



# **PROJETO DE GRADUAÇÃO**

## **IMPLEMENTAÇÃO DE MÉTRICAS ÁGEIS NO DESENVOLVIMENTO DE *SOFTWARE* APLICADO EM UM AMBIENTE EDUCACIONAL: UM ESTUDO DE CASO**

Por,

**Gabriel de Medeiros Jerônimo**

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Simone Borges Simão Monteiro

Brasília, 27 de setembro de 2022

**UNIVERSIDADE DE BRASÍLIA**

**Faculdade de Tecnologia**

**Departamento de Engenharia de Produção**

# PROJETO DE GRADUAÇÃO

## IMPLEMENTAÇÃO DE MÉTRICAS ÁGEIS NO DESENVOLVIMENTO DE *SOFTWARE* APLICADO EM UM AMBIENTE EDUCACIONAL: UM ESTUDO DE CASO

Por,

**Gabriel de Medeiros Jerônimo**

**15/0126212**

Relatório apresentado ao Departamento de Engenharia de Produção como requisito parcial para obtenção do grau de Engenheiro de Produção.

### **Banca Examinadora:**

Prof.<sup>a</sup> Dr.<sup>a</sup> Simone Borges Simão Monteiro (Orientadora)

Prof.<sup>a</sup> Isabela Neto Piccirillo (EESC-USP)

Prof.<sup>a</sup> MSc. Ana Cristina Fernandes Lima Goes (EPR/UNB).

**Brasília, 27 de setembro de 2022**

*"O óbvio é aquilo que ninguém enxerga, até que  
alguém o expresse com simplicidade".*

## AGRADECIMENTO

Primeiramente, quero agradecer a minha família por todo suporte dado nos tempos de graduação, permitindo que eu pudesse focar nos meus estudos. Esse agradecimento vai especialmente à minha mãe, Sandra, que sempre priorizou e investiu na minha educação e na de minhas irmãs, Brenda e Bárbara.

Quero agradecer também aos meus sócios, Enzo e Arthur, por me dar todo o apoio necessário neste período de finalização de curso, além da minha colega de trabalho, Sofia, pelo seu companheirismo e atenção.

Aos meus amigos, sou grato por trazerem leveza a um momento de intensa produção e esforço, me auxiliando na motivação de finalizar a graduação.

Agradeço à minha orientadora, Professora Simone Borges, por todo o seu zelo para comigo e com o meu projeto. Foi a sua pedagogia, dedicação e empatia que permitiram que eu chegasse até aqui. Um agradecimento especial também ao Professor Hilmer, que me permitiu aplicar este projeto nas turmas de suas disciplinas.

Agradeço a todos os estudantes envolvidos na Plataforma Unificada de Metodologia Ativa (PUMA) que viabilizaram este Estudo de Caso, especialmente ao Gustavo, estudante de Engenharia de Software que me auxiliou na implementação do código criado, através dos nossos pareamentos.

Por fim, sou grato a todos os estudantes e professores do meu curso que me permitiram conhecer de fato a Engenharia de Produção. Agradeço também ao Grupo Gestão, Empresa Júnior do curso, que me conectou com pessoas e com conhecimento essenciais para o profissional que sou hoje.

## RESUMO

Os métodos ágeis surgiram como uma maneira de lidar com a volatilidade do mercado e ascensão dos produtos digitais, ganhando assim uma grande representatividade desde a sua origem através do Manifesto Ágil. Contudo, por mais que sejam amplamente conhecidos, suas execuções muitas vezes são limitadas apenas a utilização de técnicas e cerimônias, não havendo a aplicação das métricas ágeis que poderiam contribuir com a qualidade da entrega. Dessa forma, o objetivo deste trabalho é elaborar uma ferramenta de visualização de métricas ágeis que possa ser utilizada no gerenciamento de projetos de desenvolvimento de software em um contexto universitário, a fim de contribuir com os fatores de sucesso no desenvolvimento do *webapp* Plataforma Unificada de Metodologia Ativa (PUMA). Para tanto, foi realizada uma pesquisa de natureza aplicada que utilizou como estratégia um estudo de caso onde, a partir de entrevistas semiestruturadas, mapeou-se as principais dificuldades relacionadas a tais métricas. Com isso, foi criado um *dashboard* atualizado simultaneamente com o Quadro Kanban do projeto, através de uma API do GitHub e ZenHub, garantindo maior visibilidade da operação do time de desenvolvimento para os gestores e clientes do projeto.

**Palavras-chaves:** *Agile; Software; Metric; GitHub; Performance.*

## **ABSTRACT**

Agile methods emerged as a way to deal with market volatility and the rise of digital products, thus gaining great representation since its origin through the Agile Manifesto. However, even though they are widely known, their executions are often limited to the use of techniques and ceremonies, without the application of agile metrics that could contribute to the quality of the delivery. Thus, the objective of this work is to elaborate a visualization tool of agile metrics that can be used in software development project management in a university context, in order to contribute with the success factors in the development of the PUMA webapp. To this end, an applied research was conducted that used as strategy a case study where, from semi-structured interviews, the main difficulties related to such metrics were mapped. As a result, a dashboard was created and updated simultaneously with the project's Kanban Board through a GitHub and ZenHub API, ensuring greater visibility of the development team's operation for the project's managers and customers.

**Keywords:** *Agile; Software; Metric; GitHub; Performance.*

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>11</b>
1.1 JUSTIFICATIVAS	13
1.2 OBJETIVOS	14
<b>1.2.1 Objetivo geral</b>	<b>14</b>
<b>1.2.2 Objetivo específico</b>	<b>14</b>
1.3 ESTRUTURAÇÃO DOS CAPÍTULOS	14
<b>2. LEVANTAMENTO DO ESTADO DA ARTE (TEMAC)</b>	<b>16</b>
2.1 PREPARAÇÃO DA PESQUISA	17
2.2 APRESENTAÇÃO E INTER-RELAÇÃO DOS DADOS	17
2.3 APLICAÇÃO DA TEORIA DO ENFOQUE META ANALÍTICO CONSOLIDADO	19
<b>2.3.1 Co-ocorrência</b>	<b>19</b>
<b>2.3.2 Co-citação</b>	<b>21</b>
<b>2.3.3 Coupling</b>	<b>22</b>
<b>2.3.4 Priorização de métricas através das pesquisas de alto impacto</b>	<b>25</b>
<b>3. REFERENCIAL TEÓRICO</b>	<b>27</b>
3.1 MODELOS TRADICIONAIS DE DESENVOLVIMENTO DE SOFTWARE	28
3.2 MODELOS ÁGEIS	29
<b>3.2.1 SCRUM</b>	<b>31</b>
<b>3.2.2 XP</b>	<b>32</b>
<b>3.2.3 LEAN</b>	<b>33</b>
<b>3.2.4 KANBAN</b>	<b>33</b>
3.3 FATORES DE SUCESSO DE SOFTWARE E MÉTRICAS ÁGEIS	34
<b>3.3.1 Dificuldade de aplicação de métricas ágeis</b>	<b>37</b>
3.3.1.1 Dificuldades na implementação de métodos ágeis	37
3.3.1.2 Dificuldade na tomada de decisão de desenvolvimento ágil	39

<b>4. METODOLOGIA</b>	<b>42</b>
4.1 MÉTODO DA PESQUISA	42
4.2 ESTRUTURAÇÃO DA PESQUISA	43
<b>5. ESTUDO DE CASO: DIFICULDADES NA IMPLEMENTAÇÃO DA FERRAMENTA DE VISUALIZAÇÃO DE MÉTRICAS ÁGEIS</b>	<b>46</b>
5.1 ENTENDIMENTO NA LITERATURA DAS MAIORES DIFICULDADES EM IMPLEMENTAR AS MÉTRICAS SELECIONADAS	48
5.2 LEVANTAMENTO DAS DIFICULDADES DA APLICAÇÃO DE MÉTRICAS ÁGEIS NA VISÃO DOS STAKEHOLDERS	49
5.3 LEVANTAMENTO DE PONTOS DE MELHORIA DE GERENCIAMENTO A PARTIR DE MÉTRICAS ÁGEIS	50
<b>6. ESTUDO DE CASO: DESENVOLVIMENTO DA FERRAMENTA DE VISUALIZAÇÃO DE MÉTRICAS ÁGEIS</b>	<b>55</b>
6.1 LEVANTAMENTO DOS DADOS NECESSÁRIOS PARA A CONSTRUÇÃO DOS GRÁFICOS	55
6.2 CODIFICAÇÃO DA INTEGRAÇÃO	56
6.3 EXECUÇÃO DO CÓDIGO	60
6.4 ANÁLISE CRÍTICA DOS DADOS COLETADOS	61
6.4.1 Overview do Projeto	62
6.4.2 Throughput e Velocity	63
6.4.3 Lead time e Cycle Time	64
6.4.4 Work in Progress	65
6.4.5 Histórico (Pontos e Número de Issues) com filtros de Sprint e Usuário	67
6.4.6 Histórico de Detalhamento das Issues	68
<b>7. CONCLUSÃO</b>	<b>71</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>73</b>
<b>APÊNDICES</b>	<b>78</b>



## LISTA DE FIGURAS

- Figura 1: Palavras-chaves adotadas inicialmente.
- Figura 2: Palavras-chaves filtradas por área de pesquisa.
- Figura 3: Representação da co-ocorrência.
- Figura 4: Co-citação.
- Figura 5: Coupling.
- Figura 6: Contribuição do TEMAC.
- Figura 7: Histórico de Abordagens de Desenvolvimento de Software.
- Figura 8: Gráficos de implementação de abordagens ágeis ao longo do tempo.
- Figura 9: Artefatos do Scrum
- Figura 10: Etapas do TEMAC
- Figura 11: Entrevistas no MIRO
- Figura 12: Categoria e quantidade de dificuldades levantadas pelos entrevistados
- Figura 13: Insights levantados pelos entrevistados
- Figura 14: Relatórios gerado pela API do Github e Zenhub
- Figura 15: Overview do Projeto (Google Data Studio)
- Figura 16: Throughput e Velocity (Google Data Studio)
- Figura 17: Lead Time e Cycle Time (Google Data Studio)
- Figura 18: Work in Progress (Google Data Studio)
- Figura 19: Histórico de Pontos e de Número de Issues (Google Data Studio)
- Figura 20: Histórico de Detalhamento de Issues
- Figura 21: Benefícios que a dashboard poderia agregar a PUMA

## **LISTA DE QUADROS**

- Quadro 1: Explicação das etapas do TEMAC
- Quadro 2: Países com maior incidência de pesquisa
- Quadro 3: Cluster da co-ocorrência.
- Quadro 4: Análise dos estudos levantados pela co-citação
- Quadro 5: Análise dos estudos levantados pelo coupling
- Quadro 6: Recorrência de citação de métricas ágeis entre as obras de maior impacto.
- Quadro 7: Modelos Tradicionais de Desenvolvimento de Software.
- Quadro 8: Artefatos do Scrum
- Quadro 9: Quadro 9: Métricas ágeis escolhidas para análise
- Quadro 10: Principais dificuldades na aplicação de métodos ágeis
- Quadro 11: Maiores dificuldades relacionadas à tomada de decisão de times ágeis
- Quadro 12: Estruturação da pesquisa
- Quadro 13: Dificuldades relacionadas à implementação de métricas ágeis
- Quadro 14: Quadro das Principais Dificuldades com Ordenação Decrescente de Recorrência.
- Quadro 15: Principais dificuldades dos entrevistados (categoria de dados)
- Quadro 16: Estrutura das métricas escolhidas para implementação
- Quadro 17: Backlog Report
- Quadro 18: Sprint Report
- Quadro 19: Daily Report
- Quadro 20: Skipped Report
- Quadro 21: Assignees Report

# 1. INTRODUÇÃO

O surgimento dos métodos ágeis ocorreu a partir de um contexto de aceleração e escala de produtos digitais entre o final da década de 1990 e o início dos anos 2000. Dessa forma, modelos de desenvolvimento de software, que antes eram centrados nos processos sendo repetitivos e determinísticos, passaram a ter abordagens mais adaptativas de maneira a garantir respostas rápidas às mudanças do mercado cada vez mais voláteis.

Previamente, os modelos tradicionais enxergavam a gestão de projetos de *software* como um processo linear e sequencial onde, de maneira escalonada, o produto é desenvolvido e, ao final do processo, é entregue ao cliente. Dessa forma, seus requisitos seriam considerados imutáveis. Diversas informações eram pré-definidas antes do entendimento de como o produto se comportaria no mercado gerando um contexto de falha quando eram entregues (DINGSOEYR, FALESSI, POWER, 2019).

Devido à necessidade de acompanhar a volatilidade do mercado aos produtos digitais, foram criados métodos que visam a melhoria contínua com maior flexibilidade e velocidade de processo, melhor comunicação da equipe e maior proximidade com o cliente, garantindo a sua entrega de valor (DINGSOEYR, FALESSI, POWER, 2019). Neste contexto, surgiram os métodos ágeis tais como o Lean, o Scrum, o Kanban e o XP, além de diversos outros sincretismos que unem as melhores práticas dessas metodologias.

Assim, os projetos que têm como objetivo a celeridade na entrega de valor passam a exigir uma nova forma de gerenciamento já que os modelos tradicionais não atendem a essas necessidades. Para isso, foram entendidos diversos fatores que corroboram com o agilismo e, assim, o fator sociotécnico passou a representar um dos principais pilares dos modelos ágeis (RODRIGUES *et al.*, 2019).

As métricas ágeis surgem para acompanhar o time de desenvolvimento corroborando com esse fator sociotécnico e trazendo à gestão de desenvolvimento de *software* maior apoio para as tomadas de decisões durante o projeto. Assim, ao se ter metrificado como o time está se comportando em termos de produtividade, os gestores são capazes de investigar as causas de tais cenários, planejar melhorias e verificar suas evoluções de maneira quantitativa e incremental. Isso contribui com o aumento do sucesso do produto, ao se ter tomada de decisão embasada, diminuição do tempo e do custo de desenvolvimento, melhoria da qualidade e garantia de satisfação do cliente.

É no contexto de dificuldade de implementação de métricas ágeis para melhoria

operacional de times de desenvolvimento de *software* em um ambiente universitário que esta pesquisa se aplica, tendo como objeto de estudo o time de desenvolvimento da webapp Plataforma Unificada de Metodologia Ativa (PUMA). Tal produto possui como objetivo unificar processos que envolvem a aplicação de metodologias ativas de aprendizado, a princípio do curso de Engenharia de Produção da Universidade de Brasília (MONTEIRO *et al.*, 2018).

Entende-se que a implementação de métricas ágeis possui relação direta com a gestão da qualidade do produto, de maneira a garantir que as expectativas do cliente sejam atendidas. Tal gestão pode ser entendida como reorientação da gestão das organizações e tem como pontos básicos: o foco no cliente, o trabalho em equipe (permeando toda a organização), as decisões baseadas em fatos e dados e a busca constante da solução de problemas e diminuição de erros (LIZARDO & RIBEIRO, 2020).

Dessa maneira, empresas se empenham para implementar programas de qualidade a fim de garantir a plena satisfação dos clientes ao reduzir custos de operação, acelerar a entrega e minimizar perdas, erros e retrabalho. É com base nisso que Abrantes (2009) traz que a gestão da qualidade foca em melhorar a competitividade, a eficácia e a flexibilidade de uma organização através da elaboração de um planejamento coerente, onde se compreende e acompanha as atividades relacionadas ao produto e seus responsáveis.

Porém, diferentemente de produtos físicos, gerir a qualidade do desenvolvimento de *software* é algo mais complexo já que não se tem um objeto pontual e visível, mas sim linhas de códigos que conseguem evidenciar o funcionamento sistêmico apenas quando estão conectadas. Essa complexidade pode gerar planejamentos e execuções longos e nebulosos, possibilitando um aumento no custo e um risco do produto não se encaixar apropriadamente no mercado.

Para conter essa questão, o uso de métricas ágeis se faz necessário, de maneira a auxiliar no gerenciamento através de uma análise focada na produtividade do time de desenvolvimento, facilitando etapas de planejamento, execução e entrega, ao se utilizar dados que sinalizam o comportamento histórico desse time para tomada de decisão. Assim, garante-se que o time entregue o que é necessário no tempo acordado e com o recurso disponível para atingir as expectativas do cliente e o então sucesso do projeto.

Diante disso, essa pesquisa busca responder à seguinte pergunta: Como gerenciar projetos de desenvolvimento de *software* através de métricas ágeis em um ambiente educacional, contribuindo com fatores de sucesso no desenvolvimento do mesmo e impactando na qualidade do produto?

## 1.1 JUSTIFICATIVA

A formação de Engenheiros de Produção pela Universidade de Brasília demanda a identificação, a caracterização e o tratamento da criação de valor de determinado produto ou serviço através do entendimento da produção e da alocação de bens e serviços nas esferas políticas, econômicas e culturais da sociedade, seja em âmbito público ou privado. Assim, tais características têm relação direta com a gestão de projetos, um dos cerne do curso (SILVA E BALTHAZAR, 2010).

O curso historicamente teve como foco o setor de serviços. Porém, à medida que o gerenciamento de projetos vem se deparando com outras esferas de produção tecnológica, a Engenharia de Produção requer maior entendimento e aprimoramento para cenários como o de desenvolvimento de *softwares*.

É neste contexto que o projeto em questão se aplica: com o fim de enriquecer o panorama de gestão de desenvolvimento de *software* por meio de um olhar analítico e gerencial da Engenharia de Produção.

Sabe-se, então, que o acompanhamento de indicadores de desempenho de times de desenvolvimento de software, como as métricas ágeis, pode contribuir com tomadas de decisões mais assertivas e com o sucesso do projeto, garantindo este olhar analítico e gerencial do Engenheiro de Produção. Contudo, embora amplamente comentadas na literatura, existe uma carência de como se investigar e implementar na prática métricas ágeis em times de desenvolvimento, fato que pode ser evidenciada na dificuldade de bibliografias que trazem com mais detalhes obstáculos e orientações da implementação de métricas ágeis no contexto de *software*. Dessa maneira, o foco deste estudo é voltado para aplicação dessas métricas, principalmente em desenvolvimentos que utilizam como plataformas de apoio o Github e Zenhub, ferramentas de código aberto amplamente utilizadas no contexto universitário para hospedagem de código fonte e para gerenciamento de atividades de desenvolvimento, respectivamente.

Assim, a contribuição acadêmica deste estudo é disponibilizar mais informações práticas relacionadas à aplicação de métricas ágeis no contexto de software, além de promover informações que auxiliem na melhoria de gestão de times ágeis de software através de pilares de análise de dados de produtividade.

O trabalho foi desenvolvido por meio da parceria realizada entre o curso de Engenharia de Produção (disciplina de Projeto de Sistemas de Produção 5) com o curso de Engenharia de Software (disciplinas de Engenharia de Produto de Software e de Métodos de

Desenvolvimento de Software) da Faculdade do Gama (FGA), que teve como objeto de estudo o desenvolvimento da Plataforma Unificada de Metodologias Ativas (PUMA).

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

O objetivo geral deste trabalho é elaborar uma ferramenta de visualização de métricas que possam ser utilizadas no gerenciamento de projetos de desenvolvimento de *software* a fim de contribuir com fatores de sucesso durante o desenvolvimento do módulo da Plataforma Unificada de Metodologias Ativas (PUMA).

### 1.2.2 Objetivos específicos

Para isso a pesquisa necessita seguir com os seguintes objetivos específicos:

- Identificar as metodologias ágeis de desenvolvimento de software e as métricas associadas a elas;
- Selecionar as métricas ágeis de maior relevância na literatura;
- Levantar as principais dificuldades de aplicação de tais métricas no contexto da PUMA;
- Viabilizar o entendimento e implementação de métricas ágeis para a melhoria da tomada de decisão da PUMA.
- Coletar feedback dos estudantes da PUMA acerca da solução implementada.

## 1.3 ESTRUTURAÇÃO DOS CAPÍTULOS

Este trabalho estrutura-se em 6 capítulos, sendo eles:

- Capítulo 1: Introdução - Neste capítulo estão abordados os aspectos gerais da pesquisa, apresentando o contexto da aplicação de métodos ágeis e de suas métricas no processo de desenvolvimento de *software*. Ademais, contempla a justificativa da pesquisa, o objetivo geral e os objetivos específicos além da estrutura dos capítulos vista nesta seção.
- Capítulo 2: Levantamento do Estado da Arte (TEMAC) - Neste capítulo, serão apresentados os resultados do levantamento do estado da arte relacionados à aplicação de métricas ágeis em *software* bem como a priorização das principais métricas que este projeto terá como foco
- Capítulo 3: Referencial Teórico - Neste capítulo será fundamentada teoricamente a pesquisa através dos estudos encontrados na revisão bibliográfica trazendo como foco os métodos de desenvolvimento de *software* (tradicionais e ágeis); fatores de sucesso de

software; e métricas ágeis.

- Capítulo 4: Método de Pesquisa - Neste capítulo serão definidos os métodos de pesquisa e sua estrutura de maneira que se consiga atingir seus objetivos propostos.
- Capítulo 5 e 6: Estudo de Caso - Neste capítulo será evidenciado o estudo de caso da pesquisa desde a aplicação das entrevistas semi estruturadas até o desenvolvimento dos planos de ação mapeados para solucionar as problemáticas encontradas. Por fim, serão coletados feedbacks dos estudantes envolvidos na PUMA para com a solução criada.
- Capítulo 7: Conclusão - Neste capítulo serão apresentadas as conclusões e as contribuições da pesquisa além de seus possíveis próximos passos.



## 2. LEVANTAMENTO DO ESTADO DA ARTE

No desenvolvimento de pesquisas que buscam a resolução de alguma problemática específica, é muito comum autores possuírem dificuldades ou não encontrarem as informações disponíveis, levando-os a uma busca incessante de levantamento de dados para explorar suas pesquisas. Contudo, tal realidade poderia ser consideravelmente facilitada, já que muitos outros estudiosos já avançaram anteriormente na respectiva temática, trazendo ao pesquisador o conhecimento e levantamento do que já foi escrito acerca de sua temática (MARIANO E ROCHA, 2017).

Nessa circunstância, surgiu a Teoria do Enfoque Meta Analítico Consolidado (TEMAC). Esta abordagem busca unificar os aportes do uso do Enfoque Meta-Analítico (EMA), sendo este um método desenhado em etapas que utiliza o critério de impacto de revistas, citações de autores e artigos e frequência de palavras-chave com um longo e árduo trabalho manual para escolha do material a ser utilizado (MARIANO E ROCHA, 2017).

Ademais, a aplicação do TEMAC garante uma maior qualidade ao trabalho já que se aproveita ao máximo linhas de pesquisa importantes, similares ao objeto do estudo, garantindo que as informações de um artigo de qualidade sejam respeitadas.

O TEMAC é fundamentado em 3 etapas: preparação da pesquisa; apresentação e integração dos dados; detalhamento do modelo integrador e validação por evidência. Esta pesquisa aplicou as 3 etapas com adaptações para garantir o devido embasamento e validação. Com embasamento na pesquisa de Mariano e Rocha (2017), o Quadro 1 evidencia resumidamente tais etapas.

Quadro 1: Explicação das etapas do TEMAC

Etapa	Explicação
1 - Preparação da pesquisa	Definição de palavras-chave acerca do tema da pesquisa; o espaço-tempo da análise; as bases de dados utilizadas; e por fim as áreas de conhecimento relacionadas à pesquisa.
2-Apresentação de interrelação de dados	Nesta etapa, os diversos estudos sobre o tema proposto são relacionados entre si. Sendo assim, pode-se escolher diversos critérios para se relacionar às pesquisas, como revistas mais relevantes; revistas que mais publicaram sobre o tema; evolução do tema ano a ano; entre outros.
3 - Detalhamento do modelo integrador e validação por evidências	São construídas análises de maneira a integralizar as pesquisas realizadas, validadas por evidência. É necessário realizar as seguintes análises bibliográficas: <ul style="list-style-type: none"><li>● <b>co-occurrence ou co-word:</b> Segundo Guedes e Borschiver (2005), tal análise identifica a estrutura conceitual sobre o estudo, trazendo uma rede de temas associada a ele através da ocorrência de palavras-chaves;</li></ul>

Etapa	Explicação	Continua
3 - Detalhamento do modelo integrador e validação por evidências	<ul style="list-style-type: none"> <li>● <b>co-citation:</b> verifica artigos citados juntos, podendo indicar semelhanças entre eles ao tratar das abordagens convergentes;</li> <li>● <b>coupling:</b> traz uma perspectiva das frentes de pesquisas mais utilizadas.</li> </ul>	

Fonte: Adaptado de Mariano e Rocha (2017).

As análises tratadas na etapa 3 foram realizadas através do **VOSViewer**, um software de construção e visualização de redes bibliométricas. Essas construções, não obstante, são possibilitadas através da geração de arquivos em **.csv** da base de dados do Scopus (PERIANES-RODRÍGUEZ, WALTMAN, VAN ECK, 2016).

Com o propósito de externalizar a importância de tal metodologia para a construção deste artigo, foram levantadas as pesquisas sugeridas pelas análises acima que tinham relação direta com o tema proposto, auxiliando na identificação e seleção das métricas ágeis, última etapa desta seção.

## 2.1 PREPARAÇÃO DA PESQUISA

Segundo Mariano e Rocha (2017), a primeira etapa do TEMAC é importante para preparar a pesquisa ao procurar as palavras chaves relacionadas a ela, seu período de análise, quais serão as bases de dados envolvidas, quais serão as áreas do conhecimento adotadas, e por fim, um refinamento dos artigos.

O estudo adotou, inicialmente, 3 tipos de palavras chaves na base SCOPUS sendo elas "**agile**", "**software**", "**metric**".

Foram encontrados um total de 664 artigos divididos nas áreas de ciência da computação, engenharia, matemática, ciência da decisão, negócios, gestão e contabilidade, ciência social, física e astronomia, e multidisciplinaridade.

## 2.2 APRESENTAÇÃO E INTER-RELAÇÃO DOS DADOS

Foram traçadas as relações que os 664 artigos encontrados tinham com o tema. Observa-se que eles foram divididos nas áreas de ciência da computação (85%), engenharia (30%), matemática (22%), ciência da decisão (12%), negócios, gestão e contabilidade (11%), ciência social (7%), física e astronomia (1%), energia (2%), ciência dos materiais (1%), e medicina (1%). Contudo, como os 5 últimos temas - ciência social; física e astronomia; energia; ciência dos materiais; e medicina - não possuíam relação com esta pesquisa, foi feito um filtro para removê-los resultando em 595 pesquisas (Figura 2).

Ademais, também foram traçadas convergências territoriais entre os artigos sendo eles, em ordem decrescente, provenientes dos seguintes países: Estados Unidos, Índia, Alemanha, Brasil, Itália, Reino Unido, Finlândia, Suécia, Canadá e Espanha (Quadro 2).

Quadro 2: Países com maior incidência de pesquisa

País	Número de Estudos Encontrados
Estados Unidos	98
Índia	47
Alemanha	38
Brasil	33
Itália	29
Finlândia	28
Reino Unido	27
Suécia	22
Canadá	18
Espanha	17

Fonte: Adaptado de Scopus (2022)

Nota-se que o Brasil é o único país da América do Sul presente nessa análise, sendo o quarto maior protagonista na temática analisada, sugerindo a relevância e incidência do tema em âmbito nacional.

