sup>. Dra. Valéria Lelli Leitão Dantas

FORTALEZA-CEARÁ

2023

---

Página reservada para ficha catalográfica.

Utilize a ferramenta *online* [Catalog!](#) para elaborar a ficha catalográfica de seu trabalho acadêmico, gerando-a em arquivo PDF, disponível para download e/ou impressão.

(<http://www.fichacatalografica.ufc.br/>)

---

LUCAS SILVA CAMELO

APLICAÇÃO DE TÉCNICAS DE VERIFICAÇÃO E VALIDAÇÃO EM EMPRESAS QUE  
USAM METODOLOGIAS ÁGEIS PARA O DESENVOLVIMENTO DE SOFTWARE

Monografia apresentada no curso de Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia da Computação. Área de concentração: Computação.

Aprovada em: \_\_\_/\_\_\_/\_\_\_\_\_.

BANCA EXAMINADORA

---

Profa. Dra. Valéria Lelli Leitão Dantas (Orientador)  
Universidade Federal do Ceará (UFC)

---

Profa. Dra. Atslands Rego da Rocha  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Ismayle de Sousa Santos  
Universidade Estadual do Ceará (UECE)

A Deus.

Aos meus pais, Benício e Ana Célia.

## **AGRADECIMENTOS**

Meu sincero agradecimento à professora Dra. Valéria Lelli Leitão que contribuiu de maneira significativa para a conclusão deste Trabalho de Conclusão de Curso.

“O que prevemos raramente ocorre; o que menos esperamos geralmente acontece.”

*Benjamin Disraeli*

## RESUMO

Em um cenário competitivo, na qual empresas devem responder de forma rápida e eficiente às mudanças propostas pelo mercado, tornou-se inviável, do ponto de vista econômico e temporal, levantar todos os requisitos de um sistema. Diante disso, os processos convencionais, que tem por objetivo priorizar os procedimentos e definir os papéis durante o ciclo de desenvolvimento, tendem ao fracasso, pois a maioria costuma ser bem demorado, indo de encontro ao novo modelo global, que prioriza entregas rápidas. Com base nisso, surgiu a necessidade de implementar processos “leves” que fossem capazes de se adequar às mudanças impostas durante o desenvolvimento. Logo, os métodos ágeis surgiram como forma de conciliar o desenvolvimento de software, junto com a flexibilidade dos requisitos, priorizando sempre entregas do produto de maneira rápida e versionada. Da mesma forma que o processo de desenvolvimento teve que ser modelado para atender as demandas do mercado, os processos de Verificação e Validação (V&V) também tiveram que se adequar para atender aos princípios propostos pelas Metodologias Ágeis. Como exemplo, podemos citar o teste em XP (Extreme Programming), que propõe que os testes sejam realizados antes mesmo de iniciar o processo de codificação, reduzindo dessa forma possíveis erros que possam ocorrer. Logo, se utilizados da forma correta, a aplicação desses processos possibilita entregas mais rápidas e de maior qualidade, trazendo confiança para a equipe e promovendo a satisfação do cliente. Nesse contexto, neste trabalho foi conduzido um estudo de caso, em uma Empresa de Software Ágil, cujo principal objetivo foi identificar quais as principais dificuldades enfrentadas no processo de desenvolvimento e, com base nelas, implantar técnicas de verificação e validação, investigadas na literatura, a fim de mitigar ou reduzir os possíveis *bugs* que ocorrem nas diferentes fases do desenvolvimento.

**Palavras-chave:** desenvolvimento de software; metodologia ágil; verificação e validação.

## ABSTRACT

In a competitive scenario, in which companies must respond quickly and efficiently to changes proposed by the market, it has become impracticable, from an economic and temporal point of view, to raise all the requirements of a system. In view of this, conventional processes, which prioritize procedures and roles are well defined during the development cycle, tend to fail, as most of them tend to take a long time, going against the new global model, which prioritizes quick deliveries. Based on this, the need arose to implement “lightweight” processes that were capable of adapting to the changes imposed during development. Soon, agile methods emerged as a way to reconcile software development, along with the flexibility of requirements, always prioritizing product deliveries in a fast and versioned way. In the same way that the development process had to be modeled to meet market demands, the Verification and Validation (V&V) methodologies also had to adapt to meet the principles proposed by the Agile Methodologies. As an example, we can mention the test in XP (Extreme Programming), which proposes that the tests be carried out even before starting the coding process, thus reducing possible errors that may occur. Therefore, if applied correctly, the application of these processes enables faster and higher quality deliveries, bringing confidence to the team and promoting customer satisfaction. In this context, a case study was carried out in a Software Company, whose main objective was to identify the main difficulties of the development process and to implement verification and validation techniques, investigated in the literature, in the different stages of development.

**Keywords:** software development; agile methodology; verification and validation

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 - Fluxograma de atividade.....                                      | 20 |
| Figura 2 - Fluxograma do modelo de processo de desenvolvimento atual.....    | 22 |
| Figura 3 - Fluxograma do modelo de processo geral atual.....                 | 22 |
| Figura 4 - Fluxograma do modelo de processo de desenvolvimento proposto..... | 24 |
| Figura 5 - Perfil de colaborador.....  | 36 |
| Figura 6 - Avaliação do estudo de caso.....                                  | 37 |

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1 - Total de estudos da revisão bibliográfica .....      | 19 |
| Tabela 2 - Identificar os requisitos da sprint.....             | 24 |
| Tabela 3 - Definir o escopo dos testes.....                     | 25 |
| Tabela 4 - Definir métodos e técnicas de testes.....            | 25 |
| Tabela 5 - Planejar ambiente de testes .....                    | 26 |
| Tabela 6 - Escrever casos de testes .....                       | 27 |
| Tabela 7 - Revisar casos de testes .....                        | 28 |
| Tabela 8 - Gerenciar ajustes.....                               | 28 |
| Tabela 9 - Criar plano de testes .....                          | 29 |
| Tabela 10 - Configurar ambiente de teste. ....                  | 29 |
| Tabela 11 - Preparar massa de testes.....                       | 30 |
| Tabela 12 - Executar caso de teste.....                         | 30 |
| Tabela 13 - Reportar defeitos.....                              | 31 |
| Tabela 14 - Avaliar resultado dos testes .....                  | 32 |
| Tabela 15 - Elaboração e análise dos indicadores de testes..... | 32 |
| Tabela 16 - Entrega do sistema para o cliente.....              | 32 |

## SUMÁRIO

|  |           |
|--|-----------|
| <b>1 INTRODUÇÃO.....</b>                             | <b>14</b> |
| <b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>                  | <b>15</b> |
| 2.1 Metodologias Ágeis.....                          | 15        |
| 2.2 Verificação e Validação.....                     | 16        |
| <b>3 TRABALHOS RELACIONADOS.....</b>                 | <b>17</b> |
| <b>4 PROCEDIMENTOS METODOLÓGICOS.....</b>            | <b>18</b> |
| <b>5 ESTUDO DE CASO.....</b>                         | <b>21</b> |
| 5.1 Concepção da abordagem.....                      | 21        |
| 5.2 Descrição da abordagem.....                      | 22        |
| 5.2.1 Modelo Atual.....                              | 22        |
| 5.2.2 Modelo do processo de teste especializado..... | 24        |
| <b>6 AVALIAÇÃO.....</b>                              | <b>34</b> |
| <b>7 CONCLUSÃO.....</b>                              | <b>38</b> |

## 1 INTRODUÇÃO

Nos últimos anos, a área de desenvolvimento de software tem passado por mudanças significativas. Os métodos tradicionais de desenvolvimento de software, como o modelo cascata, têm dado lugar a abordagens mais flexíveis e adaptáveis, conhecidas como metodologias ágeis, onde é possível destacar a utilização do *Scrum* e do *Extreme Programming* (XP). Essas metodologias enfatizam a colaboração, a entrega contínua de valor e a rápida adaptação às mudanças no ambiente de desenvolvimento. De acordo com Sommerville (2011), as empresas precisam responder a novas oportunidades e novos mercados, a mudanças nas condições econômicas e ao surgimento de produtos e serviços concorrentes.

