



MAPEAMENTO DE MÉTRICAS PARA DESEMPENHO DE PROJETOS ÁGEIS: REVISÃO SISTEMÁTICA DA LITERATURA

Mapping metrics for agile project performance: systematic literature review

José da Silva Azanha Neto, Renato Penha, Marcelo Luiz do Amaral Gonçalves

Universidade Nove de Julho, Brazil

Email: jose.azanha@gmail.com, renato.penha@uni9.pro.br, marluago@gmail.com

RESUMO

Em um contexto competitivo onde as organizações precisam entregar produtos ou serviços por meio da gestão ágil de projetos, um processo em que as métricas se tornam insumos importantes para o controle e monitoramento de projetos. O que se busca nesse artigo é compreender se existem organizações que, apesar de conduzirem projetos ágeis, utilizam controle os projetos por meio do uso de métricas tradicionais. Além disso, se busca entender se a utilização de métricas de projetos preditivos dificultam a visão e o trabalho da liderança corporativa para tomar decisões em projetos ágeis. O objetivo deste artigo foi de mapear quais são as métricas utilizadas para medir o desempenho de projetos tradicionais e ágeis. Como estratégia metodológica foi adotado uma Revisão Sistemática de Literatura para auxiliar no mapeamento e avaliação de uma estrutura intelectual específica para desenvolver um corpo de conhecimento. Para a coleta de dados foram utilizadas as bases *Web of Science* e *Scopus*. Foram encontrados 83 artigos e os resultados apontaram que os projetos tradicionais seguem controlando os projetos com métricas tradicionais, tal como *Earned Value Management* (EVM). Em projetos ágeis, as métricas mais comuns são métricas associadas *backlog* do produto, *delivery* e qualidade do produto. Este estudo contribui com outras pesquisas que desejarem identificar as métricas utilizadas entre as diferentes abordagens de gestão de projetos e orientar novas pesquisas para trabalhos futuros.

Palavras-chave: Desenvolvimento de Software; Métricas Ágeis e Tradicionais; Scrum; Kanban.

ACEITO EM: 10/12/2023

PUBLICADO EM: 30/12/2023



RISUS - Journal on Innovation and Sustainability
volume 14, número 4 - 2023
ISSN: 2179-3565
Editor Científico: Arnaldo José de Hoyos Guevara
Editor Assistente: Vitória Catarina Dib
Avaliação: Melhores práticas editoriais da ANPAD

MAPPING METRICS FOR AGILE PROJECT PERFORMANCE: SYSTEMATIC LITERATURE REVIEW

Mapeamento de métricas para desempenho de projetos ágeis: revisão sistemática da literatura

José da Silva Azanha Neto, Renato Penha, Marcelo Luiz do Amaral Gonçalves
Universidade Nove de Julho, Brazil
Email: jose.azanha@gmail.com, renato.penha@uni9.pro.br, marluago@gmail.com

ABSTRACT

In a competitive context where organizations need to deliver products or services through agile project management, a process in which metrics become important inputs for project control and monitoring. What is sought in this article is to understand whether there are organizations that, despite conducting agile projects, use project control through the use of traditional metrics. In addition, it seeks to understand whether the use of predictive project metrics hinder the vision and work of corporate leadership to make decisions in agile projects. The objective of this article was to map which are the metrics used to measure the performance of traditional and agile projects. As a methodological strategy, a Systematic Literature Review was adopted to assist in mapping and evaluating a specific intellectual structure to develop a body of knowledge. For data collection, the Web of Science and Scopus databases were used. 83 articles were found and the results showed that traditional projects continue to control projects with traditional metrics, such as Earned Value Management (EVM). In agile projects, the most common metrics are metrics associated with product backlog, delivery and product quality. This study contributes to other studies that wish to identify the metrics used between the different project management approaches and guide new research for future work.

Keywords: Software Development; Agile and Traditional Metrics; Scrum; Kanban.

INTRODUÇÃO

O Modelo em Cascata (MC) foi criado com o objetivo de representar a execução de fases de um projeto de desenvolvimento de *software* (Royce, 1970). Nesse modelo, cada fase possui dependência com a fase anterior, onde a próxima fase somente poderá ser iniciada quando a fase antecedente for concluída (Sommerville, 2007). O MC é composto pelas fases de Levantamento de Requisitos, Análise de Requisitos, Projeto, Desenvolvimento, Testes e Implantação (Pressman, 2021). Dentre elas, as fases de Levantamento de Requisitos, Análise de Requisitos e Projeto podem exigir grandes esforços durante o planejamento e concepção sobre o escopo que será desenvolvido no projeto e esperado pelo cliente (Van Casteren, 2017).

No contexto das práticas gerenciais, os projetos orientados ao MC sob as práticas tradicionais de gestão de projetos possuem incertezas no início do projeto que podem mudar o escopo e o direcionamento do *software* durante o projeto (Sutherland, 2019).

Em se tratando das práticas gerenciais e o desempenho de projetos, o uso de métricas tornou-se um fator estratégico para que o controle e o monitoramento do projeto se tornem eficazes (Glenwright, 2007). No tocante às práticas tradicionais, as métricas mais utilizadas pelas organizações é o gerenciamento do valor agregado (Junior e Junior, 2019). O gerenciamento do valor agregado ou *Earned Value Management* (EVM), foi proposto pelo *Project Management Institute* (PMI, 2021). O EVM é uma técnica de gerenciamento de projetos para medir o desempenho e o progresso de projetos combinando as medidas de escopo, tempo e custos, mediante um planejamento previamente estabelecido (Glenwright, 2007; PMI, 2021). Porém, a utilização das métricas pode causar uma percepção difusa entre as práticas tradicionais e as práticas ágeis em gerenciamento de projetos de *software*.

Em relação ao prazo do projeto, as práticas ágeis preconizam que os prazos das atividades devem ser baseados em dados históricos de entrega da equipe de projeto, em vez de estimativas baseadas no planejamento (Sutherland, 2019). No caso do MC, os prazos são baseados nas estimativas definidas na fase de planejamento e dando origem ao cronograma de atividades (Sutherland, 2019), distorcendo assim à utilização das métricas em ambas as práticas. Conforme Choudhury (2019) e Dias e Larieira (2021), essa distorção na utilização das métricas de projetos tradicionais nos projetos ágeis se justifica devido a utilização das abordagens do triângulo de ferro (escopo, prazo e custo) em projetos tradicionais e o triângulo invertido em projetos ágeis. Choudhury (2019) e Dias e Larieira (2021) adicionaram que o triângulo de ferro aborda o escopo fixo com o prazo e o custo variáveis, enquanto no ágil o triângulo invertido trata do prazo e custo fixo com o escopo variável.

No contexto das práticas ágeis, as métricas são orientadas ao valor entregue ao cliente (Gren, Goldman & Jacobsson, 2020). De acordo com Jyothi e Rao (2017) e Budacu e Pocatilu (2018), as métricas ágeis mais utilizadas são (i) *velocity*; (ii) *Work Item Age* (WIA); (iii) *Throughput*, (iv) *Cycle Time* (CT) e (v) *Lead Time* (LT). O *velocity* é uma métrica utilizada no desempenho dos desenvolvedores por meio dos pontos de história do usuário. O WIA é uma métrica para medir o tempo em dias que um requisito está em uma determinada fase do desenvolvimento. Já o *Throughput* é uma métrica para avaliar a capacidade de entrega da equipe de projeto ao final de cada iteração. O CT conta o tempo em que um requisito está em um determinado ciclo ou etapa de desenvolvimento, enquanto o LT é considerado uma medida que avalia o tempo em que um requisito é iniciado em desenvolvimento até ser entregue ao cliente (Budacu & Pocatilu, 2018).

Já as práticas tradicionais utilizam o EVM para medir o desempenho e o progresso de projetos, utilizando métricas baseadas em medidas comparativas entre o esforço planejado em relação ao esforço realizado (PMI, 2021). Glenwright (2007) apresenta as métricas mais utilizadas pelas organizações sob as práticas tradicionais de gerenciamento de projetos são (i) *Schedule Performance Index* (SPI) e (ii) *Cost Performance Index* (CPI). O SPI analisa o progresso real ou o valor agregado do trabalho realizado com o progresso do trabalho planejado até o momento da medição. Já o CPI é uma medida da eficiência de custos realizados com o progresso do custo planejado até o momento da medição. Assim, as métricas possuem relações distintas de acompanhamento durante o ciclo de vida dos projetos.

Em relação às entregas do projeto, nas práticas tradicionais, o produto gerado é entregue somente no final do projeto ou entregue em fases dentro de uma estratégia de cronograma em fases (PMI, 2021). Já nas práticas ágeis o produto é entregue em até quatro semanas (*Sprint*) na forma de incrementos funcionais. A soma dos incrementos funcionais pode viabilizar os lançamentos do *Minimum Viable Product* (MVP) (Ries, 2011). O MVP é caracterizado por ser uma versão funcional e sem erros críticos de um *software* para possível liberação aos clientes pra receber o *feedback* mais rápido possível (Sutherland, 2019).

Ao se observar a utilização das métricas para análise do desempenho de projetos de desenvolvimento de *software*, é possível notar divergências entre as práticas ágeis e tradicionais de gestão de projetos. Essa divergência consiste em um cenário em que organizações que conduzem projetos sob as práticas tradicionais decidem implementar as práticas ágeis de gestão de projetos. Porém, a liderança permanece utilizando métricas tradicionais baseadas nas abordagens de custo, prazo e escopo diferentes entre as práticas, podendo causar impactos no desempenho dos projetos ágeis.

O problema que tange e justifica nessa pesquisa é que muitas organizações que conduzem projetos, implementaram o modelo ágil de desenvolvimento de *software* em equipes isoladas e com as métricas ágeis comumente na operação (Jyothi & Rao 2017; dos Santos et al. 2018). Além disso, a liderança realiza a medição do desempenho do projeto por meio de métricas tradicionais, o que podem causar impactos no gerenciamento de projetos ágeis, dado que a gestão tradicional e a ágil de projetos operam com abordagens de custo, prazo e escopo de maneiras diferentes (Choudhury, 2019; Dias & Larieira, 2021).

Com base no apresentado sobre a divergência da utilização de métricas para o controle de projetos tradicionais e ágeis, abre-se uma lacuna para a realização desta pesquisa, procurando responder a seguinte questão: “Quais são as métricas utilizadas para medir o desempenho de projetos nas práticas ágeis?”. O objetivo deste artigo foi de mapear quais são as métricas utilizadas para medir o desempenho de projetos tradicionais e ágeis. Para responder tal indagação, foi adotado como estratégia metodológica uma Revisão Sistemática de Literatura (RSL) verificando as métricas tradicionais e ágeis utilizadas no desempenho de projetos. A justificativa para tal escolha se dá pelo papel da RSL que auxilia no mapeamento e avaliação de uma estrutura intelectual específica para desenvolver um corpo de conhecimento (Tranfield, Denyer & Smart, 2003).

Na seção seguinte serão apresentados os procedimentos metodológicos. Posteriormente são apresentados os resultados e, por fim, serão apresentadas as conclusões, discutidas as limitações desta pesquisa e indicações para estudos futuros.

1 MATERIAIS E MÉTODOS

Esta pesquisa adotou uma RSL como método a fim de compreender a convergência de duas temáticas relevantes no contexto de gestão de projetos, as métricas, tanto nas práticas tradicionais como nas práticas ágeis. A RSL se difere das tradicionais revisões narrativas por adotar um processo científico sistemático que é replicável e transparente (Tranfield, Denyer & Smart, 2003). Nesse sentido, justifica-se também a sua utilização por uma RSL minimizar o viés na construção de um *corpus* teórico, bem como a possibilidade de construir uma trilha de auditoria das decisões e procedimentos aplicados (Cook, Mulrow & Haynes, 1997).

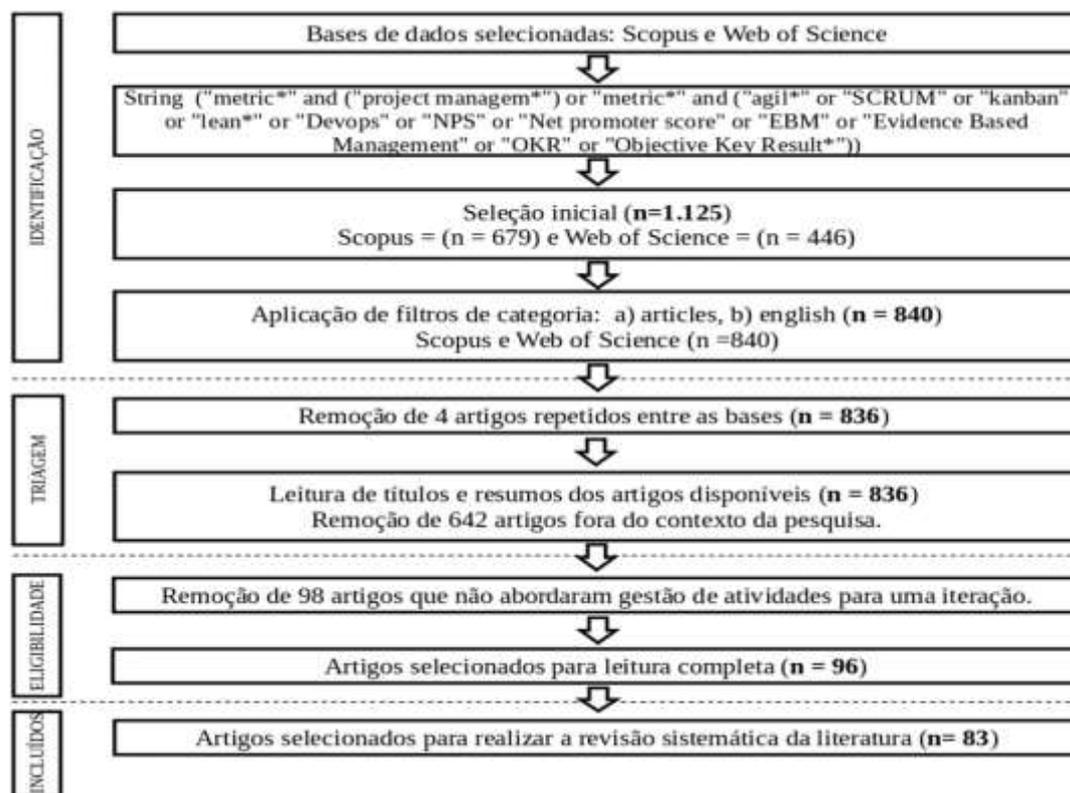
Os procedimentos para realização desta RSL seguiram seis fases prescritas por Pollock e Berge (2018): (i) esclarecer metas e objetivos de pesquisa; (ii) buscar pesquisas relevantes; (iii) coletar dados; (iv) avaliar a qualidade dos estudos; (v) sintetizar as evidências; (vi) interpretar os achados.

A primeira etapa foi motivada pela questão que norteia esta pesquisa, que foi “quais são as métricas utilizadas para medir o desempenho de projetos tradicionais e ágeis?”. Assim, foram utilizadas como fontes de pesquisa as bases de dados acadêmicas *Web of Science* e *Scopus*, por serem umas das principais bases de dados para acesso as pesquisas publicadas na área de ciências sociais.

