

**UNIVERSIDADE NOVE DE JULHO
PROGRAMA DE MESTRADO PROFISSIONAL EM ADMINISTRAÇÃO
GESTÃO DE PROJETOS**

**PROPOSTA DE MODELO BASEADO EM COMPETÊNCIAS PARA
GERENCIAMENTO DE ATIVIDADES EM PROJETOS ÁGEIS**

JOSÉ ROMUALDO DA COSTA FILHO

São Paulo
2021

José Romualdo da Costa Filho

**PROPOSTA DE MODELO BASEADO EM COMPETÊNCIAS PARA
GERENCIAMENTO DE ATIVIDADES EM PROJETOS ÁGEIS**

**PROPOSAL OF A COMPETENCIES-BASED MODEL FOR MANAGING
ACTIVITIES IN AGILE PROJECTS**

Dissertação apresentada ao Programa de Mestrado Profissional em Administração: Gestão de Projetos da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de **Mestre em Administração**.

Orientador: Prof. Dr. Renato Penha

Coorientação: Prof. Dr. Luciano Ferreira da Silva

São Paulo
2021

Costa Filho, José Romualdo da.

Proposta de modelo baseado em competências para gerenciamento de atividades em projetos ágeis. / José Romualdo da Costa Filho. 2021.

117 f.

Dissertação (Mestrado) - Universidade Nove de Julho - UNINOVE, São Paulo, 2021.

Orientador (a): Prof. Dr. Renato Penha.

1. Gerenciamento de atividades. 2. Gerenciamento de projetos preditivos. 3. Gerenciamento de projetos ágeis. 4. Competências pessoais. 5. Competências interpessoais.

I. Penha, Renato. II. Título.

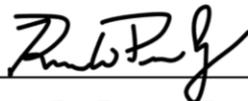
CDU 658.012.2

DEFESA DE DISSERTAÇÃO DE MESTRADO

JOSÉ ROMUALDO DA COSTA FILHO

Dissertação apresentada ao Programa de Pós-Graduação em Gestão de Projetos da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de **Mestre em Administração**, pela Banca Examinadora, formada por:

São Paulo, 29 de novembro de 2021.



Presidente: Prof. Dr. Renato Penha – Orientador



Membro: Prof. Dr. Luciano Ferreira da Silva – Coorientador



Membro: Profa. Dra. Cristina Dai Prá Martens (UNINOVE)



Assinado de forma digital por PAULO
SERGIO GONCALVES DE
OLIVEIRA:11632733846
Dados: 2021.11.29 15:31:09 -03'00'

Membro: Prof. Dr. Paulo Sérgio Gonçalves de Oliveira (ANHEMBI MORUMBI)

“O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)”

“This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)”

“A mente que se abre a uma nova ideia
jamais voltará ao seu tamanho original.”
(Albert Einstein)

DEDICATÓRIA

À minha família e aos amigos que me acompanharam durante esta minha jornada

AGRADECIMENTO

Agradeço à Universidade Nove de Julho, pela oportunidade de participar deste excelente programa de pós-graduação, fornecendo espaço, corpo diretivo e administrativo e professores de excelente qualidade. Agradeço também aos orientadores, Professor Doutor Renato Penha e Professor Doutor Luciano Ferreira da Silva, que me conduziram durante este período de forma ímpar e tornaram-se, não só professores da vida, mas grandes amigos.

Aos amigos do mestrado e doutorado, com os quais convivi e aprendi, tanto compartilhando conhecimentos quanto materiais, principalmente aqueles materiais que já vinham traduzidos! Agradeço à Professora Doutora Cristina Dai Prá Martens e ao Professor Doutor Paulo Sérgio Gonçalves de Oliveira, pelo tempo e parceria no processo de construção desta dissertação e pela avaliação crítica e sincera que fizeram do trabalho apresentado.

Aos demais professores, que ao longo deste período se dedicaram para nos entregar o que há de melhor: a capacidade de ser um pesquisador.