No entanto, mesmo com a crescente adoção de metodologias ágeis, a validação e verificação de software continuam sendo desafios significativos. Para Sommerville (2011), a verificação e validação (V&V), tem a intenção de mostrar que um software se adequa a suas especificações ao mesmo tempo que satisfaz as especificações do cliente do sistema. Esta definição dá uma perspectiva que a validação refere-se ao processo de garantir que o software atenda aos requisitos e expectativas dos usuários, enquanto a verificação se concentra em confirmar se o software foi desenvolvido corretamente de acordo com as especificações.

A validação e verificação de software têm grande importância, pois erros ou falhas podem levar a consequências graves, como perda de dados, prejuízos financeiros e até mesmo riscos à segurança dos usuários. Segundo Patton (2018), os custos são logarítmicos, ou seja, aumentam dez vezes com o aumento do tempo. Quanto mais cedo o *bug* for encontrado, menor será o prejuízo. De acordo com Laporte (2018), aproximadamente 70% dos defeitos são inseridos antes mesmo de qualquer linha de código ser produzida. Portanto, é fundamental adotar abordagens e técnicas adequadas para melhorar a qualidade do software.

Ao longo deste Trabalho de Conclusão de Curso (TCC), foi evidenciado como as metodologias ágeis se relacionam com a validação e verificação de software, por meio de uma revisão literária utilizando o auxílio de *string* de busca. Após a identificação dos artigos que se relacionavam com o tema, foi feito um refinamento da seleção por meio de critérios de inclusão e de exclusão, de modo a aproximar as referências adotadas do contexto do estudo de caso. Em seguida, foi realizado um levantamento de quais os principais gargalos enfrentados no contexto real de uma empresa que utiliza metodologias ágeis. Por fim, foi feita uma

especialização de um modelo proposto na literatura pelo Saboia et al. (2019) , de tal forma que se adequasse à necessidade da empresa.

Nesse contexto, o trabalho busca responder às seguintes questões de pesquisa:

Q1. Quais os principais problemas enfrentados pelas empresas de desenvolvimento de software que utilizam metodologias ágeis na etapa de Verificação e Validação?

Q2. Quais são as técnicas de Verificação e Validação que podem ser aplicadas no desenvolvimento de software de empresas que utilizam metodologias ágeis?

Dessa forma, este trabalho busca contribuir como um relato na aplicação de técnicas de V&V, podendo favorecer de forma direta o enfrentamento de problemas que ocorrem durante o ciclo de desenvolvimento de software em um contexto real de uma empresa que utiliza metodologias ágeis.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Nesta seção, serão apresentados alguns conceitos importantes e necessários para o entendimento desse estudo de caso. Nas seções 2.1 e 2.2, são aprofundados temas relacionados às metodologia ágeis e verificação e validação, respectivamente.

### **2.1 Metodologias Ágeis**

A utilização de metodologias ágeis vêm sendo incorporadas pelas empresas de desenvolvimento de software que buscam cada vez mais valorizar o produto a ser entregue ao cliente do que processos e ferramentas. Com isso, a implantação de processos torna-se cada vez mais comum entre a indústria.

Segundo Raul Wazlawic (2013), processo é um conjunto de regras que definem como um projeto deve ser executado. Um processo normalmente é adotado por uma empresa como um conjunto de regras específicas que seus funcionários devem seguir sempre que trabalharem em um projeto. Assim, quando um projeto precisa ser planejado na empresa, o responsável deve tomar o processo definido e, a partir dele, definir as atividades concretas, prazos e responsáveis.

Dentro os modelos de processos ágeis bastante utilizados pelas empresas de

tecnologia, destaca-se o *Scrum*. Para Raul Wazlawic (2013), o *Scrum* é um método um tanto mais radical em relação a fugir de prescrições e gerenciamento em função de investir na equipe para que ela possa se auto-organizar.

O autor destaca, ainda, que é possível delinear três perfis nesse modelo:

- a) O *Scrum master*, que não é um gerente no sentido dos modelos prescritivos. Não é um líder, já que as equipes são auto-organizadas, mas um facilitador (pessoa que conhece bem o modelo) e um solucionador de conflitos.
- b) O *product owner*, ou seja, a pessoa responsável pelo projeto em si. Tem, entre outras atribuições, a de indicar quais são os requisitos mais importantes a serem tratados em cada sprint. O *product owner* é o responsável pelo ROI (Return Of Investment) e também por conhecer e avaliar as necessidades do cliente.
- c) O *Scrum team*, que é a equipe de desenvolvimento. Essa equipe não é necessariamente dividida em papéis como analista, designer e programador, mas todos interagem para desenvolver o produto em conjunto. Em geral são recomendadas equipes de 6 a 10 pessoas. (WAZLAWIC, 2013, p.56)

Ainda nesse contexto, vale destacar o *Product Backlog* que, conforme aponta Wazlawic, concentra as funcionalidades e requisitos a serem implementadas em cada projeto. No âmbito das metodologias ágeis, isso significa que as adaptações ocorrem e são identificadas ao longo da execução do projeto, podendo haver a incorporação de novas funcionalidades.

Para compreender esse modelo, outro conceito fundamental é o *Sprint*, o ciclo de desenvolvimento curto em que se estrutura o *Scrum*. Com base nos requisitos delineados no *product backlog*, é feito um processo de priorização para o *sprint backlog*, com a reorganização dos requisitos de acordo com uma visão mais voltada ao trabalho da equipe de desenvolvimento, o que possibilita que as atividades sejam implementadas em prazos mais curtos e com maior celeridade.

O Manifesto Ágil foi criado por um grupo de desenvolvedores de software que buscavam uma abordagem mais adaptável, colaborativa e centrada no cliente para o desenvolvimento de projetos. A partir desses valores e princípios, diversas metodologias ágeis foram desenvolvidas, como Scrum, Kanban, XP (Extreme Programming), entre outras.

Os valores são:

Indivíduos e interações mais que processos e ferramentas.

Software em funcionamento mais que documentação abrangente.

Colaboração com o cliente mais que negociação contratual.

Responder a mudanças mais que seguir um plano

Com base nesses conceitos, serão abordados na seção seguinte as etapas de verificação e validação.

## **2.2 Verificação e Validação**

A validação e verificação de software são atividades essenciais no desenvolvimento de sistemas de software. Embora sejam conceitualmente consideradas coisas diferentes, têm como principal objetivo garantir a qualidade e a confiabilidade dos produtos de software. Essas atividades envolvem aplicação de métodos, técnicas e ferramentas para avaliar se um sistema de software atende aos requisitos especificados e está livre de defeitos.

De acordo com Koscianski e Soares (2007), o objetivo do teste é encontrar defeitos, de forma a expor que o comportamento do software, em algumas situações, pode não estar em conformidade.

A ISO 9000 (2015) define verificação como a confirmação, por meio do fornecimento de evidências objetivas, de que os requisitos especificados foram atendidos. De forma análoga, a IEEE<sup>1</sup> 2012 define verificação como o processo de avaliação de um sistema ou componente para determinar se os produtos de uma dada fase de desenvolvimento satisfazem as condições impostas no início dessa fase. E de forma mais sucinta, Raul Wazlawic (2013), considera a verificação como sendo a análise do software para ver se ele está sendo construído de acordo com o que foi especificado. De forma complementar, Terra e Bigonha (2008) considera que a validação se destina a mostrar que o software realiza exatamente o que o usuário espera que ele faça.

Dentre um dos métodos mais tradicionais de utilização de técnica de V&V, pode ser destacada a utilização da inspeção, cujo principal objetivo é a detecção e remoção de possíveis anomalias nos artefatos produzidos durante o processo de desenvolvimento. Para Jones (1991), o esforço com retrabalho é reduzido em média para 10% a 20% do esforço total de desenvolvimento. Dessa forma, melhorando significativamente o desenvolvimento do software.