A *string* utilizada para realização das buscas foi ("*metric**" and ("*project managem**") or "*metric**" and ("*agil**" or "*SCRUM*" or "*kanban*" or "*lean**" or "*Devops*" or "*NPS*" or "*Net promoter score*" or "*EBM*" or "*Evidence Based Management*" or "*OKR*" or "*Objective Key Result**"). A pesquisa foi realizada em 01 de maio de 2022. A utilização dos operadores booleanos “and” e “or”, além do uso do símbolo “*” permitem uma maior abrangência e controle na construção da base de pesquisa. Os operadores são aplicados levando em consideração a intersecção das duas áreas de pesquisa estudadas. O uso do asterisco incorpora todas as variações da palavra na posição posterior a que ela se encontra. Destaca-se que não foi aplicado filtro temporal para que fosse possível mapear toda a produção sobre as temáticas estudadas. Portanto, após a primeira fase que compreende o estabelecimento dos objetivos, os pesquisadores passaram para a aplicação da *string* de busca. Os resultados encontrados na primeira rodada passaram pela análise e triagem, apresentado na Figura 1, respeitando o proposto por Pollock e Berge (2018), que apresentam quatro etapas para a elaboração do *corpus* de análise. A Figura 1 apresenta a primeira etapa destaca a *string* utilizada para a pesquisa nas bases de dados. A segunda etapa é a triagem dos resultados com a finalidade de balizar os artigos correspondentes à proposta de pesquisa. Já na terceira etapa foram aplicados os critérios de elegibilidade, onde os artigos foram avaliados segundo os critérios de inclusão e

exclusão. A quarta etapa constituiu o *corpus* de análise, resultando na base de dados amostral de artigos para a pesquisa.

Figura 1 - Etapas da RSL



Fonte: Adaptado de Pollock e Berge (2018).

Para a composição da base de dados, filtros foram aplicados durante a pesquisa inicial, como o critério de somente artigos em periódicos, excluindo da base os artigos em congresso, livros, entre outros. Essa exclusão foi definida, devido esses estudos não passarem por uma revisão por pares em alguns casos, além de mitigar a redundância de obras apresentadas em congressos e publicadas em periódicos. As áreas delimitadas para esta pesquisa foram: “*Management*” e “*Business*” para *Web of Science*, “*Business*” para a *Scopus*. Com a consolidação das bases de dados, foram removidos os artigos repetidos com o objetivo de eliminar a redundância. Na sequência, a base de dados foi tratada com auxílio de planilhas eletrônicas do *LibreOffice*. O *LibreOffice* permitiu realizar as análises dos dados e apresentar os resultados a partir da combinação de informações quantitativas pela análise de frequência, bem como informações qualitativas pela categorização dos conteúdos dos artigos. Esta fase da pesquisa permitiu também apresentar uma análise descritiva relevante. A partir da leitura dos títulos e resumos na fase de triagem para verificar a elegibilidade dos artigos foram estabelecidos alguns critérios de inclusão e exclusão, destacados na Tabela 1. Após verificação e leitura dos resumos e introduções dos 840 artigos selecionados previamente, a base final foi composta por 83 artigos aceitos que formaram o *corpus* de pesquisa.

Tabela 1 - Critérios de Inclusão e Exclusão

Critérios de Inclusão	Razão para Inclusão
Artigos que conceituam os construtos estudados	Permitir abordar finalidades da pesquisa: compreender os construtos estudados conforme as obras.
Artigos que abordam as relações entre os construtos estudados	Permitir contemplar alinhamento das obras: compreender as interdependências e relações entre os construtos.
Artigos publicados	Oferecer maior rigor nos argumentos e contribuições teóricas estudadas.
Critérios de Exclusão	Razão para Exclusão
Artigos com foco em Qualidade, <i>Marketing</i> , Saúde, Finanças ou outros fins fora dos construtos determinados.	Excluir artigos que não estejam focados nas questões que oferecerão <i>insights</i> para atender os objetivos da pesquisa.
Artigos sem fundamentação teórica relevante ou de baixa relação com os construtos.	Uma das finalidades do estudo é obter perspectivas futuras de pesquisa, por meio de conhecimento teórico existente em uma estrutura, para a qual os pressupostos teóricos são pré-requisitos.

Fonte: Elaborado pelos autores, 2021.

O próximo passo foi de realizar a leitura dos 83 artigos contidos no *corpus* de análise, categorizando os conteúdos em planilhas do *LibreOffice* com o objetivo de apresentar um agrupamento dos achados e comparação das categorias. As atividades aplicadas nessa fase estão alinhadas com as prescrições de Pollock e Berge (2018) nas fases (v) sintetizar as evidências e (vi) interpretar os achados. Embora tenham sido aplicados alguns tratamentos quantitativos. Nessa pesquisa, priorizou-se a análise qualitativa dos artigos a fim de constituir uma matriz que pudesse representar os achados deste estudo.

2 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nessa seção são apresentados os resultados após a realização dos processos de análise dos artigos e as categorias evidenciadas após a análise aprofundada dos conteúdos dos artigos.

1 Evolução da publicação dos artigos

Os artigos selecionados nas bases de dados *Web of Science e Scopus* foram triados e o *corpus* de análise foi constituído por 83 artigos. Os artigos passaram por uma análise mais aprofundada, o que permitiu compreender os estudos sobre a relação entre as métricas ágeis e as métricas tradicionais. Os 83 artigos analisados estão situados temporalmente entre os anos de 2011 até maio de 2021. De acordo com esta base de dados, a publicação de artigos que tratam da relação entre as métricas ágeis e métricas tradicionais ficou entre 4 e 5 artigos de 2011 até 2013. Entre os anos de 2014 até 2017 que as publicações subiram para 8 e 9 artigos publicados. Somente em 2018 que 17 artigos foram publicados e retornando para 8 artigos em 2019, 13 em 2020 e 10 em 2021. Um aspecto relevante é que no ano de 2018 teve o número maior de publicações e correspondeu com 18% de publicações nos últimos dez anos ou de 2011 a 2021. Após a análise da distribuição temporal dos artigos, também se buscou compreender a frequência de artigos por periódico. A Tabela 2 demonstra os primeiros dez periódicos e a respectiva quantidade de artigos publicados, ordenados pelo maior número de publicações e pela ordem alfabética por periódicos.

Tabela 2 - Periódicos e número de publicações no tema

#	Periódicos	Publicações
1	IEEE Access	15
2	Journal Of Systems And Software	14
3	International Journal Of Project Management	10
4	International Journal Of Lean Six Sigma	10
5	Computer Networks	7
6	Computers & Industrial Engineering	7
7	Journal Of Modelling In Management	6
8	Journal Of Software-Evolution And Process	6
9	Journal Of Manufacturing Technology Management	6
10	International Journal Of Software Engineering And Knowledge Engineering	6

Fonte: dados da pesquisa, 2022.

Com relação ao número de artigos publicados por periódico, os periódicos considerados mais profícuos, de acordo com dados da pesquisa, são: ‘IEEE Access’ com 15 artigos; ‘*Journal Of Systems And Software*’ com 14 artigos e ‘*International Journal Of Project Management*’ com 10 artigos. Nota-se que a grande maioria dos periódicos configura com apenas 1 e 2 artigos publicados, sumarizando um total de 100 periódicos. Apesar do tema de gestão de projetos estar associado a Administração, as três revistas com mais publicações são associadas a engenharia e tecnologia (*IEEE Access* e *Journal Of Systems And Software*) e uma de gestão de projetos (*International Journal Of Project Management*). Na próxima seção serão discutidos artigos mapeados no *corpus* dessa pesquisa, a categorização que os artigos foram submetidos e a discussão dos achados dos artigos que compõem as categorias.

2 Análise e discussão das categorias encontradas

Após o mapeamento dos artigos que constituíram o *corpus* de análise, foi realizada uma análise aprofundada dos conteúdos publicados. A leitura e categorização dos conteúdos levou a classificação dos artigos em quatro categorias, apresentados na Tabela 3.

Tabela 3 - Classificação dos artigos em 4 categorias

Categoria	Qtde	Autores
Métricas de Projetos Tradicionais	40	Vanhoucke (2011); Chen (2021); Aljedaibi e Khamis (2019); Colin <i>et al.</i> (2015); Rajagopalan e Srivastava (2018); Chen, Chen e Lin (2016); Fleche <i>et al.</i> (2017); Sharma, Caldas e Mulva (2021); Batselier e Vanhoucke (2015); Chen (2018); Pajares e Lopez-Paredes (2011); Vanhoucke (2019); Colin e Vanhoucke (2014); Song, Marten e Vanhoucke (2021); Kerkhove e Vanhoucke (2017); Geng <i>et al.</i> (2018); Sato e Hirao (2013); Staron e Meding (2016); Spijkman <i>et al.</i> (2021); Ibrahim, Hanna e Kievet (2020); Chen <i>et al.</i> (2020); Ko e Cho (2020); Yousefi <i>et al.</i> (2019); Orgut <i>et al.</i> (2018); Kristiansen e Ritala (2018); Rajablu <i>et al.</i> (2017); Han, Choi e O’Connor (2017); Hanna (2016); Kim, Kim e Cho (2016); Hazır, (2015); Wauters e Vanhoucke (2015); Grau e Back (2015); Chen, H. L. (2014); Narbaev e De Marco (2014); Jethani (2013); El Asmar, Hanna e Loh (2013); Hanna, Tadt e Whited (2012); Iqbal, Naeem e Khan (2012); Pozzana <i>et al.</i> (2021); Colin e Vanhoucke (2015).
Métricas Ágeis de Fluxo	9	Grimaldi <i>et al.</i> (2016); Dennehy e Conboy (2018); Mas, Mesquida e Pacheco (2020); Ahmed <i>et al.</i> (2017); Sajedi-Badashian e Stroulia (2020); Petersen e Wohlin (2011); Choraś <i>et al.</i> (2020); Meidan <i>et al.</i> (2018); Sadler (2020).

Métricas Ágeis de Backlog do Produto	9	Azzeh e Nassif (2016); Wallace e Sheetz (2014); Parrend <i>et al.</i> (2014); Kayes, Sarker e Chakareski (2016); Zheng <i>et al.</i> (2021); Pavlova <i>et al.</i> (2021); Antinyan (2014); Prakash e Viswanathan (2018); Asha e Mani (2018).
Métricas Ágeis de Produto	25	Nidagundi e Novickis (2016); Mäkiäho, Vartiainen, e Poranen (2022); Chang (2015); Mascarenhas Hornos <i>et al.</i> (2014); Singh, Singh e Singh (2019); Simpson <i>et al.</i> (2021); Day <i>et al.</i> (2019); Üsfekes, <i>et al.</i> (2019); Abdelrahman Aljemabi, Wang e Saleh (2020); Costa, Vasconcelos, e Fragoso (2020); Ibba <i>et al.</i> (2018); Concas <i>et al.</i> (2012); Shawky e Abd-El-Hafiz (2016); Tahir, Rasool e Noman (2018); Isong e Obeten (2013); Tanveer <i>et al.</i> (2019); Pradhan e Nanniyur (2021); Kamulegeya, Mugwanya e Hebig (2018); Yamashita e Counsell (2013); Bakhtiary, Gandomani e Salajegheh (2020); Hernández <i>et al.</i> (2020); Athanasiou <i>et al.</i> (2014); Gao <i>et al.</i> (2011); Perkusich <i>et al.</i> (2017); Savola, Frühwirth e Pietikäinen (2012).
Total	83	

Fonte: dados da pesquisa, 2022.

As quatro categorias encontradas compreendem uma abstração observada nas leituras dos artigos. Cabe explicitar que o processo de análise inicial dos artigos foi individual e, depois, para seleção das categorias passou-se pela análise dos pesquisadores desta pesquisa. O processo de categorização foi realizado pela discussão e estabelecimento de um consenso sobre qual era a maior aderência de todos os artigos em cada uma das categorias constituídas.

Na próxima seção, serão discutidas as categorias para explicitar os achados desta pesquisa e as categorias aqui apresentadas representam uma forma de explicação e organização dos conteúdos estudados.

2.2.1 Métricas de Projetos Tradicionais

Como visto, as práticas de gerenciamento de projetos tradicionais se baseiam, para medir o desempenho dos projetos, a partir de um conjunto de métricas orientados frente a um planejamento prévio. Desse modo, as métricas são caracterizadas, em sua essência, utilizando as medidas de escopo, tempo e custos, mediante a um planejamento previamente estabelecido. Tais medidas são comparadas com a posição atual do projeto (trabalho realizado) em relação ao planejado até o momento da medição (trabalho realizado).

Frente ao uso das práticas tradicionais de gestão de projetos, Vanhoucke (2011) apresentou como métrica, no Monte-Carlo para simular o progresso do projeto sob a estrutura analítica do projeto (EAP), utilizando dados do gerenciamento de valor agregado (EVM) como possíveis sinais de alertas e acionadores para a necessidade de ações corretivas. Ainda sob o olhar da EAP, Colin e Vanhoucke (2015) apresentaram um modelo multivariado adotando como métrica o EVM/ES, resultando no *Earned Value Management/Earned Schedule*. A métrica é utilizada pelo gerente caso o projeto apresente algum risco de maneira preventiva. Kerkhove e Vanhoucke (2017) adotaram a técnica *Earned Incentive Metric*, empregando o EVM/ES com incentivos de custo e/ou tempo. Como resultado, os autores destacaram que *Earned Incentive Metric* apresentou em papel mais significativo que o EVM/ES em uma amostra de 4.200 projetos com tamanhos e estruturas de contrato diferentes ou parametrizando o tamanho de um projeto grande (Chen *et al.*, 2020).

Sobre a adoção do EVM como métrica, Chen, Chen e Lin (2016) concluíram que a melhoria no poder preditivo do *Planned Value* (PV), calculado pelo erro percentual médio absoluto (*Mean Absolute Percentage Error*), antes da execução do projeto fornece aos gestores informações preditivas confiáveis no desempenho do *Earned Value* (EV) e *Actual Cost Value* (ACV), permitindo uma ação proativa efetiva para garantir resultados de desempenho favoráveis. Chen (2021) e Hazır (2015) adotaram como métrica variação de desempenho na fase de planejamento que explica uma parte substancial da dificuldade do projeto na conclusão, mediante a um modelo de classificação de problemas de projetos, com o objetivo de identificar problemas nos projetos que provavelmente falharão antes de eles começarem. No mesmo sentido, Sharma, Caldas e Mulva (2021) utilizaram o EVM nas áreas de custo, cronograma, segurança, retrabalho e mudanças no projeto com o propósito de apoiar à tomada de decisão, facilitar o orçamento de capital, o impacto das condições de mercado e na melhoria contínua e melhores práticas de gestão.

Nessa mesma linha de métrica, Yousefi *et al.* (2019) apresentam o controle estatístico em índices do *Earned Duration Management* (EDM) para melhor investigar as variações do desempenho do cronograma do projeto ou para controlar projetos de inovação radical. Ainda em relação ao cronograma, Han *et al.* (2017) destacam o uso de métricas de qualidade de cronogramas baseados no limite de duração máxima (30 dias); Proporção de tarefas detalhadas para marcos; Porcentagem de tarefas no caminho crítico; Número de defasagens; Tipo de

relacionamento (terminar para começar); e Marcos sem predecessor ou sucessor. Wauters e Vanhoucke (2015) criaram um experimento computacional para avaliar a estabilidade usando a nova métrica de defasagem média proposta usando o *Coefficient of Variation* (CV) do EVM. Grau e Back (2015), no mesmo contexto, destacam que as métricas de controle do cronograma devem apoiar o gestor a ser proativo, em vez de reativo, no reconhecimento de tendências e eventos melhora significativamente a capacidade da equipe de minimizar a diferença entre os resultados previstos e reais. Chen (2014) combinou séries temporais e análise de regressão para desenvolver PV em um modelo de resposta EV para prever *Work Perform* (WP) de EV. Usando a relação entre *Budgeted Cost* (BC) e o WP estimado representado no modelo de resposta EV, Chen (2014) desenvolveu ainda EV em um modelo ACV para prever *Actual Cost* (AC) de ACV.