RESUMO

A gestão das atividades envolve o desenvolvimento e o emprego de competências técnicas (*hard skills*) e competências interpessoais (*soft skills*). Destaca-se que as práticas, ferramentas e competências empregadas na gestão de atividades se diferem em relação às práticas gerenciais preditivas e ágeis. No contexto preditivo, a associação de uma pessoa do time de desenvolvimento à determinada atividade em projetos é feita de forma manual e individualmente por uma pessoa responsável pelo gerenciamento do projeto. Nas práticas ágeis, por outro lado, o gerenciamento de atividades ocorre em um processo iterativo e incremental, no qual as pessoas do time de desenvolvimento e os demais *stakeholders* do projeto trabalham ativamente juntos para entender o problema, além de manterem priorizadas as atividades com maior valor de negócio. Nesse cenário, esta pesquisa tem como objetivo desenvolver um modelo conceitual, baseado em competências que auxiliem a autodesignação de atividades de um time ágil em uma iteração. Para alcançar esse objetivo, esta pesquisa contemplou duas fases. Na primeira fase, foi realizada uma Revisão Sistemática da Literatura (RSL) com o propósito de identificar como os estudos sobre as práticas ágeis de gestão de projetos abordam a gestão de atividades em uma iteração. Na segunda fase da pesquisa, foi realizada uma pesquisa de campo, onde o modelo proposto, foi implementado em uma ferramenta computacional e aplicado em ambiente laboral em uma empresa que atua no segmento de serviços, com desenvolvimento de seus produtos por meio do uso de práticas ágeis de gestão de projetos. A contribuição desta dissertação é oferecer uma ferramenta computacional, de concepção inédita, como ferramenta de auxílio no processo de autodesignação de atividades de um time ágil em uma iteração, uma vez que as pesquisas efetuadas nas base de dados acadêmica não apresentaram modelos computacionais compatíveis ao modelo computacional apresentado nesta pesquisa. A pesquisa apresentou limitações por abordar apenas os *frameworks* ágeis, *Scrum* e *Kanban*, e restringiu-se a atividades de dois times de desenvolvimento. Estudos futuros podem ser realizados para análise do uso da ferramenta computacional em organizações com práticas ágeis em escala ou com maiores times de desenvolvimento.

Este estudo se insere na linha de pesquisa Gerenciamento em Projetos do PPGP-UNINOVE, no contexto do projeto de pesquisa de Projetos Ágeis e Híbridos e do projeto financiado pelo Edital Universal CNPq. A partir deste estudo foram produzidos um registro de software e um modelo de utilidade.

Palavras-chave Gerenciamento de atividades; gerenciamento de projetos preditivos, gerenciamento de projetos ágeis; competências pessoais; competências interpessoais.

ABSTRACT

The management of activities involves the development and use of technical skills (hard skills) and interpersonal skills (soft skills). It is noteworthy that the practices, tools and skills used in the management of activities differ in relation to predictive and agile management practices. In the predictive context, the association of a development time person in an activity provided in projects is done manually and individually by the person responsible for managing the project. In agile practices, on the other hand, the management of activities takes place in an iterative and incremental process, people do not make the development time and the other stakeholders of the project actively work together to understand the problem, in addition to keeping them prioritized as activities with greater business value. In this scenario, this research aims to develop a conceptual model based on competences that help the self-assignment of quick-time activities in an iteration. To achieve this goal, this research included two phases. In the first phase, a Systematic Literature Review (RSL) was carried out to identify how studies on project management practices approach the management of activities in an iteration. In the second phase of the research, field research was carried out, where the proposed model, derived from RSL, was implemented in a computational tool, and applied in a work environment in a company that operates without a service segment, with the development of its products using agile project management practices. The results contributed to reduce the problems related to measurement and self-designation of activities in associations under agile project management practices. The contribution of this dissertation is to offer a computational tool, as an aid tool in the process of self-assignment of activities of agile time in an iteration, since research carried out in the academic database does not have computational models compatible with the computational model presented. in this research. The research limitations for addressing only the Scrum and Kanban agile frameworks and limited to two-stage development activities. Future studies can be carried out to analyze the use of the computational tool in associations with agile practices at scale or with longer development times.

This study is part of the PPGP-UNINOVE Project Management research line, in the context of the research project on Agile and Hybrid Projects and the project financed by the CNPq Universal Notice. From this study, a software registry and a utility model were produced.