As afiliações do País que mais publicaram estudos sobre o tema são: Universidade Federal de Campina Grande; Universidade de Brasília; Universidade Estadual de Campinas; Universidade Federal de Pernambuco; Centro de Estudos e Sistemas Avançados do Recife (CESAR); Universidade Federal do Rio de Janeiro; Universidade de São Paulo; Universidade Federal de Santa Catarina; e Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

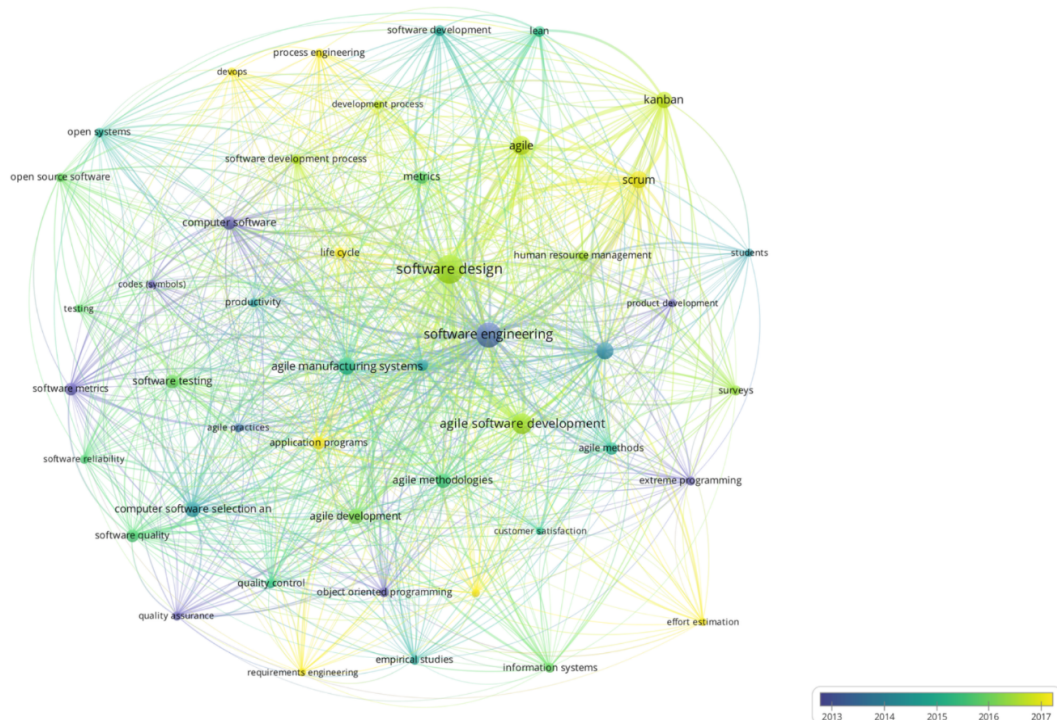
## 2.3 APLICAÇÃO DA TEORIA DO ENFOQUE META ANALÍTICO CONSOLIDADO

Feito o refinamento e o entendimento dos fatores de convergência, o próximo passo do TEMAC foi exportar a base de dados do Scopus para realizar as criações e visualizações das redes bibliométricas a partir do VOSViewer, iniciando-se então as etapas de análise (co-ocorrência, co-citação e coupling).

### 2.3.1 Co-ocorrência

Esta etapa é responsável por evidenciar os principais temas associados ao estudo em questão através da ocorrência de palavras-chave. Foram refinadas 40 palavras, conforme apresentado na Figura 3 (GUEDES E BORSCHIVER, 2005).

Figura 3: Representação da co-ocorrência.



Fonte: SCOPUS (2022)

A partir da co-ocorrência também é possível categorizar as palavras-chave encontradas em clusters - grupos de convergência das palavras-chave por meio de sua temática. Dessa maneira, os 6 clusters associados às palavras-chave encontradas podem ser evidenciados a seguir (Quadro 3).

Quadro 3: Cluster da co-ocorrência.

Número do Cluster	Palavras-chave
Cluster 1	<ul style="list-style-type: none"> <li>● application programs</li> <li>● codes (symbols)</li> <li>● computer software</li> <li>● computer software service</li> <li>● customer satisfaction</li> <li>● devops</li> <li>● life cycle</li> <li>● open source software</li> <li>● open systems</li> <li>● productivity</li> <li>● quality assurance</li> <li>● quality control</li> <li>● software metrics</li> <li>● software quality</li> <li>● software reliability</li> <li>● software testing</li> <li>● testing</li> </ul>
Cluster 2	<ul style="list-style-type: none"> <li>● agile methodologies</li> <li>● agile software development</li> <li>● effort estimation</li> <li>● extreme programming</li> <li>● iterative methods</li> <li>● object oriented programming</li> <li>● project management</li> <li>● software engineering</li> <li>● students</li> <li>● surveys</li> </ul>
Cluster 3	<ul style="list-style-type: none"> <li>● agile methods</li> <li>● development process</li> <li>● human resource management</li> <li>● metrics</li> <li>● process engineering</li> <li>● software design</li> <li>● software development</li> </ul>
Cluster 4	<ul style="list-style-type: none"> <li>● agile</li> <li>● kanban</li> <li>● lean</li> <li>● product development</li> <li>● scrum</li> </ul>
Cluster 5	<ul style="list-style-type: none"> <li>● agile development</li> <li>● agile practices</li> <li>● empirical studies</li> <li>● requirements engineering</li> <li>● software industry</li> </ul>
Cluster 6	<ul style="list-style-type: none"> <li>● agile manufacturing system</li> <li>● information system</li> </ul>

Fonte: SCOPUS (2022)

Nota-se que a clusterização garante um maior entendimento das direções voltadas para a temática da pesquisa (*agile*, *software* e *metric*). Dessa maneira, determinou-se o que cada cluster estava categorizando.

O **Cluster 1** é voltado a uma perspectiva de qualidade na área de *software*. O **Cluster 2** traz metodologias e técnicas relacionadas ao ágil. O **Cluster 3** evidencia uma perspectiva processual e de gerenciamento de pessoas, englobando também métricas. O **Cluster 4** é

referente às diferentes metodologias ágeis existentes. O **Cluster 5** está relacionado a práticas e estudos ágeis. Por fim, o **Cluster 6** engloba algumas áreas de aplicação do ágil, como a de Tecnologia da Informação e a de Sistemas Manufatureiro.

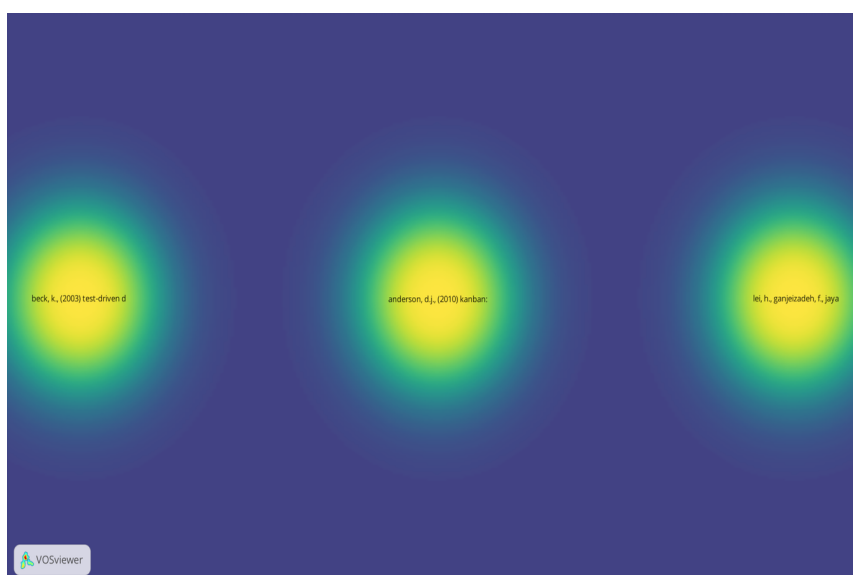
O entendimento desses clusters foi essencial para determinar se tais categorizações estavam convergindo com a temática desta pesquisa, a fim de garantir maior assertividade no levantamento do estado da arte. Dessa forma, observa-se que a convergência desejada existe, possibilitando a continuidade do TEMAC.

### 2.3.2 Co-citação

Segundo Vogel & Güttelp (2013), essa etapa obtém uma relação de similaridade entre duas publicações citadas com caráter de uma abordagem dinâmica. Assim, como visto anteriormente, a ferramenta verifica os artigos que regularmente são citados conjuntamente, podendo sugerir uma semelhança entre estes estudos, o que pode trazer mais valor à análise.

Dos estudos encontrados pela co-citação, dois geraram alto impacto nesta pesquisa ao apresentar convergência com o tema. (Quadro 4).

Figura 4: Co-citação.



Fonte: SCOPUS (2022)

É notório que dois entre os três estudos encontrados na co-citação tiveram alto impacto nesta pesquisa ao apresentar convergência com o tema. Tais estudos podem ser evidenciados no Quadro 4, que apresenta na primeira coluna o título e autor do estudo; e na segunda, um resumo da contribuição do estudo com a pesquisa.

Quadro 4: Análise dos estudos levantados pela co-citação.

Estudo	Contribuição à pesquisa
Kanban: Mudança Evolucionária de Sucesso Para Seu Negócio de Tecnologia (ANDERSON, 2011)	Traz de maneira altamente prática a aplicação do método Kanban.
A statistical analysis of the effects of Scrum and Kanban on software development projects (LEI et al., 2017)	Compara estatisticamente a eficácia dos efeitos do Scrum e de Métodos Kanban sobre o gerenciamento de projetos para desenvolvimento de software.

Fonte: Elaborado pelo autor

Parte dos estudos encontrados na co-citação (66%) tiveram relação direta com a temática desta pesquisa. Sendo assim, o primeiro estudo (ANDERSON, 2011), ao se evidenciar a aplicabilidade de uma das metodologias ágeis, traz também algumas métricas a serem exploradas relacionadas a esta metodologia e vistas adiante. Já o segundo estudo (LEI et al., 2017), traz comparativos de duas metodologias ágeis, garantindo o entendimento dos pontos positivos e negativos de cada qual.

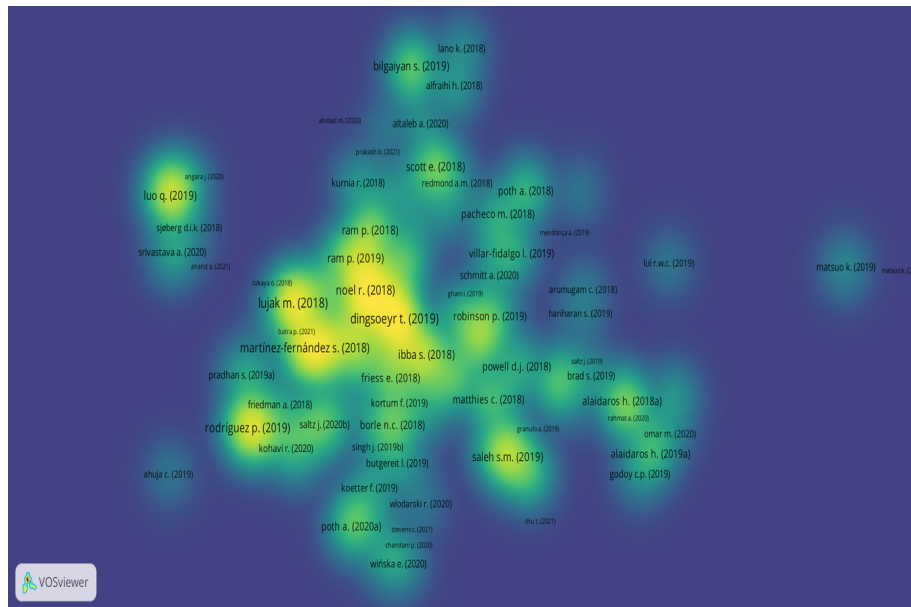
### 2.3.3 Coupling

O coupling foi desenvolvido a fim de se compreender quais são as tendências de pesquisa no momento relacionado ao cenário de métricas ágeis na gestão do desenvolvimento de *software*.

Apesar da similaridade, o coupling e a co-citação entregam resultados diferentes. Segundo Mariano e Rocha (2017), o coupling é uma medida de associação entre duas publicações citadas, já a co-citação é uma relação de similaridade entre as duas publicações citadas. Isso diferencia o que cada uma implica no âmbito analítico, sendo a co-citacao uma abordagem dinâmica, estabelecida por autores de artigos diferentes do que ele liga, sendo extrínseco aos documentos envolvidos. Já o coupling é desenvolvido a partir de referências feitas pelos autores dos documentos envolvidos, sendo intrínseca a estes. Assim, quanto mais sobreposição entre as bibliografias do coupling, maior será o seu engajamento (Vogel & Güttelp, 2013).

Dessa forma, para entender essas tendências que conduzirão as pesquisas, o coupling foi realizado no intervalo de três anos até o momento (a partir de 2018) por meio da ferramenta VOSViewer. A Figura 5 permite a sua visualização.

Figura 5: Coupling



Fonte: SCOPUS (2022)

Neste sentido, foram encontradas, através do mapa de calor evidenciado no coupling, 15 pesquisas com grande realce de cor, indicativo do seu protagonismo em termos de tendência de pesquisa. Porém, desses estudos, apenas 8 apresentaram impacto direto com esta pesquisa (Quadro 5).

Quadro 5: Análise dos estudos levantados pelo coupling.

Estudo	Contribuição à pesquisa
A quality model for actionable analytics in rapid software development (MARTINEZ-FERNANDEZ et al, 2018)	Traz um apanhado de métricas de qualidade, além de tratar da falta de confiabilidade dos profissionais de software para com as análises geradas no desenvolvimento.
Agile development at scale: the next frontier (DINGSOEYR et al, 2019)	Trata sobre a aplicação de métodos ágeis em contextos de larga escala, exemplificando diversos modelos.
Advances in using agile and lean processes for software development (RODRÍGUEZ et al, 2019)	Traz o contexto histórico para o surgimento do ágil; explica os métodos ágeis; trata objetivos das equipes; e interliga tais objetivos com métricas.
Comparative study within scrum kanban xp focused on their practise (SALEH et al., 2019)	Faz um comparativo entre Scrum, Kanban e XP, trazendo os pontos fortes e fracos de cada um.



Estudo	Contribuição à pesquisa	Continua
Success factors for effective process metrics operationalization in agile software development: a multiple case study (RAM et al, 2019)	Indica a carência de aplicação de métricas no desenvolvimento de software.	
Software process measurement and related challenges in agile software development: a multiple case study (RAM et al, 2018)	Trata sobre a escassez de investigações sobre a aplicabilidade de métricas ágeis, além de trazer um apanhado delas, além das suas convergências em algumas empresas.	
Development of an agile development method based on kanban for distributed part time teams and an introduction framework (HOFMAN et al, 2018)	Traz comparativos entre scrum e kanban, mas focado neste último. Ademais, traz algumas falhas de métodos ágeis, com proposta de solução.	
Towards an improved software project monitoring task model of agile kanban method (ALAIDARO et al, 2021)	Explica sobre gestão de software, trazendo os maiores critérios de falha relacionados ao Kanban e propondo, por fim, soluções.	

Fonte: SCOPUS (2022)

Nota-se que a aplicação da metodologia TEMAC contribuiu de forma positiva para esta pesquisa no sentido de guiá-la para as principais abordagens, tendências e frentes de trabalho, além de trazer palavras-chave que poderão ser usadas na publicação facilitando sua acessibilidade a demais pesquisadores com intuítos convergentes. Ademais, durante a leitura destes estudos foram encontradas outras referências relacionadas à temática da pesquisa, contribuindo com o seu enriquecimento.

A Figura 6 mostra a contribuição do TEMAC para esta pesquisa por meio do refinamento e aplicação da co-citação e do coupling.

Figura 6: Contribuição do TEMAC



Fonte: Elaborado pelo autor

Dessa maneira, a partir dos 15 estudos apontados através da aplicação do TEMAC, 8 deles (53%) tiveram grandes impactos para que esta pesquisa atingisse os seus objetivos. Essa filtragem foi possível através da análise individual de cada um, onde procurou-se convergências com a temática de métricas relacionadas ao agilismo e voltadas para a produtividade do time de desenvolvimento.

### 2.3.4 Priorização de métricas através das pesquisas de alto impacto

O TEMAC possibilitou que se encontrasse os principais estudos relacionados à temática deste estudo. Porém, nos estudos de maior impacto evidenciadas anteriormente foram encontradas um grande compilado de dados quantitativos que poderiam ser mais aprofundados no desenvolvimento deste projeto.

Tais dados podem ser entendidos como medidas, métricas ou indicadores. Dessa maneira, segundo McGarry et al. (2002), as medidas são avaliações de algum atributo através de um método específico de medição, ou seja, necessitam ser avaliadas seguindo um padrão (como metros, celsius, newtons etc.).

Já as métricas, são medições utilizadas para determinar se algum sistema, componente ou processo possui determinado atributo, sendo normalmente calculada através de duas ou mais medidas (IEEE). Por fim, indicadores são variáveis configuradas para sinalizar determinada condição, sendo portanto, relacionado a uma métrica, interpretando-a em determinada situação (SATO, 2007).

Os 8 principais estudos encontrados pela aplicação do TEMAC citaram diversas medidas e métricas relacionadas ao agilismo. Assim, a partir do compilado de todas essas métricas e medidas citadas, a pesquisa buscou entender quais e quantos desses estudos citaram cada uma delas (Autores e Recorrência, conforme mostrado no Quadro 6).

Esta etapa foi fundamental para trazer visibilidade das principais métricas e entender suas possíveis aplicações no estudo de caso.

Quadro 6: Recorrência de citação de métricas ágeis entre as obras de maior impacto.

Métrica ou medições	Autores	Recorrência
<i>Customer satisfaction</i>	DINGSOEYR et al, 2019; RODRÍGUEZ et al, 2019; RAM et al, 2019; RAM et al, 2018; ALAIDARO et al, 2021	5

<b>Métrica ou medições</b>	<b>Autores</b>	<b>Recorrência Contínua</b>
<i>Velocity</i>	MARTINEZ-FERNANDEZ et al, 2018; RODRÍGUEZ et al, 2019; RAM et al, 2019; RAM et al, 2018	4
<i>Effort estimate</i>	RODRÍGUEZ et al, 2019; SALEH et al, 2019; RAM et al, 2019; RAM et al, 2018	4
<i>Defect count</i>	RODRÍGUEZ et al, 2019; RAM et al, 2019; RAM et al, 2018	3
<i>Work in progress</i>	SALEH et al, 2019; HOFMAN et al, 2018; ALAIDARO et al, 2021	3
<i>Cycle time</i>	RODRÍGUEZ et al, 2019; HOFMAN et al, 2018; ALAIDARO et al, 2021	3
<i>Technical debt</i>	MARTINEZ-FERNANDEZ et al, 2018; RODRÍGUEZ et al, 2019;	2
<i>Lead time</i>	RODRÍGUEZ et al, 2019; SALEH et al, 2019	2
<i>Test coverage</i>	MARTINEZ-FERNANDEZ et al, 2018; RODRÍGUEZ et al, 2019;	2
<i>Throughput</i>	RAM et al, 2019; ALAIDARO et al, 2021	2
<i>Burndown</i>	SALEH et al, 2019; RAM et al, 2018	2
<i>Build status</i>	RODRÍGUEZ et al, 2019)	1
<i>Number of unit tests</i>	MARTINEZ-FERNANDEZ et al, 2018;	1

Fonte: Elaborado pelo autor

Do que foi exposto, aprofundou-se naquelas métricas relacionadas à produtividade do time de desenvolvimento, de maneira a relacionar artefatos da Engenharia de Produção com o desenvolvimento de software. O restante que não foi contemplado ou não correspondia ao significado de métricas ou não possui relação direta com a produtividade.

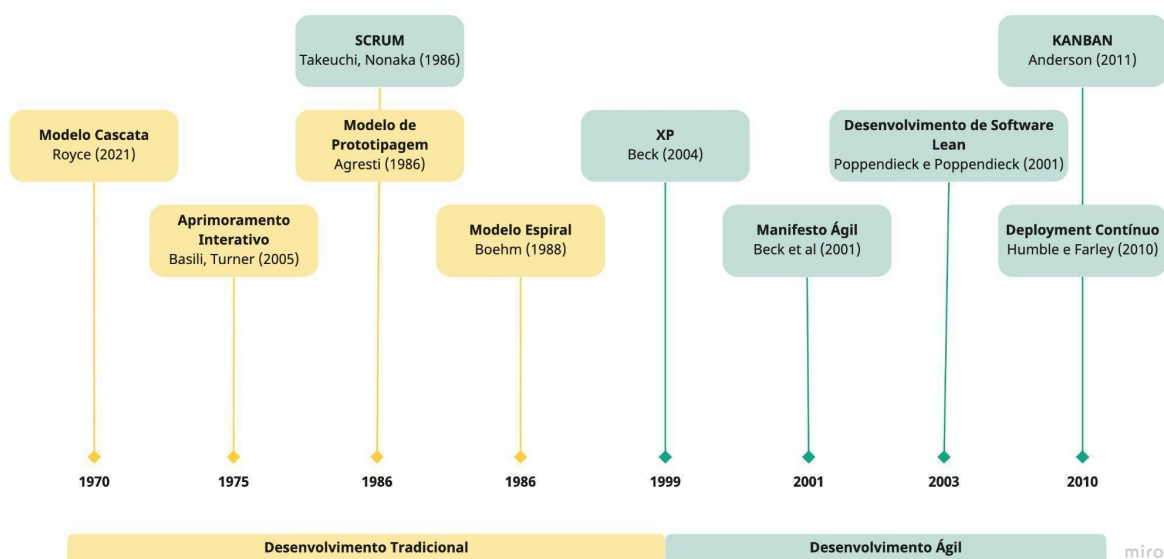
Dessa maneira, as métricas priorizadas para serem aprofundadas nesta pesquisa e que atendem ao cenário de produtividade foram o Velocity; o Work in Progress (WIP); o Cycle Time; o Lead Time; e o Throughput. Tais métricas serão devidamente explicadas no Capítulo 3.

### 3. REFERENCIAL TEÓRICO

A gestão de produtos digitais vem ganhando cada vez mais notoriedade na esfera tecnológica. Assim, sua evolução ocorre de maneira rápida e profunda, deixando para trás muitas abordagens que não permitiam um desenvolvimento de *software* efetivo dada a alta volatilidade do mercado.

Segundo Alaidaros et al (2021), o gerenciamento de projeto de *software* envolve um conjunto de conhecimentos, técnicas e ferramentas onde são planejados, implementados, monitorados e controlados os processos de implementação de *software* com o objetivo de se garantir o produto esperado. Dessa maneira, a Figura 7 mostra as abordagens históricas de gestão de software até os dias de hoje.

Figura 7: Histórico de Abordagens de Desenvolvimento de Software



Fonte: Adaptado de Rodriguez et al. (2019)

O Modelo de Cascata deu início ao desenvolvimento tradicional nos anos 1970, seguidos do Aprimoramento Interativo em 1975 e do Modelo de Prototipagem em 1986. Neste mesmo ano surgiu o Scrum, que, embora tenha nascido no contexto tradicional, só ganhou notoriedade nos anos 2000 após o Manifesto Ágil. Em 1988, o Modelo Espiral finaliza o contexto de desenvolvimento tradicional e o Extreme Programming (XP) dá início ao Desenvolvimento Ágil (2004), seguido do Manifesto Ágil em 2001; do desenvolvimento Lean, em 2003; e do Deployment Contínuo e Kanban, em 2010. Os modelos citados podem ser verificados na seção seguinte.

### 3.1 MODELOS TRADICIONAIS DE DESENVOLVIMENTO DE SOFTWARE

Pode-se afirmar que os modelos tradicionais deram início ao gerenciamento de *software* através da construção de métodos de desenvolvimento. Mas para serem considerados uma metodologia de desenvolvimento, eles deveriam apresentar determinada **aplicabilidade** e **popularidade** (ECKL BAUER, 2017). Dessa maneira, de acordo com (RODRÍGUEZ et al, 2019) os modelos tradicionais que historicamente se encaixam nesse cenário estão evidenciados no Quadro 7.

Quadro 7: Modelos Tradicionais de Desenvolvimento de Software.

Abordagem	Autor e Ano	Descrição
Modelo Cascata	Royce (2021)	O desenvolvimento é sequencial e com fases fixas, fluindo de cima para baixo, como uma cascata. São elas, respectivamente: análise de requisitos, projeto, implementação, testes (validação), integração e manutenção de software.
Aprimoramento Interativo	Basili e Turner (2005)	É uma técnica para se refinar gradualmente o desenvolvimento de software. Ela é semelhante ao modelo cascata, contudo, deve ser conduzida em diversos ciclos, podendo retornar aos ciclos anteriores, diferente da cascata que é sequencial e linear.
Modelo de Prototipagem	Agresti (1986)	Esse modelo vem como uma resposta à crítica do modelo de cascata por ele não conseguir acomodar novos paradigmas. Dessa maneira, traz a aplicação da prototipagem como etapa importante do desenvolvimento de software e do desenvolvimento de usuário final
Modelo Espiral	Boehm (1988)	É um modelo estruturado em cima de uma espiral sobreposta em 4 quadrantes, sendo o primeiro Definição de Objetivos; o segundo, Avaliação e Redução de Riscos; o terceiro, Implementação e Validação; e o último, Planejamento e Especificação. Dessa forma, ao longo do desenvolvimento do software, vai se caminhando no espiral e passando por todas essas fases constantemente, até que o software esteja pronto.

Fonte: Elaborado pelo autor

Contudo, à medida que os produtos digitais foram ficando cada vez mais presentes na vida das pessoas, exigindo mais escalabilidade das empresas e; à medida que o mercado começou a se tornar cada vez mais competitivo, colocando o cliente como centro dos produtos; tais modelos começaram a ficar obsoletos, já que impediam muitas vezes entregas de valor rápidas e efetivas com o que o mercado ansiava.

Dessa forma, modelos centrados em processos determinísticos e repetíveis foram substituídos por abordagens flexíveis, que enfatizam a evolução contínua do software, através de funcionalidade, flexibilidade, rapidez, comunicação face a face, envolvimento com cliente e número mínimo de artefatos leves (Rodríguez et al. 2019). Esses novos modelos são popularmente conhecidos como Modelos Ágeis.