Em relação a duração do projeto, Batselier e Vanhoucke (2015) adotaram duas técnicas baseadas no EVM que integram, respectivamente, retrabalho e sensibilidade à atividade no tempo de EVM prevendo as extensões do projeto. Lopez-Paredes (2011) introduziram duas novas métricas denominadas por Índice de Controle de Custos (ICC1) e Índice de Controle de Cronograma (ICC2). Ambos os índices compararam as medidas de EVM com os valores máximos que o projeto deveria apresentar se o projeto estivesse rodando sob uma hipótese de análise de risco que afetam o risco, custo e cronograma, o que pode ser aplicado nas fases do ciclo de vida do projeto (Chen, 2018).

Sato e Hirao (2013) contribuíram para estabelecer o conceito de orçamento ótimo e esclarecer essa condição e a teoria permite a análise do equilíbrio custo/risco não apenas no nível do projeto, mas também no nível da rede de atividades. Kim, Kim e Cho (2016) estudaram um processo com relação linear entre atividades em que as saídas de um negócio precedente se tornam diretamente os insumos de um sucessor. Narbaev e De Marco (2014) calcularam o *Cost Estimate At Completion* (CEAC) que é um número considerado como novidade de pesquisa e que contribui para a extensão do corpo de conhecimento EVM. Narbaev e De Marco (2014) apresentaram uma nova abordagem metodológica que foi desenvolvida para fornecer previsões CEAC mais precisas, confiáveis e eficazes para os estágios iniciais e intermediários da execução do projeto de construção.

Em relação ao resultado dos projetos, Aljedaibi e Khamis (2019) desenvolveram o *Critical Success Factors* (CSF) é um método para medir, monitorar e controlar fatores críticos de sucesso de sistemas de *software* grande, por meio de escala de metas, custos, infraestrutura de TI, gestão de projetos, entre outros. Rajagopalan e Srivastava (2018) adotaram um método único usando uma variável chamada de *Project Health Index*, capaz de medir o sucesso, em contraste com a visão tradicional de tempo e custo como único critério, onde o resultado da variável pode ser usado como monitor de progresso com base no *Corrective And Preventive Actions*. Ibrahim, Hanna e Kievet (2020) examinaram as diferenças de desempenho tempo de processamento de pedido de mudança, aumento no prazo do cronograma, aumento de custo, mudança percentual do projeto, qualidade geral dos sistemas do projeto, mudança percentual relacionada à qualidade/valor, mudança percentual relacionada ao *design*.

Durante as fases de iniciação e planejamento do projeto, métricas da abordagem tradicional são incorporadas para prever o uso de recursos humanos e outras necessidades de execução de atividades para apoio ao controle e o monitoramento de projetos (PMI, 2017). Assim, Song, Martens e Vanhoucke (2021) apresentaram a métrica de *Using Schedule Risk Analysis* (SRA) com a restrição de recursos (*Resource Constrained - RC*) para melhorar as ações corretivas e o desempenho do projeto sob o uso de recursos restritos. Pozzana *et al.* (2021) apresentaram um conjunto de dados composto por projetos de engenharia descritos por suas redes de atividades, registros do desempenho e se a atividade foi entregue no prazo.

Em se tratando do envolvimento das métricas com os envolvidos nos projetos, Rajablu *et al.* (2017) desenvolveram o *Stakeholder Metrics-integrated Management Model* que integra as partes interessadas que permite gerenciá-las identificando, comunicando, engajando, capacitando e atendendo as partes interessadas com métricas e ferramentas. Além disso, Hanna, Tadt e Whited (2012) utilizaram análise de *bootstrap* para avaliar o desempenho de um projeto de infraestrutura. Hanna (2016) avaliaram com métrica o desempenho geral mais alto no que se refere à implementação de técnicas de construção enxuta, equipe de liderança de projeto engajada e capacitada e um alto envolvimento de *stakeholders* em todo o projeto.

Algumas métricas possuem foco no desenvolvimento do produto. Fleche *et al.* (2017) implantaram métricas quantitativas e não intrusivas para os *designers*, onde cada métrica utiliza variáveis do *design* do produto planejado. Spijkman *et al.* (2021) propuseram o modelo *Requirements Engineering for Software Architecture* (RE4SA) que fornece uma conexão entre os artefatos e que facilita a comunicação dentro da equipe de desenvolvimento. Além disso, Spijkman *et al.* (2021) forneceram métricas para quantificar o alinhamento entre requisitos e arquitetura para detectar sinais de granularidade, concentrando-se em valores discrepantes em um conjunto de dados. Jethani (2013)

apresentou o esforço no nível da tarefa e monitoramento do progresso; Monitoramento de marcos e cronogramas; e Monitoramento de defeitos, com base nas atividades regulares de planejamento e monitoramento de projetos de desenvolvimento de *software*. A visão é estendida por Iqbal, Naeem e Khan (2012), que completaram que a rastreabilidade avalia os vínculos entre os requisitos dentro de um documento e métricas de integridade de requisitos, que verificam se os requisitos especificados estão completos ou não para o desenvolvimento do produto.

Em ambiente de escritório de projetos, Geng *et al.* (2018) integraram o aprendizado organizacional e o desenvolvimento do conhecimento com um processo de seleção de projetos, propondo uma estrutura para avaliar essa integração baseada no *Analytic Hierarchy Process*, ou por meio de uma avaliação de eficiência a ser realizada por meio da atualização da lista de análise de projetos (Ko & Cho, 2020). Por fim, a Tabela 4 trás uma breve análise de objetivos, restrições e relacionamentos entre as métricas de Projetos Tradicionais para auxiliar no entendimento de cada métrica e a sua contribuição à gestão tradicional.

Tabela 4 – Métricas de Projetos Tradicionais, objetivos, restrições e relacionamentos

Métrica	Objetivo	Restrição de uso	Métrica Relacionada
<i>Earned Value Management (EVM)</i>	Medir o progresso e desempenho de projetos combinando as medidas de escopo, tempo e custos.	O escopo é fixo e mantém o tempo e custos estimados.	-
<i>Earned Value Management/ Earned Schedule (EVM/ES)</i>	Medir possíveis riscos de maneira preventiva sobre o cronograma do projeto.	Restringe-se aos riscos relativos ao prazo.	EVM
<i>Earned Duration Management (EDM)</i>	Medir as variações do desempenho do cronograma do projeto ou para controlar projetos de inovação radical.	Restringe-se a medir as variações (não as causas) no desempenho do projeto.	EVM, EVM/ES
<i>Planned Value (PV)</i>	Medir o custo orçado para o trabalho programado para ser feito.	Mantém o foco no custo orçado para apenas ser utilizado como <i>baseline</i> para avaliar o desempenho do custo realizado.	EVM
<i>Actual Cost (AC)</i>	Medir o custo realizado pelo trabalho realizado.	Mantém o foco no custo realizado e não avalia os motivos pelo desempenho.	EVM
<i>Earned Value (EV)</i>	Medir o valor agregado com base no escopo, prazo e custo.	Se restringe a avaliar o desempenho do projeto com base no escopo, prazo e custo.	EVM
<i>Coefficient of Variation (CV)</i>	Experimento computacional para avaliar a estabilidade usando a nova métrica de defasagem.	Restringe-se a medir as variações (não as causas) no desempenho do projeto.	EVM
<i>Work Perform (WP)</i>	Medir a combinação de séries temporais e análise de regressão para desenvolver PV em um modelo de resposta EV para prever WP.	Restringe-se em avaliar o desempenho do cronograma.	PV/EV
<i>Budgeted Cost (BC)</i>	Medir o custo orçado estimado.	Restringe-se em avaliar o desempenho de custos.	EV/WP
<i>Actual Cost Value (ACV)</i>	Medir a relação entre BC e o WP estimado representado no modelo de resposta EV, para prever o AC de ACV	Restringe-se em avaliar o desempenho de custos.	AC/EV
Índice de Controle de Custos (ICC1)	Compara as medidas de EVM com os valores máximos que o projeto deveria apresentar se o projeto estivesse rodando sob uma hipótese de análise de risco que afetam o cronograma do projeto.	Com base em medições do EVM, restringe-se e avaliar o desempenho de custos.	EVM
Índice de Controle de Cronograma (ICC2)	Compara as medidas de EVM com os valores máximos que o projeto deveria apresentar se o projeto estivesse rodando sob uma hipótese de análise de risco que afetam o custo do projeto.	Com base em medições do EVM, restringe-se e avaliar o desempenho do cronograma.	EVM
<i>Cost Estimate at Completion (CEAC)</i>	Medir a estimativa de custo na conclusão que é um número que contribui para a extensão do corpo de conhecimento EVM.	Com base em medições do EVM, restringe-se e avaliar o desempenho de custos.	EVM
<i>Critical Success Factors (CSF)</i>	Medir, monitorar e controlar fatores críticos de sucesso por meio de escala de	Restringe-se em avaliar uma série de variáveis para determinar o	-

	metas, custos, infraestrutura de TI e gestão de projetos.	sucesso do projeto.	
<i>Project Health Index (PHI)</i>	Medir o sucesso, em contraste com a visão tradicional de tempo e custo como único critério, onde o resultado da variável pode ser usado como monitor de progresso com base em ações de prevenções e correções.	Está restrito em medir a saúde como progresso do projeto com base em ações de prevenções e correções.	-
<i>Using Schedule Risk Analysis (SRA)</i>	Utiliza a restrição de recursos para medir e melhorar as ações corretivas e o desempenho do projeto sob o uso de recursos restritos.	Restrito em medir o desempenho do projeto sob a ótica de recursos restritos.	-
<i>Stakeholder Metrics-integrated Management Model (SMMM)</i>	Integra as partes interessadas com métricas de identificação, comunicação, engajamento, capacitação e atendimento.	Restringe-se em medir o nível de integração dos <i>stakeholders</i> com o projeto.	-
<i>Requirements Engineering for Software Architecture (RE4SA)</i>	Medir a conexão entre os artefatos do projeto e que facilita a comunicação dentro da equipe de desenvolvimento de <i>software</i> .	Restringe-se em medir o relacionamento entre artefatos do projeto.	-

Fonte: Elaborado pelo autor

Os artigos dessa categoria apresentaram discussões importantes sobre as principais métricas utilizadas pelas organizações para o controle e monitoramento de projetos tradicionais. A partir dessa perspectiva, as métricas como EVM, EV, AC, PV e ES são as mais citadas na literatura e nota-se a introdução de novas métricas como EDM e métricas com *benchmarks* associados para o processo de requisição de informações em projetos.

2.2.2 Métricas Ágeis de Fluxo

As métricas ágeis de fluxo visam medir o desempenho de projetos ágeis durante o fluxo de desenvolvimento de software ou de produto (Jyothi e Rao, 2017; dos Santos et al., 2018). Assim, as métricas ágeis de fluxo procuram medir desde o ponto de comprometimento (*Sprint Backlog*) até a entrega da funcionalidade para os clientes/*stakeholder*. As métricas comumente utilizadas nesse contexto são o *Lead Time*, *WIP*, *Cycle Time*, *Work Item Age* entre outras que procuram medir as etapas de desenvolvimento de *software*.

Inserido no *Scrum*, Parrend et al. (2014) avaliaram o impacto das práticas de gerenciamento de projetos nos dois primeiros *Sprints*, medindo as práticas de elegibilidade do incremento com impacto mensurável no desempenho das equipes e nível de exigência que levam ao fracasso se não forem implementadas. Parrend et al. (2014) destacaram a importância dos princípios do *Scrum* e as propriedades *swarming* são práticas relevantes de sucesso e um sintoma de gerenciamento de desempenho. O *swarming* é uma técnica e uma dinâmica de gestão que consiste em empregar a inteligência coletiva da equipe na execução de uma atividade crítica para resultados mais eficazes do projeto (Parrend et al., 2014).

Dennehy e Conboy (2018) realizaram a avaliação por meio de um estudo empírico do uso de fluxo de trabalho com *Kanban*, obtendo dois resultados. Primeiro, um conjunto de desafios empíricos, tais como a mudança organizacional e a configuração de quadros *Kanban*. Segundo, identificaram que as métricas de *Lead Time* e *Velocity* são as mais utilizadas para o planejamento e nos processos de melhoria contínua no projeto (Dennehy & Conboy, 2018).

Os autores Mas, Mesquida e Pacheco (2020) realizaram um estudo para saber se empresa é ágil e madura, e se um conjunto de métricas de desempenho estiver em conformidade com as melhores práticas de gerenciamento de projetos. Um conjunto de métricas foram identificadas nas empresas ágeis estudadas, tais como *efficiency flow*, *lead time*, *cycle time*, *reopens* e *tasks finished flow* (Mesquida & Pacheco, 2020). A métrica *efficiency flow* mede quantos itens do *backlog* são entregues por ciclo (Mesquida & Pacheco, 2020). O *lead time* mede o tempo em que um item do *backlog* é iniciado até o momento que é entregue ao cliente (Mesquida & Pacheco, 2020). A métrica *reopens* mede o número de tarefas ou incidentes reabertos e a métrica *tasks finished flow* mede as tarefas finalizadas no fluxo de desenvolvimento, diferentemente do *Throughput*, que mede funcionalidades entregues ou histórias de usuário. (Mesquida & Pacheco, 2020).

Grimaldi et al. (2016) apresentaram como uma estrutura ágil escalável, baseada em SAFe 3.0 e mostrou como implementar e medir o desempenho considerando que as funções e responsabilidades são divididas entre a organização e o fornecedor. Grimaldi et al. (2016) utilizaram métricas como eficácia e eficiência, custo-hora real

e planejada, impedimentos e custo do atraso durante as rodadas de *Sprint*. No mesmo sentido que Ahmed et al. (2017) discutiram sobre as métricas *Sprint Burn Down*, *Epic Burn Down Velocity Chart*, *Control Chart* e a maneira como muitos dos cálculos de métricas são fortemente acoplados aos pontos de história para itens como o Épico ou *Sprint*.

Sajedi-Badashian e Stroulia (2020) mostraram que é possível melhorar e medir a eficácia do processo de atribuição de *bugs* por meio na análise de dados do projeto. Desse modo, Sajedi-Badashian e Stroulia (2020) encontraram uma maneira de atribuir a pessoa mais indicada para a resolução do problema e reduzir o *time to market* do produto.

Do ponto de vista de Petersen e Wohlin (2011), o *Lean* fornece ferramentas de análise e melhoria com foco no ciclo de vida do desenvolvimento, enquanto o ágil se concentra em soluções e prescreve conjuntos de práticas para obter agilidade. Petersen e Wohlin (2011) sugeriram análises sobre as métricas como reduzir o *lead time* para obter alta capacidade de resposta às necessidades dos clientes e fornecer um sistema de rastreamento que mostre o status do desenvolvimento de *software*. No sentido do monitoramento, as métricas de processo propostas por Choraś et al. (2020) foram utilizadas continuamente para ajudar a garantir a qualidade e estabilidade do *software* como a estimativa de tarefas e rastreamento tarefas em tempo real. Meidan et al. (2018) identificaram que o desenvolvimento ágil e empresa de pequeno/médio porte foram os contextos de pesquisa mais frequentemente identificados. Meidan et al. (2018) demonstraram que a capacidade, desempenho e maturidade foram os atributos de processo mais medidos, enquanto esforço, desempenho e risco foram os atributos de projeto mais medidos.