Keywords: Activity management; predictive project management, agile project management; personal skills; interpersonal skills

LISTA DE ABREVIATURAS E SIGLAS

CR – Criticidade

CO – Complexidade

GAP – Gestão Ágil de Projetos

PMBOK – *Project Management Body of Knowledge*

PMI – *Project Management Institute*

PMP – *Project Management Professional*

RSL – Revisão Sistemática da Literatura

TI – Tecnologia da Informação

WoS – *Web of Science*

LISTA DE TABELAS

Tabela 1. Modelo para as dimensões <i>hard skills</i> e <i>soft skills</i>	26
Tabela 2.Principais diferenças no processo de estimativa de atividades nos métodos preditivos e ágeis.....	29
Tabela 3. Participantes entrevistas e <i>focus group</i>	38
Tabela 4 Categoria de análise.....	43
Tabela 5. Lista de competências e autores	45
Tabela 6 Integrantes do Time A.....	59
Tabela 7. Integrantes do Time B	60
Tabela 8. Competências dos integrantes do Time de Desenvolvimento	64
Tabela 9. Avaliação de competências de integrantes do Time de Desenvolvimento	65
Tabela 10. Questionário de avaliação de Criticidade e Complexidade	67
Tabela 11. Definição de pesos para papéis	68
Tabela 12. Avaliação inicial da atividade	68
Tabela 13. Avaliação com pesos de papéis aplicados	69
Tabela 14. Avaliação ponderada da atividade	69
Tabela 15. Processo <i>feature engineering</i>	71
Tabela 16. Critérios de avaliação de competências	74
Tabela 17. Modelo para as dimensões <i>hard skills</i>	80
Tabela 18. Modelo para as dimensões <i>soft skills</i>	80
Tabela 19. Atividades avaliadas	83
Tabela 20. Avaliação de Criticidade e Complexidade.....	84
Tabela 21. Normalização das avaliações de Criticidade e Complexidade.....	85
Tabela 22. Classificação de Complexidade e Criticidade	86
Tabela 23. Ranking de Competências.....	89

LISTA DE FIGURAS

Figura 1. Resultados da pesquisa nas fontes de dados	35
Figura 2. Características de Complexidade e Criticidade – <i>focus group</i>	39
Figura 3. Evolução da produção literária sobre o tema	41
Figura 4. Quantidade de artigos científicos publicados por periódico	42
Figura 5. Relação das competências para times ágeis	47
Figura 6. Ferramentas para times ágeis	50
Figura 7. Modelo conceitual das competências técnicas e interpessoais dos times ágeis para a autogestão de atividades de uma iteração	55
Figura 8. Infográfico: Construção e aplicação do modelo.....	57
Figura 9. Modelo conceitual das competências técnicas e interpessoais dos times ágeis para a autogestão de atividades de uma iteração	62
Figura 10. Etapas de implementação da ferramenta computacional	63
Figura 11. Processo de avaliação dos integrantes do time de desenvolvimento pelo gestor da equipe.	65
Figura 12. Processo de avaliação de Criticidade e Complexidade de uma atividade do <i>backlog</i>	66
Figura 13. Desenvolvimento do modelo de classificação de Criticidade e Complexidade.....	70
Figura 14. Entrada de dados de Criticidade e Complexidade.	72
Figura 15. Análise de classificação de Criticidade e Complexidade.....	73
Figura 16. Modelos de classificação de Criticidade e Complexidade implementados em Python.....	78
Figura 17. Análise de competências do time	87
Figura 18. Análise de competências técnicas	88
Figura 19 – Tela inicial do software	93
Figura 20 – Lista de Analistas avaliados	93
Figura 21 - Cadastro de competências dos Analistas	94
Figura 22 – Avaliação de complexidade e criticidade das atividades	95
Figura 23 – Apresentação da classificação dos analistas	96