---

<sup>1</sup> IEEE: Institute of Electrical and Electronic Engineers

### **3 TRABALHOS RELACIONADOS**

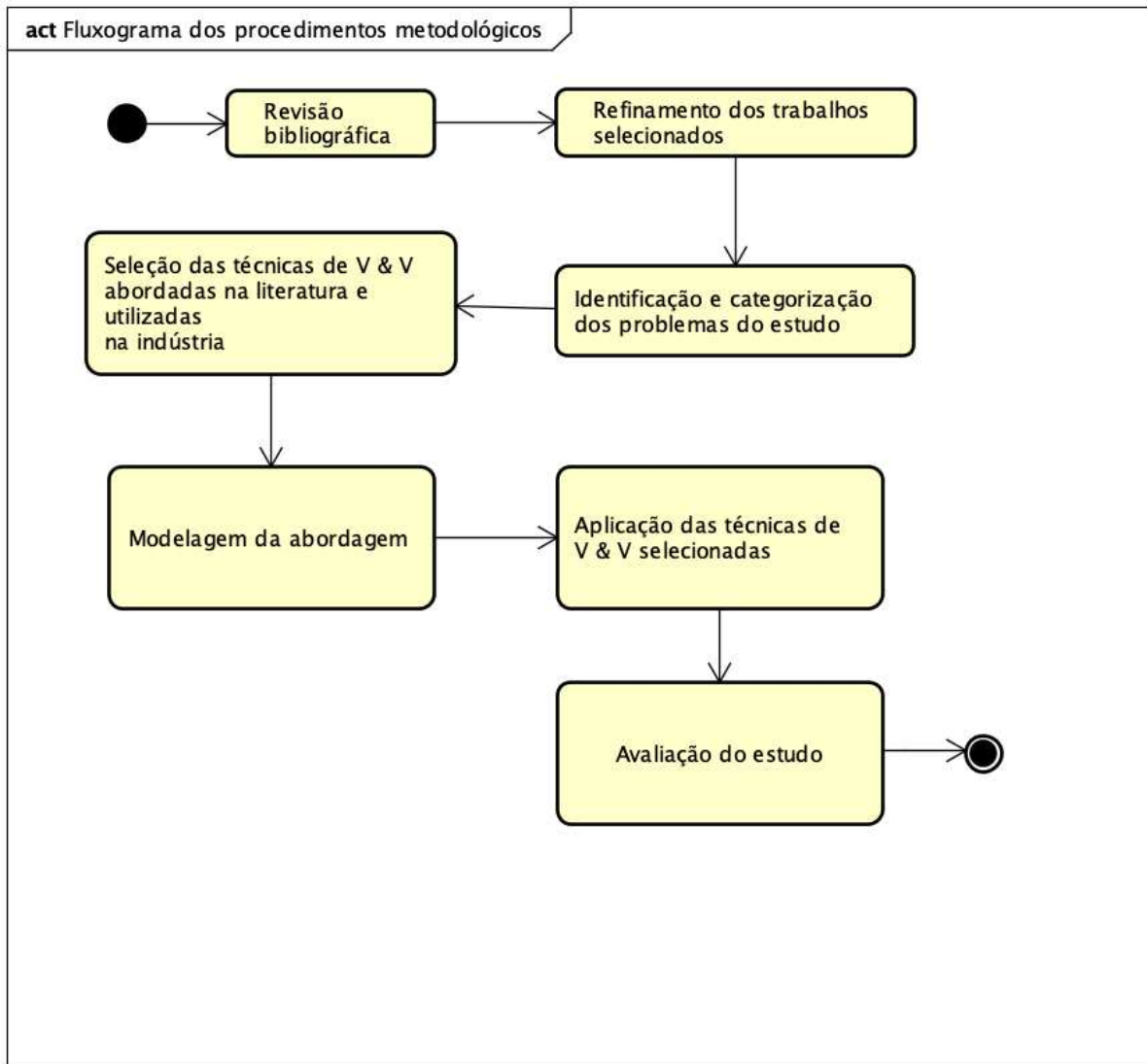
Na revisão dos trabalhos relacionados, foram analisadas diversas pesquisas, artigos científicos e projetos desenvolvidos na área em questão. Os estudos investigados abordaram temas como V&V e metodologias ágeis, explorando diferentes abordagens, metodologias e técnicas. Além disso, foram identificados trabalhos que tratam de questões semelhantes ou complementares ao objeto de estudo deste trabalho, oferecendo contribuições relevantes para o desenvolvimento da análise. A revisão dos trabalhos permitiu uma compreensão aprofundada, fornecendo uma base sólida para o desenvolvimento deste trabalho.

No artigo de Jarbele, Wilkerson e Patrícia (2019) são investigadas as práticas, estratégias e ferramentas na associação de temas relacionados a Engenharia de Requisitos com Teste de Software por meio de uma revisão bibliográfica. A partir dos 1099 artigos identificados pela *String* de Busca, foram selecionados apenas 14 documentos para serem analisados de forma mais aprofundada. Nessa análise mais rebuscada, observou-se a realização de reuniões semanais com os *stakeholders*, utilização dos Casos de Usos e documentação de Casos de Testes bem como a utilização de estratégias como *Acceptance Test Driven Development* (ATDD).

O trabalho de Saha (2008), apresenta uma abordagem conceitual sobre a definição de teste de software. Aborda quais os principais objetivos em um teste de software e qual seu custo, fazendo um comparativo entre os testes manuais e automatizados.

### **4 PROCEDIMENTOS METODOLÓGICOS**

Nesta seção, são apresentadas e detalhadas as atividades realizadas para a condução deste trabalho. A Figura 1 mostra detalhadamente o fluxo de atividades realizadas.



**Figura 1 - Fluxo de Atividades**

Inicialmente, foi realizada a **revisão bibliográfica** que tem como principal objetivo delimitar a identificação do problema a ser estudado neste trabalho. Para isso, foram definidas as questões de pesquisas a serem investigadas e formulada a *string* de busca que auxiliou na busca dos artigos a serem utilizados na revisão bibliográfica. As questões de pesquisa e a *string* de busca utilizada foram, respectivamente:

Q1. Quais os principais problemas enfrentados pelas empresas de desenvolvimento de software que utilizam metodologias ágeis na etapa de Verificação e Validação?

Q2. Quais são as técnicas de Verificação e Validação que podem ser aplicadas no desenvolvimento de software de empresas que utilizam metodologias ágeis?

*((problem OR difficult\* OR issue OR obstacle OR challenge OR "lessons learned" OR "test debt") AND ("software development" OR "software factory" OR "test factory") AND (agile OR scrum OR "extreme programming" OR "test process" OR "testing process" OR "testing processes" OR "test process")) AND (test OR testing OR verification OR validation OR inspection OR review))*

A *string* de busca foi executada na base de dados da ACM. Para melhorar a precisão da seleção dos artigos<sup>2</sup>, foi utilizado como filtro apenas o *abstract* de cada resumo, retornando assim um total de 130 artigos.

Posteriormente, foi realizado o **refinamento dos trabalhos selecionados** utilizando dois critérios, um de inclusão e um de exclusão, a saber:

- Critério de inclusão: Estudos que abordam técnicas de V&V para desenvolvimento de software ágil.
- Critério de exclusão: Excluir estudos que não são adequados para o contexto específico ou que não são relevantes para os objetivos do projeto.

Após o refinamento, foram obtidos 2 trabalhos relacionados ao tema que abordam lacunas na pesquisa existente e apenas 1 artigo foi utilizado no contexto do estudo de caso.


**Tabela 1 - Total de estudos da Revisão Bibliográfica**

| BASE | RETORNADOS | RELACIONADOS AO TEMA | UTILIZADO NO ESTUDO |
|------|------------|----------------------|---------------------|
| ACM  | 130        | 2                    | 1                   |

Em seguida, foi realizada a **identificação e categorização dos problemas do estudo**<sup>3</sup>, levantando quais os principais problemas que ocorriam na fase de testes durante o ciclo de desenvolvimento de um software, mais especificamente na empresa Credit2B Soluções em Tecnologia<sup>4</sup> que utiliza metodologias ágeis, e suas ocorrências durante o tempo de três *Sprints*, com duração de 1 mês, entre os meses de janeiro a março do ano de 2023.

Depois, foi feita a **seleção das técnicas de V&V abordadas na literatura e**

<sup>2</sup> Planilha com os artigos selecionados:  Artigos.xlsx

<sup>3</sup> Planilha com os artigos categorizados:  Relatório de Inicial.xlsx

<sup>4</sup> <https://www.credit2b.com.br/>

**utilizadas na indústria**, fornecendo uma base teórica sólida para o estudo e servindo de apoio à metodologia, provendo exemplos de estruturas utilizadas em pesquisas semelhantes.

Após a categorização, houve a **modelagem da abordagem**, que tinha como principal objetivo definir e documentar o processo, incluindo as atividades, técnicas e ferramentas a serem utilizadas.

Logo depois, foi realizada a **aplicação das técnicas de V&V selecionadas** anteriormente, adaptando os métodos e procedimentos de pesquisa estabelecidos por outros pesquisadores para o contexto específico desse estudo de caso.