### **3.2 MODELOS ÁGEIS**

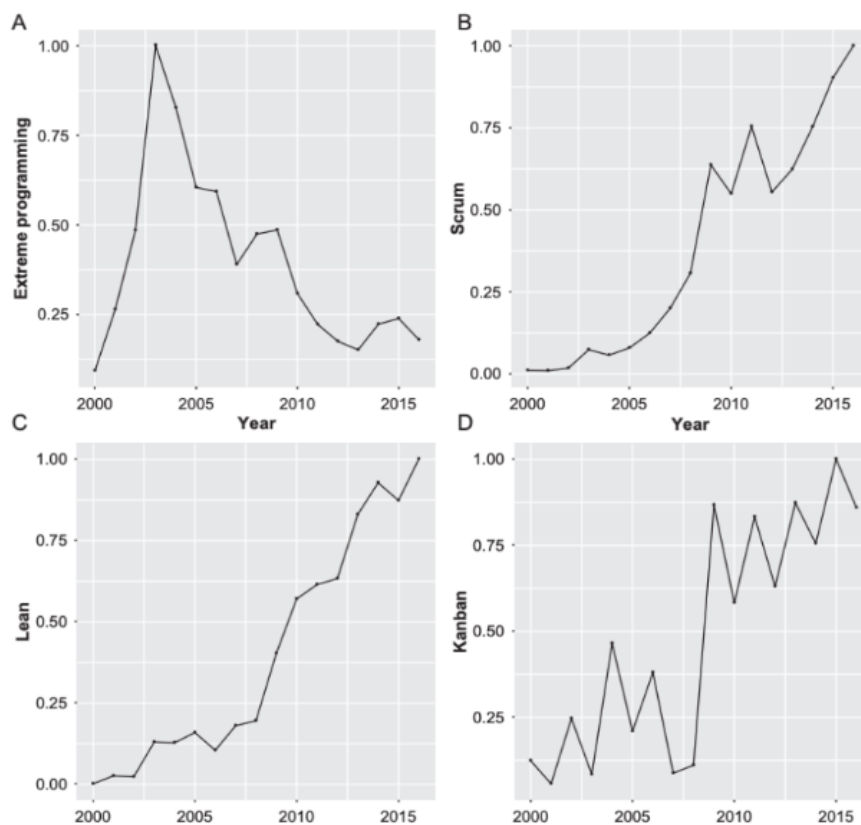
Os modelos ágeis ganharam espaço e importância no contexto de software por se apresentarem como alternativas efetivas a cenários de incerteza, fator que os modelos tradicionais não se adaptaram bem (GAETE et al, 2020).

Neste contexto, eles surgiram através do manifesto ágil no início dos anos 2000 para sanar a necessidade de maior adaptação e flexibilidade no desenvolvimento de software em contrapartida com os modelos tradicionais - inflexíveis, sequenciais e lentos demais para se adaptarem às mudanças de negócios. Catorze profissionais de software se reuniram por 2 dias em Utah, onde propuseram tal modelo que reconhece as limitações de antecipar mudanças externas de negócios e fornece meios para criar produtos mínimos viáveis (MVP). Tal manifesto entregou uma nova forma de fazer produto: entregar o básico de funcionalidade e em conjunto com o cliente, de forma incremental e iterativa, conhecendo-se gradativamente o comportamento do produto no mercado; desenvolver o produto em pequenos ciclos para se acompanhar mais detalhadamente as suas entregas; e gerar feedbacks constantes para garantir a melhoria contínua (Rodríguez et al. 2019).

Além dessas características vistas, o agilismo também enfatiza o lado humano do software. Assim, para Rodríguez et al. (2019) entendem que o fator sociotécnico no desenvolvimento é o principal impulsionador do sucesso do projeto, já que as pessoas tornam os processos menos determinísticos e repetitivos que os métodos tradicionais. Dessa maneira, entender a produtividade de um time de desenvolvimento vai muito além de aspectos quantitativos e qualitativos, mas também emocionais.

Neste contexto, a partir de tal manifesto foram criados modelos que se encaixam nessa nova realidade, como o Scrum; o método Kanban; o Extreme Programming (XP); além de adaptações de modelos já existentes, como o Lean. Tais metodologias diante de uma nova realidade de adaptabilidade e entrega contínua e rápida de valor tem se mostrado cada vez mais ascendente no contexto de software. Dessa maneira, Rodrigues et al. (2019) trouxe uma visualização desse contexto de ascendência em relação ao tempo, conforme evidenciado na Figura 8.

Figura 8: Gráficos de implementação de abordagens ágeis ao longo do tempo



Fonte: Adaptado de Rodriguez et al. (2019)

A Figura 8 mostra que as abordagens ágeis apresentam comportamentos diferentes ao longo do tempo, tendo o Extreme Programming com uma curva de implementação descendente a partir do início dos anos 2000. O Lean, o Scrum e o Kanban (que se adaptou no contexto de software em 2011) possuem curvas ascendentes. Assim, esta seção irá focar em trazer mais detalhes sobre cada uma antes de entrar no cenário de métricas associadas aos modelos ágeis.

### 3.2.1 SCRUM

Segundo HOFFMAN et al (2018) o **Scrum** é uma abordagem de desenvolvimento rígida com procedimentos claramente definidos. Dessa maneira, ele objetiva dividir o projeto em partes menores - as *sprints*- de maneira a garantir maior controle, qualidade e previsibilidade, além de diminuir a complexidade e aumentar a entrega de valor.

Os artefatos do Scrum permitem acompanhar e trazer mudanças incrementais ao desenvolvimento. Estas estão apresentadas no Quadro 8 de acordo com SALEH et al. (2019).

Quadro 8: Artefatos do Scrum

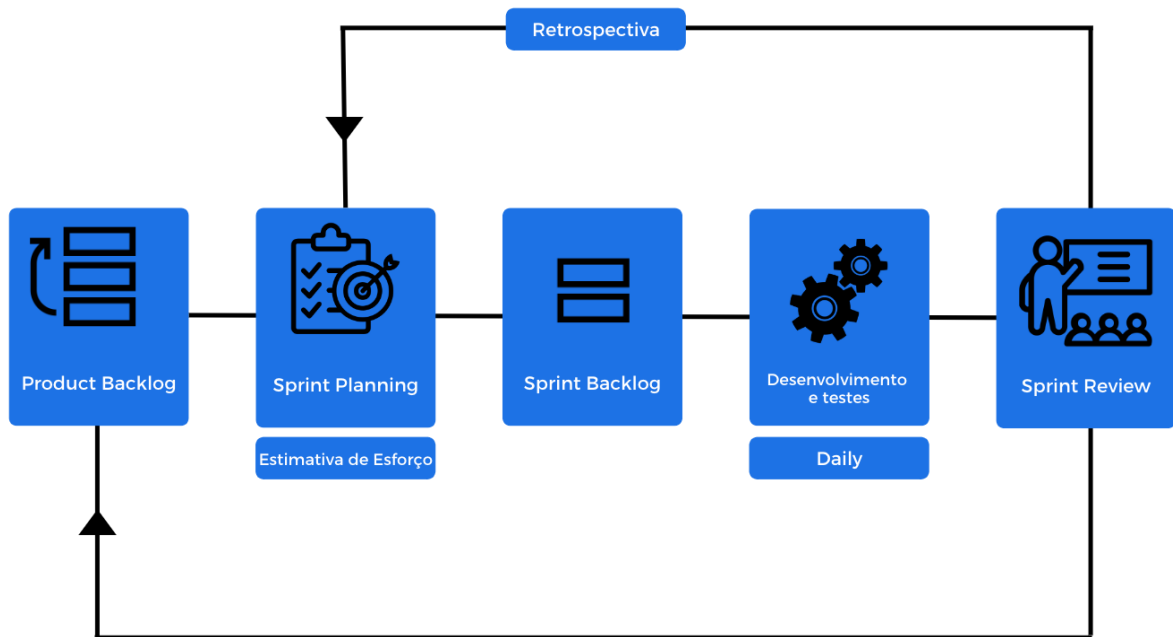
Artefato	Descrição
Daily	Reuniões curtas de alinhamento diário
Sprint Planning	Reunião de planejamento do que será feita naquela sprint
Sprint Review	Reunião de revisão do que foi feito na sprint
Retrospectiva	Reunião de entendimento dos pontos de melhoria daquela sprint para se documentar boas práticas
Product Backlog	Todas as atividades a serem feitas envolvidas na construção de determinado produto
Sprint Backlog	As atividades alocadas para serem desenvolvidas na sprint em questão
Estimativa de esforço	Avaliação do nível de dificuldade e de tempo necessário para implementar determinada atividade

Fonte: Adaptado de Saleh et al. (2018)

Um esquema de como tais artefatos do SCRUM se interagem na rotina de determinado time ágil pode também ser evidenciada a partir da Figura 9.



Figura 9: Artefatos do Scrum



Fonte: Adaptado pelo autor de Scrum.org (2020)

Apesar de grandes benefícios, também são encontrados pontos negativos no Scrum. Um deles é por se tratar de um sistema empurrado, ou seja, todas as tarefas demandadas são empurradas ao mesmo tempo para serem feitas, não importando a capacidade do time, o que gera gargalos e atrasos (ANDERSON e CARMICHAEL, 2016).

Segundo GAETE et al. (2021), o Scrum não fornece tanta flexibilidade quanto os demais métodos. Dessa forma, os times devem saber exatamente como executar cada iteração, quais são as etapas a seguir, quais regras eles devem respeitar, qual é a gama de tempo que eles devem alocar para conclusão de cada atividade, etc.

Uma outra dificuldade relacionada ao SCRUM é em relação à estimativa de esforço, principalmente quando se trata de times juniores. Tal artefato surgiu a partir do Planning Poker e possui como objetivo estimar o esforço das histórias de usuários associadas ao projeto durante a Sprint Planning. Essa estimativa é feita através da suposição de quantos pontos cada história poderia ter e para isso usa-se a sequência de Fibonacci de maneira a garantir avaliações com maiores espaçamentos entre elas, tendo assim mais facilidade em identificar a sua complexidade (GRENNING, 2002).

Contudo, o Planning Poker muitas vezes pode apresentar estimativas mal avaliadas, o que pode impactar diretamente no planejamento do projeto. Para isso, portanto, é sugerido que todo o time participe da avaliação das histórias e não só para quem de fato vai atuar na

história em questão. Isso auxilia na diminuição do otimismo do time e garante um planejamento mais realista para a sprint.

### 3.2.2 XP

Já o **Extreme Programming (XP)**, embora menos famoso que o Scrum, é um dos precursores do manifesto ágil. Dessa forma, muitas de suas práticas estão envolvidas nos pilares do agilismo, como fornecer feedbacks contínuos de ciclos de desenvolvimento curtos e incrementais; além de viabilizar maior flexibilidade nos planejamentos e nos cronogramas de maneira a trazer maior adaptabilidade às mudanças mercadológicas (Rodríguez et al, 2019).

Dessa forma, algumas técnicas envolvidas nesta abordagem são a estruturação de pequenos lançamentos (*releases*); de design simples; de testes; de refatoramento; de pareamento; senso de dono coletivo; de integração contínua; de horas de trabalho sustentável; comunicação presencial com o cliente; e de convenções de codificação (SALEH et al, 2019).

### 3.2.3 LEAN

A metodologia **Lean**, em contrapartida às outras, foi adaptada ao cenário de software, tendo sua abordagem lançada para esse contexto apenas em 2003 por Poppendieck e Poppendieck (Rodríguez et al. 2019).

Dessa maneira, um dos fatores que chamaram atenção para o Lean foi a organização como um todo, sendo também uma abordagem mais flexível que as anteriores e que traz recomendações altamente personalizáveis em prol da diminuição de desperdícios e de retrabalho e da melhoria do fluxo de trabalho. Ademais, houve uma dificuldade latente em escalar os métodos tratados anteriormente, como o Scrum e o XP, de forma sistêmica na organização. Dessa maneira, tal método se tornou uma tratativa promissora quando se unia a outros métodos ágeis de maneira a facilitar a sua escalabilidade (GAETE et al, 2021).

### 3.2.4 KANBAN

Por fim, o último método analisado nesta pesquisa é o **Kanban**. Assim como o Lean, o conceito de Kanban já existia anteriormente no Japão através da Toyota na década de 1960 para se limitar o trabalho em progresso (ANDERSON e CARMICHAEL, 2016)

Entretanto, a técnica só foi apresentada pro cenário de software entre 2003 e 2004 por J. Anderson, com o intuito de ajudar a organizar um pequeno grupo da Microsoft. Contudo,

ele só ganhou robustez em 2010, se transformando em uma abordagem de gestão (AL Aidaros, Omar e Romli, 2021).

Dessa maneira, sua adaptação para o software trouxe como objetivo maior tornar visível o trabalho intangível de conhecimento para que seja coerente com o trabalho necessário solicitado pelo cliente e com o que o time pode propor.

Para que isso seja possível, alguns conceitos vistos no seu surgimento perpetuaram, como a limitação do trabalho em progresso (WIP), de maneira a evitar tanto a sobrecarga do colaborador quanto que se tenha muito trabalho parcialmente incompleto, já que isso pode alavancar desperdícios e grandes despesas, além de que se aumenta o tempo de espera para a entrega do produto, o que evita a sensibilidade do cliente quanto ao seu comportamento no mercado, podendo afetar na entrega de valor.

Ademais, além da limitação do WIP, o método traz mais outros princípios, sendo eles: visualizar o fluxo de trabalho; medir e gerenciar tal fluxo; tornar as políticas de processo explícitas; e usar modelos para se reconhecer oportunidades de melhoria; desenvolver continuamente; e principalmente, implementar feedbacks (Saleh et al, 2019).

Não obstante, diferentemente do Scrum, o Kanban apresenta um sistema puxado, não empurrando as tarefas para seus membros. Quando o trabalho é concluído, então, a capacidade produtiva torna-se disponível ao invés de ser empurrada para o sistema quando o novo trabalho é exigido. Assim, as atividades novas só podem ser começadas quando as pendências atuais forem entregues ou passarem para a próxima fase do fluxo de trabalho (Alaidaros, Omar e Romli, 2021; Anderson e Carmichael, 2016).

Apesar das inúmeras vantagens do método, é perceptível que ele também apresenta determinadas falhas, já que, segundo Alaidaros et al (2021) ele não mostra o objetivo das informações e deixa de relacioná-las com a quantidade que deveria ser realizada para que o projeto atendesse com seus compromissos, já que é um sistema puxado. Os limites errados do Trabalho em Progresso (WIP) trazem atrasos e falhas, além de ser dificilmente mensurados assertivamente.

Existem também outros critérios que afetam o bom funcionamento do Kanban, sendo desafios encontrados na sua implementação: entender o acompanhamento do progresso, determinar os limites do WIP e por fim visualizar o workflow com as suas devidas métricas (Alaidaros et al, 2021).

### 3.3 FATORES DE SUCESSO DE SOFTWARE E MÉTRICAS ÁGEIS

As metodologias ágeis surgiram em um cenário global de necessidade de maior assertividade para com o desenvolvimento de software. Neste contexto, Chow e Cao (2008) e Misra et al. (2009), mapearam os fatores relacionados ao sucesso de um desenvolvimento de software e assim identificaram 4 fatores fundamentais para o sucesso do desenvolvimento, sendo eles:

- Qualidade (entregar um bom produto)
- Escopo (atender aos requisitos e objetivos do projeto)
- Custo (entregar dentro do custo e esforço estimado)
- Tempo (entregar dentro do prazo)

Dessa maneira, qualquer projeto que utiliza de práticas ágeis pode ser pivô de maior sucesso do que práticas tradicionais de desenvolvimento de software se ele conseguir entregar um software com melhor qualidade, em pouco tempo e com um baixo custo (Misra et al, 2009)

É notório, contudo, que existem grandes obstáculos ao se tentar acompanhar o sucesso de uma aplicação em sua fase de desenvolvimento. Diferentemente de produtos físicos, os produtos digitais possuem a informação como bem de produção, dessa forma, a complexidade aumenta já que não se tem algo pontual e visível e sim linhas de códigos conectadas.

Dessa maneira, uma das maneiras de se gerir a qualidade é através do acompanhamento de **métricas** que sinalizem como está a produtividade do time de desenvolvimento. Neste contexto, à medida que os modelos ágeis foram se formando, também surgiram medições popularmente conhecidas como **métricas ágeis**.

Tais métricas são contextualizadas em gerenciamentos ágeis e possuem como objetivo dar suporte na otimização da operação do desenvolvimento do software em questão, ao fornecer informações para que se entenda como está o andamento do projeto e assim permitir tomada de decisões com maior embasamento e assertividade, aumentando assim o sucesso do desenvolvimento.

Diante disso, este projeto terá como foco de aplicação 6 métricas ágeis visualizadas anteriormente no levantamento do estado da arte e que seguem as diretrizes tratadas anteriormente. O Quadro 9 evidencia tais métricas, detalhando seus significados, além de relacioná-las com as 4 dimensões de um projeto evidenciadas anteriormente (Qualidade, Escopo, Tempo e Custo).

Quadro 9: Métricas ágeis escolhidas para análise

Métrica	Significado	Dimensão associada	Cálculo
Velocity	Rapidez em que a equipe está implementando as atividades propostas em determinado período. A métrica é calculada a partir do número de pontos entregues em determinado período (RODRÍGUEZ et al, 2019).	Tempo e Escopo	Número de pontos entregues/sprint
Work in Progress (WIP)	Quantas tarefas estão alocadas em cada etapa da <i>pipeline</i> (mapa das etapas que compõem o processo de desenvolvimento de software) do Quadro Kanban, desde o planejamento até a execução. (VALENTE, 2022). O intuito de entender os trabalhos em progresso na pipeline é ter embasamento para limitar suas quantidades, impedindo que o time se sobrecarregue ou que influencie em um mau planejamento (HOFMAN et al, 2018).	Tempo e Escopo	Quantidade de atividades alocadas na pipeline/sprint
Cycle Time	O tempo necessário para concluir o desenvolvimento de uma atividade, ou seja, o período entre quando ela entra para ser feita e quanto ela de fato é entregue (RODRÍGUEZ et al, 2019).	Tempo	Somatória do tempo final de desenvolvimento - Somatória do tempo inicial de desenvolvimento
Lead Time	Tempo total que um cliente tem que esperar para receber um produto ou funcionalidade após fazer o pedido (RODRÍGUEZ et al, 2019).	Tempo	Somatória do tempo final de desenvolvimento - Somatória tempo inicial de entrada no Product Backlog
Throughput	Número de atividades entregues em determinado período de tempo (ALAI DARO et al, 2021).	Tempo e Escopo	Número de atividades entregues/sprint

Fonte: Elaborado pelo autor

Neste contexto, para aplicação de quaisquer métricas ágeis, Sato (2007) e Hartman e Dymond (2006) sugerem, a partir de heurísticas mapeadas, que antes de suas aplicações, é seja definido diretrizes para a escolhas de boas métricas, aumentando sua aplicabilidade no projeto. Essas diretrizes seriam:

- Reforçar princípios ágeis;
- Medir resultados e não saídas;
- Seguir tendências e não números;
- Responder uma pergunta específica para uma pessoa real;
- Pertencer a um conjunto de métricas e diagnósticos;
- Ser facilmente coletada;
- Revelar, ao invés de esconder, seu contexto, e suas variáveis;
- Incentivar a comunicação sobre os resultados das métricas;
- Fornecer feedback frequente e regular, o que torna fundamental a fácil visibilidade de tais métricas;
- Encorajar um alto nível de qualidade

Contudo, é notório que mesmo com a facilidade de entendimento e de escolha de tais métricas, ainda existem grandes dificuldades em suas aplicações, conforme visto a seguir.

### **3.3.1 Dificuldade de aplicação de métricas ágeis**

As métricas ágeis conseguem trazer visibilidade da operação do time, como por exemplo a sua produtividade. Porém, elas apenas evidenciam numericamente o comportamento de algo, sendo causas relacionadas a esse comportamento, variadas.

Dessa forma, entender métricas ágeis é muito mais do que o entendimento analítico em si, mas também saber quais questões estão ocasionando tais cenários. Então, tais métricas devem ser usadas como um suporte na tomada de decisão, tendo como paralelismo aspectos tanto qualitativos como quantitativos.

Como não foram encontrados na literatura estudos que trouxessem dificuldades relacionadas à implementação de métricas ágeis em si, este estudo irá trazer compilados de fatores associados. O primeiro relacionado à própria aplicação dos métodos ágeis, que por si só pode trazer interferências no comportamento das métricas ágeis. O segundo está relacionado à tomada de decisão em times de desenvolvimento que utilizam de métodos ágeis.

#### **3.3.1.1 Dificuldades na implementação de métodos ágeis**

Para entender as métricas ágeis é necessário entender outros componentes do agilismo que podem trazer interferência em seus resultados. Assim, LOPEZ MARTINEZ et al (2016)

através de uma revisão sistemática da literatura entenderam as principais dificuldades na aplicação de métodos ágeis.

Através desse estudo, foram identificadas 4 categorias relacionadas a essas dificuldades, sendo elas: Organização, Pessoas, Projeto e Processos, conforme apresenta o Quadro 10.

Quadro 10: Principais dificuldades na aplicação de métodos ágeis

<b>Categoria</b>	<b>Principais dificuldades na adoção do Agilismo</b>
Projeto	Tamanho do projeto
Processo	Grau de agilismo
	Definição de práticas ineficazes para melhoria do processo.
Organização	Cultura da organização não está adaptada para a abordagem ágil
	Cultura com carência na capacidade de mudança
	Problemas organizacionais
	Carência de suporte de gestão
	Pressão externa para uso de práticas tradicionais
Pessoas	Indisponibilidade de pessoas treinadas
	Carência de experiência com métodos ágeis
	Carência de comunicação e colaboração com o cliente
	Carência de treinamento do Product Owner e do cliente
	Tamanho do time
	Carência de comunicação efetiva
	Carência de entender valores ágeis
	Capacidade técnica
	Treinamentos inadequados e disfuncionais
	Desalinhamento do time
	Resistência geral de mudança
	Carência de compromisso com a decisão
Rotatividade dos membros do time	

Categoria	Principais dificuldades na adoção do Agilismo	Continua
	Envolvimento contínuo com o cliente	

Fonte: Adaptado de LOPEZ MARTINEZ et al (2016).

Algumas dessas dificuldades relacionadas à adoção do agilismo possuem conexão com a falta de acompanhamento do time de desenvolvimento através de métricas ágeis. As dificuldades relacionadas a categoria Processo, por exemplo, são um exemplo disso, onde a falta de dados para dar suporte para a operação do time pode prejudicar a sua agilidade e permitir definição de práticas ineficazes, já que não possuem embasamento e contextualização, impedindo a melhoria contínua do processo de desenvolvimento.

### 3.3.1.2 Dificuldade na tomada de decisão de desenvolvimento ágil

Outra questão relacionada à projetos ágeis é a dificuldade na tomada de decisão neste tipo de desenvolvimento. Dessa maneira, DRURYA et al (2012), a partir de conversas com 18 entrevistados que tinham contato com métodos ágeis, trouxeram em sua pesquisa diversos fatores que podem ocasionar nessas dificuldades, tendo um olhar processual nas cerimônias do time, sendo elas, o Planejamento, a Execução, a Revisão e a Retrospectiva de uma interação (ou comumente chamada como *sprint*).

O planejamento de uma interação consiste na cerimônia em que serão mapeadas as atividades em que a equipe ágil irá desenvolver em determinado período. Segundo os autores, tais reuniões geralmente não são embasadas de dados e possuem como cerne palpites do time, o que pode representar incoerências nos resultados. Nesse momento, a estimativa de histórias - momento em que se mensura o esforço das atividades que serão desempenhadas - também apresenta diversas dificuldades. Dessa maneira, tais questões podem afetar diretamente as métricas analisadas.

A execução é o momento em que a equipe trabalha no desenvolvimento do que foi planejado, onde parte do produto é desenvolvida e muitas vezes testada. Dessa maneira, as decisões relacionadas a tal etapa são mais de nível técnico (como será desenvolvido, quais serão os testes, qual será o design de interface, entre outros).

A revisão de uma interação é o momento em que a equipe de desenvolvimento apresenta o resultado do que foi desenvolvido para as partes interessadas, onde é evidenciado o que cumprido e o que não, diante do planejado. É nessa etapa, portanto, que a equipe entende se a funcionalidade está atendendo as expectativas do cliente, se as estimativas



precisam ser modificadas, quais histórias devem ser priorizadas para a próxima interação, entre outros. Assim, nessa etapa, é importante a presença de dados para que os stakeholders consigam ver o progresso do time, o que auxilia na qualidade do projeto.

Por fim, a retrospectiva de uma interação é um momento de reflexão, onde a equipe mapeia pontos positivos e negativos da última interação e traça planos de ação para melhorar os pontos mapeados. Dessa maneira, as tomadas de decisão não são dependentes de dados e são mais uma maneira de trazer melhorias em relação às lições aprendidas no projeto.

Diante de tudo isso, DRURYA et al (2012) a partir das 18 entrevistas realizadas com pessoas em contato com o agilismo de 5 organizações diferentes, conseguiram entender as maiores dificuldades relacionadas à tomada de decisão de times, conforme evidenciado no Quadro 11.

Quadro 11: Maiores dificuldades relacionadas à tomada de decisão em times ágeis

Princípios da Tomada de Decisão	Obstáculos da tomada de decisão
A incerteza afeta o tomador de decisões	<ul style="list-style-type: none"> <li>Os membros da equipe ágil não estão dispostos a se comprometer com uma decisão e confiam no scrum master para tal</li> <li>Os membros da equipe ágil não estão tomando decisões apesar da sua autonomia</li> </ul>
As informações não são coletadas de forma racional	<ul style="list-style-type: none"> <li>As decisões são baseadas na disponibilidade instável de pessoal durante uma iteração ágil</li> <li>Os membros da equipe ágil não estão implementando decisões e estão terceirizando tais decisões para outrem.</li> </ul>
Determinadas tomadas de decisão afetam o que foi acordado na <i>planning</i> , ao se priorizar outras tarefas durante a interação.	<ul style="list-style-type: none"> <li>Os membros da equipe ágil não estão dispostos a se comprometer com uma decisão e confiam no scrum master para tomar decisões.</li> <li>Equipes ágeis enfrentam prioridades conflitantes para as decisões</li> </ul>
As diferenças contextuais afetam a tomada de decisões	<ul style="list-style-type: none"> <li>As decisões são baseadas na disponibilidade instável de pessoal durante uma iteração ágil</li> <li>A tomada de decisão em grupo colaborativo sobre equipes ágeis impede que os especialistas tomem decisões</li> </ul>

Fonte: Adaptado de DRURYA et al. (2012)

Nota-se, assim, que dos princípios mapeados, alguns poderiam ser facilitados com a presença de modelos metrificados que evidenciasse o comportamento operacional do projeto em termos de produtividade do time de desenvolvimento, sendo eles: A incerteza afeta o tomador de decisão; As informações não são coletadas de forma racional; e Determinadas tomadas de decisão afetam o que foi acordado na *planning*, ao se priorizar outras tarefas durante a interação.

Sobre o primeiro princípio, é evidente que muitas das tomadas de decisões necessitam de um entendimento analítico para serem definidas. Por exemplo, com o entendimento do comportamento do time de desenvolvimento em termos de produtividade, gestores poderiam passar prazos mais assertivos para seus clientes, já que possuem histórico sobre a capacidade do seu time, evitando esta incerteza.

O segundo princípio também se relaciona com essa necessidade de entendimento analítico do time, já que a presença de métricas ágeis pode trazer informações mais assertivas e objetivas do que pressuposições, garantindo maior embasamento para a tomada de decisão.

Por fim, o terceiro princípio também se relaciona com a temática deste estudo, já que a documentação histórica de métricas de produtividade do time pode auxiliá-lo na priorização de atividades da *sprint* através da *planning*, possibilitando a alocação de uma margem de segurança para atividades que poderiam surgir durante a *sprint*, garantindo que as expectativas dos clientes sejam alcançadas.