Sumário

1	INTRODUÇÃO	15
1.1.	PROBLEMA DE PESQUISA	18
1.2.	OBJETIVOS	21
<i>1.2.1.</i>	<i>Objetivo Geral</i>	<i>21</i>
<i>1.2.2.</i>	<i>Objetivos Específicos</i>	<i>21</i>
1.3.	JUSTIFICATIVA PARA ESTUDO DO TEMA	21
1.4.	ESTRUTURA DO TRABALHO.....	24
2	REFERENCIAL TEÓRICO	25
2.1	COMPETÊNCIAS PARA GESTÃO DE PROJETOS	25
2.2	GERENCIAMENTO DE ATIVIDADES EM GESTÃO DE PROJETOS	27
2.3	COMPETÊNCIAS PARA O GERENCIAMENTO ÁGIL DE ATIVIDADES.	30
3	MÉTODO E TÉCNICAS DE PESQUISA	33
3.1	DELINEAMENTO DA PESQUISA	33
3.1.1	Procedimentos da Revisão Sistemática de Literatura.....	33
3.1.2	Procedimentos da pesquisa de campo	36
4	RESULTADOS E APRESENTAÇÃO DO MODELO CONCEITUAL.....	41
4.1	RESULTADOS DA REVISÃO SISTEMÁTICA DA LITERATURA.....	41
4.1.1	Competências	44
4.1.2	Ferramentas	48
4.1.3	Relação com a atividade.....	50
4.1.4	Características do time de desenvolvimento	52
4.1.5	Proposta do modelo conceitual.....	53
5	CONSTRUÇÃO E APLICAÇÃO DA FERRAMENTA COMPUTACIONAL	57
5.1	CONTEXTO DE APLICAÇÃO DO MODELO PROPOSTO	58

5.1.1	Caracterização da organização	58
5.1.2	Descrição dos times de desenvolvimento.....	58
5.1.3	Descrição da ferramenta computacional	61
6	RESULTADOS E DISCUSSÃO	77
6.1	CRIAÇÃO DOS MODELOS DE ANÁLISE	77
6.2	ANÁLISE DAS AVALIAÇÕES DO TIME DE DESENVOLVIMENTO	79
6.3	COLETA E AVALIAÇÃO DOS DADOS DA ITERAÇÃO	82
6.4	PROCESSAMENTO DOS DADOS E ANÁLISE DOS RESULTADOS	83
6.5	<i>OUTPUT</i> DO MODELO: CLASSIFICAÇÃO DOS ANALISTAS	86
7	CONCLUSÃO.....	90
7.1	CONSIDERAÇÕES FINAIS	90
7.2	CONTRIBUIÇÕES PARA PRÁTICA	92
7.3	CONTRIBUIÇÕES PARA TEORIA.....	97
7.4	LIMITAÇÕES E PROPOSTA DE ESTUDOS FUTUROS	98
	REFERÊNCIAS	99
	APÊNDICE A – PROTOCOLO DE PESQUISA E COLETA DE DADOS.....	111
	APÊNDICE B – ROTEIRO DE ENTREVISTAS	114
	APÊNDICE C - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....	115
	APÊNDICE D – FORMULÁRIO DE PESQUISA.....	117

1 INTRODUÇÃO

Um projeto pode ser definido como um esforço temporário que objetiva a criação de um produto, serviço ou resultado único (PMI, 2017). Para Dinsmore e Cabanis-Brewin (2009), a Gestão de Projetos (GP) trata-se da aplicação de técnicas para que o objetivo do projeto seja alcançado. Complementando essa definição, Meredith e Mantel Jr. (2011) destacam que as técnicas e ferramentas de GP têm como finalidade assegurar que os objetivos previamente definidos para que um determinado projeto seja respeitado, leve em consideração o prazo e os custos. Dentro da gestão de projetos, existem as abordagens tradicional e ágil, que possuem ferramentas e processos específicos para cada etapa do projeto. Cabe destacar que, nesta pesquisa, a nomenclatura abordagem tradicional será empregada para tratar de metodologias orientadas ao planejamento como o *stage gate* (Cooper, Edgett, & Kleinschmidt, 2002). Para tratar da abordagem ágil serão reconhecidas as metodologias influenciadas, principalmente, pelo manifesto ágil (Beck *et al.*, 2001).