O modelo de ciclo de vida de desenvolvimento e manutenção de sistemas proposto por Sadler (2020) aplicou efetivamente diferentes níveis de rigor e métricas de qualidade mais rígidas, dependendo do risco associado aos lançamentos. O modelo de Sadler (2020) foi baseado nas normas relacionadas ao gerenciamento de riscos denominado ER²C SDMLC (*Enterprise Release Risk-Centric Systems Development and Maintenance Life Cycle*). Dentre as métricas avaliadas por Sadler (2020), tivemos a Clareza do Artefato e Envolvimento, Propriedade do Cliente e Aplicabilidade e Rastreabilidade.

Asha e Mani (2018) apresentaram uma metodologia de teste denominada por Teste Simples de Teste de Aceitação aplicando as técnicas de Desenvolvimento Orientado a Testes de Aceitação que podem ser executados como casos de teste de *software* no contexto do DevOps. Além disso, Asha e Mani (2018) utilizaram os algoritmos Tarjan e Breadth-First Search que renderizam 100% de cobertura dos cenários e histórias de usuário que auxiliaram no desempenho do processo de teste aplicando as métricas de componentes funcionais, aumentando a frequência de entrega contínua e reduzindo o *time to market*.

A Tabela 5 a seguir realiza uma breve análise de objetivos, restrições e relacionamentos entre as métricas de Projetos Ágeis de Fluxo para auxiliar no entendimento de cada métrica e a contribuição de cada uma delas à gestão ágil de projetos.

Tabela 5 – Métricas Ágeis de Fluxo, objetivos, restrições e relacionamentos

Métrica	Objetivo	Restrição	Métrica Relacionada
Elegibilidade do Incremento	Avalia o impacto das práticas de gerenciamento de projetos nos dois primeiros <i>Sprints</i> .	Com foco na avaliação entre escopo e incremento, não avalia a agregação de valor do incremento.	-
<i>Lead Time</i> (LT)	Mede a entrega a funcionalidade desde o comprometimento até a entrega ao cliente	Com foco em prazo, não avalia a agregação de valor da entrega da funcionalidade.	-
<i>Cycle Time</i> (CT)	Mede etapas de desenvolvimento do <i>software</i> dentro da iteração.	Com foco em desempenho de prazo, não avalia o valor da entrega.	-
<i>Velocity</i>	Mede quantos pontos de história foram feitos por <i>Sprint</i> .	Com foco em desempenho de esforço, não avalia o valor da entrega da HU.	-
<i>Efficiency Flow</i> (EF)	Mede quantos itens do <i>backlog</i> são entregues por ciclo.	Não avalia a agregação de valor da entrega da funcionalidade.	LT
<i>Reopens</i>	Mede o número de atividades reabertas.	Somente avalia o retrabalho e não mede as causas para aprendizado.	TFF
<i>Tasks Finished Flow</i> (TFF)	Mede as tarefas finalizadas dentro do fluxo de desenvolvimento.	Com foco em tarefas realizadas, não avalia a agregação de valor da entrega do item do <i>backlog</i> .	<i>Reopens</i>
Eficácia e eficiência	Eficácia e eficiência de entregas do projeto durante a rodada da <i>Sprint</i>	Necessário medir as entregas na perspectiva do cliente e/ou do mercado.	LT, TFF, <i>Velocity</i> , EF
Custo hora real e planejada de	Custo hora real e planejada de	Medida que contradiz a expectativa de	LT, EF

planejada	entregas do projeto durante a <i>Sprint</i> .	custos fixos em projetos ágeis.	
Impedimentos	Impedimentos mapeados durante a rodada da <i>Sprint</i> .	Somente avalia o retrabalho e não mede as causas para aprendizado.	-
Custo do atraso	Custo de atraso de entregas do projeto durante a rodada da <i>Sprint</i> .	Está concentrado em medir o custo do atraso e não mede as causas para reduzir riscos.	LT, TFF, <i>Velocity</i> , EF
<i>Sprint Burn Down (SBD)</i>	Métrica sobre pontos de história no <i>Sprint</i>	Com foco em desempenho de esforço, não avalia o valor da entrega da HU.	<i>Velocity</i> , LT
<i>Epic Burn Down (EBD)</i>	Métrica sobre pontos de história no Épico	Com foco em desempenho de esforço, não avalia a agregação de valor da entrega do épico.	<i>Velocity</i> , LT
<i>Control Chart (CC)</i>	Mede o <i>Cycle Time</i> (ou <i>Lead Time</i>) na <i>Sprint</i>	Com foco em prazo, não avalia a agregação de valor da entrega da funcionalidade.	LT, CT
Atribuição de bugs	Atribui a pessoa mais indicada para a resolução do problema e reduz o <i>time to market</i> do produto.	Apesar de auxiliar na redução do <i>time to market</i> , não mede o valor entregue.	TFF, <i>Reopens</i>
Reduzir <i>Lead Time</i>	Reduz o <i>lead time</i> para obter alta capacidade de resposta às necessidades dos clientes	Com foco em reduzir o prazo da entrega, não avalia a agregação de valor da entrega do item do backlog.	LT
Status Desenvolvimento	Fornece um rastreamento do status do desenvolvimento de <i>software</i> .	O status do desenvolvimento não garante que o escopo e/ou o valor será entregue.	CT, <i>Velocity</i> , SBD, CC
Estimativa de Tarefas (ET)	Estimativa de tarefas para auxiliar na qualidade e estabilidade do <i>software</i>	A estimativa de tarefas tem foco apenas em prazo.	-
Rastreamento de Tarefas (RT)	Rastreamento de tarefas para ajudar na qualidade e estabilidade do <i>software</i>	O rastreamento de tarefas tem foco apenas em status de tarefas.	AR
Capacidade do processo	Medir a capacidade do processo ágil.	Com restrição em medir a capacidade do processo de desenvolvimento, não avalia o produto.	LT, CT, EF, CC, SBD, EBD
Desempenho do processo	Medir o desempenho do processo ágil.	Com restrição em medir o desempenho do processo de desenvolvimento, não avalia o produto.	LT, CT, EF, CC, SBD, EBD
Maturidade do Processo	Medir a maturidade do processo ágil.	Com restrição em medir a maturidade do processo de desenvolvimento, não avalia o produto.	LT, CT, EF, CC, SBD, EBD
Esforço no projeto	Medir o esforço do projeto ágil.	O esforço no projeto tem foco apenas em prazo.	LT, CT, EF, CC, SBD, EBD, ET
Desempenho no projeto	Medir o desempenho do projeto ágil.	O desempenho no projeto tem foco apenas em custo e prazo.	LT, CT, EF, CC, SBD, EBD, ET
Risco no projeto	Medir os riscos do projeto ágil.	Medir riscos não garante sucesso ou entrega de valor pelo projeto.	-
Clareza do Artefato (CA)	Medir a clareza do artefato produzido pelo projeto.	Medir a clareza do artefato pode auxiliar na gestão do <i>backlog</i> , mas não garante entrega de valor.	EPC
Envolvimento e Propriedade do Cliente (EPC)	Medir o envolvimento e propriedade do cliente sobre o artefato produzido pelo projeto.	Medir o envolvimento e propriedade do cliente sobre o artefato auxilia na gestão do <i>backlog</i> . Não garante entrega de valor.	CA
Aplicabilidade e Rastreabilidade (AR)	Medir a aplicabilidade e rastreabilidade do artefato produzido pelo projeto.	Medir a aplicabilidade e rastreabilidade do artefato não garante entrega de valor.	RT
Componentes	Cobertura dos cenários e histórias	Medir a cobertura de cenários e histórias usuário auxilia na qualidade do <i>software</i> .	-

funcionais	de usuário que auxiliaram no desempenho do processo de teste.	mas não garante entrega de valor.	
------------	---	-----------------------------------	--

Fonte: Elaborado pelo autor

Os artigos dos autores apresentaram as métricas ágeis relacionadas ao fluxo operacional de projetos utilizando o *framework Scrum* e o *Kanban*. As métricas mais citadas nos estudos foram as de *Efficiency Flow*, *Lead Time*, *Cycle Time*, *Velocity*, *Sprint Burn Down*, *Epic Burn Down* e *Control Chart*. Verificou-se também que muitos dos cálculos de métricas são fortemente acoplados aos pontos de história como estimativa e monitoramento. Além disso, métricas ligadas aos itens de *backlog*, como *reopens*, *tasks finished flow* e *time spent extremes*, e outras métricas baseadas no SAFe 3.0 como eficácia, eficiência, custo-hora real e planejada, resíduos, impedimentos e custo do atraso foram encontradas.

2.2.3 Métricas Ágeis de Backlog do Produto

As métricas ágeis de *Backlog do Produto* visam medir o desempenho de projetos ágeis durante a etapa de definição do escopo do projeto que, em projetos ágeis, são organizados no *backlog* do produto (Mosser, Pulgar & Reinharz, 2022). O *backlog* do produto é composto pelo escopo do projeto dividido em diversos itens ou HUs que são descobertos e detalhados ao longo do desenvolvimento do produto do projeto (Mosser, Pulgar & Reinharz, 2022). Assim, as métricas ágeis de *Backlog do Produto* procuram medir o esforço de desenvolvimento da funcionalidade, nível de priorização e complexidade por *Use Case Points*, e estimativas por pontos de história.

Kayes, Sarker e Chakareski (2016) propuseram uma métrica no *Product Backlog Rating (PBR)* para avaliar o processo de teste no *Scrum* que forneceu uma estrutura adicional para o gerenciamento da qualidade e uma visão abrangente do processo de teste ao longo do ciclo de desenvolvimento do produto. O PBR de Kayes, Sarker e Chakareski (2016) considera a complexidade dos recursos a serem desenvolvidos em uma iteração do *Scrum*, avalia as classificações dos testes e oferece uma pontuação numérica do processo de teste. No mesmo contexto do processo *Scrum*, Zheng et al. (2021) mapearam as métricas de processo e do *Service-Oriented Architecture (SOA)* para auxiliar nas decisões de planejamento e *design* durante a *Sprint*. Zheng et al. (2021) sugeriram os *Key Performance Indicators (KPIs)* para ajudar os *stakeholders* a entender os imperativos da técnica proposta no planejamento de recursos, avaliação do desempenho da agilidade, qualidade e valor dos negócios, eficiência da equipe e complexidade de projetos baseados em *Scrum* e *SOA*.

Pavlova et al. (2021) determinaram os parâmetros do projeto como as características da entrega futura como o custo, prazo e possíveis desvios de prazo e custo. Tais parâmetros do estudo de Pavlova et al. (2021) resultaram em métricas para avaliar o grau de singularidade do projeto e o grau de complexidade do projeto. Para a métrica de complexidade, Pavlova et al. (2021) consideraram o número de variantes de soluções tecnológicas, o número de operações para cada variante e o número de alternativas para execução da operação para cada variante da tecnologia do produto do projeto. Wallace e Sheetz (2014) destacaram que a utilidade percebida de uma medida de software pode ser medida por uma avaliação da aplicabilidade da medida ao longo do ciclo de vida, dependência da linguagem de programação, capacidade de prescrever soluções e a validade da medida para definir a complexidade dos requisitos.

Azzeh e Nassif (2016) propuseram um modelo híbrido voltado para medir o esforço e a produtividade para apoiar o processo de estimativa de esforço inicial baseada no tamanho do *Use Case Point (UCP)*. As análises de Azzeh e Nassif (2016) contaram com um número significativo de projetos coletados que utilizaram como base o UCP em suas estimativas.

A ontologia desenvolvida por Prakash e Viswanathan (2018) acumulou conhecimento sobre os critérios de avaliação da qualidade das histórias de usuários (HU), sobre artefatos de requisitos e tipos de requisitos. Prakash e Viswanathan (2018) incluíram axiomas que determinam a qualidade da redação das HUs, a definição de prioridade e o risco das HUs, os relacionamentos de HU com os requisitos funcionais e não-funcionais e versões de produtos.

A Tabela 6 realiza uma breve análise de objetivos, restrições e relacionamentos entre as métricas de Projetos Ágeis de *Backlog* para auxiliar no entendimento de cada métrica e a contribuição de cada uma delas à gestão ágil de projetos.

Tabela 6 – Métricas Ágeis de *Backlog*, objetivos, restrições e relacionamentos

Métrica	Objetivo	Restrição	Métrica Relacionada
<i>Product Backlog Rating</i> (PBR)	Mede o nível de prioridade do item do <i>backlog</i>	Medir o nível de priorização dos itens do <i>backlog</i> não garante a priorização no sentido de valor entregue.	-
KPI de Imperativos da Proposta Técnica (IPT)	Auxilia no planejamento de recursos, testar o desempenho da agilidade, valor dos negócios, eficiência da equipe e complexidade de projeto.	Falta de foco na avaliação do KPI com abrangência em diversas fases do projeto num único KPI.	-
Grau de singularidade do projeto	Características da entrega futura como o custo, prazo, possíveis desvios de prazo e custo.	Não faz sentido medir a singularidade de projetos ágeis por custo e prazo (fixos).	IPT, GCP
Grau de Complexidade do Projeto (GCP)	Número de variantes de soluções tecnológicas, o número de operações para cada variante e o número de alternativas para execução da operação para cada variante.	Avalia somente questões tecnológicas para verificar a complexidade sem considerar o mercado e escopo para medir a complexidade.	IPT, GCP
Esforço de desenvolvimento de <i>software</i> (EDS)	Avaliação da aplicabilidade ao longo do ciclo de vida, dependência da linguagem de programação específica, capacidade de prescrever soluções.	Falta de foco na avaliação da métrica com abrangência em diversas fases do projeto numa única métrica.	IPT, GCP
Esforço e produtividade	Apoiar a estimativa de esforço inicial baseada no tamanho do <i>Use Case Point</i> (UCP)	Usar UCP como métrica de planejamento versus realizado não garante a entrega o valor esperado.	IPT, GCP, EDS
Avaliação da qualidade das HUs	Avalia a qualidade das HUs, prioridade e o risco das HUs, relacionamentos de HU com fontes de requisitos, requisitos não-funcionais e versões de produtos.	Não realiza a medição das HUs que entregaram o valor esperado.	PBR, GCP

Fonte: Elaborado pelo autor

As métricas de *Backlog* de Produto que foram apresentadas nos artigos, trazem métricas importantes para a gestão de requisitos em projetos ágeis. Assim, destacam-se as métricas como o *Product Backlog Rating*, qualidade e valor dos negócios, complexidade de projetos baseados em Scrum e SOA, complexidade do requisito, complexidade da arquitetura de solução e complexidade da mudança. Outros achados tivemos com a estimativa de esforço inicial baseada no tamanho do UCP, avaliação da qualidade das histórias de usuários (HU), sobre artefatos de requisitos e tipos de requisitos e a medição do tamanho do requisito a partir do comportamento criado pelo *Behavior Driven Development* (BDD) usando a notação de Gherkin (Moult & Krijnen, 2020).