Dessa forma, a partir do mapeamento de dificuldades relacionadas tanto a aplicação de metodologias ágeis quanto com a tomada de decisão em contexto ágil, foi possível entender no que as métricas ágeis relacionadas a produtividade do time de desenvolvimento poderiam contribuir, fornecendo mais embasamento para a elaboração do estudo de caso.

## **4. METODOLOGIA**

Segundo Selltiz et. al (1974), a metodologia científica garante ao estudo maior confiabilidade e capacidade de replicação ao se levantar o conjunto de caminhos executados para se alcançar respostas mais precisas e não tendenciosas, garantindo maior acurácia aos resultados.

Desta maneira, tal seção apresenta a metodologia relacionada a este projeto, através do detalhamento do seu método e a estruturação da pesquisa.

### **4.1 MÉTODO DA PESQUISA**

A pesquisa é caracterizada como um estudo de caso, que consiste na utilização de um ou mais métodos qualitativos de captação de informação, não seguindo uma linha rígida de investigação. Esse método ocorre como uma maneira de penetrar uma realidade social, não alcançada apenas por um levantamento amostral e quantitativo (COSTA et al, 2008).

Dessa maneira, este método envolve o entendimento do comportamento humano e surgiu como uma maneira de se explicar o "como?" e "por que?" de questões complexas que envolvem o contexto humano da vida real. Assim, ele é uma ferramenta investigativa utilizada para compreender processos complexos no âmbito social, como situações problemáticas, análise de dificuldades e entre outros (YIN, 2001).

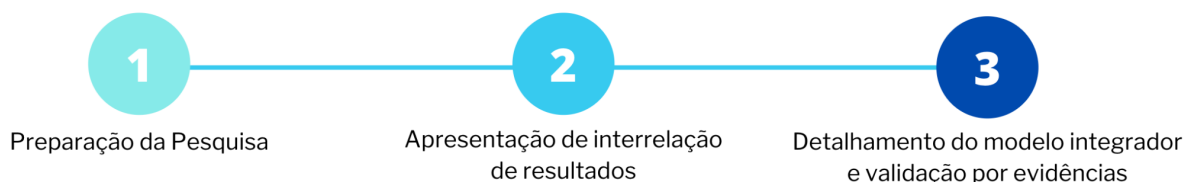
A pesquisa também trata-se de natureza aplicada - quando existe a resolução prática da questão da pesquisa - e qualitativa, onde através da localização do observador é possível interpretar subjetivamente comportamentos sociotécnicos em grupos, envolvendo assim materiais empíricos, como experiências pessoais; relatos de introspecção; produções e artefatos do projeto; interações; e tudo que detalhe a rotina da vida humana em grupo (DENZIN e LINCOLN, 2000).

O caráter exploratório da pesquisa busca de maneira investigativa uma proximidade com a realidade do objeto de estudo, justamente pela carência de informações em relação à temática analisada. Ela também foi desenvolvida com o intuito prático de resolução de problemas específicos através da exploração de uma determinada problemática (RUY, 2002).

As técnicas utilizadas para a sua aplicação foram análise documental, entrevistas semi-estruturadas e utilização do programa Google DataStudio para construção do dashboard.

A análise documental utilizou como estratégia a Teoria do Enfoque Meta Analítico Consolidado de maneira a selecionar os estudos de maior relevância da literatura acerca da temática da pesquisa, além de garantir o direcionamento das métricas ágeis que foram utilizadas no projeto (MARIANO e ROCHA, 2017). Para isso, o TEMAC foi estruturado com base nas etapas visualizadas na Figura 10.

Figura 10: Etapas do TEMAC



Fonte: Elaborado pelo autor

Na primeira etapa do TEMAC foram definidas as palavras chaves escolhidas para se captar as pesquisas relacionadas à temática da pesquisa nas bases de dados (Scopus, Web of Science e entre outras), selecionando-se também o espaço tempo e as áreas de conhecimento desejadas.

Na segunda etapa do TEMAC foram inter relacionados os resultados alcançados na etapa anterior, filtrando-se com critérios definidos pelo autor, como, por exemplo, as revistas mais relevantes, as que mais publicaram sobre o tema. Essa etapa foi fundamental para lapidar ainda mais os estudos que direcionaram o projeto.

A última etapa foi importante para integrar as pesquisas realizadas, validadas por evidência. Para isso, criou-se 3 análises através da plataforma VOSViewer. A primeira delas é a **co-ocorrência**, que identificou a estrutura conceitual sobre o estudo, trazendo uma rede de temas associada a ele através da ocorrência de palavras-chaves. A segunda delas foi a **cocitação**, que verificou artigos citados juntos, podendo indicar semelhanças entre eles ao tratar das abordagens convergentes. E por fim, criou-se o **coupling**, que traz uma perspectiva das frentes de pesquisas mais utilizadas (GUEDES e BORSCHIVER, 2005). Essas análises garantem a seleção dos estudos de maior relevância para com a pesquisa, trazendo mais acurácia no levantamento do estado da arte.

Além disso, as entrevistas foram realizadas através de estudos relacionados à temática e contextualizadas com o desenvolvimento do software da plataforma PUMA. Assim, foram divididas em três grupos de entrevistados, sendo o primeiro grupo, estudantes da disciplina que gerencia o desenvolvimento do software no semestre vigente (disciplina de Engenharia de Produto de Software - EPS - do curso de Engenharia de Software da Universidade de

Brasília), onde foram entrevistados 4 estudantes. O segundo grupo também possui como público alvo estudantes desta disciplina, porém do semestre anterior, e contemplou 3 entrevistados. E o terceiro grupo, são os clientes que auxiliam na visão do produto em questão, onde houveram 2 entrevistados.

Nos grupos entrevistados, aqueles relacionados aos estudantes (Grupo 1 e Grupo 2), possuíam as mesmas questões utilizadas no roteiro das entrevistas semi-estruturadas. Tais questões são relacionadas aos métodos ágeis; à tomada de decisão; às ferramentas de acompanhamento de dados; às métricas ágeis; às cerimônias do time; e ao gerenciamento do projeto. Já os relacionados aos clientes (Grupo 3), tiveram como foco aspectos como suas expectativas em relação ao projeto; dados que recebem e que gostariam de receber; acompanhamento operacional do time; comunicação; priorização de atividades; e gerenciamento do projeto. As questões do roteiro podem ser visualizadas no Apêndice 1.

## 4.2 ESTRUTURAÇÃO DA PESQUISA

Para a realização deste trabalho, os objetivos foram divididos em 4 etapas, sendo a primeira, a própria revisão da literatura. A segunda etapa está relacionada à priorização das métricas relacionadas aos projetos - ambas etapas consolidadas no levantamento do estado da arte. A terceira etapa se refere às entrevistas. E por fim, a quarta etapa está relacionada com a implementação dos planos de ação formulados durante o estudo de caso. Tais etapas, associadas com seus respectivos objetivos, podem ser visualizadas no Quadro 12.

Quadro 12: Estruturação da pesquisa

Objetivo geral		
Elaborar uma ferramenta de visualização de métricas ágeis que possa ser utilizada no gerenciamento de projetos de desenvolvimento de software, a fim de contribuir com fatores de sucesso do módulo da Plataforma Unificada de Metodologias Ativas (PUMA).		
Objetivos Específicos	Etapas da Pesquisa	Procedimentos Técnicos
Identificar as metodologias ágeis de desenvolvimento de software e as métricas associadas a elas	<b>Etapa 1: Revisão da literatura</b>	
	1.1 Preparação de Pesquisa	Acessos às bases de dados indexadas (Scopus)
	1.2 Levantamento do Estado da Arte (TEMAC)	
	1.3 Análise dos Estudos	

	<b>Etapa 2 - Seleção de métricas</b>	
Selecionar as métricas ágeis de maior relevância na literatura	2.1 Priorização das métricas mais citadas na literatura	Base de dados priorizadas: Scopus
	<b>Etapa 3 - Levantamento de dificuldades de aplicação de métricas ágeis na visão dos stakeholders</b>	
Levantar as principais dificuldades de aplicação de tais métricas no contexto do PUMA	3.1 Entendimento na literatura das maiores dificuldades em aplicar as métricas selecionadas	Análise Documental
	3.2 Levantamento das dificuldades da aplicação de métricas ágeis na visão dos stakeholders	Entrevistas semi-estruturada
	<b>Etapa 4 - Implementação da Solução</b>	
Viabilizar o entendimento e implementação de métricas ágeis para a melhoria da tomada de decisão da PUMA	4.1 Levantamento de pontos de melhoria de gerenciamento a partir de métricas ágeis	Compilação dos dados das entrevistas semi-estruturadas
	4.2 Desenvolvimento da ferramenta de visualização de métricas ágeis	Dashboard (Google DataStudio)
	4.3 Análise crítica dos dados coletados	Análise Documental
	4.4 Feedback dos entrevistados sobre a solução proposta	Entrevistas semi-estruturada

Fonte: Elaborado pelo autor

A Etapa 1 teve como objetivo a Revisão da Literatura de maneira a garantir o embasamento teórico necessário para que esta pesquisa pudesse acontecer. Sendo assim, essa etapa foi dividida em 3 sub etapas vistas a seguir:

Sub Etapa 1.1 - Preparação da pesquisa: consiste em determinar os termos que seriam pesquisados (palavras chaves); as bases de dados; o intervalo de tempo das pesquisas; além de filtrar as áreas que mais tinham relação com a temática.

Sub Etapa 1.2 - Levantamento do Estado da Arte (TEMAC): Através do TEMAC foi possível entender as pesquisas de maior popularidade que se relacionavam com a temática do projeto.

Sub Etapa 1.3 - Análise dos Dados: A partir da leitura dos artigos mais relevantes, conforme evidenciado na etapa anterior, foi entendido, daqueles evidenciadas, as que mais tinham impacto com a temática do projeto.

A Etapa 2 teve como objetivo a seleção das métricas ágeis que guiarão a pesquisa, através da sub etapa vista a seguir:

Sub Etapa 2.1 - Priorização de métricas mais citadas na literatura: Através da identificação dos estudos de maior relevância para este projeto, é possível analisar quais

medidas e métricas relacionadas ao agilismo possuem maior recorrência dentre os principais estudos. Tal passo é essencial para definição das métricas cujo autor atuará com foco no estudo de caso.

A Etapa 3 teve como objetivo o levantamento das principais problemáticas relacionadas ao tema desta pesquisa, de maneira a garantir o embasamento para a elaboração da solução. Ela foi possibilitada através das seguintes sub etapas:

Sub Etapa 3.1 - Entendimento na literatura das maiores dificuldades em implementar as métricas selecionadas: Esta etapa é fundamental para embasar melhor o autor na construção do roteiro das entrevistas relacionadas ao tema.

Sub Etapa 3.2 - Levantamento das dificuldades da implementação de métricas ágeis na visão dos stakeholder: Através das entrevistas semiestruturadas, é possível entender as maiores dificuldades relacionadas ao tema, além de entender o grau de recorrência dessas dificuldades mediante os entrevistados, fator que auxiliará na escolha de oportunidades com maior impacto.

A Última Etapa teve como objetivo a Implementação da solução através do mapeamento das principais dificuldades relacionadas ao tema, observados na etapa anterior. Para isso, foram necessárias as seguintes sub etapas:

Sub Etapa 4.1 - Levantamento de pontos de melhoria de gerenciamento a partir de métricas ágeis: Através das dificuldades de maior impacto relacionado ao tema do autor, é possível formular possibilidades de planos de ação que poderiam solucionar ou dar suporte na problemática em questão.

Sub Etapa 4.2 - Desenvolvimento da ferramenta de visualização de métricas ágeis: implementação prática da proposta definida na etapa anterior.

Sub Etapa 4.3 - Análise crítica dos dados coletados: Através da elaboração do Data Studio com as métricas ágeis do projeto, foi discutido criticamente o comportamento de cada dashboard, já configuradas com os dados da equipe de desenvolvimento, provindos a partir da API do Github e do Zenhub.

Sub Etapa 4.4 - Feedback dos entrevistados sobre a solução proposta: Através de entrevistas semi estruturadas realizadas após a implementação da solução, foram coletados feedbacks com estudantes da manteria de Engenharia de Produto de Software do semestre vigente e do anterior, a fim de entender suas possíveis contribuições e pontos de melhoria com a solução proposta.

## 5. ESTUDO DE CASO: DIFICULDADES NA IMPLEMENTAÇÃO DA FERRAMENTA DE VISUALIZAÇÃO DE MÉTRICAS ÁGEIS

O objeto deste estudo de caso é a plataforma *web* PUMA - Plataforma Unificada de Metodologia Ativa. Essa plataforma está sendo desenvolvida através de parceria entre o curso de Engenharia de Produção da Faculdade de Tecnologia (FT) e o curso de Engenharia de Software da Faculdade do Gama (FGA), ambas sediadas na Universidade de Brasília.

As partes interessadas envolvidas no desenvolvimento da PUMA são as disciplinas de Método de Desenvolvimento de Software (MDS) e de Engenharia de Produto de Software (EPS) da FGA, que focam, respectivamente, no desenvolvimento técnico e gerencial da plataforma. Além disso, a disciplina de Projeto de Sistema de Produção 5 da FT auxilia na melhoria do produto - através de testes de funcionalidade - e os mestres de Computação Aplicada garantem suporte estratégico, como clientes do projeto (MONTEIRO; CAMPOS; LIMA; MARIANO, 2018).

Neste contexto, a plataforma PUMA possui como objetivo unificar e dar suporte aos processos que envolvem a aplicação de metodologias de aprendizado ativa, principalmente às relacionadas às disciplinas de Projeto de Sistemas de Produção (PSP) do curso de Engenharia de Produção da Universidade de Brasília. Tais metodologias possuem como foco garantir o real aprendizado do estudante, tornando-os protagonistas no processo do seu próprio conhecimento, ao aproximá-lo da abordagem PBL (Problem Based on Learning), onde através de problemas reais, os alunos podem aprender na prática, não somente competências técnicas, mas também interdisciplinares (MONTEIRO; CAMPOS; LIMA; MARIANO, 2018).

Dessa maneira, a plataforma PUMA deseja integrar propostas de projetos submetidas pelos stakeholders - sendo elas, problemas reais - aos alunos, de maneira que eles encontrem uma solução prática para tais problemas, possibilitando assim que eles resolvam os problemas por meio da execução de projetos. Com isso, os estudantes possuem suas competências transversais sendo acompanhadas e avaliadas para que o real aprendizado ocorra.

Para que a plataforma PUMA tenha seu desenvolvimento finalizado, é necessário que ela seja dividida em partes, onde, a cada semestre, um módulo é idealizado, detalhado e desenvolvido pelos estudantes. Essa idealização e detalhamento ocorre no início do projeto através de uma metodologia denominada *Lean Inception*, onde os alunos, através de dinâmicas, decidem a sequência de atividades que definirão as funcionalidades do Mínimo Produto Viável (MVP) (CAROLI, 2018).

As atividades do *Lean Inception* são programadas para acontecer de segunda a sexta,



onde no primeiro dia constrói-se a visão de produto - onde o produto quer chegar -, definindo a essência do valor do negócio para os clientes, além de se entender o que o produto é, o que ele não é, o que ele faz e o que ele não faz. O segundo dia é entendido as personas (usuários almejados para utilizar o produto) e como se dará a jornada para cada perfil de usuário. No terceiro dia é entendido as funcionalidades que o produto deve ter, além de uma revisão técnica, de negócio e de UX, a fim de amadurecer o entendimento do que foi levantado (CAROLI, 2018).

No quinto dia é identificado os pontos de interação dos usuários com o produto e, a partir disso, é verificado a sequência de utilização das funcionalidades pelos usuários, onde é entendido o que será entregue no MVP em questão e o que será incrementado. O último dia é necessário para detalhar o MVP, descrevendo-se as personas, as jornadas, a proposta de valor, o custo, o cronograma, os resultados esperados e as métricas que serão utilizadas para validar a hipótese de negócio (CAROLI, 2018).

Os times de desenvolvimento da PUMA, tanto de MDS quanto de EPS, necessitam executar a Lean Inception em todos os semestres para colocarem em prática essas técnicas de definição de um MVP. O processo, porém, é consolidado através de um escopo já existente, sendo fundamental para aprofundar o conhecimento das funcionalidades. Este escopo integral do produto contempla os seguintes épicos:

- Divulgação dos melhores projetos desenvolvidos;
- Submissão, avaliação e alocação de propostas de projeto;
- Avaliação *Peer*, criação de times, recomendação de treinamentos e criação de cursos;
- Atribuição de notas do curso;
- Feedbacks dos stakeholders;

A partir desse escopo, a PUMA já possui alguma de suas funcionalidades desenvolvidas pelos semestre anteriores, sendo elas: Cadastro, Login e Esqueci a senha; o nível do usuário; CRUD de palavras chaves

Dessa forma, o módulo escolhido para ser desenvolvido neste semestre de desenvolvimento da PUMA tem como escopo inicial melhorias de funcionalidades desenvolvidas no semestre anterior, como a submissão de propostas de projeto, a página de detalhamento da proposta, o CRUD de disciplinas e a avaliação das propostas de projetos. Em seguida, será implementada a *homepage* da PUMA, o CRUD - possibilidade de criar, ler, atualizar e deletar informações - de semestres; e a alocação simplificada de projetos.

Neste contexto, o desenvolvimento e melhoria dessas funcionalidades foram foco

deste estudo de caso, onde o autor, através de entrevistas, entendeu as principais dificuldades relacionadas à gestão do time de desenvolvimento, especificamente a aspectos analíticos relacionados a métricas ágeis.

Após tais entrevistas, os dados foram compilados de maneira a se entender as questões mais recorrentes entre os entrevistados. Em seguida, foi feita uma análise com possíveis planos de ação relacionados à temática do estudo em questão. Por fim, foi desenvolvido uma proposta de solução diante do que foi encontrado, não tendo, contudo, a aplicação no semestre vigente.

## **5.1 ENTENDIMENTO NA LITERATURA DAS MAIORES DIFICULDADES EM IMPLEMENTAR AS MÉTRICAS SELECIONADAS**

O primeiro passo para preparação da pesquisa foi entender através da literatura as principais dificuldades relacionadas à aplicação de métricas ágeis no contexto de software. Conforme evidenciado anteriormente, a literatura carece ainda de informações relacionadas a tais dificuldades, contudo, dificuldades relacionadas à implementação de métodos ágeis e tomada de decisão em contexto de agilismo serviram de embasamento para a formulação do roteiro de entrevista, sendo eles evidenciados no Quadro 13.

Quadro 13: Dificuldades relacionadas à implementação de métricas ágeis

<b>Dificuldades relacionadas à aplicação de métricas ágeis</b>
Grau de agilismo
Definição de práticas ineficazes para melhoria do processo.
A incerteza afeta o tomador de decisões
As informações não são coletadas de forma racional
Determinadas tomadas de decisão afetam o que foi acordado na <i>planning</i> , ao se priorizar outras tarefas durante a interação.

Fonte: Elaborado pelo autor

As maiores dificuldades relacionadas à aplicação de métricas ágeis encontradas na literatura são relacionadas ao grau de agilismo da equipe, ou seja, o quanto aquele time está contextualizado com os métodos ágeis; a definição de práticas ineficazes para melhoria de processo e a incerteza ao se tomar decisões, ocasionadas principalmente pela falta de embasamento dos tomadores de decisão; as informações não serem coletadas de forma

racional, e sim subjetivamente pelo gestor; e o fato de que existe despriorização de atividades da planning pelo tomador de decisão, ocasionadas principalmente por não haver um histórico que detalhe como e no que o time atuou anteriormente.

Assim, a partir desse entendimento, foi possível construir o roteiro das entrevistas, contemplados tais pontos. A construção do roteiro foi fundamental para guiar os estudantes apropriadamente com o contexto do projeto.

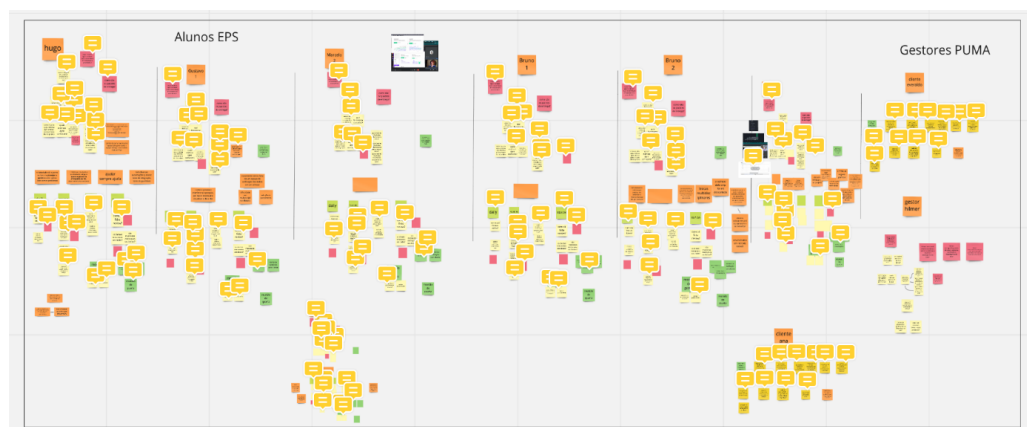
## 5.2 LEVANTAMENTO DAS DIFICULDADES DA APLICAÇÃO DE MÉTRICAS ÁGEIS NA VISÃO DOS STAKEHOLDERS

Após a confecção do roteiro de entrevistas, as mesmas foram realizadas tanto com estudantes de EPS quanto com mestres de Computação Aplicada, garantindo também o entendimento do ponto de vista estratégico do projeto. Em relação aos estudantes de EPS, foram escolhidos alunos do semestre vigente e do anterior, de maneira a captar mais informações sobre as principais dificuldades.

As entrevistas tiveram como média de tempo 40 minutos de duração, sendo elas feitas sincronamente através da plataforma do Google Meets. Conforme evidenciado na metodologia, elas passaram por pontos relacionados à tomada de decisão através de dados (com foco em métricas ágeis), a cerimônias ágeis e a questões gerenciais do projeto, tendo como coletar tanto feedbacks quanto insights dos estudantes. É importante ressaltar que o anonimato de todos os entrevistados foi respeitado.

As entrevistas tiveram suas documentações realizadas na ferramenta MIRO, conforme evidenciado na Figura 10.

Figura 10: Dados das entrevistas no MIRO



Fonte: Elaborado pelo autor

Através dessa ferramenta, foi possível transcrever os pontos coletados de maneira

visual e facilitada para assim ter uma visão holística sobre as informações, fator importante para entender os pontos convergentes evidenciados na seção 5.3.

### 5.3 LEVANTAMENTO DE PONTOS DE MELHORIA DE GERENCIAMENTO A PARTIR DE MÉTRICAS ÁGEIS

Após a coleta dos dados dos entrevistados, o pesquisador analisou individualmente cada uma das entrevistas de maneira a identificar pontos de convergência em relação às dificuldades encontradas. Assim, através de 3 pilares - **processo, produto e dados** -, foi possível categorizar as principais dores dos entrevistados, o que facilitou a sua análise e refinamento para a temática deste estudo. Além disso, também foram revisitados os insights levantados através das entrevistas. Tais pilares junto com o número de dificuldades encontradas em cada um deles são evidenciados na Figura 11.

Figura 11: Categoria e quantidade de dificuldades levantadas pelos entrevistados



Fonte: Elaborado pelo autor

Além de tais categorizações, também foi entendido o quão recorrente eram essas dificuldades em relação a cada grupo de entrevistados (estudantes do semestre atual, estudantes de semestres anteriores e clientes). O resultado da compilação de dificuldades juntos com suas respectivas recorrências podem ser vistos no **Quadro das Principais Dificuldades com Ordenação Decrescente de Recorrência** (Quadro 14), onde o Grupo 1 são os estudantes de EPS do semestre atual; o Grupo 2, estudantes de EPS do semestre anterior; e o Grupo 3, os clientes.

Quadro 14: Quadro das Principais Dificuldades com Ordenação Decrescente de Recorrência.

<b>Dificuldades encontradas pelos entrevistados</b>	<b>Grupo 1</b>	<b>Grupo 2</b>	<b>Grupo 3</b>	<b>Total</b>
Não existe uma definição clara de dono do produto (Product Owner) no projeto	3	1	2	6
Há sobrecarga de atividades para alguns membros da equipe	2	2	2	6
Pessoas inexperientes não sabem como aplicar métodos ágeis na íntegra	2	3	0	5
Manipulação do Zenhub	2	2	1	5
Não existe um histórico da evolução do produto atualizado (erros, melhoria, growth)	3	0	2	5
Alunos matriculados pós pandemia entram nas matérias menos preparados	3	1	0	4
Elaboração do <i>Lean Inception</i> em um projeto já em andamento	2	1	1	4
Previsibilidade dos prazos	2	1	0	3
Estimativas são desalinhadas com a experiência individual dos desenvolvedores	3	0	0	3
Necessidade de maior presença dos clientes nas priorizações	2	0	1	3
Comunicação com os gestores	2	0	1	3
Existe a necessidade, por parte dos desenvolvedores, de maior envolvimento estratégico	2	1	0	3
Inconsistência nos gráficos gerados pelo Zenhub	2	1	0	3
Existe a necessidade de análise individual da performance dos desenvolvedores	2	0	1	3
Existe a necessidade da centralização da documentação do projeto	0	2	1	3
Não existem horários convergentes para as Dailies	1	1	0	2
Necessidade de mais trocas multidisciplinares entre a equipe	1	1	0	2
Necessidade, por parte dos desenvolvedores, de auxiliar os clientes nas tomadas de decisão	2	0	0	2
Existe a necessidade de maior visibilidade da operação do time	0	0	2	2
Existe a necessidade de documentação de algumas cerimônias	0	1	1	2
Conhecimento do produto pelo próprio cliente	2	0	0	2
Wiki atualizada	0	0	2	2
Necessidade de centralizar os artefatos do projeto	0	0	2	2
Não há utilização de dados nas Sprint Planning	1	0	0	1
Existe a necessidade de dar e receber mais feedbacks entre os desenvolvedores	1	0	0	1
Existem falhas na contextualização do produto	1	0	0	1
Existem falhas com o projeto de escopo aberto	0	1	0	1
Possuem dificuldade na disciplina de acompanhamento de métricas	0	1	0	1
Não mapeiam dependências técnicas	0	1	0	1
As entregas finais são muito próximas do final do semestre o que compromete efetiva finalização e qualidade do que foi entregue	0	0	1	1
Necessidade de melhor alinhamento entre o corpo estratégico	1	0	0	1
Existem discordâncias relacionadas à responsabilização de atividades para outras disciplinas	1	0	0	1
Existe um maior foco na metodologia do que no produto em si	0	0	1	1
Gostariam de ter maior participação dos estudantes nas Reviews	1	0	0	1
Gostariam de ter mais visibilidade do que conseguiram entregar	1	0	0	1

Dificuldades encontradas pelos entrevistados	Grupo 1	Grupo 2	Grupo 3	Total Continua
Gostariam de visualizar o histórico de atividades desenvolvidas	1	0	0	1
Quadro de conhecimento e plano de comunicação não estão sendo utilizados	1	0	0	1
Possuem dificuldades na montagem de gráficos	0	1	0	1
Possuem dificuldades na atualização diário do Quadro Kanban	0	1	0	1

Fonte: Elaborado pelo autor

Além das dificuldades, também foram levantados insights pelos entrevistados aos quais foram analisados para a elaboração dos planos de ação. Tais *insights* podem ser evidenciados na Figura 12.