Tratando-se da realização efetiva dos projetos, Barcaui (2012) define que as práticas de GP requerem o envolvimento de pessoas no planejamento, nas organizações, nos processos de controle e monitoramento de atividades e das pessoas destinadas a ele. No que se refere ao uso de pessoas, Penha *et al.* (2014) destacam que, na gestão de projetos tradicional, o cronograma é a ferramenta que garante o planejamento das pessoas para a execução das diversas atividades de um projeto. Em sua forma mais tradicional, o PMI (2017) salienta que para a construção de um cronograma são necessários: (1) lista de atividades e suas respectivas especificações, interdependências e estimativas para sua realização; (2) lista de pessoas disponíveis agrupadas por competências; (3) informações do escopo, da fase de planejamento de custos, dos riscos e etc.

Portanto, em paralelo às práticas tradicionais de gestão de projetos, a publicação do Manifesto Ágil em 2001 apresentou abordagens alternativas para a comunidade internacional para o gerenciamento e execução de projetos de softwares – origem do termo Gerenciamento Ágil de Projetos (GAP) (Beck *et al.*, 2001; Amaral *et al.*, 2011). Para Babenko *et al.* (2019), a origem do GAP está fundamentada em um processo iterativo e incremental, no qual, os membros do time e os demais *stakeholders* do projeto trabalham ativamente juntos para entenderem o domínio do problema, identificarem o que precisa ser construído, além de manterem priorizados os requisitos com maior valor de negócio.

A GAP, apesar de existir há quase duas décadas, ainda é tratada como algo novo dentro de muitas organizações. Diferente da gestão de projetos tradicional, como ressalta Grapenthin

et al. (2015), em projetos ágeis deve existir um planejamento inicial capaz de sustentar todo o desenvolvimento da iteração, partindo do pressuposto de que as equipes são auto-organizadas. Nessa mesma linha, Beck *et al.* (2001) apontam que é comum que um projeto ágil seja dividido em fases menores, onde o planejamento é realizado para cada uma das iterações, visando ao final, entregar mais valor ao negócio e aos clientes. Para isso, o planejamento de cada fase de um projeto ágil é realizado seguindo os ritos do *framework* que foi escolhido para a execução do projeto.

Atualmente, existem diversos *frameworks* que objetivam auxiliar o GAP, conforme destacado por Penha, da Silva e Russo (2020), podem ser destacados o *Scrum* e o *KanBan*. O *Scrum* utiliza ciclos de entrega com característica de *time box*, possuindo duração fixa para as atividades de cada iteração. Cada iteração deve possuir o prazo de quinze ou trinta dias, denominados de *sprints*. Quanto às responsabilidades, os ritos e papéis são bem definidos e exercem influências distintas nos projetos. Ao tratar dos papéis, o *Scrum Master* (SM) possui o papel de garantir que as boas práticas do *Scrum* sejam empregadas por todos os envolvidos no projeto. O *Product Owner* (PO) é o responsável por interagir com o *stakeholder*, a fim de garantir que as entregas feitas pelo time de desenvolvimento agreguem valor ao negócio. Por fim, o *Development Team* (DT), tem a característica de ser um grupo multifuncional auto-organizado com a responsabilidade de gestão das atividades de uma iteração (Colomo-Palacios *et al.*, 2012).

O *framework KanBan* tem como principal objetivo apresentar uma visão panorâmica das atividades a serem realizadas para toda a equipe (Hoda & Murugesan, 2016). As atividades são dispostas em um *backlog*¹ da interação, onde os integrantes do time de desenvolvimento são responsáveis pela retirada de uma atividade do *backlog* e por colocá-la em execução sob sua responsabilidade. No *KanBan* o gerenciamento da atividade é de responsabilidade de cada integrante do time.

Tratando-se do gerenciamento de atividades, o processo ocorre de maneira distinta quando comparado ao gerenciamento tradicional frente ao ágil. No gerenciamento de projetos tradicional, Kerzner (2009) define que o gerenciamento de atividades é realizado por meio de um cronograma, onde a partir de um conjunto de restrições, uma pessoa é alocada para executar uma determinada atividade. Para destacar a importância do desenvolvimento do cronograma nos projetos tradicionais, Jones (1994) afirma que a elaboração de cronogramas exagerados ou

¹ *Backlog* da interação é um conjunto de atividades planejadas de forma estratégica para a execução dentro de uma iteração.

insensatos são, possivelmente, o principal fator negativo do desenvolvimento de produtos ou serviços nas organizações.