2.2.4 Métricas Ágeis de Produto

As métricas ágeis de produto visam medir a qualidade de construção e de ações para produzir um produto de melhor qualidade e para reduzir o seu *time to market* (Kristensen & Shafiee, 2019; Ries, 2011). Dessa forma, as métricas ágeis de produto nessa sessão medem tanto atividades para melhorar a qualidade do código-fonte, melhoria do processo de testes de *software*, a gestão e rastreabilidade de *bugs* e o processo de desenvolvimento orientado a testes. Além disso, a categoria buscou apresentar as métricas para medir o nível de satisfação do cliente/*stakeholder* com o produto recebido.

Mäkiahio, Vartiainen e Poranen (2022) apresentaram uma ferramenta de monitoramento de métricas que foi desenvolvida por alunos de graduação e pós-graduação para apoiar os membros do projeto, gerentes de projeto e gestores. Segundo Mäkiahio, Vartiainen e Poranen (2022) a ferramenta permitiu acompanhar a quantidade de *commits*, horas trabalhadas e se os requisitos permaneciam muito tempo em estágios no processo de desenvolvimento.

Os experimentos de Chang (2015) foram usados para detectar itens de risco de software a partir do *cluster* de dados oriundos de projetos. Tais dados foram apresentados por Chang (2015) no formato de métricas com os riscos foram mitigados e fornece informações sobre modelos de risco que podem ser usados para entender as características dos riscos de *software*. Dentre as métricas apresentadas por Mascarenhas et al. (2014), temos o crescimento nas vendas de novos produtos, o orçamento dedicado à análise ou verificação do cliente, projetos orientados para o cliente, projetos e programas respeitando custos e orçamento, marcos respeitados e componentes reutilizáveis.

A pesquisa de Singh et al. (2019) propuseram análises de complexidade de programas orientados a objetos em Java com base na métrica *Chidamber-Kemerer* (CK) e o desempenho são medidos com uso da métrica *Mean Time to Execution*. Como resultado indicado por Singh et al. (2019), realizar reengenharia usando práticas ágeis é benéfico em termos de implementação de requisitos, estimativa de custo e melhoria de desempenho. Nesse sentido, Isong e Obeten (2013) realizaram uma validação empírica das métricas CK e *Source Lines of Code* (SLOC). A validação empírica de Isong e Obeten (2013) das métricas CK e SLOC e os modelos que as utilizaram em três aspectos: (1) nível de significância e insignificância com propensão a falhas (PF), (2) estado de validação e (3) utilidade na qualidade de software. Segundo Isong e Obeten (2013), as métricas SLOC, *Coupling between Objects* (CBO), *Response for a Class* (RFC), *Weighted Methods per Class* (WMC) possuem forte relação com PF e são consideradas os melhores preditores de PF com impacto na qualidade de *software*.

Nesse sentido que Concas et al. (2012) também estudou diversas iniciativas em projetos ágeis que utilizaram as métricas CK, SLOC, CBO, RFC e WMC para medir a qualidade de *software* em diversos projetos. Ainda sobre a qualidade de código-fonte, Yamashita e Counsell (2013) investigaram o potencial de *code smells* (potenciais problemas no código) para refletir indicadores de manutenibilidade em nível de *software*. Ao comparar os *code smells* com outras abordagens, Yamashita e Counsell (2013) descobriram que as medidas estruturais forneceram mais informações sobre qual sistema tinha o *design* equilibrado, mas essa medida ignorou quando as tarefas de manutenção eram de tamanho pequeno ou médio. O estudo de Athanasiou et al. (2014) apresentou um modelo que avalia a qualidade do código de teste combinando métricas de código-fonte que refletem três aspectos principais da qualidade do código de teste: integridade, eficácia e manutenibilidade. Athanasiou et al. (2014) encontraram uma correlação positiva e estatisticamente significativa entre a qualidade do código de teste, rendimento e produtividade. As contribuições do estudo de Athanasiou et al. (2014) iniciaram a partir do modelo que combina métricas para fornecer uma medida da qualidade do código de teste e a investigação empírica que demonstrou uma correlação positiva significativa entre a qualidade do código de teste e na resolução de *bugs*.

Sobre a experiência do usuário do produto, de Day et al. (2019) utilizaram o *Net Promoter Score* (NPS) para avaliar a viabilidade de um sistema de apoio à decisão no Registro Eletrônico de Saúde (RES). O NPS foi avaliado como resultado de esforços em testes de usabilidade e a implementação em larga escala com efeito positivo no paciente e na tomada de decisão compartilhada (Day et al., 2019). Simpson et al. (2021) procurou reduzir a carga do RES com um *Sprint* de otimização de duas semanas que trabalhou com 19 provedores de prática avançada em uma unidade de especialidade. Simpson et al. (2021) mediram o NPS o tempo gasto no RES com base nos dados de log do usuário, onde o NPS aumentou de 6 para 60 e para o resultado da *Sprint* de duas semanas o NPS foi de 93.

Üsfekes et al. (2019) investigaram por meio de uma revisão da literatura a utilidade de mecanismos de incentivo para rastreamento eficiente de *bugs* no *Application Lifecycle Management*. Para Üsfekes et al. (2019), isso colabora positivamente para os resultados de métricas que olham para a quantidade de *bugs* resolvidos por unidade de tempo. Abdelrahman Aljemabi et al. (2020) mostraram que a *Bug Tracking System-based*, exibem três diferentes níveis de coordenação (Plano, Consciente e Reflexivo) com atividades de colaboração e na métrica de tempo médio para reparo para avaliar o tempo de resolução de *bugs*.

Costa, Vasconcelos e Fragoso (2020) destacaram as etapas de identificar as métricas aplicáveis à arquitetura dos projetos e para quantificar o nível de alinhamento entre os projetos e os objetivos estratégicos. As métricas foram propostas por Costa, Vasconcelos e Fragoso, B. (2020) foram organizadas em métricas de Funcionalidade, Confiabilidade, Eficiência, Manutenibilidade, Portabilidade e Alinhamento.

Ibban et al. (2018) entenderam as atividades de engenharia de software relacionadas a Ofertas Iniciais de Moedas (*Initial Coin Offering* – ICO) que é uma forma inovadora de financiar projetos baseados em *blockchain*. Além disso, Ibban et al. (2018) buscaram fazer uma comparação entre ICOs e ICOs ágeis. Cerca de 5% das ICOs examinadas por Ibban et al. (2018) aplicam práticas ágeis e analisaram as ICOs ágeis em termos de métricas de código-fonte, versões de idioma e uso de ferramentas de teste. Nidagundi e Novickis (2016) descobriram

possibilidades de adoção dos modelos e métricas de transformação *Lean* no plano de teste de software para simplificar o processo de teste para uso posterior dessas métricas de teste em Canvas. Com o Canvas, Nidagundi e Novickis (2016) afirmaram que as métricas de teste podem nos dar *feedback* para o processo de teste de *software*. Ainda no tema de qualidade, Shawky e Abd-El-Hafiz (2016) apresentaram uma abordagem que estuda o efeito do processo de desenvolvimento na qualidade *software* e verificaram se existe relação entre o tipo de método de desenvolvimento utilizado e os valores de algumas métricas indicadoras de qualidade. Shawky e Abd-El-Hafiz (2016) afirmaram que as métricas usadas incluem métricas de complexidade, acoplamento e coesão, sendo as duas últimas sendo utilizadas para medir a complexidade. Ainda sobre processo de software, Tahir, Rasool e Noman (2018) identificaram 29 modelos de medição de software e 4 ferramentas projetadas para gerentes de projeto em pequenas e médias empresas. Assim, o estudo de Tahir, Rasool e Noman (2018) apresentou que 51% dos modelos de medição são referentes a melhoria de processo de software, 60% sobre uso de especialistas em medição e experiência, 40% sobre o uso de padrões de medição e 22% sobre o uso de ferramentas automatizadas.

Tanveer et al. (2019) desenvolveram um método híbrido que utiliza informações de análise de impacto de mudança na estimativa de esforço e do modelo de estimação baseado em árvores impulsionadas por gradiente (AIG). A avaliação subjetiva de Tanveer et al. (2019) sobre o *Hybrid Effort Estimation in Agile Software Development* (HyEEASe) e do *Planning Poker* (PK) foi sobre medir a facilidade de uso percebida, utilidade e aprendizagem. Tanveer et al. (2019) avaliaram que o PK obteve classificações superiores ao HyEEASe em facilidade de uso percebida e capacidade de aprendizado, entretanto, os avaliados consideraram que HyEEASe requer menos esforço mental. Em relação à métrica de precisão da estimativa, Tanveer et al. (2019) concluíram uma estimativa puramente baseada no modelo AIG não teve um desempenho melhor do que uma estimativa baseada no HyEEASe.

Pradhan e Nanniyur (2021) apresentaram um estudo de caso sobre a transformação em larga escala de um sistema de gestão de qualidade legado para um sistema moderno com aprendizado de máquina, dados conectados, operações integradas e *Big Data*. Pradhan e Nanniyur (2021) identificaram que as métricas de qualidade precisam abranger a *Sprint* ou Projeto, Lançamentos, Portfólio e Visão Empresarial. Pradhan e Nanniyur (2021) validaram a eficácia da transformação nos perfis de defeitos recebidos, taxas de falhas e cronograma de maturidade de lançamento. Kamulegeya, Mugwanya e Hebig (2018) buscaram entender o uso e os benefícios percebidos da medição em *startups* de *software* na África Oriental e identificaram que as *start-ups* estão usando uma série de métricas orientadas a negócios e produtos e as *startups* mais antigas utilizam as métricas orientadas a projetos e as métricas organizacionais são as métricas menos utilizadas.

Bakhtiary, Gandomani e Salajegheh (2020) mediram a aplicação das abordagens de *Test-Driven Development* (TDD) e Não-TDD (NTDD) para compará-los, dada a qualidade do resultado do trabalho e a produtividade dos desenvolvedores em uma empresa de *software*. Bakhtiary, Gandomani e Salajegheh (2020) mostraram que ao aumentar a escala do projeto, o número de defeitos descobertos antes e após o lançamento aumentou mais na abordagem NTDD em relação ao TDD. Bakhtiary, Gandomani e Salajegheh (2020) observaram que houve uma redução na satisfação do cliente no NTDD em comparação com a abordagem TDD em cada ciclo, e teve desempenho melhor. Outra análise de Bakhtiary, Gandomani e Salajegheh (2020) e Antinyan (2014) trouxeram que medir a complexidade ciclomática que corresponde o nível de complexidade de um programa de computador e resultou que a abordagem TDD reduziu a complexidade no código em comparação com a abordagem NTDD. No mesmo sentido da produtividade, Hernández, et al. (2020) estabeleceu o nível de relevância das métricas de produtividade de equipes de desenvolvimento ágil de *software*. Para atingir o objetivo, o nível de relevância das métricas de produtividade da equipe foi identificado por Hernández et al. (2020) em três aspectos fundamentais: (i) entrega antecipada e frequente de software, (ii) agregação de valor e (iii) aspectos relacionados ao esforço. Hernández et al. (2020) sugeriram que é relevante para os profissionais medirem a produtividade entregando *software* com antecedência e frequência com alta relevância o esforço de implantação de *software* que resolva os problemas de clientes. Hernández et al. (2020) afirmaram que o cálculo sistemático do retorno do investimento é visto como uma forma relevante de medir a agregação de valor.

Com o enfoque no problema de seleção de atributos no contexto da estimativa de qualidade de *software*, o estudo de Gao et al. (2011) investigou comparativa é apresentada para avaliar nossa proposta de abordagem híbrida de seleção de atributos. Dado esse contexto, Gao et al. (2011) destacaram que uma seleção inteligente de métricas de *software* antes de treinar um modelo de previsão de defeitos provavelmente melhorará o resultado, removendo recursos redundantes e menos importantes.

Perkusich et al. (2017) sugeriram que o modelo com Redes Bayesianas e que abordasse a avaliação da qualidade de processo e de produto a partir de métricas de qualidade do incremento, dos eventos do *Scrum* e do código-fonte. Assim, Perkusich et al. (2017) identificaram nos nós uma oportunidade para analisar a qualidade do trabalho da equipe do projeto e avaliar as chances de sucesso do projeto.

Savola, Frühwirth e Pietikäinen (2012) demonstraram o potencial prático das métricas de segurança orientadas a riscos, particularmente ao oferecer alguma visibilidade inicial da eficácia e eficiência da segurança. A principal contribuição do estudo de Savola, Frühwirth e Pietikäinen (2012) está na avaliação do potencial e significado de uma metodologia de desenvolvimento de métricas de segurança hierárquicas orientadas a risco no contexto do desenvolvimento ágil de *software*. Assim, a Tabela 7 apresenta uma breve análise de objetivos, restrições e relacionamentos entre as métricas de Projetos Ágeis de Produto para auxiliar no entendimento de cada métrica e a sua contribuição na gestão ágil de projetos.

Tabela 7 – Métricas Ágeis de Produto, objetivos, restrições e relacionamentos

Métrica	Objetivo	Restrição	Métrica Relacionada
Quantidade de <i>Commits</i> (QC)	Quantidade de novos códigos-fonte publicados.	Avaliar código-fonte publicados não é uma avaliação de produtividade.	-
Horas Trabalhadas (HT)	Quantidade de horas trabalhadas durante um período.	Não avalia horas produzidas que é diferente de horas trabalhadas.	TED
Tempo nos Estágios em Dev (TED)	Tempo do requisito nos estágios de desenvolvimento.	Como o <i>Work Item Age</i> do <i>Kanban</i> , restringe-se na medição dos estágios para apontar problemas pontuais.	HT
Características dos riscos de software	Fornece informações sobre modelos de risco que podem ser usados para entender as características dos riscos de <i>software</i> .	Modelos de risco certamente não avaliam hipóteses de negócio para avaliar riscos de <i>software</i> .	-
Crescimento nas vendas de novos produtos (CVNP)	Mede o aumento nas vendas de novos produtos.	Apesar da boa prática da métrica, restringe-se a venda e não avalia as partes do produto que os clientes mais gostam.	POC, ODVC, NPS
Orçamento Dedicado à Verificação do Cliente (ODVC)	Mede os orçamentos com dedicação de análise e verificação do cliente.	Restringe-se em validar o orçamento junto ao cliente. Caso o escopo mude (que é bem provável que ocorra), o orçamento teria que ser revalidado.	POC, CVNP, NPS
Projetos Orientados para o Cliente (POC)	Mede o número de projetos criados e orientados para o cliente.	Dado que todos os projetos deveriam ser orientados ao cliente, não existe muita aplicabilidade dessa métrica.	ODVC, NPS
Projetos respeitando custos e orçamento (PRCO)	Mede o número de projetos e programas respeitando custos e orçamento.	Em projetos ágeis o custo é fixo. Essa medida restringe a avaliação da relação de valor entregue versus custos realizado.	TED, POC, ODVC, HT, MR
Marcos Respeitados (MR)	Mede o número de marcos respeitados e entregues.	Marcos ou épicos, restringe-se em avaliar a entrega sem avaliar também os resultados do <i>feedback</i> dos clientes.	POC, ODVC
Componentes Reutilizáveis (CR)	Mede o número de componentes de software reutilizáveis dentro ou entre projetos.	Restringe-se na reutilização de componentes de software, mas poderia verificar o quanto desse reuso agregou na redução de falhas e desempenho.	NI, NM, NP, BRUT
<i>Chidamber-Kemerer</i> (CK)	Análises de complexidade de código orientado a objetos em Java.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	NI, NM, NP, BRUT
<i>Mean Time to Execution</i> (MTE)	Mede o desempenho do código com o tempo médio de execução	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	SLC