Por fim, na **avaliação do estudo**, foi analisado o impacto da utilização de técnicas e processos propostos pela literatura em um contexto real. Para isso, foi realizado um novo levantamento durante uma *Sprint*, de duração de 1 mês, entre o mês de maio, cujo objetivo era mensurar o impacto do estudo realizado.

## **5 ESTUDO DE CASO**

Esta seção foi organizada da seguinte forma: Subseção 5.1 apresenta como foi concebida a abordagem, com base na revisão literária e no estudo do processo de desenvolvimento da empresa Credit2B. A subseção 5.2 descreve com detalhes quais os procedimentos, técnicas e instrumentos foram utilizados para coletar, analisar e aplicar no processo de V&V no estudo de caso.

### **5.1 Concepção da abordagem**

O trabalho apresentado em Saboia et al. (2019) serviu como base para a definição das atividades do processo de teste. Ele fornece uma concepção na segmentação do processo de teste e a qualificação e quantificação das dívidas de teste. O trabalho relata como a fase de testes é segmentada, em três grandes fases, sendo elas Planejamento, Elaboração e Execução, conforme descrito abaixo:

- Planejamento: Tem por finalidade planejar o teste de uma determinada demanda. No planejamento são identificados quais tipos de testes devem ser executados, o cronograma a ser seguido e os principais riscos. As

principais atividades dessa fase são: (i) Iniciar projeto; (ii) Planejar projeto; e (iii) Monitorar projeto nos marcos especificados.

- **Elaboração:** Nessa fase são elaborados os cenários e casos de testes para a demanda de teste. Também está prevista a revisão dos testes elaborados. Essa fase desenvolve os insu- mos para a fase de Execução. As principais atividades são: (i) Iniciar fase; (ii) Projetar testes; (iii) Criar scripts de testes automatizados (se aplicável); e (iv) Monitorar projeto nos marcos especificados.

- **Execução:** Objetiva executar os testes previamente elaborados e reportar os resultados obtidos. Nessa fase também é finalizado o projeto de teste com a documentação das lições aprendidas. As principais atividades são: (i) Iniciar fase; (ii) Executar testes; (iii) Analisar incidentes de teste; e (iv) Finalizar projeto de teste.

Nesse mesmo artigo, o autor aborda o conceito de dívida de teste e propõe uma abordagem para identificar e gerenciar essas dívidas em uma fábrica de testes. A identificação das dívidas, sua categorização e o estabelecimento de um plano de ação para corrigi-las são etapas essenciais para melhorar a qualidade e a eficiência dos testes de software realizados nesse ambiente.

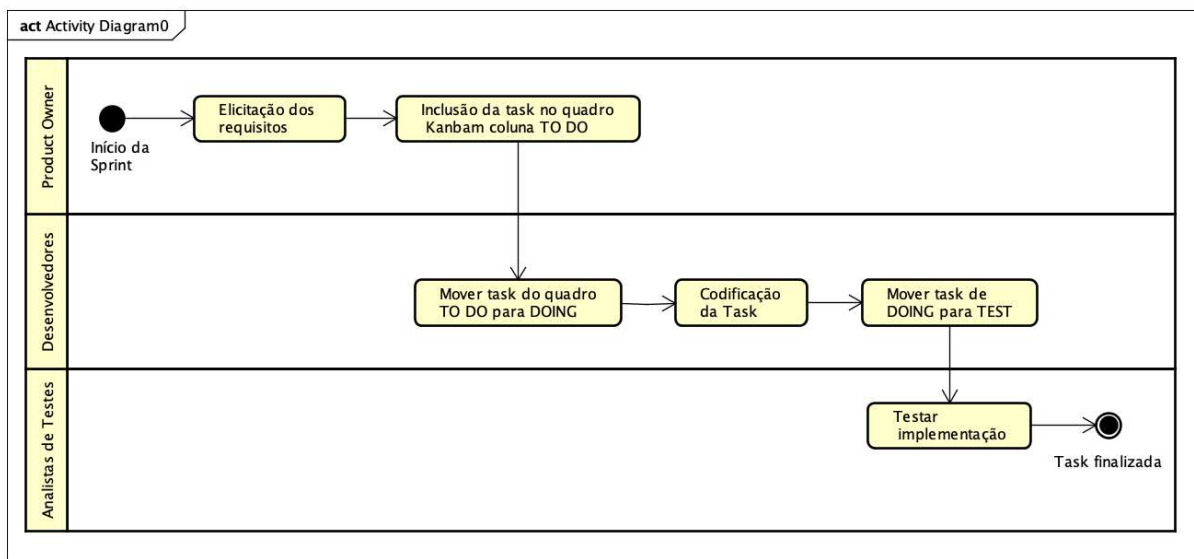
A concepção envolveu uma pesquisa de 3 meses, abrangendo 3 *sprints*, da qual foram extraídas por meio da ferramenta de gerenciamento de tarefas ClickUp, as tarefas relacionadas ao processo de desenvolvimento, para que pudessem ser qualificadas de acordo com seu gênero e quantificadas.

## **5.2 Descrição da abordagem**

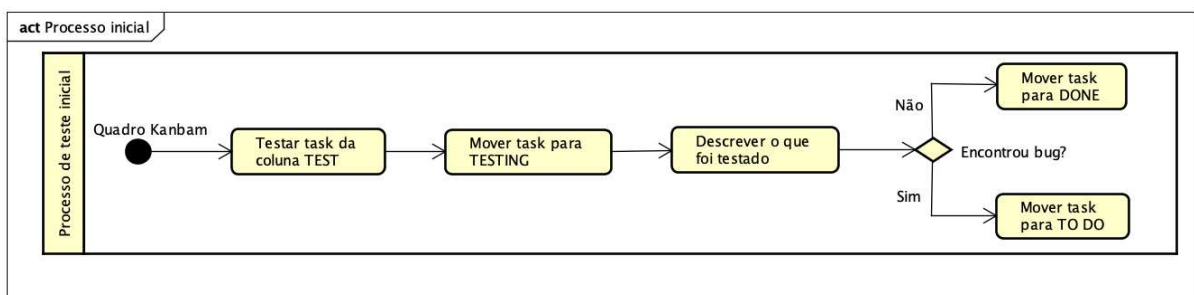
Nesta seção, será apresentado como a abordagem foi utilizada pelo trabalho, sendo dividida em duas subseções. A 5.2.1 apresenta o atual processo da empresa e a 5.2.2 apresenta o modelo especializado a partir do modelo proposto no artigo do Saboia et al. (2019).

### **5.2.1 Modelo Atual**

A Figura 2 ilustra o modelo de processo utilizado atualmente pelo estudo de caso na empresa Credit2B e a Figura 3 ilustra o subprocesso de testes.



**Figura 2 - Modelo atual do processo utilizado**



**Figura 3 - Subprocesso de teste do modelo atual**

Conforme ilustram as Figuras 2 e 3, a *sprint* se inicia com a definição do *Backlog da Sprint*, onde todas as tarefas (*tasks*) que irão compor o produto durante aquele *sprint* corrente são elucidadas pelo dono do Produto (*Product Owner*).

Logo após a documentação completa de todas as *tasks*, as demandas são inseridas em um quadro Kanban que contém as seguintes colunas: TO DO; DOING; TEST; e DONE. Esse quadro é monitorado por parte do time, ficando sob responsabilidade do time de desenvolvimento a movimentação das *tasks* que se encontram na coluna *TO DO* para *DOING* utilizando metodologia de versionamento por meio do *GitFlow*<sup>5</sup>. A partir do momento em que a *task* é finalizada, o desenvolvimento fica responsável de move-lá para a coluna *TEST*.

<sup>5</sup> É um modelo de fluxo de trabalho que busca simplificar e organizar o versionamento de ramificações de um projeto de desenvolvimento no Git.

Dessa forma, o time de testes fica responsável pela validação da demanda realizando testes manuais na aplicação e, caso seja aprovada, a demanda é movida para a coluna *DONE*, caso contrário, a *task* é retornada para a coluna *TO DO*, onde o *bug* é especificado pelo time de testes e corrigido pelo time de desenvolvimento.

### 5.2.2 Modelo do processo de teste especializado

Com base na segmentação da fase de teste proposta pelo artigo do Saboia et al. (2019), foi definido um modelo que se adequasse ao contexto da Credit2B. A Figura 4 ilustra o modelo proposto pelo o estudo.