Ao contrário do que é praticado na metodologia tradicional, no GAP, ao início de cada ciclo de desenvolvimento, as atividades já possuem detalhamento e especificação definidos, mas não existe uma atribuição prévia de uma determinada pessoa do time de desenvolvimento para execução de cada atividade. Entretanto, segundo Grapenthin *et al.* (2015), selecionar as atividades para o *backlog* é uma atividade complexa, pois o time precisa ter conhecimentos detalhados de diversos aspectos, tanto técnicos quanto de negócios para poder mensurar a quantidade de esforço que cada atividade irá demandar (Gren *et al.*, 2018).

Para o processo de atribuição das atividades, o time de desenvolvimento que utiliza metodologias ágeis, deve ser auto-organizado (Gjøystdal & Karunaratne, 2020), o que significa que não há uma centralidade de comando. Em relação ao comportamento do time, Medeiros *et al.* (2020) destacam que cada pessoa deve selecionar as atividades que serão executadas na iteração atual, de acordo com as competências individuais de cada pessoa em relação à criticidade e à complexidade de realização de cada atividade. Para Pascoa, Telha e Santos (2019), a criticidade para a realização de uma atividade está relacionada com a realização de uma atividade até a sua entrega final, devendo possuir todos os requisitos de negócio previamente acordados com o cliente. Em relação à complexidade, López-Martínez *et al.* (2018) caracterizam que a complexidade para a realização de uma atividade está condicionada aos aspectos técnicos de estimativas, dificuldade técnica e entendimento funcional de um requerimento, ou seja, relacionados ao como fazer.

Os requerimentos iniciais de um projeto no contexto das práticas ágeis surgem durante a fase de ideação com o cliente (Ganesh & Thangasamy, 2012). A partir da sua relação com a necessidade de negócio, os requerimentos compõem o *backlog* do produto (Fontana *et al.*, 2015; Licorish & MacDonell, 2015; Hsieh & Chen, 2015). Para Jia *et al.* (2019), a criação do *backlog* é considerada uma atividade crítica dentro de times que utilizam a metodologia ágil, pois, caso uma atividade não seja compreendida, ou resulte no desmembramento para outras atividades, certamente o time terá mais trabalho do que a sua capacidade mensurada, levando dessa forma a incerteza para a entrega da iteração.

Outro ponto que distingue o gerenciamento de atividades nas práticas tradicionais e ágeis está relacionado ao sequenciamento, priorização e dependência das atividades (Birgin, Ferreira, & Ronconi, 2015). Tratando-se de criticidade nas práticas tradicionais de GP, Laslo (2010) determina que ela esteja relacionada à data de início das atividades de um determinado projeto que satisfaça as relações de precedência e prioridade de pessoas com o objetivo de

minimizar o prazo total do projeto. Esse cenário de planejamento das atividades é asseverado por Hartmann (2013) e Fang *et al.* (2015). Os autores destacam que a relação de dependência das atividades define se uma atividade vai ser iniciada após a conclusão das atividades antecessoras, em um processo em que cada uma delas não pode ser interrompida até o término de sua duração. Nesse cenário, Mingozi *et al.* (1998) explicam que existe a formação de conjuntos de pessoas de forma variada, em que uma quantidade limitada e permanente de pessoas está disponível ao longo do ciclo de vida do projeto. Segundo os autores, os efeitos desta variabilidade são as restrições das pessoas que limitam as atividades que podem ser executadas simultaneamente, em cada período de duração de cada projeto.

Para o mesmo cenário inserido no GAP, Golfarelli, Rizzi e Turricchia (2013) sustentam que a pessoa responsável por maximizar o valor do produto ao cliente, revisa e prioriza os requisitos (*user stories*), que são organizados em um *backlog*. Em relação aos *frameworks* ágeis, Lindsjörn *et al.* (2016) definem que no Scrum essa atividade é realizada pelo *Product Owner*, enquanto no *KanBan* essa responsabilidade é compartilhada por todos os integrantes do time. Com base no que foi apresentado até aqui, nesta pesquisa será entendido que o time de desenvolvimento possui a responsabilidade de compreender e selecionar um conjunto de atividades de maior valor no *backlog* que julguem capazes de implementar dentro de cada iteração.