Source Lines of Code (SLC)	Mede o número de linhas do código-fonte do programa.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	MTE
Coupling between Objects	Representa o número de classes acopladas a uma determinada classe.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	MS, QCF, CC
Response for a Class	Número total de métodos que potencialmente podem ser executados em resposta a uma mensagem recebida por um objeto de uma classe.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	CS, QCF, CC
Weighted Methods per Class (WMC)	Mede a soma da complexidade dos métodos em uma classe.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	NI, NM, NP, BRUT
Bugs Resolvidos por Unidade de Tempo (BRUT)	Métrica que olha para a quantidade de <i>bugs</i> resolvidos por unidade de tempo.	Com restrição de falhas resolvidas, não faz avaliação se a redução contribuiu para o <i>time-to-market</i> .	NM
Nível de Alinhamento da Funcionalidade (NAF)	Mede o nível de alinhamento das funcionalidades com as estratégias organizacionais.	Com restrição de alinhamento funcional com estratégia organizacional, não faz avaliação se tal alinhamento resulta em valor de negócio.	NC, NPS
Nível de Confiabilidade (NC)	Mede o nível de confiabilidade em termos de qualidade de software.	O nível de confiabilidade pode ser uma parte da avaliação de valor para segurança da informação.	NAF, NM
Nível de Eficiência e Eficácia (NEE)	Mede o nível de eficiência em termos de qualidade de software.	O nível de eficiência pode ser uma parte da avaliação de valor.	NE, NC, NM, NI
Nível de Eficácia (NE)	Mede o nível de eficácia em termos de qualidade de <i>software</i> .	O nível de eficácia pode ser uma parte da avaliação de valor.	NRR, NM, NI
Nível de Integridade (NI)	Mede o nível de integridade em termos de qualidade de software.	O nível de integridade pode ser uma parte da avaliação de valor para segurança da informação.	BRUT, NC, NP
Nível de Manutenção (NM)	Mede o nível de manutenibilidade em termos de qualidade de <i>software</i> .	O nível de manutenibilidade pode ser uma parte da avaliação de <i>time-to-market</i> .	BRUT, NC, NP
Nível de Portabilidade (NP)	Mede o nível de portabilidade do software com outras arquiteturas de <i>software</i> .	Restringe-se na reutilização de componentes de software, mas poderia verificar o quanto desse reuso agregou na redução de falhas e desempenho.	NEE, NM
Teste em Canvas	Métrica de transformação <i>Lean</i> no plano de teste de software para simplificar o processo de teste.	Enxugar o processo de teste pode contribuir apenas ao <i>time-to-market</i> .	NEE, NM, BRUT
Net Promoter Score	Mede o nível de satisfação dos clientes em relação do produto recebido e sua experiência com ele.	Métrica comumente utilizada para avaliar e experiência geral dos clientes e não por determinadas partes do produto.	CVNP, ODVC, POC, PRCO, MR
Complexidade por acoplamento	Mede a complexidade por acoplamento que define o número de recursos de <i>software</i> por solução.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	CS, QCF, CC
Complexidade por coesão	Mede a complexidade por coesão que define o número de coesão	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria	CS, QCF, CC

	intra componentes de <i>software</i> .	ao produto.	
<i>Previsão por Planning Poker</i>	Mede o nível de compreensão e assertividade da estimativa.	Restringe-se em contribuir com o planejamento das <i>Sprints</i> .	-
Qualidade por <i>Sprint</i>	Mede o nível de qualidade do <i>software</i> por <i>Sprint</i> .	O nível de qualidade pode ser uma parte da avaliação de valor.	BRUT, NC
Qualidade por Incremento (QI)	Mede o nível de qualidade do <i>software</i> por <i>release</i> ou incremento.	O nível de qualidade por <i>release</i> ou incremento pode ser uma parte da avaliação de valor.	BRUT, NC
Qualidade por Solução (QS)	Mede o nível de qualidade do <i>software</i> por Solução.	O nível de qualidade por solução pode ser uma parte da avaliação de valor.	BRUT, NC
Qualidade por Portfólio (QP)	Mede o nível de qualidade do <i>software</i> por Portfólio de projetos.	O nível de qualidade por portfólio pode ser uma parte da avaliação de valor organizacional.	BRUT, NC
Qualidade por Visão Empresarial	Mede o nível de qualidade do <i>software</i> alinhado com a Visão Empresarial.	O nível de qualidade por visão empresarial pode ser uma parte da avaliação de valor organizacional.	BRUT, NC, CVNP, ODVC,POC, PRCO, MR
<i>Code smells</i> (CS)	Mede potenciais problemas em código-fonte.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	QCF, QC
Usou TDD (UT)	Mede se o projeto utilizou o <i>Test-Driven Development</i>	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	NUT
Não usou TDD (NUT)	Mede se o projeto não utilizou o <i>Test-Driven Development</i>	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	UT
Complexidade ciclomática (CC)	Mede a complexidade ciclomática (complexidade do código-fonte).	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	CS, QCF
Qualidade do código-fonte (QCF)	Mede a qualidade do código-fonte.	Restringe-se em ser uma métrica técnica de engenharia que não sabemos se agregaria ao produto.	CS, QC
Qualidade dos Eventos do <i>Scrum</i> (QES)	Mede a qualidade dos eventos do <i>Scrum</i> , como <i>Sprint Plannig</i> , <i>Daily</i> , <i>Sprint Review</i> e <i>Sprint Retrospective</i> .	Restringe-se em avaliar os eventos do processo do <i>Scrum</i> .	TED
Entrega Antecipada e Frequente de <i>Software</i> (EAFS)	Mede a relevância da produtividade entregando <i>software</i> com antecedência e frequência.	Avaliar a produtividade e a eficiência versus eficácia, sendo que a entrega antecipada e frequente reduz o <i>time-to-market</i> e aumenta o <i>feedback</i> .	QI, QS, QP
Agregação de valor	Mede o retorno do investimento como uma forma relevante de medir a agregação de valor	Medir o retorno do investimento pode ser uma parte da avaliação de valor.	NC, VCNP, ODVC,POC, PRCO,MR
Aspectos relacionados ao esforço	Mede a relevância do esforço de implantação de <i>software</i> que resolva os problemas de clientes.	Usar relevância do esforço de implantação de <i>software</i> não garante que a produção algo que não entrega o valor esperado.	ODVC, POC,PRCO
Modelo de previsão de defeitos	Modelo que mede a previsão de defeitos de <i>software</i> com base em outras métricas.	Restrito apenas para reduzir o risco de falhas durante as fases de teste de <i>software</i> .	EAFS, QI, QS, BRUT.

Segurança orientada a riscos	Métrica de segurança hierárquicas orientadas a risco no contexto do desenvolvimento ágil de <i>software</i> .	Medir a segurança orientada a risco pode ser uma parte da avaliação de valor para segurança da informação.	-
Melhoria de processo de <i>software</i>	Métricas para avaliar melhorias no processo de desenvolvimento de <i>software</i> .	Melhorar o processo de desenvolvimento pode contribuir apenas a entrega contínua e frequente de <i>software</i> .	EAFS, QES

Fonte: Elaborado pelo autor

Os artigos apresentaram métricas que avaliaram diferentes aspectos na avaliação no desenvolvimento de produto, seu *feedback* e o valor entregue. Assim, estudos apresentaram métricas baseadas em modelos de risco que podem ser usados para entender as características dos riscos de *software*, crescimento nas vendas de novos produtos, medição de projetos orientados para o cliente, projetos e programas respeitando custos e orçamento, marcos respeitados e componentes de *software* reutilizáveis. No tocante a complexidade de programas orientados a objetos, autores dessa categoria propuseram as métricas de CK e CBO para avaliar o nível de significância e insignificância com propensão a falhas, estado de validação e utilidade na qualidade de *software* como *bugs* resolvidos por unidade de tempo. Outras métricas aplicáveis à arquitetura dos projetos e para quantificar o nível de alinhamento entre os projetos e os objetivos estratégicos, como métricas de funcionalidade, confiabilidade, eficiência, manutenibilidade, portabilidade e alinhamento de negócio. Para rastrear as abordagens ágeis, outros estudos apresentaram as métricas de complexidade, acoplamento e coesão. Outros autores mediram o nível de satisfação de clientes em projetos que usam o TDD e avaliaram o nível de satisfação de clientes em relação ao produto, onde identificou-se uma grande adoção da medida do NPS. Por fim, a seção seguinte apresenta, para cada categoria, a classificação das métricas utilizadas para cada prática, tanto ágil como preditiva.

2.2.5 Classificação das métricas por categoria

A Tabela 8 destaca a classificação das métricas encontradas e suas respectivas categorias mapeadas durante o processo de RSL, além das subcategorias e processos de medição mapeados após a análise dos resultados. A classificação por Fase do Ágil apresentada na Tabela 8 foi inspirada nas etapas *Initiate*, *Discovery*, *Delivery e Release* dos projetos ágeis sugerido pelo *The Agile Landscape – Deloitte* (TAL). Dessa forma, a categoria de Métricas de Projetos Tradicionais não foram classificadas segundo o TAL.

A partir das etapas *Initiate*, *Discovery*, *Delivery e Release* do modelo TAL e fazendo um relacionamento com as métricas identificadas na Tabela 8, são identificadas algumas lacunas no âmbito da medição em projetos ágeis. Dessa maneira, observa-se que a etapa de *Initiate* não foi incluída na classificação devido à ausência de métricas que avaliassem as ideias iniciais do produto para o projeto ágil como, por exemplo, medir o número de hipóteses de produto que se confirmaram como importantes para os clientes versus as que refutaram. Outra lacuna identificada, apesar de etapa de *Release* ser identificada na Tabela 8, muitas métricas de lançamento de produto com ou sem problemas de funcionamento, com *feedback* em nível de valor e funcional da parte do mercado e de clientes.

Assim, a etapa de *Release* somente contemplou métricas relacionadas a versão de lançamento, sem se preocupar com os resultados após o uso de tal versão no mercado e pelos clientes. Em suma, as lacunas se apresentam nas etapas de *Initiate e Release*, onde são cruciais para se identificar quais ideias (*Initiate*) foram realmente convertidas em valor para o cliente (*Release*). Desde a gestão tradicional de projetos que a etapa de *Initiate* dá suporte à gestão de portfólio e programas. A etapa de *Discovery*, *Delivery e Release* suportam as iniciativas de projetos ou times. O nível de portfólio comumente gerencia um escopo e custos macro dos projetos. O nível de programa gerencia os projetos que possuem dependência técnica e/ou funcional (Lappi et al., 2019). O nível de projeto ou times gerenciam as iniciativas em nível de time de desenvolvimento (Kock & Gemünden, 2019).

Tabela 8 - Classificação das métricas por categoria e subcategoria

Categoria	Subcategoria	Fase	Métrica	Resumo das Métricas
Métricas Ágeis de Backlog do Produto	Complexidade	<i>Delivery</i>	Grau de complexidade do projeto e KPI de Imperativos da técnica proposta.	Com o número de variantes de soluções tecnológicas, auxilia no planejamento de recursos, testar o desempenho da agilidade, qualidade e valor dos negócios, eficiência da equipe e complexidade de projetos.
	Prioridade	<i>Delivery</i>	<i>Product Backlog Rating</i> e Avaliação da qualidade das histórias de usuários (HU).	Mede o nível de prioridade do item do <i>backlog</i> e ajuda na qualidade da redação das HUs, a definição de prioridade e o risco das HUs, os relacionamentos de HU com fontes de requisitos, requisitos não-funcionais e versões de produtos.
	Monitoramento do Projeto	<i>Delivery</i>	Esforço de desenvolvimento de software, Esforço e produtividade e Grau de singularidade do projeto.	Avaliação da aplicabilidade ao longo do ciclo de vida, dependência da linguagem de programação específica e capacidade de prescrever soluções. Determina as características da entrega como o custo, prazo, possíveis desvios de prazo e custo.
Métricas Ágeis de Fluxo	Fluxo ou Processo	<i>Delivery</i>	Reduzir Lead Time, <i>Lead Time</i> , <i>Cycle Time</i> , <i>Efficiency flow</i> , <i>Tasks finished flow</i> , <i>Control Chart</i> , Capacidade do processo, Desempenho do processo e Maturidade do Processo.	Mede a entrega de uma funcionalidade desde o comprometimento até a entrega ao cliente e permite medir quantos itens do <i>backlog</i> são entregues por ciclo. Permite reduzir o <i>lead time</i> para alta capacidade de resposta às necessidades de clientes.
	Iteração ou <i>Sprint</i>	<i>Delivery</i>	Eficácia e eficiência, <i>Sprint Burn Down</i> , impedimentos, Custo do atraso, Custo hora real e planejada e Previsão por <i>Planning Poker</i> .	Realiza a medição por <i>Sprint</i> ou iteração considerando os pontos de história, nível de assertividade das estimativas, impedimentos, custos para medir a eficiência e eficácia do projeto.
	Incremento ou Artefato	<i>Delivery</i>	Clareza do Artefato, Envolvimento e Propriedade do Cliente, Aplicabilidade e Rastreabilidade	Medir a clareza, rastreabilidade e aplicabilidade do artefato do projeto, os níveis de envolvimento e propriedade do cliente aos entregáveis do projeto.
	Estimativa e Esforço	<i>Delivery</i>	<i>Epic Burn Down</i> , <i>Velocity</i> , Estimativa de tarefas e Esforço no projeto e Status Desenvolvimento.	Métrica sobre pontos de história no Épico (abrange uma ou muitas <i>Sprints</i>), mede o desempenho (<i>velocity</i>) através dos pontos de história feitos por <i>Sprint</i> . Realiza a estimativa de tarefas para ajudar na qualidade do software.
	Monitoramento do Projeto	<i>Delivery</i>	Componentes funcionais, Desempenho no projeto e Risco no projeto.	Cobertura dos cenários e histórias de usuário que auxiliaram no desempenho do processo de teste e medir o desempenho do projeto. Medir os riscos do projeto ágil para dar visibilidade ao <i>backlog</i> do produto e sua priorização.
	Qualidade do Produto	<i>Delivery</i>	Rastreamento de tarefas, Atribuição de <i>bugs</i> e <i>Reopens</i> .	Rastreamento de tarefas para ajudar a garantir a qualidade e estabilidade do software. Através da atribuição da pessoa mais indicada para a resolução do problema aberto ou reaberto e reduzindo o <i>time to market</i> do produto.