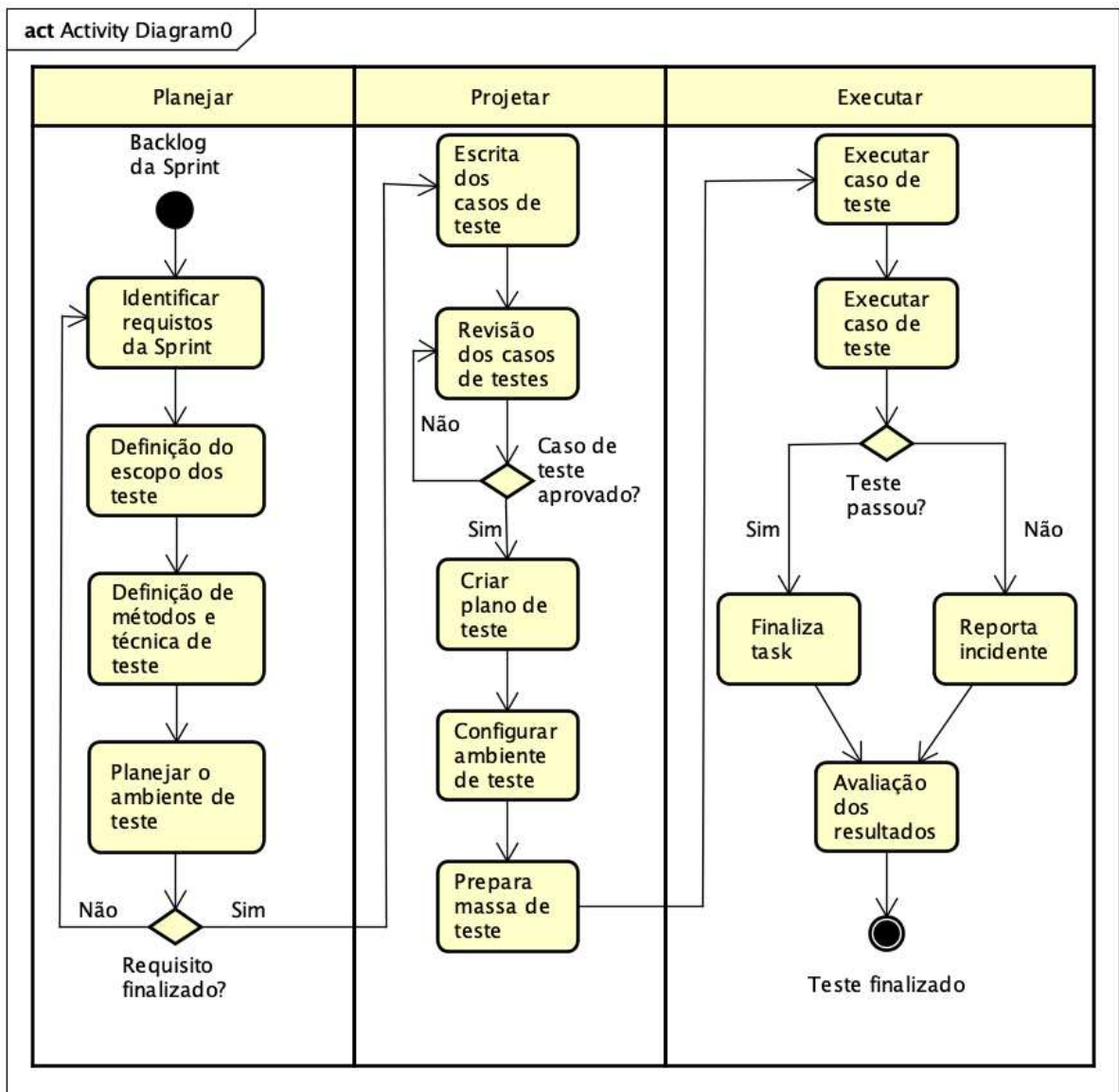


Figura 4 - Modelo especializado

Cada fase do modelo especializado segue detalhada nos tópicos abaixo:

A **Fase de Planejamento** é a fase inicial do processo, na qual são definidos os objetivos do teste, registrado através do artefato Plano de Teste. Essa etapa também é responsável pela identificação das técnicas de testes que serão aplicadas na *sprint* atual. Ela contém as seguintes tarefas:

- I. Identificar os requisitos da Sprint;
- II. Definir o escopo dos testes;
- III. Definir métodos e técnicas de testes;
- IV. Planejar o ambiente de teste.

Cada atividade é descrita no processo com base na sua descrição, pré-tarefa, critérios de entrada e saída, responsáveis, artefatos produzidos e requeridos, bem como as ferramentas e as pós-tarefas.

A seguir, cada tarefa da fase de planejamento é descrita de acordo com os campos citados, conforme detalhado nas tabelas 2, 3, 4 e 5.

Tabela 2 - Identificar os requisitos da *Sprint*

| <b>Tarefa</b>        | <b>Identificar Requisitos da Sprint</b>  |
|----------------------|--|
| Descrição            | Identificar os requisitos que serão implementados na Sprint. Deve ser realizado antes ou na primeira semana da <i>sprint</i> |
| Pré-tarefa           | Identificar Requisitos da <i>Sprint</i>  |
| Critérios de entrada | <i>Sprint Backlog</i> definido   |
| Critérios de saída   | Entendimento da lista de Requisitos a serem implementados na <i>Sprint</i>   |
| Responsáveis         | Analista de Teste  |
| Artefatos requeridos | Backlog da <i>Sprint</i>   |
| Artefatos produzidos | Nenhum   |
| Ferramentas          | Nenhum   |
| Pós-tarefa           | Definir Escopo dos Testes  |

Tabela 3 - Definir o escopo dos testes

| <b>Tarefa</b>        | <b>Definir Escopo dos Testes</b>   |
|----------------------|--|
| Descrição            | Identificar quais Tipos de Testes e atividades que devem ser realizadas durante a Sprint |
| Pré-tarefa           | Nenhum   |
| Critérios de entrada | Sprint Backlog e Objetivos da Sprint definidos   |
| Critérios de saída   | Lista de Tipos de Testes e atividades de Testes a serem realizadas na Sprint             |
| Responsáveis         | Líder Técnico, Analista de Testes  |
| Artefatos requeridos | Especificação de Requisitos, Sprint Backlog  |
| Artefatos produzidos | Lista de Tipos de Testes e atividades a serem realizados na Sprint                       |
| Ferramentas          | ClickUp ou Editor de texto   |
| Pós-tarefa           | Definir Métodos e Técnicas de Testes   |

Tabela 4 - Definir métodos e técnicas de testes

| <b>Tarefa</b>        | <b>Definir Métodos e Técnicas de Testes</b>   |
|----------------------|---|
| Descrição            | Definir quais métodos e técnicas de Testes serão aplicados para realizar os testes definidos na tarefa anterior |
| Pré-tarefa           | Definir Escopo dos Testes   |
| Critérios de entrada | Atividades de Testes da <i>Sprint</i> definidas   |
| Critérios de saída   | Métodos e técnicas de Testes definidos  |
| Responsáveis         | Analista de Testes, Líder Técnico   |
| Artefatos requeridos | Lista de Atividades de Testes a serem realizados na <i>Sprint</i>   |
| Artefatos produzidos | Lista de métodos e técnicas de testes a serem aplicados   |
| Ferramentas          | <i>ClickUp</i> ou Editor de texto   |
| Pós-tarefa           | Planejar Ambiente de Testes   |

Tabela 5 - Planejar ambiente de teste

| <b>Tarefa</b>        | <b>Planejar Ambiente de Teste</b>   |
|----------------------|---|
| Descrição            | Identificar quais as configurações necessárias para o Ambiente de Testes para que seja possível realizar as Atividades de Testes planejados para Sprint |
| Pré-tarefa           | Definir Métodos e Técnicas de Testes  |
| Critérios de entrada | Atividades de Testes da Sprint definidas  |
| Critérios de saída   | Especificação das configurações do Ambiente de Testes   |
| Responsáveis         | Líder Técnico, Analista de Testes   |
| Artefatos requeridos | Lista de Atividades de Testes a serem realizados na Sprint  |
| Artefatos produzidos | Lista de Configurações do Ambiente de Testes  |
| Ferramentas          | ClickUp ou Editor de texto  |
| Pós-tarefa           | Escrever Casos de Teste   |

A **Fase de Projetar** ocorre depois de realizado o planejamento, o próximo passo é colocar em prática tudo que foi especificado. É o momento de projetar os Casos de Testes e preparar o ambiente de execução dos testes. Contém as seguintes atividades:

- I. Escrever Casos de Teste;
- II. Revisar Casos de Teste;
- III. Gerenciar Ajustes;
- IV. Criar Plano de Testes;
- V. Configurar Ambiente de Teste;
- VI. Preparar Massa de Teste;

Cada etapa de atividade possui diversas tarefas fundamentais, conforme detalhamento realizado a seguir nas tabelas 6, 7, 8, 9, 10 e 11.