Neste contexto, surge a oportunidade de compreender como a gestão das atividades pode ser melhorada, pois é preciso destacar que há uma diferença entre as abordagens preditiva e ágil, sendo que esta pesquisa se delimita a estudar a gestão de atividades da abordagem ágil dentro de suas iterações.

1.1. PROBLEMA DE PESQUISA

Nas práticas tradicionais de GP, o controle e o monitoramento das atividades são efetuados por meio do desenvolvimento do cronograma. O desafio está nas competências que o gestor precisa adequar com a duração das atividades, além dos custos, prazos de entrega e a relação de precedência entre as atividades (PMI, 2017). Para as práticas ágeis, Gren, Knauss e Stettina (2018) apontaram que a responsabilidade é do time, em efetuar o controle e monitoramento das atividades em uma iteração. Esse processo de auto-organização das atividades está centrado na autodesignação das atividades. Para isso, como ressalta Kakar (2017), o time deve possuir as competências necessárias e ser auto-organizado para que uma

determinada atividade seja atribuída por uma pessoa sem impactar o andamento e dependências das demais atividades planejadas.

No contexto previamente apresentado, há uma distinção entre abordagens e a gestão de atividades. A dependência de atividades nas práticas de GP tradicionais é marcada pela alocação de uma determinada pessoa por um prazo e custo previamente planejados (PMI, 2017). Caso alguma alteração relacionada à composição da atividade seja evidenciada, automaticamente as atividades dependentes poderão sofrer desvio de prazo, impactando negativamente o desempenho do projeto (Archer & Ghasemzadeh, 1998; Cooper, 2001; Penha *et al.*, 2014).

Para as práticas ágeis, a pesquisa de Gren *et al.* (2018) destacam que durante uma iteração, o maior desafio de um gestor (no caso do Scrum de um Scrum Master) é identificar a relação de dependência de atividades e o processo de monitoramento do progresso de cada atividade sendo executada por cada pessoa do time. Os autores salientam que os gráficos atuais não representam esse cenário, aumentando o risco de impedimentos de atividades e ou possíveis cancelamentos de entregas de uma determinada iteração. Por fim, Beck *et al.* (2001) e Amaral *et al.* (2011) apontaram que devido às competências dos integrantes dos times ágeis, em média 26% das atividades sofrem algum tipo de alteração durante a iteração. Isso acontece porque o time não compreende corretamente o escopo funcional e técnico da atividade, o que leva ao aumento dos riscos de atraso e cancelamento de entrega, além de ser contra os propósitos de entrega de valor ao cliente das práticas ágeis (Beck *et al.*, 2001; Amaral *et al.*, 2011).

Portanto, a relação entre atividade e a pessoa do time que irá executá-la passa a ter outro comportamento nos métodos ágeis (Ioannou *et al.*, 2020; Luong, Sivarajah & Weerakkody, 2019). Nos métodos tradicionais, para a realização de uma determinada atividade, a pessoa com o papel de gerente do projeto estabelece quem será a pessoa que irá realizar determinada atividade, em um determinado período e custo associado, levando em consideração as competências (*hard skills* e *soft skills*) de cada pessoa em relação à complexidade de execução de cada atividade (PMI, 2017). Nos métodos ágeis, essa relação é autodesignada pelos membros do time de desenvolvimento. O time deve ser auto-organizado a ponto de selecionarem a atividade de acordo com suas competências, sem perder o foco na iteração. Desse modo, o time deverá apresentar competências técnicas e interpessoais para identificar a complexidade relacionada à execução de cada atividade e o seu nível de criticidade em relação ao término de uma iteração.

A complexidade está relacionada às competências necessárias que cada pessoa do time deve possuir para a execução de uma atividade em uma iteração. O time deverá apresentar competências como capacidade de abstração (Gren, Knauss & Stettina, 2018) e análise de

requerimentos (Medeiros *et al.*, 2020), necessárias para criar generalizações e abstrações das necessidades de negócio dos clientes, tornando tangível o desenvolvimento de requerimentos abstratos em funcionalidades sistêmicas. De posse do entendimento, o time deverá decompor as necessidades de negócios dos clientes (Tan & Vicente, 2019) em atividades mensuráveis para execução em uma atividade (Ramirez-Mora, Oktaba & Perez, 2020). Para tanto, cada pessoa do time deverá possuir conhecimento em linguagens de programação, assim como padrões de projeto e padrões arquiteturais voltadas ao desenvolvimento de soluções durante uma iteração (Pascoa, Telha & Santos, 2019).