Métricas Ágeis do Produto	Monitoramento do Projeto	<i>Delivery</i>	Horas trabalhadas em atividades, Tempo nos estágios de desenvolvimento, Aspectos relacionados ao esforço, Características dos riscos de software, Segurança orientada a riscos, Projetos/programas respeitando custos e orçamento e Marcos respeitados.	Analisa a quantidade de horas trabalhadas durante um período (<i>Sprint</i> ou ciclo), o tempo do requisito nos estágios de desenvolvimento, marcos e esforço para entregar ao cliente. Fornece informações sobre modelos de risco que podem ser usados para entender as características dos riscos de software.
	Qualidade do Código-Fonte	<i>Delivery /Release</i>	<i>Chidamber-Kemerer, Source Lines of Code, Coupling between Objects, Response for a Class, Weighted Methods per Class, Mean Time to Execution</i> , Componentes reutilizáveis, Complexidade por acoplamento, Complexidade por coesão, <i>Code smells</i> , Qualidade do código-fonte, Complexidade ciclomática, Usou <i>Test-Driven Development</i> , Não usou <i>Test-Driven Development e</i> Quantidade de <i>commits</i> .	Análises de complexidade de programas orientados a objetos, códigos-fonte publicados, de número de linhas de código, reúso de componentes e a avaliação da qualidade de implementação por melhores práticas orientadas a objetos com baixo acoplamento e alta coesão.
	Qualidade de Software	<i>Delivery /Release</i>	<i>Bugs</i> resolvidos por unidade de tempo, Nível de Confiabilidade, Nível de Eficiência/Eficácia, Nível de Eficácia, Nível de Integridade, Nível de <i>Manutenabilidade</i> , Nível de Portabilidade, Teste em <i>Canvas</i> , Qualidade por <i>Sprint</i> , Qualidade por <i>Release/Incremento</i> , Qualidade por Solução, Qualidade por Portfólio, Qualidade por Visão, Empresarial, Qualidade dos Eventos do Scrum e Modelo de previsão de defeitos.	Realiza a medição da quantidade de <i>bugs</i> resolvidos por unidade de tempo e mede o nível de confiabilidade, eficiência, eficácia, integridade, manutenibilidade e portabilidade em termos de qualidade de software. Utiliza práticas <i>Lean</i> no plano de teste de software para simplificar o processo de teste.

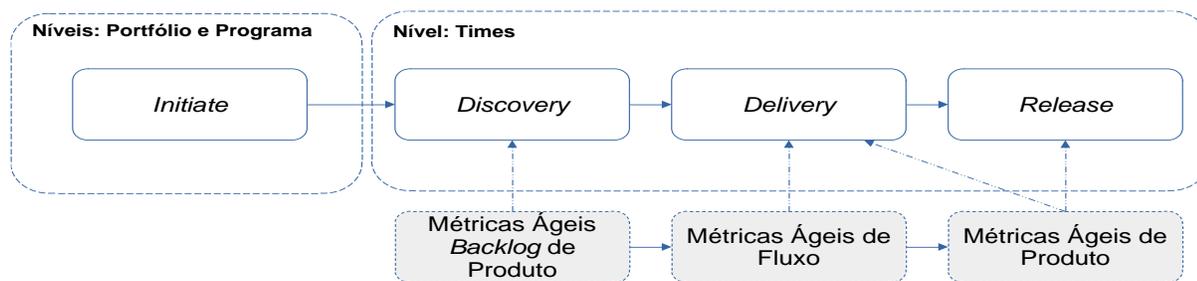
	Relação Clientes e Stakeholders	<i>Delivery/Release</i>	Crescimento nas vendas de novos produtos, Projetos orientados para o cliente, Orçamento dedicado à análise ou verificação do cliente, Nível de Alinhamento da Funcionalidade, Agregação de valor e <i>Net Promoter Score</i> .	Mede os orçamentos com dedicação de análise e verificação do cliente e o número de projetos criados e orientados para o cliente. Mede o nível de satisfação dos clientes em relação do produto recebido, sua experiência e agregação de valor.
	Entrega Contínua	<i>Delivery/Release</i>	Melhoria de processo de software e Entrega antecipada e frequente de software.	Utilizou métricas de produto para avaliar melhorias no processo de desenvolvimento de <i>software</i> e a relevância da produtividade entregando software com antecedência e frequência.
Métricas de Projetos Tradicionais	Análise do Valor Agregado	Não se aplica	EVM (<i>Earned Value Management</i>) e <i>Work Perform</i> .	Mede o desempenho e o progresso de projetos, com a capacidade de combinar medidas de escopo, tempo e custos, mediante um planejamento previamente estabelecido. Combina séries temporais e análise de regressão para desenvolver PV (<i>planned value</i>) e resposta EV (<i>earned value</i>) para prever <i>work perform</i> .
	Qualidade do Produto	Não se aplica	Monitoramento de defeitos, Qualidade geral dos sistemas do projeto e Mudança percentual relacionada à qualidade/valor.	Monitora os defeitos identificados durante o desenvolvimento do projeto e monitora a mudança do percentual relacionado à qualidade/valor do produto.
	Monitoramento do Projeto	Não se aplica	Variação de Desempenho, Número de defasagens, EDM (<i>Earned Duration Management</i>), Índice de Controle de Cronograma, Desempenho de cada atividade, Monitoramento de marcos e cronogramas, Análise <i>bootstrap</i> , <i>Project Health Index</i> , Esforço da tarefa e monitoramento, Índice de Controle de Custos, Análise do equilíbrio custo/risco, Análise de Risco do Cronograma com restrição de recursos e <i>Critical Success Factors</i> .	Análise de inicialização para avaliar o desempenho de um projeto e a classificação de problemas de projetos para identificar problemas em projetos que provavelmente falharão antes de começarem. Calcula o número de defasagens (atrasos) no cronograma e mede os custos do projeto. Utiliza o conceito de orçamento ótimo no nível do projeto e no nível da rede de atividades.
	Alinhamento Organizacional/Conhecimento	Não se aplica	Seleção de Projetos.	Integraram o aprendizado organizacional e o desenvolvimento do conhecimento adquirido através do processo de seleção de projetos.

Requisitos do Produto	Não se aplica	Modelo RE4SA e Integridade de requisitos.	Modelo <i>Requirements Engineering for Software Architecture</i> (RE4SA) que fornece uma conexão entre os artefatos e que facilita a comunicação dentro da equipe de desenvolvimento. Avalia se os requisitos especificados são completos ou não para o desenvolvimento.
Análise da Qualidade do Cronograma	Não se aplica	Proporção de tarefas detalhadas para marcos, Porcentagem de tarefas no caminho crítico, Tipo de relacionamento, Marcos sem predecessor ou sucessor e Impacto Predecessor no Sucessor.	Calcula o percentual de tarefas do projeto no caminho crítico do cronograma e investiga a proporção de tarefas do projeto que foram detalhadas para marcos específicos. Avalia os marcos do cronograma sem definição de predecessor e sucessor e calcula o tipo de relacionamento de tarefas predecessoras.
Gestão de Mudanças	Não se aplica	Tempo de resolução de pedido de mudança e Mudança percentual relacionada ao design.	Calcula o tempo de processamento de pedido de mudança no projeto e avalia a mudança percentual relacionada ao design.
Relação Cliente/ <i>Stakeholders</i>	Não se aplica	<i>Stakeholder Metrics-integrated Management Model</i>	Integra as partes interessadas com métricas de identificação, comunicação, engajamento, capacitação e atendimento.

Fonte: dados da pesquisa, 2022.

Dessa forma, a Figura 2 apresenta o relacionamento entre as etapas do TAL e as categorias identificadas durante essa pesquisa e destaca uma lacuna na ausência de métricas na etapa de *Initiate*. Assim, identificou-se métricas de Gestão Orçamentária (GO) que não se relacionam com todas as fases do TAL.

Figura 2 – Relacionamento entre as etapas do TAL e a categorização realizada na Tabela 5



Fonte: Dados da pesquisa, 2022.

Conforme os achados das métricas da Tabela 8, a GO é compreendida por métricas como Custo do Atraso, Custo-Hora Real e Planejada, Desempenho no Projeto (Métricas Ágeis de Fluxo) e Projetos/Programas respeitando Custos e Orçamento (Métricas Ágeis do Produto). Conforme o PMI (2021), a gestão de projetos conta com três medidas básicas de gestão de custos, sendo o Custo Real (CR) comparado ao Custo Planejado (CP), Variação de Custo (*Cost Variance* - CV) e o Índice de Desempenho de Custo (*Cost Performance Index* - CPI). Assim, PMI (2021) determina que o CR comparado ao CP, trata-se de uma medida de custo que compara o custo realizado de mão de obra ou recursos com o custo estimado; o CV é uma variação de custo simples sendo determinada comparando o custo realizado de uma entrega com o custo estimado e, quando usado com gerenciamento de valor agregado, é a diferença entre o valor agregado e o custo real; e o CPI é uma medida de gerenciamento de valor agregado que indica a eficiência com que o

trabalho está sendo executado em relação ao custo orçado do trabalho. Assim, entende-se a oportunidade de medir o custo planejado *versus* o realizado dos entregáveis do projeto por ciclos menores iniciando desde a execução de Iterações ou *Sprints*, dos entregáveis realizados por trimestre ou *release* pelo programa, e as entregas esperadas em nível de portfólio de projetos.

Grimaldi *et al.* (2016) apresentaram como uma estrutura ágil escalável, baseada em SAFe 3.0 e mostrou como implementar e medir o desempenho considerando que as funções e responsabilidades são divididas entre a organização e o fornecedor. Grimaldi *et al.* (2016) utilizaram métricas como eficácia e eficiência, custo-hora real e planejada, impedimentos e custo do atraso durante as rodadas de *Sprint*. No mesmo sentido que Ahmed *et al.* (2017) discutiram sobre as métricas *Sprint Burn Down*, *Epic Burn Down Velocity Chart*, *Control Chart* e a maneira como muitos dos cálculos de métricas são fortemente acoplados aos pontos de história para itens como o Épico ou *Sprint*.

Do ponto de vista de Petersen e Wohlin (2011), o *Lean* fornece ferramentas de análise e melhoria com foco no ciclo de vida do desenvolvimento, enquanto o ágil se concentra em soluções e prescreve conjuntos de práticas para obter agilidade. Petersen e Wohlin (2011) sugeriram análises sobre as métricas como reduzir o *lead time* para obter alta capacidade de resposta às necessidades dos clientes e fornecer um sistema de rastreamento que mostre o status do desenvolvimento de *software*. No sentido do monitoramento, as métricas de processo propostas por Choraś *et al.* (2020) foram utilizadas continuamente para ajudar a garantir a qualidade e estabilidade do *software* como a estimativa de tarefas e rastreamento tarefas em tempo real. Assim, pode-se identificar um *gap* sobre a ausência de métricas que aborde todas as etapas do projeto com objetivo de obter novas discussões sobre um processo de medição desde o portfólio, programa e time de projetos para trabalhos futuros é necessário. Esse contraponto sobre as métricas apresentadas pela Tabela 8 pode ser dividido em dois cenários dentro do contexto das práticas de gestão ágil de projetos: 1) escassez de propostas de métricas que venham medir as fases de *initiate*, *discovery*, *delivery* e a entrega contínua (*release*), mesmo que isoladamente, no desenvolvimento de um produto e 2) escassez de propostas de um processo de medição das etapas de *discovery*, *delivery* e entrega contínua para que assim seja possível medir e monitorar o projeto desde a iniciação até a entrega em partes (*Sprints*) e/ou no encerramento do projeto.

Com base na Tabela 5 e aos cenários 1 e 2 acima apresentados, verifica-se ainda a lacuna de processos de gestão ágil de projetos que contribuam como um *framework* de métricas ágeis capaz de ser um instrumento gerencial de maneira a orientar a liderança na gestão de projetos ágeis e para a tomada de decisão a partir de métricas estratégicas para a organização e para o desenvolvimento ágil de produtos. Os projetos ágeis que desenvolvem produtos, precisam testar hipóteses para resolver problemas organizacionais ou de clientes. Nesse sentido que o resultado oriundo das hipóteses deve ser medido para a tomada de decisão de refutar ou seguir adiante com a estratégia organizacional.

CONSIDERAÇÕES FINAIS

A partir da pesquisa realizada nas bases *Web of Science* e *Scopus*, a RSL encontrou-se 83 artigos que foram selecionados, dos quais apenas 42 abordavam aspectos relacionados a métricas de projetos tradicionais, 10 sobre as métricas ágeis de fluxo, 7 sobre as métricas ágeis de *backlog* do produto e 24 sobre as métricas ágeis de produto.

Apesar do histórico da adoção dos projetos tradicionais existir há décadas, o total dos artigos sobre as métricas em projetos tradicionais (40) encontradas nesse estudo não apresentaram um número tão maior que os resultados das métricas ágeis (43 artigos). Assim, percebeu-se que as métricas tradicionais focaram em apresentar a métrica do EVM com a adoção de melhorias na métrica, tal como melhorar a previsibilidade de riscos e alcançar o melhor valor agregado e o sucesso do projeto. Outras métricas muito citadas foram as adaptações inspiradas no EVM como o EDM e a utilização do SPI buscando medir o desempenho do cronograma.

Nas métricas ágeis, percebeu-se que as métricas se concentraram no *discovery* e no *delivery*, ou seja, em relação a medir o *backlog* do produto (9 artigos) no *discovery* e no fluxo e produto, tivemos 34 artigos no *delivery*. As métricas de *backlog* do produto se concentraram em medir a qualidade do requisito, enquanto as de produto, verificou-se uma grande concentração de métricas com preocupações com a qualidade do produto, ou seja, no tocante a quantidade de *bugs* resolvidos, tempo de resolução de problemas, qualidade do código-fonte e sobre a satisfação do cliente. Sobre

as métricas de fluxo, identificou-se métricas como o *Lead Time*, *Velocity*, *Efficiency Flow*, *Cycle Time*, *Sprint Burn Down*, *Epic Burn Down*, *Control Chart*, entre outras. Assim, a Figura 2 apresenta uma lacuna de métricas que suporte um processo de gestão de projetos a partir do portfólio, programa e times de projeto e não apenas a gestão e métricas isoladas nas etapas de *discovery*, *delivery* e *release*. Além disso, na fase de *Initiate*, pode-se identificar um *gap* sobre a ausência de métricas para a gestão de iniciativas para o portfólio de projeto. Por fim, entende-se com esse estudo que ainda existe uma lacuna de processos de gestão ágil de projetos que contribuam como um *framework* capaz de ser um instrumento gerencial com métricas estratégicas para a organização e para projetos de desenvolvimento ágil de produtos. Assim, projetos ágeis precisam testar hipóteses para resolver problemas organizacionais ou de clientes e precisam ser medidas para a tomada de decisão do que fazer a seguir.

Espera-se que este estudo possa contribuir com outras pesquisas que desejam identificar as métricas utilizadas entre as abordagens ágil e tradicional de gestão de projetos. Além disso, orientar novas pesquisas para trabalhos futuros no sentido de analisar uma possível lacuna entre na utilização de métricas tradicionais em projetos ágeis, que se compreende também em ser uma limitação dessa pesquisa. Além disso, é preciso estender a pesquisa para uma verificação em bases de patentes e na literatura cinzenta.