Figura 12: Insights levantados pelos entrevistados



Fonte: Elaborado pelo autor

É possível concluir que algumas das dificuldades encontradas estão relacionadas intrinsecamente com **questões processuais**, onde diversos entrevistados relataram a necessidade de uma melhor organização das funções, por exemplo, definir alguém para ser o *Product Owner* do projeto - fato justificado pela metodologia ágil adotada pelo professor de

MDS e EPS em suas disciplinas -; questões relacionadas à própria estruturação do **produto** - como a necessidade de otimizar o *Lean Inception* (processo imersivo de idealização do produto que ocorre no início do semestre) -; e questões relacionadas ao uso de **dados** para tomada de decisões, ponto focal deste estudo de caso. Neste contexto, as dificuldades com maior recorrência dos entrevistados relacionados a esta última categoria estão apresentadas no Quadro 15.

Quadro 15: Principais dificuldades dos entrevistados (categoria de dados)

Nº	Dificuldades	Pontuação
1	Dificuldade de manipulação dos dados do Zenhub	5
2	Não existe um histórico da evolução do produto atualizado (erros, melhoria, growth)	5
3	Previsibilidade dos prazos	3
4	Estimativas são desalinhadas com a experiência individual dos desenvolvedores	3
5	Inconsistência nos gráficos gerados pelo Zenhub	3
6	Existe dificuldade em obter uma análise individual de desempenho dos desenvolvedores	3
7	Necessidade de centralizar os artefatos do projeto	2

Fonte: Elaborado pelo autor

Dificuldade 1 - Existe dificuldade de manipulação do Zenhub: uma das críticas relacionadas ao GitHub - e sua integração com o Zenhub - é tanto relacionada à acessibilidade e confiabilidade dos dados quanto à dificuldade em flexibilizar e gerar novas análises que não sejam as já fixadas na plataforma - Velocity e BurnDown.

Dificuldade 2 - Não existe um histórico da evolução do produto atualizado (erros, melhoria, *growth*): Esse ponto está associado à necessidade de se visualizar um histórico que armazena informações como bugs, melhorias e novas funcionalidades, facilitando a gestão do conhecimento dos membros do projeto.

Dificuldade 3 - Existe dificuldade com a previsibilidade dos prazos: as dificuldades com os prazos muitas vezes se dão por falta de dados para a tomada de decisão.

Dificuldade 4 - Estimativas são desalinhadas com a experiência individual dos desenvolvedores: isso ocorre por muitas vezes não haver a união efetiva entre a estimativa de esforço das atividades que serão desenvolvidas na sprint e a curva de aprendizado dos estudantes, o que leva a grandes diferenças de notas entre os avaliadores.

Dificuldade 5 - Existe inconsistência nos gráficos gerados pelo Zenhub: tal problemática traz à tona a necessidade de outras análises para maior embasamento do comportamento dos dados gerados no Zenhub.

Dificuldade 6 - Existe dificuldade em obter uma análise individual de desempenho dos desenvolvedores: alguns entrevistados citaram que gostariam de ter um histograma por indivíduo, bem como a visualização do que eles estão atuando (por parte dos clientes) para auxiliar com maior precisão em cada sujeito, garantindo assim planos de ações personalizados.

Dificuldade 7 - Existe a necessidade de centralizar os artefatos do projeto: foi visualizado uma problemática relacionada ao projeto ter diversos artefatos descentralizados e desconectados. Assim, alguns stakeholders, por exemplo, utilizam o Zenhub para documentação, enquanto outros utilizam a Wiki; alguns utilizam o Slack para comunicação, enquanto outros utilizam o Discord.

A partir das dificuldades mencionadas, é possível identificar sua relação com a necessidade dos stakeholders do projeto de possuir mais dados para a tomada de decisão e de uma melhor documentação. Estabelecer planos de ações que auxiliem na melhoria de tais pontos pode contribuir com os fatores de sucesso do projeto, garantindo sua real aplicabilidade no contexto universitário.



## 6. ESTUDO DE CASO: DESENVOLVIMENTO DA FERRAMENTA DE VISUALIZAÇÃO DE MÉTRICAS ÁGEIS

Em face das 7 dificuldades levantadas pelo time de desenvolvimento da PUMA na seção 5.3, propôs-se como plano de ação o desenvolvimento de uma ferramenta de visualização de métricas ágeis no Google DataStudio que permitisse unificar as principais informações relacionadas a métricas ágeis do time de desenvolvimento, garantindo a possibilidade de uma análise crítica sobre os dados provindos do Github e do Zenhub, além de possibilitar a construção de novos gráficos, dando maior flexibilidade para os *stakeholders* do projeto de gerar novas análises. Nesse dashboard, foi incluído, além das métricas ágeis escolhidas nesta pesquisa, outras informações desejadas por tais stakeholders.

Dessa maneira, as métricas e informações selecionadas através da quantidade de recorrência dos estudos levantados no estado da arte para serem implementadas são: *Throughput; Velocity; Lead time; Cycle Time; Work in Progress; Histórico de Pontos e de Número de Issues Alocados (com filtros de Sprint e Usuário); e Histórico de Detalhamento das Issues.*

Para gerar a ferramenta para leitura desses dados (*Google Data Studio*), foi necessário fomentar uma integração entre o Quadro Kanban do Zenhub - local onde as atividades relacionadas ao produto são gerenciadas e direcionadas aos responsáveis. Assim, foi necessário construir, através da API do Github e do Zenhub, uma implementação que integrasse as informações sobre as issues (Github) e sobre as *pipelines* (Zenhub), garantindo que tais dados pudessem ser utilizados para a configuração desses gráficos. As tecnologias utilizadas para tal implementação foram NodeJS (plataforma para rodar Javascript fora do navegador) e Json, utilizando como editor de código o Visual Studio Code.

A partir de tal integração, foi possível baixar os arquivos *.csv* com todos os dados necessários para a construção do Google Data Studio. Dessa maneira, o único ponto de interação para atualização dos gráficos no dashboard é executar o código no final da sprint que se queira analisar, baixar o *.csv* com os dados e centralizá-los num banco de dados que unifique tais informações (o *Google Sheets* foi utilizado).

## 6.1 LEVANTAMENTO DOS DADOS NECESSÁRIOS PARA A CONSTRUÇÃO DOS GRÁFICOS

A primeira etapa foi entender como as medidas e cálculos relacionados às métricas ágeis escolhidas se configuraram. Além disso, foi entendido também a configuração daquelas relacionadas aos outros pontos visualizados anteriormente, como o que cada usuário está atuando e o histórico individual deles.

Para isso, foi necessário entender todos os dados e cálculos relacionados a essas métricas e medidas, que estão evidenciados no Quadro 16.

Quadro 15: Estrutura das métricas escolhidas para implementação

Métrica	Medidas relacionadas	Cálculo
Velocity	<ul style="list-style-type: none"> <li>• Número de pontos entregue</li> <li>• Intervalo de tempo analisado</li> </ul>	<ul style="list-style-type: none"> <li>• Número de pontos entregues/sprint</li> </ul>
WIP	<ul style="list-style-type: none"> <li>• Quantidade de atividades em aberto na <i>pipeline</i></li> <li>• Intervalo de tempo analisado</li> </ul>	<ul style="list-style-type: none"> <li>• Quantidade de atividades alocadas na <i>pipeline</i>/sprint</li> <li>• Quantidade de atividades alocadas na pipeline /dia</li> </ul>
Cycle Time	<ul style="list-style-type: none"> <li>• Tempo em que uma atividade entrou em progresso</li> <li>• Tempo em que uma atividade foi finalizada</li> </ul>	<ul style="list-style-type: none"> <li>• Somatória do tempo final de desenvolvimento - Somatória do tempo inicial de desenvolvimento</li> </ul>
Lead Time	<ul style="list-style-type: none"> <li>• Tempo em que uma atividade entrou em Product Backlog</li> <li>• Tempo em que uma atividade foi finalizada</li> </ul>	<ul style="list-style-type: none"> <li>• Somatória do tempo final de desenvolvimento - Somatória tempo inicial de entrada no Product Backlog</li> </ul>
Throughput	<ul style="list-style-type: none"> <li>• Quantidades de atividades entregues</li> <li>• Intervalo de tempo analisado</li> </ul>	<ul style="list-style-type: none"> <li>• Número de atividades entregues/sprint</li> </ul>
Análise por indivíduo	<ul style="list-style-type: none"> <li>• Histórias em atuação por indivíduo</li> <li>• Pontos alocados para o indivíduo</li> <li>• Número de atividades alocadas para o indivíduo</li> <li>• Intervalo de tempo analisado</li> </ul>	<ul style="list-style-type: none"> <li>• Quantidade de pontos alocados/sprint (com filtro para cada desenvolvedor)</li> <li>• Quantidade de atividades alocadas/sprint (com filtro para cada desenvolvedor)</li> </ul>

Fonte: Elaborado pelo autor

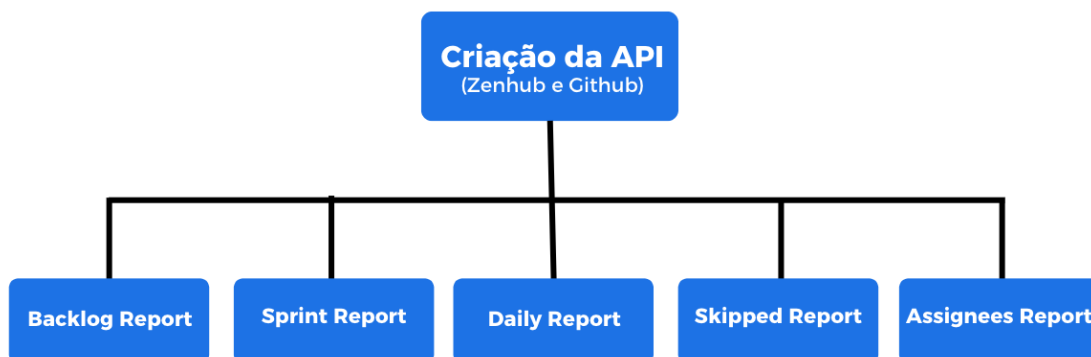
A partir do entendimento de todas as medidas relacionadas às métricas que serão apresentadas no Google Data Studio, foi possível dar início a codificação. Dessa forma, essas medidas direcionaram o entendimento de quais informações do Zenhub e do Github deveriam

ser integradas na API, conforme visto na seção 5.4.2.

## 6.2 CODIFICAÇÃO DA INTEGRAÇÃO

As informações evidenciadas anteriormente foram contempladas no desenvolvimento do código que faz a geração de dados do Quadro Kanban em um arquivo `.csv`, possibilitando a construção do Google Data Studio. Dessa maneira, a execução do código gera 5 relatórios que fornecem as informações do `.csv`. Tais relatórios podem ser vistos na Figura 13, sendo explicados em seguida quais dados são relacionados a cada um deles.

Figura 13: Relatórios gerados pela API do Github e Zenhub



Fonte: Elaborado pelo autor

No *Backlog Report*, existem informações e métricas relacionadas a cada issue da backlog. Essas informações geram um arquivo no formato `{datetime}_backlog_report.csv`. O Quadro 17 evidencia os dados relacionados a este relatório.

Quadro 17: Backlog Report

Coluna	Descrição
number	Número da issue no GitHub.
title	Título da issue.
state	Estado da issue(open ou closed).
url	URL de acesso da issue.
sprint	Sprint(s) em que a issue permaneceu IN PROGRESS.
points	Estimativa de pontos da issue (Story points).
todo_at	Data em que a issue foi criada ou entrou no pipeline TODO.
doing_at	Data em que a issue entrou no pipeline IN PROGRESS.

Coluna	Descrição	Continua
done_at	Data em que a issue entrou no pipeline DONE.	
lead_time	Lead time em dias da issue.	
cycle_time	Cycle time em dias da issue.	

Fonte: Elaborado pelo autor

Já o relatório de *Sprint Report* possui informações e métricas relativas ao progresso do projeto em cada sprint, fornecendo dados relacionados às atividades em progresso e já calculando automaticamente algumas métricas. O nome do arquivo gerado possui o formato {datetime}\_sprints\_report.csv (Quadro 18).

Quadro 18: Sprint Report

Coluna	Descrição
sprint	Número de identificação da sprint.
started_at	Data em que a sprint foi iniciada.
ended_at	Data em que a sprint foi concluída.
total_issues	Número de issues IN PROGRESS durante a sprint.
total_points	Quantidade de pontos das issues IN PROGRESS durante a sprint.
throughput	Número de issues concluídas durante a sprint.
velocity	Quantidade de pontos das issues concluídas durante a sprint.

Fonte: Elaborado pelo autor

O relatório de *Daily Report* tem informações relativas ao progresso do projeto desde o dia START\_DATE até o dia END\_DATE, fornecendo a quantidade de atividade e de pontos alocadas na *pipeline*. O nome do arquivo gerado possui o formato {datetime}\_daily\_report.csv (Quadro 19).

Quadro 19: Daily Report

Coluna	Descrição
day	Dia da análise.
sprint	Sprint do dia em questão.
number_todo	Número de issues que entraram no pipeline TODO no dia em questão.
number_doing	Número de issues que entraram no pipeline IN PROGRESS no dia em questão.

Coluna	Descrição	Continua
number_done	Número de issues que entraram no pipeline DONE no dia em questão.	
points_doing	Quantidade de pontos que entraram no pipeline IN PROGRESS no dia em questão.	
points_done	Quantidade de pontos que entraram no pipeline DONE no dia em questão.	
number_wtd	Número de issues que estão no pipeline TODO no dia em questão ("Work in TODO").	
number_wip	Número de issues que estão no pipeline IN PROGRESS no dia em questão ("Work in PROGRESS").	
number_wdn	Número de issues que estão no pipeline DONE no dia em questão ("Work in DONE").	
points_wip	Quantidade de pontos que estão no pipeline IN PROGRESS no dia em questão ("Work in PROGRESS").	
points_wdn	Quantidade de pontos que estão no pipeline DONE no dia em questão ("Work in DONE").	

Fonte: Elaborado pelo autor

Para fins de validação, o relatório de Skipped Report possui as issues ignoradas, ou seja, issues que não estão relacionadas ao backlog do produto. O nome do arquivo gerado possui o formato {datetime}\_skipped\_report.csv (Quadro 20).

Quadro 19: Skipped Report

Coluna	Descrição
number	Número da issue no GitHub.
title	Título da issue.
state	Estado da issue(open ou closed).
url	URL de acesso da issue.

Fonte: Elaborado pelo autor

Por fim, o relatório de Assignees Report possui informações referentes à análise individual dos desenvolvedores, como o número de issues e a quantidade de pontos total alocadas para cada um em dada sprint, além de especificar quais issues estão em seu domínio nesse período. O nome do arquivo gerado possui o formato {datetime}\_assignees\_report.csv (Quadro 20).

Quadro 21: Assignees Report

Coluna	Descrição
sprint	Sprint em questão
user	Usuário relacionado à issue
Total_issues	Número de issues associadas ao usuário na sprint em questão
total_points	Número de pontos associados ao usuário na sprint em questão
issues	Issues associadas ao usuário

Fonte: Elaborado pelo autor

O código implementado que permite, através da criação da API do Github e do Zenhub, a geração de dados do Quadro Kanban em um arquivo `.csv` a partir desses relatórios, são visualizados nos apêndices 2 a 7, onde foram divididos nos seguintes arquivos:

- `configuration.js`
- `constants.js`
- `environments.js`
- `index.js`
- `package.json`
- `package-lock.json`

A etapa de desenvolvimento de tais arquivos foi obtida através de pareamentos do orientando com alunos da Engenharia de Software da Universidade de Brasília e podem ser encontrados através do seguinte repositório do [github](https://github.com) encontrado no link: <https://bityli.com/MABrzjk>.

### 6.3 EXECUÇÃO DO CÓDIGO

Após realizar o *download* do repositório que inclui os arquivos com o código anteriormente citado, foi necessário, a partir do Terminal, navegar dentro deste repositório. Em seguida baixar o Node JS, ferramenta que interpreta o código em JavaScript.

Depois disso, iniciou-se a configuração de variáveis de ambiente. Existem 3 motivos para isso:

- Coletar os tokens que permitirá o acesso às APIs (Zenhub e Github);
- Impedir que tokens pessoais e secretos sejam expostos, evitando certas operações nos repositórios por terceiros;
- Garantir a flexibilidade de utilizar o projeto em outros cenários e repositórios, ou seja, não tem uma configuração fixa apenas no repositório da PUMA.

Dessa maneira, essa etapa se iniciou criando um novo arquivo denominado `.env` do editor de código, onde acrescentou-se nele o link do board de gestão (Quadro Kanban) a ser integrado, bem como os tokens do Github e do Zenhub, conforme evidenciado seguir e também disponível no arquivo do Apêndice 4.

```
REPOSITORY_URL={url do board de gestao}  
ZENHUB_TOKEN={token do zenhub}  
GITHUB_TOKEN={token do github}
```

Após a etapa de variáveis de ambiente, foi importante configurar as *pipelines*. Dessa forma, o arquivo `configuration.js` foi aberto e assim ajustado todos os campos referentes ao Quadro Kanban que foi conectado com a implementação. Para isso, portanto, adaptou-se o nome das colunas da pipeline referentes ao projeto, criando ou alterando outras (`zenhub_pipelines`); configurou-se as pipelines (especificando-se quais colunas concerne ao To Do [`todo_pipeline`], Doing [`doing_pipeline`] e Done [`done_pipeline`]), as labels referentes a essa nova pipeline (backlog labels), os dias que a sprint do projeto possui (`sprint duration`) e a data de início e fim do projeto (`start_date end_date`). Tais informações podem ser evidenciadas no arquivo exposto no Apêndice 2.

Após a configuração das pipelines, foi necessário instalar as dependências antes de executar o código. Para isso, abriu-se o repositório, a partir do terminal, e executa-se o comando:

```
npm install
```

Após instalar as dependências, foi possível baixar os dados através do seguinte comando no Terminal:

```
node index.js
```

Esse comando faz o download do histórico dos dados, que ficarão disponíveis em relatórios gerados no VS Code, onde é possível salvar automaticamente para a versão `.csv`.

Como última etapa, foi necessário criar uma base de dados unificada para alocar os dados disponíveis e que seja compatível com o DataStudio. Dessa maneira, a base escolhida

foi a do Google Sheets, onde a cada vez que o código é rodado, gera-se novas informações, devendo colar tais dados nessa mesma base unificada que será integrada ao Google Datastudio.

## **6.4 ANÁLISE CRÍTICA DOS DADOS COLETADOS**

A partir dos dados gerados com a API, foi montado a ferramenta de visualização das métricas de maneira a garantir o acesso e rastreabilidade mais simplificados dos dados, auxiliando na tomada de decisões do time (Google Data Studio). Tal ferramenta também pode ser acessada através de navegador celular, trazendo ainda mais facilidade para os usuários.

O [Google Data Studio](https://datastudio.google.com/reporting/10402124-0cf0-4c92-a23b-cfa72c5f441c) pode ser encontrado através do link <https://datastudio.google.com/reporting/10402124-0cf0-4c92-a23b-cfa72c5f441c>, onde estará aberto para ser duplicado e utilizado por qualquer indivíduo interessado na temática apresentada. Ademais, ele também pode ser visualizado no Apêndice 8. Devido a limitação de prazo de encerramento do semestre vigente, as análises só puderam ser realizadas até a Sprint de número 6. Contudo, o projeto teve ao todo 8 Sprints.

Nesta seção, por fim, os gráficos gerados a partir da API criada serão discutidos criticamente, organizados através das seguintes divisões:

1. Overview do Projeto
2. Throughput e Velocity
3. Lead time e Cycle Time
4. Work in Progress
5. Histograma (Pontos e Número de Issues) com filtros de Sprint e Usuário
6. Histórico de Detalhamento das Issues

### **6.4.1 Overview do Projeto**

Os primeiros dados disponibilizados no dashboard são referentes a uma visão geral do projeto PUMA referente ao semestre vigente, onde pode-se ter como análise o número de issues que estão em cada coluna da pipeline (To Do, Doing e Done); além do número de pontos que estão sendo desenvolvidos (Doing) e Entregues (Done). A Figura 14 evidencia essas informações.



Figura 14: *Overview* do Projeto (Google Data Studio)



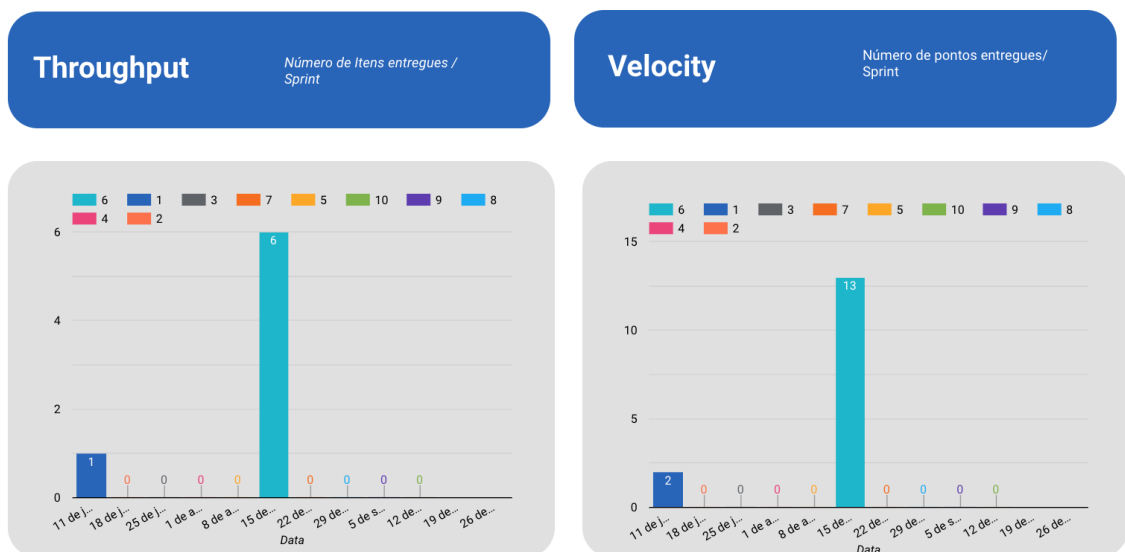
Fonte: Elaborado pelo autor

As informações deste overview são referentes a quantidade de atividades e de pontos espalhados ao longo do Quadro Kanban do projeto do semestre vigente. Sendo assim, a primeira caixa evidencia o número de itens que estão na coluna de To Do, já a segunda caixa representa o número de atividades em Doing e na terceira, o número de itens em Done. Em seguida, nas caixas azuis escuras, são evidentes o número de pontos estimados pelo time que está em desenvolvimento (Doing) e os que já foram entregues (Done).

### 6.4.2 Throughput e Velocity

As próximas análises são referentes às métricas de Throughput e de Velocity. Conforme evidenciado anteriormente, a primeira métrica refere-se à quantidade de tarefas que são entregues em dado período (Sprint). Já a segunda, refere-se à quantidade de pontos entregues nesse período. A Figura 15 evidencia tais informações.

Figura 15: Throughput e Velocity (Google Data Studio)



Fonte: Elaborado pelo autor

É notório que a partir da Figura 15 as atividades que estão sendo desenvolvidas não possuem cadência em suas entregas. Percebe-se que existem picos de entregas, um referente à primeira sprint, onde foi entregue uma atividade relacionada a adição de documentação dinâmica do projeto (esta atividade não está diretamente associada à entrega de valor para o cliente, mas sim à organização inicial do projeto); e outro pico à sexta sprint, onde foram entregues o conserto de alguns *bugs* encontrados (validação dos campos obrigatórios no cadastro, telefone de usuário no cadastro, campos em Análise da Proposta e a alteração do rótulo de botão editar disciplina); bem como melhorias na plataforma (sinalização visual dos campos obrigatórios no cadastro e alteração da ordenação da lista de palavras-chaves), conforme mostra a legenda na parte superior dos gráficos. No momento destas análises, nenhuma história de usuário havia sido entregue.

A falta de constância das entregas são ocasionadas, principalmente, pela grande curva de aprendizado que os alunos possuem no início do projeto, o que necessita um período de capacitação e de entendimento técnico durante o desenvolvimento das histórias. Dessa maneira, tais gráficos auxiliam no entendimento de como está se comportando as entregas do time, seja em termos de constância, seja em termos de quantidade.

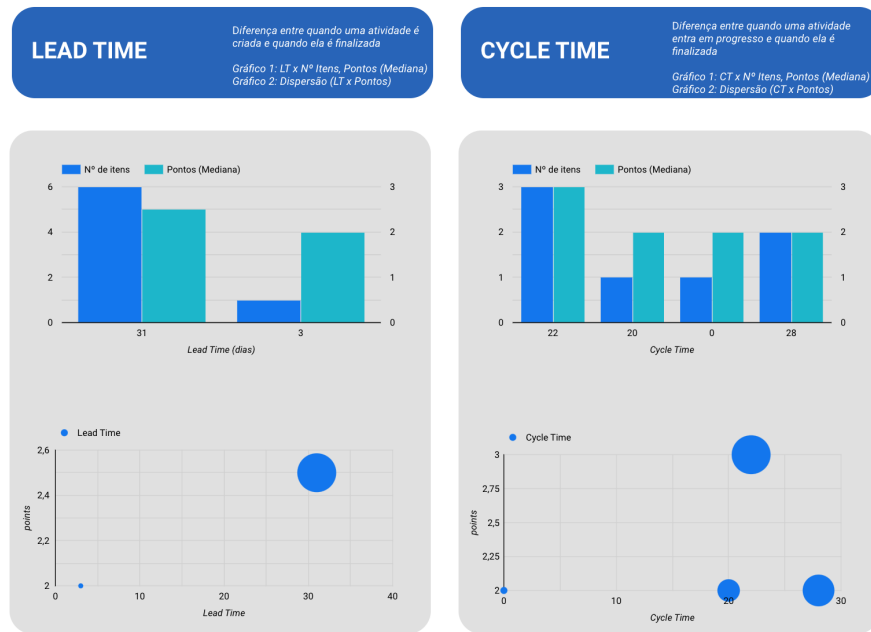
O agilismo, porém, recomenda entregas constantes e incrementais. O acesso a esses dados, portanto, possibilita ao gestor enxergar tais sinalizações para entender se isso de fato está ocorrendo, para assim, buscar entender os motivos desses comportamentos e assim gerar planos de ações.

### **6.4.3 Lead time e Cycle Time**

Conforme visto anteriormente, o Lead Time e Cycle Time estão associadas à quantidade de dias em que uma tarefa é colocada na backlog e de fato entregue para o cliente e a quantidade de dias que uma tarefa demora para ser elaborada, contando a partir do momento que ela entra em progresso, respectivamente.

A Figura 16 evidencia tais informações, colocando dois tipos de gráficos. O primeiro é um gráfico de barra que apresenta a quantidade média de dias necessários para entregar determinados pontos de esforço (mediana) ou atividades (média). Já o segundo, é um gráfico de dispersão que facilita a visualização da quantidade de dias necessários para entregar determinado número de pontos (mediana).