Tabela 6 - Escrever Casos de Teste

| <b>Tarefa</b>        | <b>Escrever Casos de Testes</b>  |
|----------------------|--|
| Descrição            | Criação dos casos de testes (definindo entradas, saídas esperadas e procedimentos de testes relacionados) com base nos Requisitos do Produto |
| Pré-tarefa           | Planejar Ambiente de Teste   |
| Critérios de entrada | Requisitos criados e validados com Time de Produto e Desenvolvimento   |
| Critérios de saída   | Casos de Testes especificados  |
| Responsáveis         | Analista de Testes   |
| Artefatos requeridos | Especificação de Requisitos do Produto   |
| Artefatos produzidos | Casos de Testes  |
| Ferramentas          | TestLink   |
| Pós-tarefa           | Revisar Casos de Testes  |

Tabela 7 - Revisar Casos de Teste

| <b>Tarefa</b>        | <b>Revisar Casos de Testes</b>  |
|----------------------|---|
| Descrição            | Revisar os Casos de Teste criados, revisando a consistência dos Cenários descritos e os Requisitos do Produto a ser testado |
| Pré-tarefa           | Escrever Casos de Testes  |
| Critérios de entrada | Casos de Testes especificados   |
| Critérios de saída   | Casos de Testes aprovados   |
| Responsáveis         | Um membro da equipe diferente do que elaborou os Casos de Testes  |
| Artefatos requeridos | Casos de Testes   |
| Artefatos produzidos | Laudo da revisão (Issue no ClickUP com indicações de melhorias)   |
| Ferramentas          | TestLink ou ClickUP   |

|            |  |
|------------|--|
| Pós-tarefa | Gerenciar Ajustes (caso existam não conformidades) |
|------------|--|

Tabela 8 - Gerenciar Ajustes

| <b>Tarefa</b>        | <b>Gerenciar ajustes</b>  |
|----------------------|---|
| Descrição            | O responsável pela qualidade controla até que todas as não conformidades tenham sido solucionadas |
| Pré-tarefa           | Revisar Casos de Testes   |
| Critérios de entrada | Ter sido realizada a verificação dos casos de teste e terem sido identificadas não conformidades  |
| Critérios de saída   | Não-conformidades resolvidas  |
| Responsáveis         | Um membro da equipe diferente do que elaborou os Casos de Testes                                  |
| Artefatos requeridos | Laudo da revisão  |
| Artefatos produzidos | Laudo da revisão atualizado (caso ainda existam) ou aprovação do caso de teste                    |
| Ferramentas          | TestLink ou ClickUP   |
| Pós-tarefa           | Variada   |

Tabela 9 - Criar Plano de Testes

| <b>Tarefa</b>        | <b>Criar Plano de Testes</b>  |
|----------------------|---|
| Descrição            | Nesta atividade é criado o Plano de Testes da Sprint contendo os Casos de Testes que devem ser executados no decorrer da mesma. |
| Pré-tarefa           | Revisar Casos de Testes   |
| Critérios de entrada | Casos de Testes aprovados   |
| Critérios de saída   | Plano de Testes da Sprint criado  |
| Responsáveis         | Analista de Testes  |
| Artefatos requeridos | Casos de Testes, Sprint Backlog   |
| Artefatos produzidos | Plano de Testes da Sprint   |
| Ferramentas          | TestLink  |

|            |                               |
|------------|-------------------------------|
| Pós-tarefa | Configurar ambiente de Testes |
|------------|-------------------------------|

Tabela 10 - Configurar Ambiente de Teste

| <b>Tarefa</b>        | <b>Configurar ambiente de Testes</b>   |
|----------------------|--|
| Descrição            | Nesta atividade deve-se configurar um ambiente de testes isolado do ambiente de desenvolvimento. |
| Pré-tarefa           | Criar Plano de Testes  |
| Crítérios de entrada | Casos de Testes aprovados  |
| Crítérios de saída   | Ambiente de Testes configurado   |
| Responsáveis         | Analista de Testes   |
| Artefatos requeridos | Lista de Configurações do Ambiente de Testes   |
| Artefatos produzidos | Nenhum   |
| Ferramentas          | Nenhum   |
| Pós-tarefa           | Preparar massa de testes (se pertinente) ou Executar Casos de Testes                             |

Tabela 11 - Preparar Massa de Teste

| <b>Tarefa</b>        | <b>Elaborar Casos de Testes</b>  |
|----------------------|--|
| Descrição            | Preparar a massa de dados necessária para executar os testes, fundamentando-se nos Casos de Teste. |
| Pré-tarefa           | Criar ambiente de teste  |
| Crítérios de entrada | Ambiente de teste criado   |
| Crítérios de saída   | Ambiente de teste com massa de teste carregada   |
| Responsáveis         | Analista de Testes   |
| Artefatos requeridos | Casos de Testes  |
| Artefatos produzidos | Nenhum   |
| Ferramentas          | Nenhum   |
| Pós-tarefa           | Executar Casos de Testes ou Criar scripts de Testes  |

|  |                               |
|--|-------------------------------|
|  | Automatizados (se pertinente) |
|--|-------------------------------|

Após a fase e Projetar é conduzida a **fase Executar**. O objetivo dessa fase é colocar em prática tanto o que foi planejado quanto o que foi projetado no processo. É o momento de executar os testes. Um cuidado especial deve ser tomado antes de iniciar essa etapa. É preciso verificar se o ambiente está propício e estável, para que a execução dos testes seja possível. As tabelas 12, 13 e 14 apresentam de forma detalhada cada passo.

- I. Executar Casos de Teste;
- II. Reportar de Defeitos;
- III. Avaliar Resultados do Teste.

Tabela 12 - Executar Casos de Teste

| <b>Tarefa</b>        | <b>Executar Casos de Testes</b>  |
|----------------------|--|
| Descrição            | Executar os casos de teste manualmente ou os scripts de teste automatizado, verificando se os resultados observados correspondem ao resultado esperado |
| Pré-tarefa           | Configurar Ambiente de Testes  |
| Critérios de entrada | Ambiente de teste criado, Massa de Dados criada (se pertinente), Casos de Testes aprovado  |
| Critérios de saída   | Casos de Testes executados   |
| Responsáveis         | Testador, Analista de Testes   |
| Artefatos requeridos | Casos de Testes  |
| Artefatos produzidos | Registro de execução de teste (Log de Testes no Testlink)  |
| Ferramentas          | TestLink   |
| Pós-tarefa           | Reportar Defeitos  |

Tabela 13 - Reportar de Defeitos

| <b>Tarefa</b>        | <b>Reportar Defeitos</b>  |
|----------------------|---|
| Descrição            | Documentar os defeitos encontrados durante a execução dos Casos de Testes |
| Pré-tarefa           | Executar Casos de Testes  |
| Critérios de entrada | Casos de teste executados que tenham apresentado qualquer                 |

|                      |   |
|----------------------|---|
|                      | diferença entre resultado observado e planejado |
| Critérios de saída   | Defeitos registrados                            |
| Responsáveis         | Testador, Analista de Testes                    |
| Artefatos requeridos | Registro de execução de teste                   |
| Artefatos produzidos | Relatório de defeitos de testes (issue bug)     |
| Ferramentas          | ClickUp   |
| Pós-tarefa           | Avaliar Resultados do Teste                     |

Tabela 14 - Avaliar Resultados do Teste

| <b>Tarefa</b>        | <b>Avaliar Resultados do Teste</b>   |
|----------------------|--|
| Descrição            | O Líder de Testes deve analisar todos os defeitos para verificar o que pode ser decorrente de uma situação específica ocorrida durante os testes ou realmente de falhas decorrentes de defeitos. |
| Pré-tarefa           | Reportar Defeitos  |
| Critérios de entrada | Casos de Testes executados, Defeitos reportados  |
| Critérios de saída   | Defeitos analisados e issues no ClickUP atualizadas  |
| Responsáveis         | Líder Técnico, Analista de Testes  |
| Artefatos requeridos | Relatório de defeitos de testes (issue bug)  |
| Artefatos produzidos | Nenhum   |
| Ferramentas          | ClickUp  |
| Pós-tarefa           | Variada  |

O objetivo da **Fase de Executar** é assegurar que os testes foram executados conforme projetado, permitindo assim que o sistema esteja apto para entrega e garantindo que os objetivos planejados foram alcançados. As tabelas 15 e 16 mostram de forma detalhada cada passo.