REFERÊNCIAS

- Abdelrahman Aljemabi, M. et al. (2020). Mining social collaboration patterns in developer social networks. *IET Software*, 14(7), 839-849.
- Ahmed, A. R. et al. (2017). Impact of story point estimation on product using metrics in scrum development process. *Intern. Journal of Advanced Computer Science and Applications*, 8(4).
- Aljedaibi, W. & Khamis, S. (2019). Towards Measuring the Project Management Process During Large Scale Software System Implementation Phase. *The ISC Intern. Journal of Info. Security*, 11(3), 161-172.
- Antinyan, V. (2014). Monitoring evolution of code complexity and magnitude of changes. *Acta Cybernetica*, 21(3), 367-382.
- Asha, N. & Mani, P. (2018). Knowledge-based Acceptance Test driven agile Approach for Quality Software Development. *International Journal of Resent Tech. and Engineering*, 7.
- Athanasiou, D. et al. (2014). Test code quality and its relation to issue handling performance. *IEEE Transac. on Software Eng*, 40(11), 1100-1125.
- Azzeh, M. & Nassif, A. B. (2016). A hybrid model for estimating software project effort from Use Case Points. *Applied Soft Computing*, 49, 981-989.
- Bakhtiary, V. et al. (2020). The effectiveness of test-driven development approach on software projects: A multi-case study. *Bulletin of Electrical Eng and Info.*, 9(5), 2030-2037.
- Batselier, J. & Vanhoucke, M. (2015). Evaluation of deterministic state-of-the-art forecasting approaches for project duration based on earned value management. *International Journal of Project Management*, 33(7), 1588-1596.
- Budacu, E. N. & Pocatilu, P. (2018). Real Time Agile Metrics for Measuring Team Performance. *Informatica Economica*, 22(4), 70-79.
- Chang, C. P. (2015). Software risk modeling by clustering project metrics. *International Journal of Software Engineering and Knowledge Engineering*, 25(06), 1053-1076.
- Chen, H. L. (2018). Early prediction of project duration: A longitudinal study. *Engineering Management Journal*, 30(3), 191-202.
- Chen, H. L. (2021). Early identification of distressed capital projects: A longitudinal approach. *International Journal of Managing Projects in Business*.
- Chen, H. L. (2014). Improving forecasting accuracy of project earned value metrics: Linear modeling approach. *Journal of Management in Engineering*, 30(2), 135-145.
- Chen, H. L., Chen, W. T. & Lin, Y. L. (2016). Earned value project management: Improving the predictive power of planned value. *International Journal of Project Management*, 34(1), 22-29.

- Chen, Z. et al. (2020). A Bayesian approach to set the tolerance limits for a statistical project control method. *International Journal of Production Research*, 58(10), 3150-3163.
- Choraś, M. et al. (2020). Measuring and improving agile processes in a small-size software development company. *IEEE access*, 8, 78452-78466.
- Choudhury, I. (2019). *Agile Methods for Engineering*. In *Management for Scientists*. Emerald Publishing Limited.
- Costa, J., Vasconcelos, A. & Fragoso, B. (2020). An Enterprise Architecture Approach for Assessing the Alignment Between Projects and Goals. *International Journal of Information Technology Project Management (IJITPM)*, 11(3), 55-76.
- Colin, J. et al. (2015). A multivariate approach for top-down project control using earned value management. *Decision Support Systems*, 79, 65-76.
- Colin, J. & Vanhoucke, M. (2014). Setting tolerance limits for statistical project control using earned value management. *Omega*, 49, 107-122.
- Colin, J. et al. (2015). A comparison of the performance of various project control methods using earned value management systems. *Expert Systems with Applications*, 42(6), 3159-3175.
- Concas, G. et al. (2012). An empirical study of software metrics for assessing the phases of an agile project. *International Journal of Software Engineering and Knowledge Engineering*, 22(04), 525-548.
- Day, F. C. et al. (2019). Feasibility study of an EHR-integrated mobile shared decision making application. *International journal of medical informatics*, 124, 24-30.
- Dennehy, D. & Conboy, K. (2018). Identifying challenges and a research agenda for flow in software project management. *Project Management Journal*, 49(6), 103-118.
- Dias, K. R. S. et al. (2021). Hybrid Project Management Method for managing ICT project's scope: a case study in a Brazilian company. *Brazilian Journal of Develop.*, 7(8), 75984-76014.
- El Asmar, M. et al. (2013). Quantifying performance for the integrated project delivery system as compared to established delivery systems. *Journal of construction engineering and management*, 139(11), 04013012.
- Fleche, D. et al. (2017). Collaborative project: Evolution of computer-aided design data completeness as management information. *Concurrent Engineering*, 25(3), 212-228.
- Gao, K. et al. (2011). Choosing software metrics for defect prediction: an investigation on feature selection techniques. *Software: Practice and Exp.*, 41(5), 579-606.
- Geng, S. et al. (2018). Knowledge contribution as a factor in project selection. *Project Management Journal*, 49(1), 25-41.
- Glenwright, E. (2007). A survey of the 30 most serious flaws in scheduling. In *PMICoS 2007 Annual Conference*.
- Grau, D. & Back, W. E. (2015). Predictability index: Novel metric to assess cost and schedule performance. *Journal of Construction Eng. and Management*, 141(12), 04015043.
- Gren, L., Goldman, A. & Jacobsson, C. (2020). Agile ways of working: a team maturity perspective. *Journal of Software: Evolution and Process*, 32(6), e2244.
- Grimaldi, P. et al. (2016). An agile, measurable and scalable approach to deliver software applications in a large enterprise. *Intern. Journal of Agile Systems and Manag.* 9(4), 326-339.
- Han, S., Choi, J. O. & O'Connor, J. T. (2017). Quality of baseline schedules: Lessons from higher education capital facility projects. *Journal of Professional Issues in Engineering Education and Practice*, 143(1), 04016017.
- Hanna, A. S. (2016). Benchmark performance metrics for integrated project delivery. *Journal of Construction Engineering and Management*, 142(9), 04016040.
- Hanna, A. S. et al. (2012). Request for information: benchmarks and metrics for major highway projects. *Journal of construction engineering and management*, 138(12), 1347-1352.
- Hazır, Ö. (2015). A review of analytical models, approaches and decision support tools in project monitoring and control. *International Journal of Project Manag.* 33(4), 808-815.
- Hernández, G. et al. (2019). Productivity Metrics for an Agile Software Develop. Team: A Systematic Review. *TecnoLógicas*, 22(SPE), 63-81.

- Ibba, S. et al. (2018). Initial coin offerings and agile practices. *Future Internet*, 10(11), 103.
- Ibrahim, M. W. et al. (2020). Quantitative comparison of project performance between project delivery systems. *Journal of management in engineering*, 36(6), 04020082.
- Iqbal, S. et al. (2012). Yet another set of requirement metrics for software projects.
- Isong, B. & Obeten, E. (2013). A systematic review of the empirical validation of object-oriented metrics towards fault-proneness prediction. *International Journal of Software Engineering and Knowledge Engineering*, 23(10), 1513-1540.
- Jethani, K. (2013). Software metrics for effective project management. *International Journal of System Assurance Engineering and Management*, 4(4), 335-340.
- Junior, V. B. et al. (2019). Elaboração de um Plano de Gerenciamento de Projeto para Implantação de Laboratório de Metrologia Utilizando o Guia PMBOK 5ª Edição. *Cippus*, 7(2), 127-145.
- Jyothi, V. E. & Rao, K. N. (2017). Effective Implementation of Agile Software Development with a Framework, Metric Tool, and in Association with Cloud and Lean Kanban. *International Journal of Advanced Eng. Research and Science*, 4(3), 237085.
- Kamulegeya, G. et al. (2018). Measurements in the early stage software start-ups: a multiple case study in a nascent ecosystem. *Found. of Computing and Decision Sciences*, 43(4), 251-280.
- Kayes, I. et al. (2016). Product backlog rating: a case study on measuring test quality in scrum. *Innovations in Systems and Software Eng.* 12(4), 303-317.
- Kerkhove, L.P. & Vanhoucke, M. (2017). Extensions of earned value management: Using the earned incentive metric to improve signal quality. *Intern. Journal of Project Management*, 35(2), 148-168.
- Kristiansen, J. N. & Ritala, P. (2018). Measuring radical innovation project success: typical metrics don't work. *Journal of Business Strategy*.
- Kim, T., Kim, Y. W. & Cho, H. (2016). Customer earned value: performance indicator from flow and value generation view. *Journal of management in Eng.* 32(1), 04015017.
- Ko, B. S. & Cho, M. S. (2020). Evaluating the relative efficiency of defense R&D projects in a multi-project environment using EVM and CAIV measures. *ICIC Express Letters, Part B: Applications*, 11(7), 675-682.
- Kock, A. & Gemünden, H. G. (2019). Project lineage management and project portfolio success. *Project Management Journal*, 50(5), 587-601.
- Yamashita, A. & Counsell, S. (2013). Code smells as system-level indicators of maintainability: An empirical study. *Journal of Systems and Software*, 86(10), 2639-2653.
- Yousefi, N. et al. (2019). Using statistical control charts to monitor duration-based performance of project. *arXiv preprint arXiv:1902.02270*.
- Lappi, T. M. et al. (2019). Project governance and portfolio management in government digitalization. *Transforming Government: People, Process and Policy*, 13(2), 159-196.
- Mas, A. et al. (2020). Supporting the deployment of ISO-based project manag. processes with agile metrics. *Computer Standards & Interfaces*, 70, 103405.
- Mascarenhas Hornos da Costa, J., Oehmen, J., Rebentisch, E. & Nightingale, D. (2014). Toward a better comprehension of Lean metrics for research and product development management. *R&D Management*, 44(4), 370-383.
- Mäkiahho, P. et al. (2022). MMT: a tool for observing metrics in software projects. In *Research Anthology on Agile Software, Software Develop and Testing* (pp. 1077-1089). IGI Global.
- Meidan, A. et al. (2018). Measuring software process: a systematic mapping study. *ACM Computing Surveys (CSUR)*, 51(3), 1-32.
- Narbaev, T. et al. (2014). Combination of growth model and earned schedule to forecast project cost at completion. *Journal of Construction engineering and management*, 140(1), 04013038.
- Nidagundi, P. & Novickis, L. (2016). Introduction to Lean Canvas Transformation Models and Metrics in Software Testing. *Appl. Comput. Syst.*, 19(1), 30.

- Orgut, R. E. et al. (2018). Metrics that matter: Core predictive and diagnostic metrics for improved project controls and analytics. *Journal of Construction Engineering and Management*, 144(11), 04018100.
- Pajares, J.& Lopez-Paredes, A. (2011). An extension of the EVM analysis for project monitoring: The Cost Control Index and the Schedule Control Index. *International Journal of Project Management*, 29(5), 615-621.
- Parrend, P. et al. (2014, May). Swarm Projects: Beyond the Metaphor. In *International Conference on Swarm Intelligence Based Optimization* (pp. 131-138). Springer, Cham.
- Pavlova, N. et al. (2021). Creating the Agile Model to Manage the Activities of Project Oriented Transport Companies. *Eastern-European Journal of Enterprise Tech*, 1(3), 109.
- Perkusich, M. et al. (2017). Assisting the continuous improvement of Scrum projects using metrics and Bayesian networks. *Journal of Software: Evolution and Process*, 29(6), e1835.
- Petersen, K.& Wohlin, C. (2011). Measuring the flow in lean software development. *Software: Practice and experience*, 41(9), 975-996.
- Pradhan, S.& Nanniyur, V. (2021). Large scale quality transformation in hybrid development organizations—A case study. *Journal of Systems and Software*, 171, 110836.
- Prakash B. et al. (2018). Ontology based risks management model for agile software development. *Journal of Advanced Research in Dynamical and Control Systems*, 10(11 Special Issue):319–330.
- Pressman, R. S.& Maxim, B. R. (2021). *Engenharia de software-9*. McGraw Hill Brasil.
- Project Management Institute (PMI) (2021): *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, PMI, Newtown Square, PA, 2021.
- Pozzana, I. et al. (2021). Spreading of performance fluctuations on real-world project networks. *Applied Network Science*, 6(1), 1-15.
- Rajablu, M. et al. (2017). Managing for stakeholders: Introducing stakeholder metrics-integrated model to lead project ethics and success. *International Journal of Project Organisation and Management*, 9(1), 31-56.
- Rajagopalan, J. et al. (2018). Introduction of a new metric “Project Health Index”(PHI) to successfully manage IT projects. *Journal of Organizational Change Management*.
- Ries, E. (2011). *The Lean Startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. New York: Crown Business.
- Royce, W. (1970). The software lifecycle model(waterfall model). In *Proc. Westcon*, V.314.
- Sampaio, R. F.& Mancini, M. C. (2007). Systematic review studies: a guide for careful synthesis of the scientific evidence. *Brazilian Journal of Physical Therapy*, 11, 83-89.
- Sato, T.& Hirao, M. (2013). Optimum budget allocation method for projects with critical risks. *International Journal of Project Management*, 31(1), 126-135.
- Savola, R. et al. (2012). Risk-driven security metrics in agile software development: An industrial pilot study. *Journal of Universal Computer Science*, 18(12), 1679-1702.
- Sadler, H. (2020). ER2C SDMLC: enterprise release risk-centric systems development and maintenance life cycle. *Software Quality Journal*, 28(4), 1755-1787.
- Sajedi-Badashian, A. & Stroulia, E. (2020). Investigating the info. value of different sources of evidence of developers' expertise for bug assignment in open-source projects. *IET Software*, 14(7), 748-758.
- Sharma, V. et al. (2021). Development of metrics and an external benchmarking program for healthcare facilities. *International Journal of Construction Management*, 21(6), 615-630.
- Shawky, D. M.& Abd-El-Hafiz, S. K. (2016). Characterizing software development method using metrics. *Journal of Software: Evolution and Process*, 28(2), 82-96.
- Staron, M. & Meding, W. (2016). Mesram—a method for assessing robustness of measurement programs in large software development organizations and its industrial evaluation. *Journal of Systems and Software*, 113, 76-100.
- Singh, J., Singh, K.& Singh, J. (2019). Reengineering framework to enhance the performance of existing software. *System*, 1139, 120-3.

- Simpson, J. R. et al. (2021). Optimizing the electronic health record: An inpatient sprint addresses provider burnout and improves electronic health record satisfaction. *Journal of the American Medical Informatics Association*, 28(3), 628-631.
- Spijkman, T. et al. (2021). Alignment and granularity of requirements and architecture in agile development: A functional perspective. *Information and Software Technology*, 133, 106535.
- Sommerville, I.(2007).*Eng. de Software*, 9. Ed. Pearson, Addison Wesley,8(9),10,p.44-45.
- Song, J., Martens, A.& Vanhoucke, M. (2021). Using schedule risk analysis with resource constraints for project control. *European Journal of Operational Research*, 288(3), 736-752.
- Sutherland, J. (2019). *Scrum: A arte de fazer o dobro do trabalho na metade do tempo*. Rio de Janeiro: Sextante, 2019.
- Tahir, T. et al. (2018). A systematic mapping study on software measurement programs in SMEs. *E-Informatica Software Engineering Journal*, 12(1).
- Tanveer, B. et al. (2019). An evaluation of effort estimation supported by change impact analysis in agile software develop. *Journal of Software: Evolution and Process*, 31(5), e2165.
- Tranfield,D. et al. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review.*British journal of management*,14(3),207-222.
- Wallace, L. G.& Sheetz, S. D. (2014). The adoption of software measures: A technology acceptance model (TAM) perspective. *Information & Management*, 51(2), 249-259.
- Wauters, M.& Vanhoucke, M. (2015). Study of the stability of earned value management forecasting. *Journal of Construction Engineering and Management*, 141(4), 04014086.
- Van Casteren, W. (2017). *The Waterfall Model and the Agile Methodologies: A comparison by project characteristics*. Research Gate, 1-6.
- Vanhoucke, M. (2011). On the dynamic use of project performance and schedule risk information during projecttracking. *Omega*, 39(4), 416-426.
- Vanhoucke, M. (2019). Tolerance limits for project control: An overview of different approaches. *Computers & Industrial Engineering*, 127, 467-479.
- Üsfekes, Ç. et al. (2019). Auction-based serious game for bug tracking. *IET Software*, 13(5), 386-392.
- Zheng, M. et al. (2021). Key performance indicators for the integration of the service-oriented architecture and scrum process model for IOT. *Scientific Programming*, 2021.