Figura 16: Lead Time e Cycle Time (Google Data Studio)



Fonte: Elaborado pelo autor

Em relação ao Lead Time, é possível concluir que houve um tempo médio de 31 dias para a entrega de 6 itens - tendo como medianas 2,5 pontos cada. Além disso, um item com estimativa de esforço de 2 pontos demorou 3 dias para ser entregue, desde o momento que foi colocada na backlog.

Já em relação ao Cycle Time, 3 itens que possuíam como mediana 3 pontos de esforço, demoraram 22 dias para serem concluídas, do momento que entraram em progresso; 1 item de 2 pontos demorou 20 dias para ser desenvolvida; 1 item de 2 pontos foi desenvolvida no mesmo dia em que foi para em progresso; e 2 itens de 2 pontos de esforço foram desenvolvidas em 28 dias.

Com tais dados, é possível entender que houve uma falha na atualização do Quadro Kanban, onde, segundo a análise, atividades com a mesma quantidade de pontos tiveram um grande número de diferença no tempo de sua entrega, o que traz uma dificuldade na acurácia desses dados. Caso não houvesse essa falha, seria possível concluir que o time estaria demorando muito (Cycle Time) para entregar itens com estimativa de esforço baixa, o que poderia ter como hipótese uma dificuldade de pontuação do esforço ou ociosidade do time.

O Lead Time indica também que diversas atividades tiveram muito tempo para serem realizadas, desde o momento que foram incluídas na backlog. Dessa forma, um lead time muito alto pode indicar um erro de priorização das atividades, o que acaba por deixar algumas

dessas atividades paradas, enquanto outras são priorizadas no lugar. Além da possibilidade de falhas do time (dificuldades de estimativa ou ociosidade) que pode acabar gerando frustração ao cliente, afetando sua satisfação e confiabilidade com o projeto.

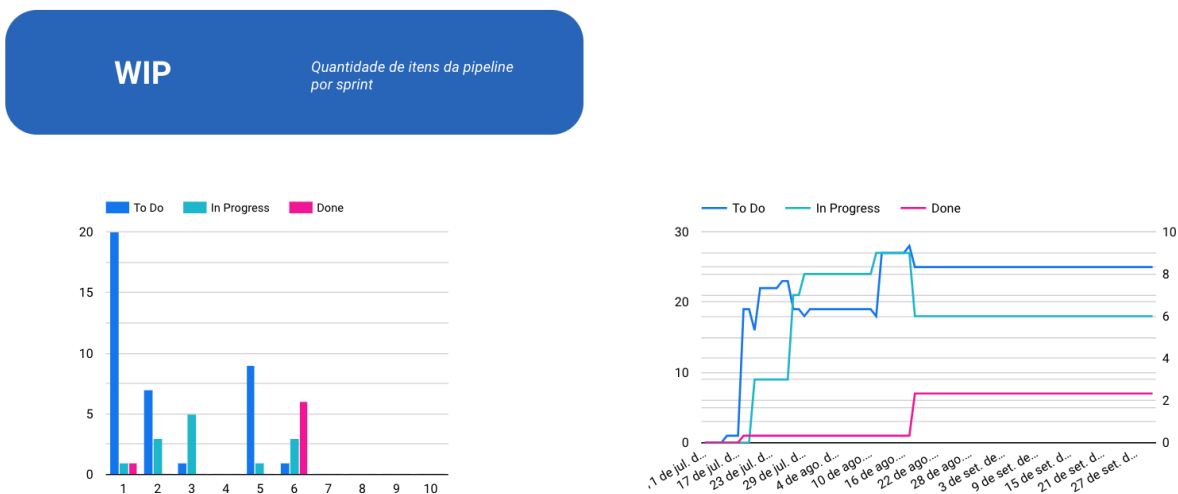
#### 6.4.4 Work in Progress

O Work In Progress é uma métrica que permite visualizar a quantidade de atividades que estão em cada coluna da pipeline do projeto (por exemplo, quantos atividades estão alocadas em To Do, Doing e Done).

O objetivo de se buscar tais dados é entender qual é o número ótimo de quantidade de atividades a ser alocada ao time em toda a pipeline. Tendo o entendimento desse número, é possível limitar o Trabalho em Progresso, de maneira a trazer mais assertividade ao planejamento e a entrega, além de garantir que o time está operando em um ritmo saudável, evitando tanto ociosidade quanto sobrecarga (VALENTE, 2022).

Neste contexto, foram desenvolvidos dois gráficos, um primeiro que traz essa evidência da quantidade de dados existentes na pipeline ao longo das sprints de 7 dias e outro que mostra essa evidência diária, conforme mostrado na Figura 17.

Figura 17: Work in Progress (Google Data Studio)



Fonte: Elaborado pelo autor

Diante de tais dados, é possível concluir que a alocação de tarefas a serem feitas (To Do) durante as sprints é maior do que de fato o time conseguiu entregar (Done). Conforme visto na Figura 17, a Sprint 1 teve um total de 20 itens a serem feitos, sendo apenas 1 entregue: adição de documentação dinâmica do projeto, conforme sinalizado na seção 6.4.2.. À medida que o projeto foi se desenvolvendo, a quantidade de atividades em To Do foi diminuindo, exceto na 5ª Sprint, onde teve novamente uma subida. Porém, apenas na 6ª Sprint

que de fato o número de atividades que foram finalizadas (Done) superou as atividades que estavam em To Do. Tais atividades foram mencionadas na seção 6.4.2 e são relacionadas ao conserto de bugs e a melhorias.

Tal cenário ocorre por uma falha de planejamento, provavelmente ocasionada pela falta de entendimento da capacidade de trabalho do time e também da alta curva de aprendizado necessária para implementação. Assim, uma das formas de se acompanhar tais dados é diminuir a quantidade de atividades que entram em To Do de maneira a evitar o descompasso entre atividades planejadas e de fato cumpridas, além de utilizar de lições aprendidas de projetos anteriores, de maneira a garantir um planejamento mais efetivo ao projeto vigente.

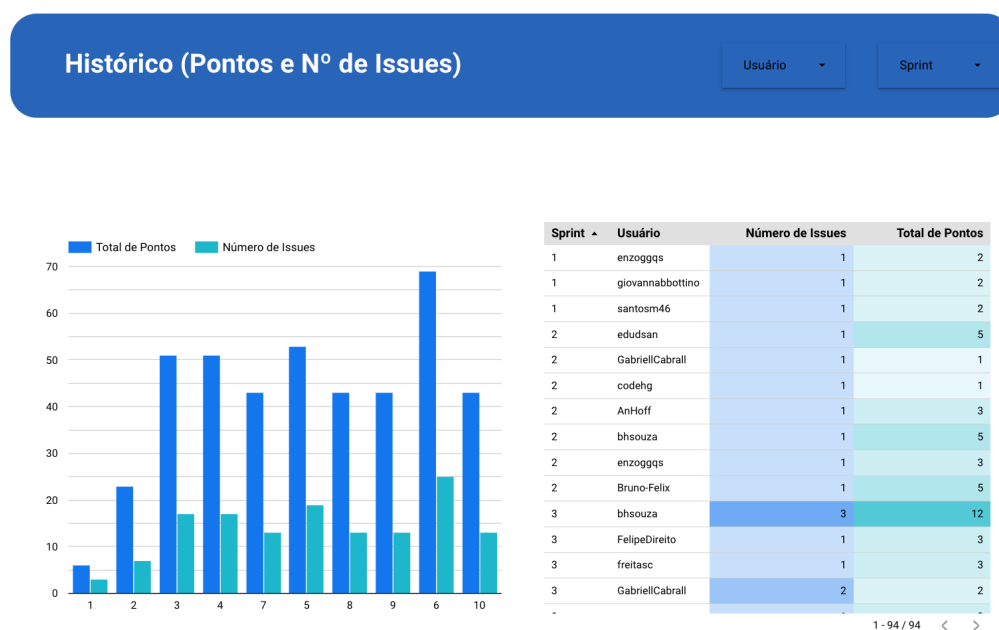
Essa falha de planejamento comprova a necessidade de se entender melhor a produtividade do time, tanto em uma visão sistêmica quanto individual. Esta última perspectiva será contemplada na análise a seguir.

#### **6.4.5 Histórico (Pontos e Número de Issues) com filtros de Sprint e Usuário**

Durante as entrevistas, foi percebido uma necessidade dos estudantes e clientes do projeto de entender melhor a performance individual dos desenvolvedores, de maneira a se construir um planejamento personalizado, garantindo assim um desenvolvimento do webapp PUMA conforme definido com os clientes.

Desse modo, foram criados gráficos que ilustram o histórico dos desenvolvedores tanto para o número de pontos que ele desenvolveu quanto para o número de issues. Conforme evidenciado no canto superior da Figura 18, também foram criados filtros de maneira a possibilitar essa perspectiva individual. Assim, o filtro "Usuário" apresenta o nome do usuário que está auxiliando no desenvolvimento do projeto (tanto estudantes de Métodos de Desenvolvimento de Software quanto os de Engenharia de Produto de Software).

Figura 18: Histórico de Pontos e de Número de Issues



Fonte: Elaborado pelo autor

Diante do exposto na Figura 18, a primeira análise é referente ao gráfico de barras, que evidencia o histórico do somatório de pontos relacionados a estimativa de esforço e do somatório do número de issues alocados para o time em determinada sprint, com possibilidade de filtrar para uma análise individual.

Já a segunda análise é referente a uma tabela onde é possível entender o quanto de pontos e de atividades, em cada sprint, cada desenvolvedor está responsável, além de fornecer um mapa de calor, acentuando as cores daqueles que estão atuando em maiores quantidades.

Tais análises são importantes para nivelar melhor os esforços do time, evitando que alguém se sobrecarregue, além de auxiliar na curva de aprendizado daqueles que possuem mais dificuldades, onde através desse histórico, pode-se montar planos personalizados para aumentar gradativamente a quantidade de pontos e tarefas alocadas.

#### 6.4.6 Histórico de Detalhamento das Issues

Por fim, a última análise da ferramenta de visualização de métricas ágeis também é referente a um desejo provindo dos estudantes e dos clientes, descoberto através das entrevistas. Assim, evidencia-se o histórico de issues que foram executadas durante o projeto, mostrando em qual sprint ela está relacionada, quantos pontos elas possuem e quem é o responsável - é normal que uma issue apresente mais de um responsável, já que algumas implementações necessitam de pareamentos por parte do time de desenvolvimento. Tal análise pode ser evidenciada na Figura 19.

Figura 19: Histórico de Detalhamento de Issues

### Histórico de detalhamento das issues

Sprint	Nome da Issue	Responsável	Pontos
1	[#62] Adiciona documentação dinâmica	enzoggqs	2
1	[#62] Adiciona documentação dinâmica	giovannabottino	2
1	[#62] Adiciona documentação dinâmica	santasm46	2
2	[Bugfix] Validação nome de Disciplina em Criar Disciplina	edudsan	5
2	[Bugfix] Alterar rotulo de botão editar disciplina	GabrielCabral	1
2	[Bugfix] Alterar rotulo de botão editar disciplina	codehg	1
2	[Bugfix] Validar telefone de usuário no cadastro	AnHoff	3
2	[Bugfix] Validação nome de Disciplina em Criar Disciplina	bhsouza	5
2	[Bugfix] Validar telefone de usuário no cadastro	enzoggqs	3
2	[Bugfix] Validação nome de Disciplina em Criar Disciplina	Bruno-Felix	5
3	[Bugfix] Corrigir estilo cabeçalho da tabela listagem projetos;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validação nome de Disciplina em Criar Disciplina	bhsouza	12
3	[Melhoria] Alterar Ordenação da lista de palavras-chaves	FelipeDireito	3
3	[Melhoria] Alterar Ordenação da lista de palavras-chaves	freitasc	3
3	[Bugfix] Validar campos em Análise da Proposta;[Bugfix] Alterar rotulo de botão editar disciplina	GabrielCabral	2
3	[Bugfix] Validar telefone de usuário no cadastro	enzoggqs	3
3	[Bugfix] Validar campos obrigatórios no cadastro	YuriBre	3
3	[Bugfix] Validação nome de Disciplina em Criar Disciplina	Bruno-Felix	5
3	[Bugfix] Validar campos em Análise da Proposta;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validar telefone de usuário no cadastro;[Bugfix] Validar campos obrigatórios no cadastro	AnHoff	9
3	[Bugfix] Corrigir estilo cabeçalho da tabela listagem projetos;[Bugfix] Validação nome de Disciplina em Criar Disciplina	edudsan	10
3	[Bugfix] Alterar rotulo de botão editar disciplina	codehg	1
4	[Bugfix] Validar telefone de usuário no cadastro	enzoggqs	3
4	[Bugfix] Corrigir estilo cabeçalho da tabela listagem projetos;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validação nome de Disciplina em Criar Disciplina	bhsouza	12
4	[Bugfix] Validar campos em Análise da Proposta;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validar telefone de usuário no cadastro;[Bugfix] Validar campos obrigatórios no cadastro	AnHoff	9
4	[Bugfix] Alterar rotulo de botão editar disciplina	codehg	1
4	[Melhoria] Alterar Ordenação da lista de palavras-chaves	freitasc	3
4	[Bugfix] Validar campos obrigatórios no cadastro	YuriBre	3
4	[Melhoria] Alterar Ordenação da lista de palavras-chaves	FelipeDireito	3
4	[Bugfix] Validação nome de Disciplina em Criar Disciplina	Bruno-Felix	5

1 - 94 / 94 < >

Fonte: Elaborado pelo autor

Apesar do Zenhub fornecer esses dados, sua visualização é dificultada principalmente para os clientes, que não possuem o acesso a tal plataforma. Assim, diante do desejo de entender em tempo real quais atividades estão sendo implementadas pelo time de desenvolvimento, foram criadas tais análises.

## 6.5 FEEDBACK DA FERRAMENTA DE VISUALIZAÇÃO DE MÉTRICAS ÁGEIS

Após a construção da dashboard juntamente com a análise crítica vista no capítulo anterior, este estudo elaborou uma validação com alguns entrevistados para com o processo de construção e análises fomentados pelo autor. Para isso, foram reunidos 3 entrevistados, sendo

2 estudantes de Engenharia de Produto de Software do semestre vigente e 1 do semestre anterior.

Foram então, entendidas, questões e feedbacks relacionadas a cada uma das etapas do processo de construção dos gráficos, desde o mapeamento de dificuldades, até a análise crítica dos dados. Além disso, também foi entendido o quanto esta proposta de solução poderia auxiliar num melhor desenvolvimento da PUMA.

Neste contexto, o primeiro ponto foi relacionado às dificuldades de maior recorrência mapeadas entre os entrevistados para dar sequência com o projeto, visto no final da seção 5.3. Segundo os entrevistados, todas as dificuldades mapeadas fizeram sentido, com destaque a Dificuldade 5: **Inconsistência nos gráficos gerados pelo Zenhub.**

Grande parte dos desenvolvedores que participaram do projeto tinha dificuldade em atualizar o board de gestão quando atuavam ou finalizavam alguma atividade, o que os levava a atualizar de última hora, apoiando na inconsistência nos dados.

Ao se buscar entender melhor os motivos para a falha de atualização, foram mapeadas algumas hipóteses entre os entrevistados: a primeira é a falta de disciplina, principalmente por se tratar de estudantes que estão tendo um primeiro contato real com desenvolvimento de software. A segunda hipótese está relacionada à falta de engajamento, ocorrida provavelmente pela desmotivação de alguns integrantes. A terceira hipótese está relacionada à falta de conhecimento da *pipeline* do Zenhub, o que faz com os estudantes não dominem para qual coluna alocar as atividades, principalmente no que diz respeito a coluna de *done* (finalizado sem revisão) e *closed* (revisado e pronto para lançamento).

Em seguida, foram coletados feedbacks relacionados a cada um dos gráficos e entendimento se a análise crítica do autor fez sentido. No que diz respeito aos gráficos de **Throughput e Velocity**, vistos na seção 6.4.2, foram visualizados picos de entrega, um relacionado a primeira sprint e outro a sexta. Esses picos de fatos aconteceram, segundo os integrantes, onde para a construção do MVP tiveram 3 grandes picos, no início, no meio e no final do semestre universitário. Esses picos provavelmente ocorreram pela dependência dos testes de funcionalidade para que as atividades fossem revisadas e de fato fechadas.

Porém, os entrevistados trouxeram que provavelmente houve alguma divergência dos gráficos com a realidade já que, mesmo tendo picos, quase todas as sprints referentes a janela de análise foram entregues atividades, e não apenas na primeira e sexta. A causa disso pode ter sido pela falta de atualização no board no tempo da análise dos dados - que seguiu apenas até a sexta sprint -, mas também outras hipóteses discutidas podem ser o mau planejamento e a demora nas entregas, vistas a seguir.



O **Lead Time** e **Cycle Time** - vistos na seção 6.4.3 - foram os gráficos de menor entendimento e acurácia na perspectiva dos entrevistados. Porém, apesar da necessidade de torná-lo mais intuitivo, foi conferido que tiveram informações coerentes, já que algumas atividades demoraram de fato muito tempo para ser executadas por haverem dependências, como por exemplo, atividades relacionadas a CI/CD (práticas combinadas de integração contínua e entrega contínua) e a DEvOps (metodologia de desenvolvimento de software que utiliza a comunicação para integrar desenvolvedores (dev) de software e profissionais de infraestrutura (ops) de TI), além de outras que tinham dependências de testes de funcionalidade.

Além disso, foi verificado pelos entrevistados que houve uma dificuldade em avaliar no início do projeto o esforço estimado para desempenhá-las através do *planning poker*, o que também poderia apoiar a baixa acurácia dos dados. A falta de atualização do board também contribuiu com esse cenário.

O **Work In Progress** - visto na seção 6.4.4 - representou um grande entendimento e acurácia para com as análises dos entrevistados, que verificaram que de fato a Sprint 4 foi uma sprint sem implementação, por questões internas, como mudança de time e de planejamento.

Outro ponto levantado para essa métrica estava relacionado com a atualização do board, onde alguns integrantes esqueceram de colocar as atividades em progresso e colocavam diretamente de *to do* para *done*, o que poderia afetar a acurácia dos dados.

O **Histórico (Pontos e Número de Issues) com filtros de Sprint e Usuário**, visto na seção 6.4.5, foi o dado que, de acordo com os entrevistados, poderia fornecer maior impacto para a pesquisa, por apresentar quais atividades cada usuário estava atuando e mostrar quantas atividades eles também estão desenvolvendo ao longo do tempo, o que pode sinalizar a necessidade de apoiá-los quando não estivesse havendo evoluções. Tais dados também poderiam facilitar a avaliação dos estudantes para com as disciplinas, já que reflete também o quanto eles estão envolvidos ao longo do semestre.

O último dado analisado foi o de Histórico de Detalhamento das Issues, visto na seção 6.4.5. Segundo os entrevistados, tais dados complementam o anterior, fornecendo os detalhes das atividades que cada estudante está atuando e seus números de pontos, o que também é importante para a avaliação nas disciplinas. Ademais, esses dados evidenciam de maneira sintetizada a gestão do conhecimento do projeto em termos técnicos.

Dessa maneira, a última questão levantada com os entrevistados foi quais pontos eles imaginavam que a dashboard poderia agregar com o desenvolvimento da PUMA no contexto

universitário. Assim, foi entendido que a dashboard, com todas as perspectivas individuais, poderia facilitar o processo de avaliação dos professores para com os alunos, trazendo uma avaliação mais justa, já que, muitas vezes, os critérios de avaliação utilizam apenas dados do Sonar e do Github, que podem acabar não mostrando com detalhes a participação dos membros, diferente da dashboard apresentada.

Ademais, as métricas de Work In Progress ganharam bastante notoriedade entre os entrevistados, pois com elas é possível entender o quanto o time de desenvolvimento está conseguindo entregar, entendendo, assim, se é válido acelerar ou não a operação nas planejings, o que evita sobrecarga do time e auxilia a equilibrar as expectativas do cliente ao trazer um desenvolvimento mais assertivo.

Os entrevistados também trouxeram que os dados da dashboard poderiam ser facilitadores nas tomadas de decisões para com os estudantes de EPS, que, a partir do gerenciamento, também poderiam agregar para os estudantes da disciplina de MDS, principalmente porque os dados coletados não são possíveis de encontrar no Github e Zenhub. Por fim, outros ganhos diretos com o projeto seriam a melhoria na previsibilidade e no planejamento das sprints. Todos esses fatores podem contribuir com os fatores de sucesso do desenvolvimento de software

Figura 19: Benefícios que a dashboard poderia agregar a PUMA



Fonte: Elaborado pelo autor

Dessa forma, com a ferramenta de visualização de métricas ágeis, criada através do Google Data Studio, pode-se auxiliar o gerenciamento do projeto de desenvolvimento de software e contribuir com seus fatores de sucesso.

A facilitação no gerenciamento, através desses dados, pode se dar ao se resolver algumas questões que times de desenvolvimento de software possuem no seu dia a dia, principalmente relacionadas a tomada de decisões a partir de dados, contribuindo com sua melhoria operacional.

No contexto da plataforma PUMA, a criação da ferramenta de visualização de métricas ágeis pode solucionar diversas dificuldades encontradas durante as entrevistas, sendo elas: uma visualização melhor e mais completa dos dados da equipe, ao facilitar a acessibilidade de leitura (é possível acessar o Google Data Studio a partir de um link, sendo responsivo para diversos equipamentos) e a flexibilidade na geração de novos gráficos, resolvendo as dificuldades 1 e 5, conforme apresentadas no Quadro 15.

Além disso, as informações contidas nesta ferramenta podem garantir mais assertividade na previsibilidade dos prazos, ao fornecer aos gestores o histórico da produtividade do time, possibilitando planejamentos empiricamente mais condizentes. Elas também fornecem uma análise individual dos desenvolvedores, podendo garantir planejamentos personalizados que seja coerente com a realidade de cada um e também um acompanhamento mais próximo das suas curvas de aprendizados, o que pode contribuir com o aumento da produtividade individual e coletiva do time. Tais contribuições auxiliam na resolução das dificuldades 3, 6 e 4 do Quadro 15, respectivamente.

Por fim, a ferramenta de visualização de métricas ágeis é um possível local para centralização dos principais artefatos do projeto, onde é documentado além do histórico operacional do time, as atividades desenvolvidas durante o projeto e alocações referentes a elas, possibilitando a unificação e acessibilidade de informações essenciais para a tomada de decisão. Tais contribuições podem solucionar as dificuldades 2 e 7 do Quadro 15.

## 7. CONCLUSÃO

Esta pesquisa contribui para a visibilidade de uma temática disseminada na literatura, mas pouco aplicada na prática: as métricas ágeis. Esse ponto foi percebido logo de início do projeto, no levantamento do referencial teórico, onde pouco se tinha informações sobre a aplicação dessas métricas no contexto de software, embora fossem amplamente citadas. Além disso, no estudo de caso também, foi evidenciado que a maioria das métricas de conhecimento dos estudantes eram aquelas já pré-fixadas nas plataformas de gestão de projetos do zenhub (velocity e burndown).

Dessa maneira, com o entendimento da equipe de desenvolvimento da PUMA bem como de seus clientes, foi possível notar que algumas dificuldades estavam relacionadas a forma como esses dados eram evidenciados, não existindo uma ferramenta acessível e que pudesse flexibilizar a geração de novas análises pelas plataformas de código aberto Github e a sua então integração Zenhub; a não existência de dados para tomada de decisão; e a descentralização de artefatos relacionados ao projeto.

Assim, os planos de ações propostos, além de centralizar tais dados em um só lugar, facilitando suas análises pelos stakeholders, também abriu a possibilidade que os gestores do projeto consigam construir novas métricas e informações para tomada de decisão, por meio dos relatórios gerados através da codificação vista nesse estudo e da dashboard evidenciada no Google DataStudio, além de acessibilizar as visualizações destes dados. Tais planos, segundo os próprios entrevistados na validação da solução podem contribuir com um mecanismo de avaliação mais eficiente para os estudantes das disciplinas de EPS e MDS; diminuir a sobrecarga de estudantes, ao se ter históricos de atuação de cada qual; trazer um melhor alinhamento com o cliente, ao se garantir um melhor planejamento e previsibilidade das atividades.

Essa implementação e a então evidência de métricas do time contribuem com maior assertividade na tomada de decisões ao longo do projeto, porém, elas não devem ser generalizadas, pois a própria escolha de métricas, conforme visto, deve ser feita conforme a necessidade e objetivo do gestor, sendo um passo adotado subjetivamente. Porém, esse projeto traz uma primeira facilitação para com as métricas escolhidas, já que ele não conseguiu ter seu acompanhamento aplicado em tempo real com o projeto por limitações relacionadas ao tempo de entrega.

A partir deste estudo de caso também se notou que as métricas ágeis, embora tragam um reflexo da operação do time desenvolvimento, auxilia apenas como um suporte na tomada

de decisões. Porém, elas não trazem as reais causas para comportamentos inesperados, devendo portanto, ser acompanhada de outros dados, principalmente associados aos 4 fatores vistos nesta pesquisa - Escopo, Tempo, Satisfação e Custo. Os dados fomentados com essa pesquisa, contudo, se concentram apenas nos fatores de escopo e tempo, dada as limitações de ser um projeto universitário.

Além disso, é importante ressaltar que o seu ganho é voltado especificamente para o ambiente acadêmico, onde os estudantes ainda apresentam diversas dificuldades relacionadas ao contexto ágil, além de ainda apresentarem uma crescente curva de aprendizado em termos de desenvolvimento.

Porém, acima de todas essas dimensões de sucesso, é notório também tanto pelo referencial teórico quanto pelo próprio estudo de caso, que o fator sociotécnico é o que de fato vai impulsionar o sucesso do projeto (RODRIGUES et al, 2019). Assim, é evidenciado que acompanhar o indivíduo e sua satisfação com o projeto, a sua curva de aprendizado e o seu nível de estresse, podem ser boas estratégias para evitar qualquer gargalo relacionada a esse fator.

Nota-se, por conseguinte, que este estudo de caso conseguiu atingir todos os objetivos propostos, consolidando a ferramenta de visualização de métricas ágeis que possa ser utilizada no gerenciamento de projetos de desenvolvimento de software, a fim de contribuir com fatores de sucesso do módulo da Plataforma Unificada de Metodologias Ativas (PUMA). Isso foi possível através do entendimento das metodologias ágeis de desenvolvimento de software e das métricas associadas a elas; da seleção das métricas ágeis de maior relevância na literatura; do levantamento das principais dificuldades de aplicação de métricas ágeis no contexto da PUMA; e do entendimento e implementação de métricas ágeis para melhoria da tomada de decisão da PUMA, através da ferramenta e das análises criadas.

A pesquisa, porém, foi limitada apenas à implementação dessas métricas, utilizando de dados reais do projeto do semestre vigente, mas não as tendo aplicadas conjuntamente com o time de desenvolvimento e de gestão da PUMA para de fato comprovar as suas reais contribuições com os fatores de sucesso do projeto. Dessa maneira, é sugerido para os próximos estudos, a aplicação prática dessas métricas por tais times, acompanhando e entendendo seus comportamentos nas tomadas de decisão realizadas a partir das informações disponibilizadas na ferramenta criada.