- I. Elaboração e Análise dos Indicadores de Teste;
- II. Entrega do Sistema para o Cliente.

Tabela 15 - Elaboração e Análise dos Indicadores de Teste

| <b>Tarefa</b>        | <b>Elaboração e Análise dos Indicadores de Teste</b>   |
|----------------------|--|
| Descrição            | Analisar os resultados obtidos no ciclo de testes executados, observando pontos que causaram problemas ou gargalos e que possam ser melhorados na próxima sprint |
| Pré-tarefa           | Avaliar Resultados do Teste  |
| Critérios de entrada | Testes da sprint finalizados   |
| Critérios de saída   | Pontos de melhoria para próxima sprint ou atividades benéficas que devem ser mantidas  |
| Responsáveis         | Todos os integrante do Time de Testes  |
| Artefatos requeridos | Qualquer artefato um dos artefatos gerados durante a sprint, depende do contexto em que o projeto se encontra.   |
| Artefatos produzidos | Nenhum   |
| Ferramentas          | Miro, Planilha, editor de texto, etc.  |
| Pós-tarefa           | Entrega do Sistema para o Cliente  |

Tabela 16 - Entrega do Sistema para o Cliente

| <b>Tarefa</b>        | <b>Entrega do Sistema para o Cliente</b>   |
|----------------------|--|
| Descrição            | Acompanhar a entrega do sistema (versão) ao cliente, garantindo que a implantação ocorreu com sucesso e registrando o possível surgimento de bugs. |
| Pré-tarefa           | Avaliar Resultados do Teste ou Elaboração e Análise dos Indicadores de Teste   |
|                      |  |
| Critérios de entrada | Testes da versão finalizado e versão pronta para ser integrada ao ambiente de produção   |
| Critérios de saída   | Versão rodando em produção   |
| Responsáveis         | Analista de Testes   |
| Artefatos requeridos | Nenhum   |
| Artefatos produzidos | Issues de bug (Se houverem bugs em produção)   |

|             |        |
|-------------|--------|
| Ferramentas | Nenhum |
| Pós-tarefa  | Nenhum |

## 6 AVALIAÇÃO

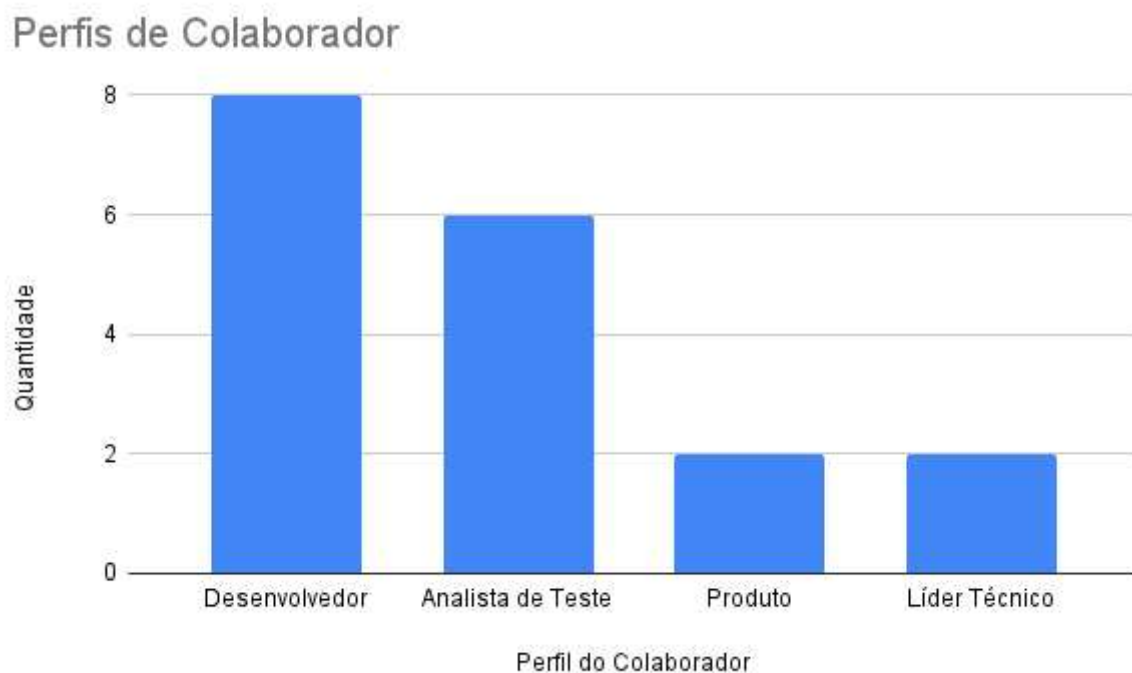
Após o lançamento do primeiro modelo especializado para o contexto do estudo de caso, foi realizado um treinamento interno para 2 *Squads*<sup>6</sup> do time de tecnologia, explicando toda a metodologia envolvida no processo e a execução do arcabouço ao longo de 3 *Sprints* (3 meses), compreendendo entre os meses de abril, maio e junho. O objetivo desse experimento é medir os resultados obtidos e avaliar o modelo.

Inicialmente, foi realizado um levantamento dos tipos de perfis de colaboradores existentes nos times por meio de um formulário<sup>7</sup>. A distribuição dos perfis, conforme ilustrado na figura 5, foi apontada da seguinte forma: 8 desenvolvedores, 6 testadores, 2 analistas de produto e 2 líderes técnicos, conforme apresentado na figura 5.

---

<sup>6</sup> É um pequeno grupo multidisciplinar, ou seja, uma equipe enxuta, mas composta por colaboradores com qualidades e competências diferentes.

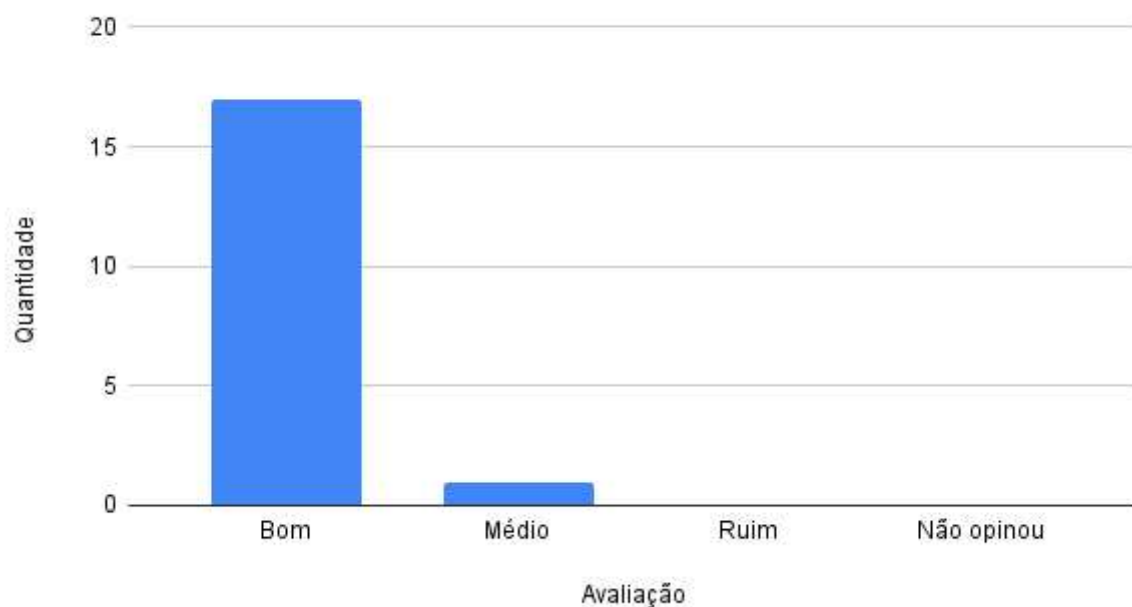
<sup>7</sup> Link do formulário: <https://forms.gle/Wv5xChCfgHgnweeq9>



**Figura 5:** Perfil de Colaboradores

O objetivo desse levantamento é ressaltar que a percepção de perspectiva do experimento pode diferir de acordo com o perfil. Diante disso, foi feita uma avaliação através de um questionário a respeito do que cada colaborador achou sobre a aplicação prática do modelo, conforme pode ser ilustrado pela Figura 6.