## REFERÊNCIAS BIBLIOGRÁFICAS

- AGRESTI, W. W. *New Paradigms for Software Development: Tutorial*. [s.l.] **IEEE Computer Society Press**, 1986.
- AL Aidaros, H.; Omar, M.; Romli, R. **Towards an improved software project monitoring task model of agile kanban method**. Disponível em: <<https://core.ac.uk/download/pdf/230745993.pdf>>.
- ANDERSON, D. J. **Kanban: Mudança Evolucionária De Sucesso Para Seu Negócio De Tecnologia**. Tradução: Andrea Pinto. [s.l.] **Blue Hole Press**, 2011.
- ANDERSON, D. J.; CARMICHAEL, A. *Essential Kanban Condensed*. [s.l.] **Lean-Kanban University**, 2016.
- ABRANTES, J. *Gestão da Qualidade*. Rio de Janeiro: **Interciência**, 2009.
- BASILI, V. R.; TURNER, A. J. Iterative enhancement: A practical technique for software development. In: *Foundations of Empirical Software Engineering*. [s.l.] **Springer Berlin Heidelberg**, 2005. p. 28–39.
- BECK, K.; ANDRES, C. *Extreme Programming Explained: Embrace Change*. [s.l.] **AddisonWesley Professional**, 2004.
- BOEHM, B. W. A spiral model of software development and enhancement. **Computer**, v. 21, n. 5, p. 61–72, 1988.
- CAROLI, P. **Lean Inception: como alinhar pessoas e construir o produto certo**. 1a Edição atualizada - São Paulo: Editora Caroli, 2018.
- CHOW, T; CAO, D-B; A survey study of critical success factors in agile software projects. **Journal of Systems and Software**, 81(6), 961-971, 2008
- COSTA, N., et al. “Propriedades Das Séries Temporais de Lucros Trimestrais Das Empresas Brasileiras.” **Revista de Contabilidade E Organizações**, vol. 2, no. 2, 1 Apr. 2008, 10.11606/rco.v2i2.34703. Accessed 26 Apr. 2022.
- CROLL, A. *Lean Analytics: Use data to build a better Startup faster*. **Sebastopol, CA, USA: O’Reilly Media**, 2013.
- DENZIN, Norman K. ; LINCOLN, Yvonna S. (Editores). *Handbook of qualitative research*. (2 Ed.). **Thousand Oaks, Califórnia: Sage Publications**. 2000.
- DINGSOEYR, T.; FALESSI, D.; POWER, K. Agile development at scale: The next frontier. **IEEE software**, v. 36, n. 2, p. 30–38, 2019.
- DRURY, M., et al. “Obstacles to Decision Making in Agile Software Development Teams.” **Journal of Systems and Software**, vol. 85, no. 6, June 2012, pp. 1239–1254, 10.1016/j.jss.2012.01.058.

GAETE, J. et al. **Agile application approach with Scrum, Lean and Kanban**. Disponível em: <<https://scielo.conicyt.cl/pdf/ingeniare/v29n1/0718-3305-ingeniare-29-01-141.pdf>>.

GRENNING, J., 2002. **Planning poker or how to avoid analysis paralysis while release planning**. Disponível em: <<http://www.renaissancesoftware.net/files/articles/PlanningPokerv1.1.pdf>>

GROVE, A. S. *High Output Management*. 1. ed. Nova Iorque, NY, USA: Random House, 1988.

GUEDES, V.L., & Borschiver, S. (2005). Bibliometria: uma ferramenta estatística para a gestão da informação e do conhecimento, em sistemas de informação, de comunicação e de avaliação científica e tecnológica. **Encontro Nacional de Ciência da Informação**, 6,1-18.

HARTMAN, D.; DYAMOND, R.; Appropriate agile measurements: Using metrics and diagnostics to deliver business value. **In Agile 2006 Conference**, pages 126–131, 2006.

HEIDENBERG, J. et al. A metrics model to measure the impact of an agile transformation in large software development organizations. In: *Lecture Notes in Business Information Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 165–179.

HIROTAKA, T.; NONAKA, I. The new product development game. **The Journal of product innovation management**, v. 3, n. 3, p. 205–206, 1986.

HOFMANN, C. et al. Development of an agile development method based on Kanban for distributed part-time teams and an introduction framework. *Procedia Manufacturing*, v. 23, p. 45–50, 2018.

HUMBLE, J.; FARLEY, D. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. [s.l.] Pearson Education, 2010.

IEEE standard glossary of software engineering terminology, 1983. IEEE Std 729.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. E. *The Unified Software Development Process—The Complete Guide to the Unified Process from the Original Designers*. [s.l.] Addison-Wesley, 1999.

KUPIAINEN, E.; MÄNTYLÄ, M. V.; ITKONEN, J. Using metrics in Agile and Lean Software Development – A systematic literature review of industrial studies. *Information and software technology*, v. 62, p. 143–163, 2015.

LEI, H. et al. A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and computer-integrated manufacturing*, v. 43, p. 59–67, 2017.

LIZARDO, C.; RIBEIRO, P. (2020). A importância da gestão da qualidade e aplicação das suas ferramentas na logística com vista à satisfação dos clientes. *Gestão E Desenvolvimento*, (28), 3-28.

LIMA, A. C. F. Ferramenta de gestão de riscos aplicada a ambientes de desenvolvimento de software com foco na garantia da qualidade do produto. 2019. XVIII, 185 f., il. Dissertação (Mestrado Profissional em Computação Aplicada) — Universidade de Brasília, Brasília, 2019.

LOPEZ-MARTINEZ, J, et al. “Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review.” **2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)**, Apr. 2016, 10.1109/conisoft.2016.30

MARIANO, A.M; ROCHA, M.S. Revisão da Literatura: Apresentação de uma Abordagem Integradora. **AEDM International Conference – Economy, Business and Uncertainty: Ideas for a European and Mediterranean industrial policy**. Reggio Calabria (Italia), 2017.

MARINO, L. Gestão da qualidade e gestão do conhecimento: fatores-chave para produtividade e competitividade empresarial. **8º SIMPEP**, Bauru, São Paulo (Brasil), 2006

MARTINEZ-FERNANDEZ, S. et al. A quality model for actionable analytics in rapid software development. **2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. Anais...IEEE, 2018

MASON-JONES, R.; NAYLOR, B.; TOWILL, D. R. Engineering the leagile supply chain. *International journal of agile management systems*, v. 2, n. 1, p. 54–61, 2000.

MCGARRY et al. *Practical Software Measurement: Objective Information for Decision Makers*. Addison-Wesley Professional, 2002.

MISRA, S. C.; KUMAR, V.; KUMAR, U., Identifying some important success factors in adopting agile software development practices. **Journal of Systems and Software**, 82(11), 1869-1890, 2009.

MONTEIRO, S, et al.; Evaluating direct and indirect results of the active methodology in learning: proposal of an integrative design in 360° via unified platform. **In International Symposium on Project Approaches in Engineering Education (PAEE)** (Vol. 8, pp. 768-775), 2018.

MULLER, A.; VÄLIKANGAS, L.; MERLYN, P. Metrics for innovation: guidelines for developing a customized suite of innovation metrics. *Strategy and leadership*, v. 33, n. 1, p. 37–45, 2005.

PERIANES, Rodriguez, A., Waltman, L., & van Eck, N.J. (2016). Constructing bibliometric networks: A comparison between full and fractional counting. **Journal of Informetrics**, 10 (4), 1178-1195.

POPPENDIECK, M.; POPPENDIECK, T. *Lean Software Development: An Agile Toolkit*. [s.l.] Addison-Wesley Professional, 2003.

POPPENDIECK, M.; POPPENDIECK, T. Introduction to lean software development. In: *Extreme Programming and Agile Processes in Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 280–280.



RAM, P. et al. Success factors for effective process metrics operationalization in agile software development: A multiple case study. **2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)**. Anais...IEEE, 2019.

RODRÍGUEZ, P. et al. Advances in using agile and lean processes for software development. In: *Advances in Computers*. [s.l.] Elsevier, 2019. p. 135–224.

ROYCE, W. W. Managing the development of large software systems (1970). In: *Ideas That Created the Future*. [s.l.] **The MIT Press**, 2021. p. 321–332.

RUY, M. Aprendizagem organizacional no processo de desenvolvimento de produtos: estudo exploratório em três empresas manufatureiras. Dissertação de Mestrado, EESC - São Carlos, 2002.

SALEH, S. M.; HUQ, S. M.; RAHMAN, M. A. Comparative study within scrum, kanban, XP focused on their practices. **2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)**. Anais...IEEE, 2019.

SATO, D. Uso Eficaz de Métricas em Métodos Ágeis de Desenvolvimento de Software. Dissertação de Mestrado, Universidade de São Paulo, 2007.

SILVA, J. BALTHAZAR, J. Projeto político pedagógico do curso de engenharia de produção. Universidade de Brasília, 2010.

SCOTT, E.; PFAHL, D. Using developers' features to estimate story points. **Proceedings of the 2018 International Conference on Software and System Process - ICSSP '18**. Anais...New York, New York, USA: ACM Press, 2018.

SCRUM.ORG, Scrum.org, 2020. About: What's Scrum? Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>

SELLTIZ, C. E. Métodos de Pesquisa nas Relações Sociais. Tradução de Dant Moreira Leite, v. 3ed, [s.d.].

SHANBHAG, N.; PARDEDE, E. A metrics framework for product development in software startups. **Journal of enterprising culture**, v. 27, n. 03, p. 283–307, 2019.

SLACK, N.; BRANDON-JONES, A. Slack: Operations and process management: Slack:OPM 6th ed. 6. ed. Londres, England: Pearson Education, 2021.

SILVA & ZAWISLAK, O Processo de Desenvolvimento de Produtos: um Estudo de Casos de Três Empresas Fornecedoras da Cadeia Automotiva do Rio Grande do Sul. RAC-Eletrônica, v. 1, n. 2, art. 4, p. 51-65, Maio/Ago. 2007, Disponível em: <[https://www.researchgate.net/publication/26503676\\_O\\_Processo\\_de\\_Developimento\\_de\\_Produtos\\_um\\_Estudo\\_de\\_Casos\\_de\\_Tres\\_Empresas\\_Fornecedoras\\_da\\_Cadeia\\_Automotiva\\_do\\_Rio\\_Grande\\_do\\_Sul](https://www.researchgate.net/publication/26503676_O_Processo_de_Developimento_de_Produtos_um_Estudo_de_Casos_de_Tres_Empresas_Fornecedoras_da_Cadeia_Automotiva_do_Rio_Grande_do_Sul)>. .

VALENTE, M. Engenharia de Software Moderna. 1ª edição. Brasil: Editora Independente, 1º de janeiro de 2022. Disponível em: <<https://engsoftmoderna.info/cap2.html>>

VOGEL, R.; GUTTEL, W. H. The dynamic capability view in strategic management: A bibliometric review. **International Journal of Management Reviews**, 15(4), 426-446, 2013

## APÊNDICES

### APÊNDICE A - QUESTÕES CONTIDAS NO ROTEIRO DAS ENTREVISTAS

Questões respondidas pelos entrevistados do Grupo 1 e do Grupo 2	
Q1	Quais as maiores dificuldades na implementação de métodos ágeis?
Q2	Como é a tomada de decisão no projeto?
Q3	Utilizam de dados para tomada de decisão?
Q4	Se sim, qual ferramenta utilizar para visualizar tais dados e existe alguma dificuldade nessa visualização?
Q5	Quais métricas ágeis conhece?
Q6	Quais maiores dificuldades relacionadas a coleta, leitura e plano de ação dessas métricas?
Q7	Quais maiores dificuldades na daily?
Q8	Na planning são utilizados dados?
Q9	Na planning são estimados pontos e quais as maiores dificuldades encontradas nisso?
Q10	Como são feitas as reviews e quais as maiores dificuldades encontradas?
Q11	Quais as maiores dificuldades relacionadas à gestão do projeto?
Q12	Como imagina que poderia ser melhorada a velocidade de entrega?

Questões respondidas pelos entrevistados do Grupo 3	
Q1	O que você espera receber nas reuniões de interação com o time de desenvolvimento?
Q2	Você enxerga como o time está operando? O quão importante é entender isso?
Q3	Quais dados você gostaria de enxergar constantemente?
Q4	Como seria um formato interessante para você ter facilidade em visualizar esses dados?
Q5	O que poderia ser feito para aprimorar o acompanhamento das entregas?
Q6	Como a comunicação com o time de desenvolvimento poderia ser facilitada?
Q7	Você apoia o time na priorização do que deve ser feito por sprint? como?
Q8	Como você imagina que poderia melhorar a velocidade de entrega?
Q9	Quais as maiores falhas de gestão que vocês identificam no projeto?
Q10	Você sente que suas expectativas estão sendo atendidas apropriadamente? Por que?

## APÊNDICE B - ARQUIVO CONFIGURATION.JS

```
const CONFIGURATIONS = {
  /* Zenhub Default */
  /* ZENHUB_PIPELINES: ['New Issues', 'Epics', 'Icebox', 'Product
Backlog', 'Sprint Backlog', 'In Progress', 'Review/QA', 'Done', 'Closed'],
  TODO_PIPELINE: ['New Issues', 'Product Backlog'],
  DOING_PIPELINE: ['Sprint Backlog', 'In Progress', 'Review/QA'],
  DONE_PIPELINE: ['Done', 'Closed'], */

  ZENHUB_PIPELINES: ['New Issues', 'Epics', 'Bugfix', 'Product Backlog',
'Sprint Backlog', 'In Progress', 'Review', 'Melhorias', 'Closed'],
  TODO_PIPELINE: ['New Issues', 'Product Backlog'],
  DOING_PIPELINE: ['Sprint Backlog', 'In Progress', 'Review'],
  DONE_PIPELINE: ['Closed'],
  BACKLOG_LABELS: ['bugfix', 'US', 'enhancement'],
  SPRINT_DURATION: 7,
  START_DATE: '07/01/2022',
  END_DATE: '09/30/2022',
};

CONFIGURATIONS.END_DATE = CONFIGURATIONS.END_DATE || new
Date().toDateString();

export default CONFIGURATIONS;
```

## APÊNDICE C - ARQUIVO CONSTANT.JS

```
const CONSTANTS = {
  WEEKDAYS: ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday',
'Friday', 'Saturday'],
  DAY_MS: 86400000,
};

export default CONSTANTS;
```

## APÊNDICE D - ARQUIVO ENVIRONMENTS.JS

```
import dotenv from 'dotenv';
dotenv.config();

const ENVIRONMENTS = {
  REPOSITORY_URL: process.env.REPOSITORY_URL,
  ZENHUB_TOKEN: process.env.ZENHUB_TOKEN,
  GITHUB_TOKEN: process.env.GITHUB_TOKEN,
};

export default ENVIRONMENTS;
```

## APÊNDICE E - ARQUIVO INDEX.JS

```
import chalk from 'chalk';
import axios from "axios";
import fs from 'fs/promises';
import papaparse from 'papaparse';
import CONSTANTS from './constants.js';
import ENVIRONMENTS from './environments.js';
import CONFIGURATIONS from './configurations.js';

/**
 * Backlog Issue Object
 * { number, url, title, state, labels, assignees, created_at, closed_at,
todo_at, doing_at, done_at, events, points, sprints }
 *
 * Skipped Issue Object
 * { number, url, title, state, labels, created_at, closed_at }
 *
 * Sprint Object
 * { number, started_at, ended_at }
 */

String.prototype.replaceAll = function (search, replacement) {
  var target = this;
  return target.replace(new RegExp(search, 'g'), replacement);
};

function sleep(ms) {
```

```

        return new Promise(resolve => setTimeout(resolve, ms));
    }

    async function extractDataFromRepository() {
        try {
            const [owner, repo] =
ENVIRONMENTS.REPOSITORY_URL.split('/').slice(3);
            const GH_AUTH = { 'Authorization': `token
${ENVIRONMENTS.GITHUB_TOKEN}` };
            const ZH_AUTH = { 'X-Authentication-Token':
`${ENVIRONMENTS.ZENHUB_TOKEN}` };

            /* Retrieve repository data */
            process.stdout.write(chalk.white('Recuperando dados do
repositório...'));
            const resRepository = await
axios.get(`https://api.github.com/repos/${owner}/${repo}`, { headers: {
...GH_AUTH } });
            const repositoryId = resRepository.data.id;
            process.stdout.write(chalk.green(`OK\n`));

            /* Retrieve all issues of repository */
            process.stdout.write(chalk.white('Recuperando as issues do
repositório...'));
            let page = 1;
            let recoveredAll = false;
            const issues = [];
            while (!recoveredAll) {
                const params = `state=all&per_page=100&page=${page}`;
                const url =
`https://api.github.com/repos/${owner}/${repo}/issues?${params}`;
                const res = await axios.get(url, { headers: { ...GH_AUTH } });
                const serverTime = res.headers['date'];
                const resetTime = res.headers['x-ratelimit-reset'];
                const remaining = res.headers['x-ratelimit-remaining'];
                const usecReset = (new Date(resetTime * 1000) - new
Date(serverTime)) + 1000;

                if (res.data.length === 0) { recoveredAll = true; continue; }

                const pageIssues = res.data.map((issue) => ({
                    number: issue.number,
                    title: issue.title,
                    state: issue.state,
                    url: issue.html_url,
                    labels: issue.labels,

```

```

        assignees: issue.assignees,
        created_at: new Date(issue.created_at),
        closed_at: issue.closed_at && new Date(issue.closed_at),
    }));
    issues.push(...pageIssues);

    if (remaining <= 1) {
        process.stdout.write(chalk.yellow(`\nLimite de requisições
atingido. Aguarde ${usecReset / 1000} segundos...`));
        await sleep(usecReset);
        process.stdout.write(chalk.yellow(`Voltando...`));
    }

    page++;
}
process.stdout.write(chalk.green(`OK\n`));

/* Separate issue types */
process.stdout.write(chalk.white('Separando as issues de backlog e
skipped...'));
const backlogIssues = issues.filter((issue) =>
    issue.labels.reduce((pre, curr) => pre ||
CONFIGURATIONS.BACKLOG_LABELS.includes(curr.name), false));
const skippedIssues = issues.filter((issue) =>
    !issue.labels.reduce((pre, curr) => pre ||
CONFIGURATIONS.BACKLOG_LABELS.includes(curr.name), false));
process.stdout.write(chalk.green(`OK\n`));

/* Retrieve events from issues */
process.stdout.write(chalk.white('Recuperando eventos das
issues...'));
for (const issue of backlogIssues) {
    const url =
`https://api.zenhub.com/p1/repositories/${repositoryId}/issues/${issue.number}/e
vents`;
    const res = await axios.get(url, { headers: { ...ZH_AUTH } });
    const serverTime = res.headers['date'];
    const resetTime = res.headers['x-ratelimit-reset'];
    const remaining = res.headers['x-ratelimit-limit'] -
res.headers['x-ratelimit-used'];
    const usecReset = (new Date(resetTime * 1000) - new
Date(serverTime)) + 1000;

    if (res.data && res.data.length > 0) {
        issue['events'] = res.data;
    } else {

```

```

        issue['events'] = [];
    }

    if (remaining <= 1) {
        process.stdout.write(chalk.yellow(`\nLimite de requisições
atingido. Aguarde ${usecReset / 1000} segundos...`));
        await sleep(usecReset);
        process.stdout.write(chalk.yellow(`Voltando...`));
    }
}

process.stdout.write(chalk.green(`OK\n`));

/* Handle issue events */
process.stdout.write(chalk.white('Tratando eventos das
issues...'));

for (const issue of backlogIssues) {
    // finding todo at
    let todo_at = null;
    for (const event of issue.events) {
        if (event.type === 'transferIssue') {
            if
(CONFIGURATIONS.TODO_PIPELINE.includes(event.to_pipeline.name)) {
                todo_at = new Date(event.created_at);
            }
        }
    }
    issue['todo_at'] = todo_at || issue.created_at;

    // finding doing at
    let doing_at = null;
    for (const event of issue.events) {
        if (event.type === 'transferIssue') {
            if
(CONFIGURATIONS.DOING_PIPELINE.includes(event.to_pipeline.name)) {
                doing_at = new Date(event.created_at);
            }
        }
    }
    issue['doing_at'] = doing_at;

    // finding done at
    let done_at = null;
    for (const event of issue.events) {
        if (event.type === 'transferIssue') {
            if
(CONFIGURATIONS.DONE_PIPELINE.includes(event.to_pipeline.name)) {

```



```

        done_at = new Date(event.created_at);
    }
}
}
issue['done_at'] = done_at || issue.closed_at;

// fix doing at if necessary
issue['doing_at'] = !issue['doing_at'] && issue['done_at'] ?
issue['done_at'] : issue['doing_at'];

// finding point estimate
let points = null;
for (const event of issue.events) {
    if (event.type === 'estimateIssue' &&
event.to_estimate?.value) {
        points = event.to_estimate.value;
        break;
    }
}
issue['points'] = points;
}
process.stdout.write(chalk.green(`OK\n`));

/* Make completed sprints */
process.stdout.write(chalk.white('Gerando sprints...'));
let sprintNumber = 1;
const sprints = [];
let currentDate = new Date(CONFIGURATIONS.START_DATE);
const endDate = new Date(CONFIGURATIONS.END_DATE);
while (currentDate <= endDate) {
    const sprintStart = currentDate;
    const sprintEnd = new Date(sprintStart.getTime() +
CONSTANTS.DAY_MS * CONFIGURATIONS.SPRINT_DURATION);
    sprints.push({ number: sprintNumber++, started_at: sprintStart,
ended_at: sprintEnd });
    currentDate = sprintEnd;
}
process.stdout.write(chalk.green(`OK\n`));

/* Find issue sprints */
process.stdout.write(chalk.white('Definindo sprints das
issues...'));
backlogIssues.forEach((issue) => {
    let issueSprints = [];
    if (issue.doing_at) {
        const doing_at = new Date(issue.doing_at);

```

```

        const done_at = issue.done_at && new Date(issue.done_at);
        issueSprints = sprints.filter((s) => {
            const hasIntersec = done_at && !(new Date(s.started_at)
> done_at || doing_at > new Date(s.ended_at));
            const isPending = !done_at && !(doing_at > new
Date(s.ended_at));
            return (hasIntersec || isPending);
        }).map(({ number }) => number).sort();
    }
    issue['sprints'] = issueSprints;
});
process.stdout.write(chalk.green(`OK\n`));

return { backlogIssues, skippedIssues, sprints };

} catch (error) {
    process.stdout.write(chalk.red(`${JSON.stringify({ error })}\n`));
    return null;
}
}

async function makeBacklogReport(issues, datetime) {
    const report = issues.map((issue) => {
        let lead_time = null;
        let cycle_time = null;
        if (issue.todo_at && issue.done_at) {
            lead_time = Math.floor(Math.abs(new Date(issue.done_at) - new
Date(issue.todo_at)) / CONSTANTS.DAY_MS);
        }
        if (issue.doing_at && issue.done_at) {
            cycle_time = Math.floor(Math.abs(new Date(issue.done_at) - new
Date(issue.doing_at)) / CONSTANTS.DAY_MS);
        }
        return {
            number: issue.number,
            title: issue.title,
            state: issue.state,
            url: issue.url,
            sprint: issue.sprints.join('-'),
            points: issue.points,
            todo_at: issue.todo_at && new
Date(issue.todo_at).toLocaleString(),
            doing_at: issue.doing_at && new
Date(issue.doing_at).toLocaleString(),
            done_at: issue.done_at && new
Date(issue.done_at).toLocaleString(),

```

```

        lead_time,
        cycle_time
    };
});

const csv = papaparse.unparse(report, { delimiter: ',' });
await fs.writeFile(`reports/${datetime}_backlog_report.csv`, csv);
}

async function makeSkippedReport(issues, datetime) {
    const report = issues.map((issue) => {
        return {
            number: issue.number,
            title: issue.title,
            state: issue.state,
            url: issue.url,
        };
    });
    const csv = papaparse.unparse(report, { delimiter: ',' });
    await fs.writeFile(`reports/${datetime}_skipped_report.csv`, csv);
}

async function makeSprintsReport(issues, sprints, datetime) {
    const report = sprints.map((sprint) => {
        const sprintIssues = issues.filter((i) =>
i.sprints.includes(sprint.number));
        const total_issues = sprintIssues.length;
        const total_points = sprintIssues.reduce((acc, i) => (acc +
i.points), 0);
        const throughput = sprintIssues.filter((i) => i.done_at &&
(sprint.started_at <= new Date(i.done_at) && new Date(i.done_at) <=
sprint.ended_at)).length;
        const velocity = sprintIssues.filter((i) => i.done_at &&
(sprint.started_at <= new Date(i.done_at) && new Date(i.done_at) <=
sprint.ended_at))
            .reduce((acc, i) => (acc + i.points), 0);
        return {
            number: sprint.number,
            started_at: new Date(sprint.started_at).toLocaleDateString(),
            ended_at: new Date(sprint.ended_at).toLocaleDateString(),
            total_issues,
            total_points,
            throughput,
            velocity,
        };
    });
    const csv = papaparse.unparse(report, { delimiter: ',' });

```

```

    await fs.writeFile(`reports/${datetime}_sprints_report.csv`, csv);
  }

  async function makeDailyReport(issues, sprints, datetime) {
    const report = [];
    let currentDate = new Date(CONFIGURATIONS.START_DATE);
    const endDate = new Date(CONFIGURATIONS.END_DATE);
    while (currentDate <= endDate) {
      const day = new Date(currentDate).toLocaleDateString();
      const sprint = sprints.find((s) => new Date(s.started_at) <=
currentDate && currentDate <= new Date(s.ended_at))??.number;
      const number_todo = issues.filter((i) => new
Date(i.todo_at).toDateString() === currentDate.toDateString()).length;
      const number_doing = issues.filter((i) => new
Date(i.doing_at).toDateString() === currentDate.toDateString()).length;
      const number_done = issues.filter((i) => new
Date(i.done_at).toDateString() === currentDate.toDateString()).length;
      const points_doing = issues.filter((i) => new
Date(i.doing_at).toDateString() === currentDate.toDateString())
      .reduce((acc, i) => (acc + i.points), 0);
      const points_done = issues.filter((i) => new
Date(i.done_at).toDateString() === currentDate.toDateString())
      .reduce((acc, i) => (acc + i.points), 0);
      const number_wtd = issues.filter((i) => (i.todo_at && new
Date(i.todo_at) <= currentDate) && (!i.doing_at || currentDate <= new
Date(i.doing_at))).length;
      const number_wip = issues.filter((i) => (i.doing_at && new
Date(i.doing_at) <= currentDate) && (!i.done_at || currentDate <= new
Date(i.done_at))).length;
      const number_wdn = issues.filter((i) => (i.done_at && new
Date(i.done_at) <= currentDate)).length;
      const points_wip = issues.filter((i) => (i.doing_at && new
Date(i.doing_at) <= currentDate) && (!i.done_at || currentDate <= new
Date(i.done_at)))
      .reduce((acc, i) => (acc + i.points), 0);
      const points_wdn = issues.filter((i) => (i.done_at && new
Date(i.done_at) <= currentDate))
      .reduce((acc, i) => (acc + i.points), 0);

      report.push({ day, sprint, number_todo, number_doing, number_done,
points_doing, points_done, number_wtd, number_wip, number_wdn, points_wip,
points_wdn });

      currentDate = new Date(currentDate.getTime() + CONSTANTS.DAY_MS);
    }

    const csv = papaparse.unparse(report, { delimiter: ',' });
  }

```

```

    await fs.writeFile(`reports/${datetime}_daily_report.csv`, csv);
  }

  async function makeAssigneesReport(issues, sprints, datetime) {
    const report = [];
    sprints.forEach((sprint) => {
      const sprintIssues = issues.filter((issue) =>
issue.sprints.includes(sprint.number));
      sprintIssues.forEach((issue) => {
        issue.assignees.forEach((assignee) => {
          const sprintUserIndex = report.findIndex((line) =>
line.sprint === sprint.number && line.user === assignee.login);
          if (sprintUserIndex >= 0) {
            const sprintUser = report[sprintUserIndex];
            report[sprintUserIndex] = {
              ...sprintUser,
              issues: `${sprintUser.issues};${issue.title}`,
              total_issues: sprintUser.total_issues + 1,
              total_points: sprintUser.total_points +
issue.points,
            }
          } else {
            report.push({
              sprint: sprint.number,
              user: assignee.login,
              total_issues: 1,
              total_points: issue.points,
              issues: issue.title
            });
          }
        });
      });
    });
  });
  const csv = papaparse.unparse(report, { delimiter: ',' });
  await fs.writeFile(`reports/${datetime}_assignees_report.csv`, csv);
}

async function main() {
  const datetime = new Date().toLocaleString().replaceAll('/',
'-').replaceAll(':', '-').replaceAll(' ', '_');

  const { backlogIssues, skippedIssues, sprints } = await
extractDataFromRepository();

  process.stdout.write(chalk.white('Gerando Backlog Report...'));
  await makeBacklogReport(backlogIssues, datetime);
}

```

```

process.stdout.write(chalk.green(`OK\n`));

process.stdout.write(chalk.white('Gerando Skipped Report...'));
await makeSkippedReport(skippedIssues, datetime);
process.stdout.write(chalk.green(`OK\n`));

process.stdout.write(chalk.white('Gerando Sprints Report...'));
await makeSprintsReport(backlogIssues, sprints, datetime);
process.stdout.write(chalk.green(`OK\n`));

process.stdout.write(chalk.white('Gerando Daily Report...'));
await makeDailyReport(backlogIssues, sprints, datetime);
process.stdout.write(chalk.green(`OK\n`));

process.stdout.write(chalk.white('Gerando Assignees Report...'));
await makeAssigneesReport(backlogIssues, sprints, datetime);
process.stdout.write(chalk.green(`OK\n`));
}

main();

```

## APÊNDICE F - ARQUIVO PACKAGE.JSON

```

{
  "name": "agile-metrics-reports",
  "version": "1.0.0",
  "description": "Generates agile metric reports from ZenHub and GitHub
APIs",
  "type": "module",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "axios": "^0.27.2",
    "chalk": "^5.0.1",
    "dotenv": "^16.0.1",
    "loadsh": "0.0.4",
    "papaparse": "^5.3.2"
  }
}

```

## APÊNDICE G - ARQUIVO PACKAGE-LOCK.JSON

```
{
  "name": "agile-metrics-reports",
  "version": "1.0.0",
  "lockfileVersion": 2,
  "requires": true,
  "packages": {
    "": {
      "name": "agile-metrics-reports",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "axios": "^0.27.2",
        "chalk": "^5.0.1",
        "dotenv": "^16.0.1",
        "loadsh": "0.0.4",
        "papaparse": "^5.3.2"
      }
    },
    "node_modules/asynckit": {
      "version": "0.4.0",
      "resolved": "https://registry.npmjs.org/asynckit/-/asynckit-0.4.0.tgz",
      "integrity": "sha512-Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoIdYLxrKim2MjW0Q=="
    },
    "node_modules/axios": {
      "version": "0.27.2",
      "resolved": "https://registry.npmjs.org/axios/-/axios-0.27.2.tgz",
      "integrity": "sha512-t+yRIyySRTp/wua5xEr+z1q60QmLq8ABsS5O9Me1AsE5dfKqgnCFzwiCZZ/cGNd11q4/7akDWMxdhVlucjmnOQ==",
      "dependencies": {
        "follow-redirects": "^1.14.9",
        "form-data": "^4.0.0"
      }
    },
    "node_modules/chalk": {
      "version": "5.0.1",

```

```

    "resolved": "https://registry.npmjs.org/chalk/-/chalk-5.0.1.tgz",
    "integrity":
"sha512-Fo07WOYgMfCWHzSXOt2CxDbC6skS/jO9ynEcmpANMoPrD+W1r1K6Vx7iNm+AQmETU1Xr2t
+n8nzkV9t6xh3w==",
    "engines": {
      "node": "^12.17.0 || ^14.13 || >=16.0.0"
    },
    "funding": {
      "url": "https://github.com/chalk/chalk?sponsor=1"
    }
  },
  "node_modules/combined-stream": {
    "version": "1.0.8",
    "resolved":
"https://registry.npmjs.org/combined-stream/-/combined-stream-1.0.8.tgz",
    "integrity":
"sha512-FQn4MRfuJeHf7cBbBMJFXhKSDq+2kAArBlmRBvcvFE5BB1HZKXtSFASDhdlz9zOYwxh81Ddn
vmMOe/+5cdoEdg==",
    "dependencies": {
      "delayed-stream": "~1.0.0"
    },
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/delayed-stream": {
    "version": "1.0.0",
    "resolved":
"https://registry.npmjs.org/delayed-stream/-/delayed-stream-1.0.0.tgz",
    "integrity":
"sha512-ZySD7Nf91aLB0RxL4KGrKHBX17Eds1DAmEdcoVawXnLD7SDhpNgtuII2aAkg7a7QS41jxPSZ
17p4VdGnMHk3MQ==",
    "engines": {
      "node": ">=0.4.0"
    }
  },
  "node_modules/dotenv": {
    "version": "16.0.1",
    "resolved": "https://registry.npmjs.org/dotenv/-/dotenv-16.0.1.tgz",
    "integrity":
"sha512-1k6hR6wtk2FviQ4kEiSjFiH5rpzEVi8WW0x96azthVMhEspNpc4DVOUTEhtEva5VThQ8IaBX
1Pe4gSzpVVUsKQ==",
    "engines": {
      "node": ">=12"
    }
  },

```



```

"node_modules/follow-redirects": {
  "version": "1.15.1",
  "resolved":
"https://registry.npmjs.org/follow-redirects/-/follow-redirects-1.15.1.tgz",
  "integrity":
"sha512-yLAMQs+k0b2m7cVxpS1VKJVvoz7SS9Tdlzss3XRwXj+ZDH00RJgnuLx7E44wx02kQLrdM3aO
Oy+FpzS7+8OizA==",
  "funding": [
    {
      "type": "individual",
      "url": "https://github.com/sponsors/RubenVerborgh"
    }
  ],
  "engines": {
    "node": ">=4.0"
  },
  "peerDependenciesMeta": {
    "debug": {
      "optional": true
    }
  }
},
"node_modules/form-data": {
  "version": "4.0.0",
  "resolved":
"https://registry.npmjs.org/form-data/-/form-data-4.0.0.tgz",
  "integrity":
"sha512-ETEkLSGi5t0QMZuiXoA/Q6vcnxcLQP5vdugSpuAyi6SVGi2clPPp+xEhuMaHC+zGgn31Kd2
35W35f7Hykkaww==",
  "dependencies": {
    "asynckit": "^0.4.0",
    "combined-stream": "^1.0.8",
    "mime-types": "^2.1.12"
  },
  "engines": {
    "node": ">= 6"
  }
},
"node_modules/loadsh": {
  "version": "0.0.4",
  "resolved": "https://registry.npmjs.org/loadsh/-/loadsh-0.0.4.tgz",
  "integrity":
"sha512-U+wLL8InpfRalWrr+0SuhWgGt10M4OyAk6G8xCYo2rwpHtxZkWiFpjei0vO463ghW8LPCdh
qQxXlMy2qicAEw==",
  "deprecated": "This is a typosquat on the popular Lodash package.
This is not maintained nor is the original Lodash package."
}

```

```

    },
    "node_modules/mime-db": {
      "version": "1.52.0",
      "resolved":
"https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",
      "integrity":
"sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETV1
OoK1UC2nU3gYvg==",
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/mime-types": {
      "version": "2.1.35",
      "resolved":
"https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",
      "integrity":
"sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICyPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1
zxuWL50hzaeXiw==",
      "dependencies": {
        "mime-db": "1.52.0"
      },
      "engines": {
        "node": ">= 0.6"
      }
    },
    "node_modules/papaparse": {
      "version": "5.3.2",
      "resolved":
"https://registry.npmjs.org/papaparse/-/papaparse-5.3.2.tgz",
      "integrity":
"sha512-6dNZu0Ki+gyV0eBsFKJhYr+MdQYAzFUGlBMNj3GNrmHxmz11fRa24CjFObPXtj cet1Ov5Ad2
99MhIK0znp3afw=="
    },
    "dependencies": {
      "asnckit": {
        "version": "0.4.0",
        "resolved":
"https://registry.npmjs.org/asnckit/-/asnckit-0.4.0.tgz",
        "integrity":
"sha512-Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoI
dYLxrKim2MjW0Q=="
      },
      "axios": {
        "version": "0.27.2",

```

```

    "resolved": "https://registry.npmjs.org/axios/-/axios-0.27.2.tgz",
    "integrity":
"sha512-t+yRIyySRTp/wua5xEr+z1q60QmLq8ABsS509Me1AsE5dfKqgnCFzwiCZZ/cGNd1lq4/7akD
WMxdhVlucjmnOQ==",
    "requires": {
      "follow-redirects": "^1.14.9",
      "form-data": "^4.0.0"
    }
  },
  "chalk": {
    "version": "5.0.1",
    "resolved": "https://registry.npmjs.org/chalk/-/chalk-5.0.1.tgz",
    "integrity":
"sha512-Fo07WOYGqMfCWHOzSXOt2CxDbC6skS/jO9ynEcmpANMoPrD+W1r1K6Vx7iNm+AQmETU1Xr2t
+n8nzkv9t6xh3w=="
  },
  "combined-stream": {
    "version": "1.0.8",
    "resolved":
"https://registry.npmjs.org/combined-stream/-/combined-stream-1.0.8.tgz",
    "integrity":
"sha512-FQON4MRfuJehf7cBbBMJFXhKSDq+2kAArBlmRBvcvFE5BB1HZKXtSFASDhdlz9zOYwxh81Ddn
vmMOe/+5cdoEdg==",
    "requires": {
      "delayed-stream": "~1.0.0"
    }
  },
  "delayed-stream": {
    "version": "1.0.0",
    "resolved":
"https://registry.npmjs.org/delayed-stream/-/delayed-stream-1.0.0.tgz",
    "integrity":
"sha512-ZySD7Nf91aLB0RxL4KGrKHBXl7Eds1DAmEdcoVawXnLD7SDhpNgtuII2aAkg7a7QS41jxPSZ
17p4VdGnMHk3MQ=="
  },
  "dotenv": {
    "version": "16.0.1",
    "resolved": "https://registry.npmjs.org/dotenv/-/dotenv-16.0.1.tgz",
    "integrity":
"sha512-1k6hR6wtk2FviQ4kEiSjFiH5rpzEVi8WW0x96aztHVMhEspNpc4DVOUTEHTtEva5VThQ8IaBX
1Pe4gSzpVVUsKQ=="
  },
  "follow-redirects": {
    "version": "1.15.1",
    "resolved":
"https://registry.npmjs.org/follow-redirects/-/follow-redirects-1.15.1.tgz",

```

```

"integrity":
"sha512-yLAMQs+k0b2m7cVxpS1VKJVvoz7SS9Tdlzss3XRwXj+ZDH00RJgnuLx7E44wx02kQLrdM3aO
Oy+FpzS7+8OizA=="
  },
  "form-data": {
    "version": "4.0.0",
    "resolved":
"https://registry.npmjs.org/form-data/-/form-data-4.0.0.tgz",
    "integrity":
"sha512-ETekLSGi5t0QMZuiXoA/Q6vcnxcLQP5vdugSpuAyi6SVGi2clPPp+xEhuMaHC+zGgn31Kd2
35W35f7Hykkaww==",
    "requires": {
      "async-kit": "^0.4.0",
      "combined-stream": "^1.0.8",
      "mime-types": "^2.1.12"
    }
  },
  "loadsh": {
    "version": "0.0.4",
    "resolved": "https://registry.npmjs.org/loadsh/-/loadsh-0.0.4.tgz",
    "integrity":
"sha512-U+wLL8InpfRalWrr+0SuhWgGt10M4OyAk6G8xCY02rwpIHtxZkWiFpjei0vO463ghW8LPCdh
qQxXlMy2qicAEw=="
  },
  "mime-db": {
    "version": "1.52.0",
    "resolved":
"https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",
    "integrity":
"sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIBe+1ETV1
OoK1UC2nU3gYvg=="
  },
  "mime-types": {
    "version": "2.1.35",
    "resolved":
"https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",
    "integrity":
"sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICyPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1
zxuWL50hzaeXiw==",
    "requires": {
      "mime-db": "1.52.0"
    }
  },
  "papaparse": {
    "version": "5.3.2",
    "resolved":

```

```
"https://registry.npmjs.org/papaparse/-/papaparse-5.3.2.tgz",  
    "integrity":  
    "sha512-6dNZu0Ki+gyV0eBsFKJhYr+MdQYAzFUGlBMNj3GNrmHxmz1lfRa24CjF0bPXtjcet1Ov5Ad2  
    99MhIK0znp3afw=="  
  }  
}  
}
```

# APÊNDICE H - FERRAMENTA DE VISUALIZAÇÃO DE MÉTRICAS ÁGEIS



## PUMA

Agile Metrics Report | PUMA Project 2022/1

por Gabriel de M. Amorim | Engenharia de Produção - Faculdade de Tecnologia

**Nº Itens (To Do)**  
43

**Pontos (Doing)**  
220

**Nº Itens (Doing)**  
35

**Pontos (Done)**  
70

**Nº Itens (Done)**  
9

**Successo de Pontos (%)**  
77,67

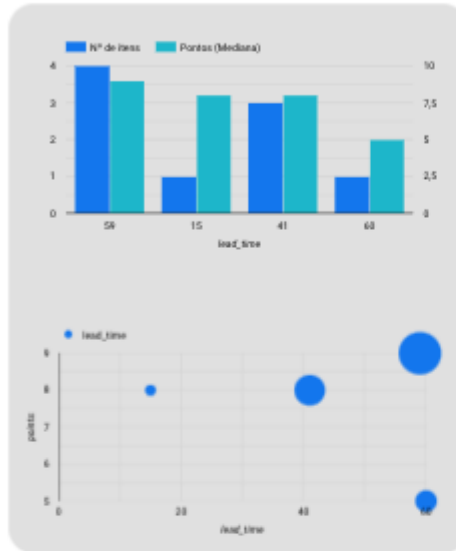
**Contribuição com o projeto**  
3,33



### LEAD TIME

Diferença entre quando uma atividade é criada e quando ela é finalizada

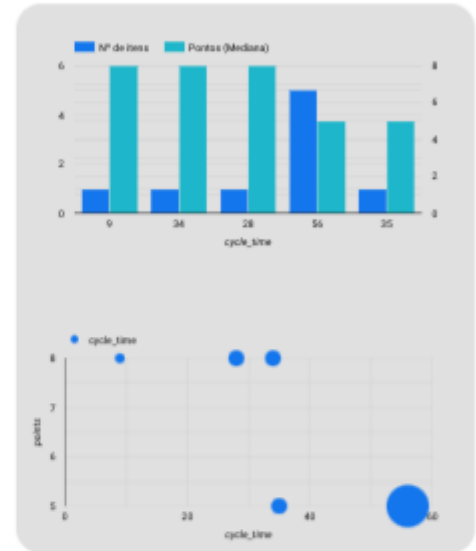
Gráfico 1: LT x Nº Itens, Pontos (Mediana)  
Gráfico 2: Dispersão (LT x Pontos)



### CYCLE TIME

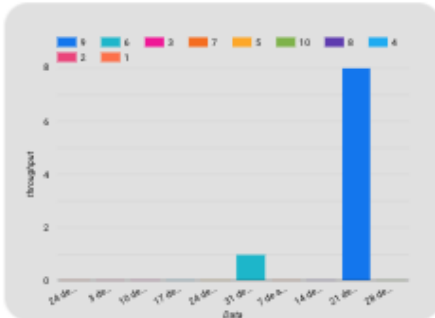
Diferença entre quando uma atividade entra em progresso e quando ela é finalizada

Gráfico 1: CT x Nº Itens, Pontos (Mediana)  
Gráfico 2: Dispersão (CT x Pontos)



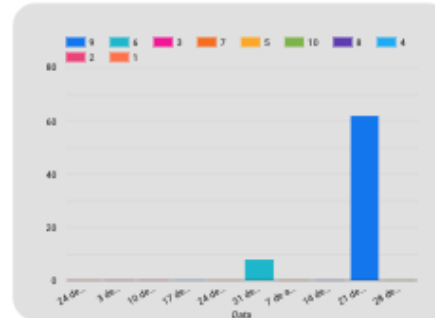
### Throughput

Número de Itens entregues / Sprint



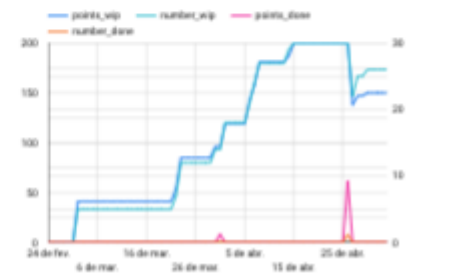
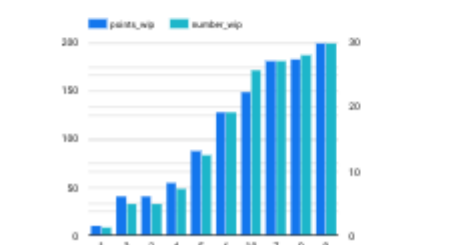
### Velocity

Número de pontos entregues / Sprint



### WIP

Número de Itens entregues / Sprint





# PUMA



## Agile Metrics Report | PUMA Project 2022/1

por Gabriel M. de Jesus | Engenharia de Produto | Foco: Estado de Trabalho

### Overview do Projeto

Nº Itens (To Do)  
38

Nº Itens (Doing)  
13

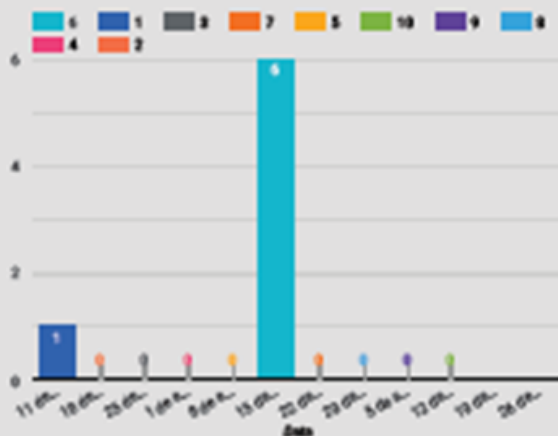
Nº Itens (Done)  
7

Pontos (Doing)  
34

Pontos (Done)  
15

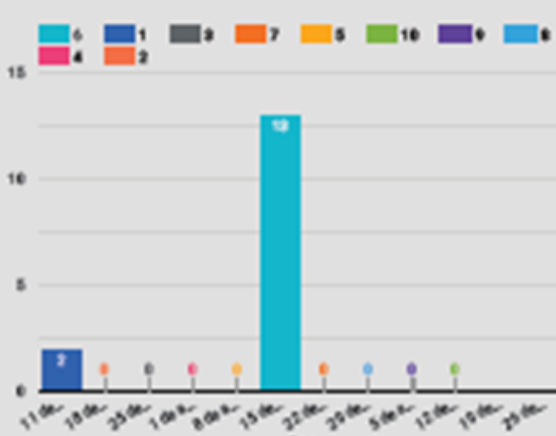
### Throughput

Número de itens entregues / Sprint



### Velocity

Número de pontos entregues / Sprint



### LEAD TIME

Diferença entre quando uma atividade é criada e quando ela é finalizada

Gráfico 1: CT x Nº Itens, Pontos (Mediana)  
Gráfico 2: Dispersão (CT x Pontos)

### CYCLE TIME

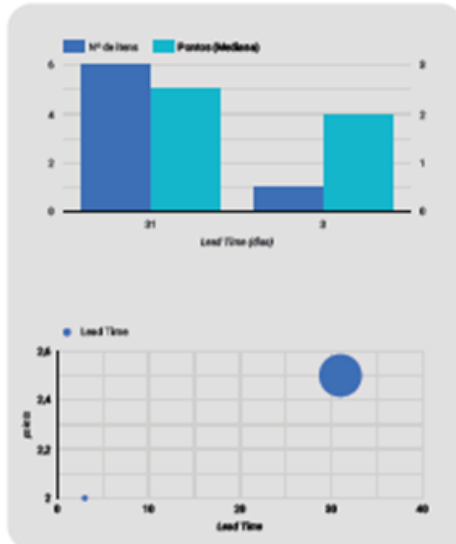
Diferença entre quando uma atividade entra em progresso e quando ela é finalizada

Gráfico 1: CT x Nº Itens, Pontos (Mediana)  
Gráfico 2: Dispersão (CT x Pontos)

## LEAD TIME

Diferença entre quando uma atividade é criada e quando ela é finalizada

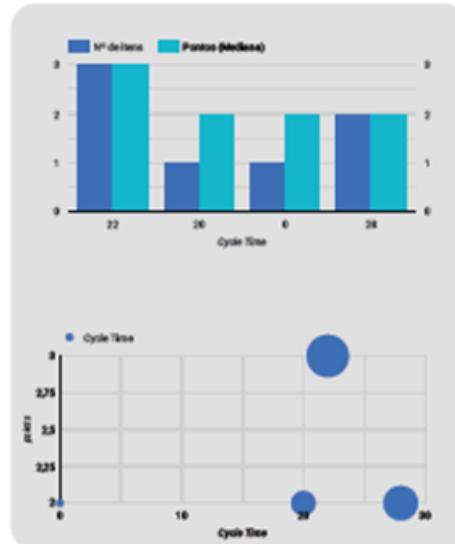
Gráfico 1: CT x Nº Items, Pontos (Mediana)  
Gráfico 2: Dispersão (CT x Pontos)



## CYCLE TIME

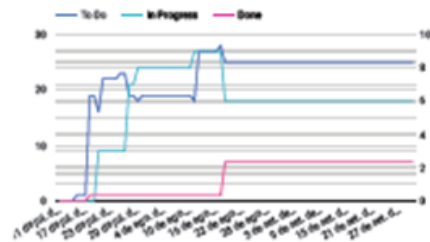
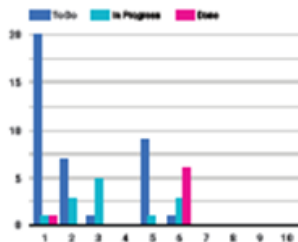
Diferença entre quando uma atividade entra em progresso e quando ela é finalizada

Gráfico 1: CT x Nº Items, Pontos (Mediana)  
Gráfico 2: Dispersão (CT x Pontos)



## WIP

Quantidade de itens de pipeline por sprint



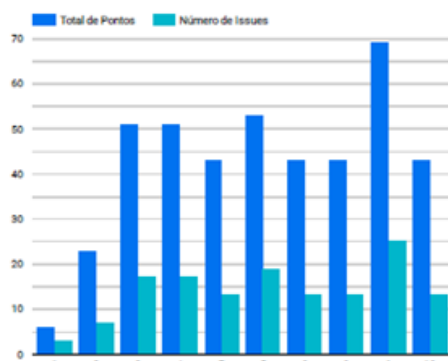
## Análise Individual | PUMA Project 2022/1

por André M. M. de Jesus | Engenharia de Produção - Faculdade de Tecnologia

## Histórico (Pontos e Nº de Issues)

Usuário -

Sprint -



Sprint	Usuário	Número de Issues	Total de Pontos
1	enzoggs	1	2
1	giovannabotti...	1	2
1	santosm46	1	2
2	edudson	1	5
2	GabrielCabral	1	1
2	codehg	1	1
2	AnHoff	1	3
2	bhsouza	1	5
2	enzoggs	1	3
2	Bruno-Felix	1	5
3	bhsouza	3	12
3	FelipeDreito	1	3
3	feitasc	1	3
3	GabrielCabral	2	2



## Histórico de detalhamento das issues

Sprint	Nome da Issue	Responsável	Pontos
1	[#62] Adiciona documentação dinâmica	enzoggs	2
1	[#62] Adiciona documentação dinâmica	giovannabbottino	2
1	[#62] Adiciona documentação dinâmica	santosm46	2
2	[Bugfix] Validação nome de Disciplina em Criar Disciplina	edudson	5
2	[Bugfix] Alterar rotulo de botão editar disciplina	GabrielCabrall	1
2	[Bugfix] Alterar rotulo de botão editar disciplina	codehg	1
2	[Bugfix] Validar telefone de usuário no cadastro	AnHoff	3
2	[Bugfix] Validação nome de Disciplina em Criar Disciplina	bhsouza	5
2	[Bugfix] Validar telefone de usuário no cadastro	enzoggs	3
2	[Bugfix] Validação nome de Disciplina em Criar Disciplina	Bruno-Felix	5
3	[Bugfix] Corrigir estilo cabeçalho da tabela listagem projetos;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validação nome de Disciplina em Criar Disciplina	bhsouza	12
3	[Melhoria] Alterar Ordenação da lista de palavras-chaves	FelipeDireito	3
3	[Melhoria] Alterar Ordenação da lista de palavras-chaves	freitasc	3
3	[Bugfix] Validar campos em Análise da Proposta;[Bugfix] Alterar rotulo de botão editar disciplina	GabrielCabrall	2
3	[Bugfix] Validar telefone de usuário no cadastro	enzoggs	3
3	[Bugfix] Validar campos obrigatórios no cadastro	YuriBre	3
3	[Bugfix] Validação nome de Disciplina em Criar Disciplina	Bruno-Felix	5
3	[Bugfix] Validar campos em Análise da Proposta;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validar telefone de usuário no cadastro;[Bugfix] Validar campos obrigatórios no cadastro	AnHoff	9
3	[Bugfix] Corrigir estilo cabeçalho da tabela listagem projetos;[Bugfix] Validação nome de Disciplina em Criar Disciplina	edudson	10
3	[Bugfix] Alterar rotulo de botão editar disciplina	codehg	1
4	[Bugfix] Validar telefone de usuário no cadastro	enzoggs	3
4	[Bugfix] Corrigir estilo cabeçalho da tabela listagem projetos;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validação nome de Disciplina em Criar Disciplina	bhsouza	12
4	[Bugfix] Validar campos em Análise da Proposta;[Melhoria] Sinalizar Visualmente os campos obrigatórios no cadastro;[Bugfix] Validar telefone de usuário no cadastro;[Bugfix] Validar campos obrigatórios no cadastro	AnHoff	9
4	[Bugfix] Alterar rotulo de botão editar disciplina	codehg	1
4	[Melhoria] Alterar Ordenação da lista de palavras-chaves	freitasc	3
4	[Bugfix] Validar campos obrigatórios no cadastro	YuriBre	3
4	[Melhoria] Alterar Ordenação da lista de palavras-chaves	FelipeDireito	3
4	[Bugfix] Validação nome de Disciplina em Criar Disciplina	Bruno-Felix	5