## Avaliação da pesquisa



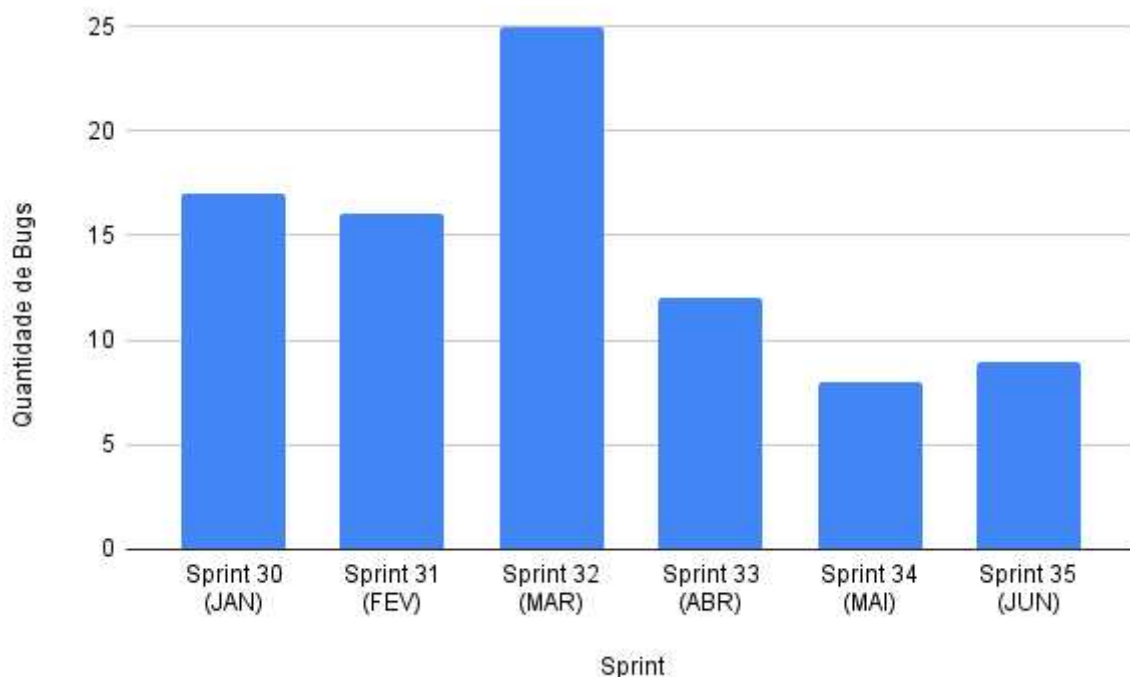
**Figura 6:** Avaliação dos colaboradores

Em relação a avaliação do modelo utilizado nas três *Sprints*, constatou-se que 17 dos 18 colaboradores avaliaram o modelo como bom, porém apenas 1 participante opinou como média. De forma subjetiva, na seção de comentário do formulário, foi analisado o motivo pelo qual essa resposta ficou fora da curva e constatou-se que o mesmo achou o processo mais demorado, impactando diretamente em um resultado mais expressivo na *Sprint*.

Após a análise de percepção do modelo por parte das duas *Squads*, foi realizado um comparativo entre o levantamento de *bugs* realizado no início da abordagem<sup>8</sup>, compreendido entre as 3 primeiras *Sprints* do ano de 2023, antes da aplicação do modelo, com os dados apontados<sup>9</sup> nas 3 últimas *Sprints*, após a aplicação do processo especializado.a. A Figura 6 mostra de forma quantitativa os dados colhidos antes e após o processo.

<sup>8</sup> [Relatório Inicial.xlsx](#)

<sup>9</sup> [Relatório Final.xlsx](#)



Fonte: elaborado pelo autor.

**Figura 6:** Avaliação do experimento

Com base nos resultados obtidos durante as 3 *sprints* da aplicação, observa-se com clareza que a quantidade de bugs diminuiu significativamente, logo pode-se inferir que a aplicação do modelo proposto pelo Saboia et al. (2019) serviu como base para mitigar possíveis erros ocorridos durante o processo de desenvolvimento..

Com base na análise realizada, foi possível elucidar as questões de pesquisa apresentadas na Seção 1 (Introdução), chegando às seguintes conclusões:

**Q1. Quais os principais problemas enfrentados pelas empresas de desenvolvimento de software que utilizam metodologias ágeis na etapa de Verificação e Validação?**

Os custos envolvidos com testes na fase de desenvolvimento acabam fazendo com que as empresas supram a utilização de técnicas a fim de agilizar a entrega do produto, tornando-se esse um dos principais motivos de gargalo.

**Q2. Quais são as técnicas de Verificação e Validação que podem ser aplicadas no**

## **desenvolvimento de software de empresas que utilizam metodologias ágeis?**

Com base no artigo do modelo proposto pelo Saboia et al. (2019) e com o resultado da aplicação do processo de forma especializada para a empresa, foi identificado que a utilização da técnica de revisão pode contribuir para evitar a presença de possíveis *bugs* e mitigar retrabalho para o time de desenvolvimento, validando dessa forma a o pensamento de Laporte (2018), onde aproximadamente 70% dos defeitos são inseridos antes mesmo de qualquer linha de código ser produzida.

## **7 CONCLUSÃO**

A partir deste estudo foi possível identificar quais eram os principais entraves que ocorriam durante o processo de desenvolvimento de software que acabavam injetando *bugs* no código do produto, deixando-o suscetível a falhas e como a literatura poderia auxiliar a propor um modelo viável que pudesse mitigar grande parte dos erros identificados e melhorar as métricas com base nos atributos identificados.

## REFERÊNCIAS

Sommerville, Ian Engenharia de Software / Ian Sommerville; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiramã. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.

Laporte, Claude Y., author. | April, Alain, author. Software quality assurance / by Claude Y. Laporte, Alain April.

Wazlawick, Raul Sidnei, 1967- Engenharia de software: conceitos e práticas / Raul Sidnei Wazlawick. - Rio de Janeiro p.11-14

Bruno Sabóia Aragão, Rute Nogueira Silveira de Castro, Ismayle de Sousa Santos, Valéria Lelli, and Rossana M. C. Andrade. 2019. Test debts identification in a test factory. In XVIII Brazilian Symposium on Software Quality (SBQS'19), October 28-November 1, 2019, Fortaleza, Brazil. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3364641.3364676>

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 9000: Sistemas de gestão da qualidade — Fundamentos e vocabulário. Rio de Janeiro. 2015.

IEEE Std1012-2012.IEEEStandardforSystemandSoftwareVerification and Validation, IEEE, The Institute of Electrical and Electronics Engineers, New York, NY, 2012.

Jarbele C. S. Coutinho, Wilkerson L. Andrade, and Patrícia D. L. Machado. 2019. Requirements Engineering and Software Testing in Agile Methodologies: a Systematic Mapping. In XXXIII Brazilian Symposium on Software Engineering (SBES 2019), September 23–27, 2019, Salvador, Brazil. ACM, New York, NY, USA, 10 pages.

Saha, Goutam Kumar. 2008. Understanding Software Testing Concepts

KOSCIANSKI, A.; SOARES, M. dos S. Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2a. ed. Novatec, 2007

TERRA, R.; BIGONHA, R. S. Ferramentas para análise estática de códigos java. In: III Encontro Brasileiro de Teste de Software. Recife: EBTS, 2008. p. 1–5.

JONES, C., 1991, “Applied Software Measurement”, McGraw Hill

PATTON, Ron. Software testing. Indianapolis: SamsPublishing, 2001. p.18