A criticidade está relacionada às competências necessárias para que o time possua o senso de organização de acordo com a sua rotina de trabalho, de forma que seja possível cumprir com a execução das atividades que o mesmo se comprometeu a realizar em uma iteração (Gjoystdal & Karunaratne, 2020). Para a realização das atividades de uma iteração, o time deverá possuir habilidade de reconhecer a importância de eventos, atividades e comportamentos do time que podem afetar a iteração, e assim, os requerimentos de negócios cliente previamente comprometidos para entrega ao final da iteração (Grapenthin *et al.*, 2015).

Vale ressaltar que no processo de identificação das complexidades e criticidades por um time em uma interação, o time ainda deverá apresentar algumas competências essenciais, desde a fase de planejamento até a fase de finalização de uma iteração. O time deverá apresentar a capacidade de negociar a adição, extinção ou modificações de requerimentos de negócios e atividades de uma iteração (Adolph, Kruchten & Hall, 2012), apresentando, em certos momentos, a capacidade de influenciar pessoas do time para alcançar objetivos pessoais ou organizacionais (Busse & Weidner, 2020). Para que tais competências obtenham resultados positivos em uma iteração, o time deverá, a cada momento, transmitir uma mensagem de forma clara e objetiva sem perda de informações por meio de uma comunicação efetiva (Gjoystdal & Karunaratne, 2020).

Como visto, cada iteração tem seu relacionamento com a entrega de atividades relacionadas ao negócio do cliente. Desse modo, o time deverá possuir habilidade de se adaptar, de forma rápida, às mudanças que ocorrem durante uma iteração. Ao desenvolver essa habilidade, o integrante deixa de ver as mudanças de forma negativa e passa a enxergar a mudança como ação estratégica para alcançar resultados e entregar valor ao final da iteração Colomo-Palacios *et al.* (2012).

Deste modo, com a adoção de *frameworks* ágeis o time passa a ser o responsável pela autodesignação de atividades do *backlog*, seguindo uma priorização que visa à entrega de valor para o negócio. A execução dessas atividades requer conhecimentos e habilidades que, segundo

Gren *et al.* (2018), podem ser desenvolvidas ao longo do tempo e podem impactar no projeto. Assim, a partir deste cenário marcado pela validação de conhecimentos e habilidades frente ao gerenciamento de atividades em projetos ágeis, esta pesquisa buscará responder a seguinte questão de pesquisa: **Como o uso de um modelo baseado em competências pode contribuir na autodesignação de atividades de um time ágil em uma iteração?**

1.2. OBJETIVOS

1.2.1. Objetivo Geral

Desenvolvimento de um modelo baseado em competências que auxilie a autodesignação de atividades de um time ágil em uma iteração.

1.2.2. Objetivos Específicos

Como objetivos específicos, esta pesquisa pretende:

- Descrever o processo de gerenciamento de atividades de um time ágil durante as iterações;
- Identificar as competências necessárias às pessoas de um time ágil para a autodesignação das atividades em uma iteração;
- Verificar a adequação das competências das pessoas para a autodesignação das atividades de um time ágil em uma iteração;
- Construir um modelo baseado em competências que auxilie na autodesignação de atividades de time ágil em uma iteração;
- Validar o modelo baseado em competências que auxilie na autodesignação de atividades de um time ágil em uma iteração.

1.3. JUSTIFICATIVA PARA ESTUDO DO TEMA

O gerenciamento de projetos segue uma dinâmica de gestão de atividades relacionada aos *frameworks* adotados. Durante o gerenciamento de atividades, as técnicas GPA determinam

que as pessoas devem possuir todas as competências técnicas e gerenciais (Medeiros *et al.*, 2020), além de serem auto-organizadas (Grapenthin *et al.*, 2015) para efetuarem a atribuição e o monitoramento do progresso de cada uma das atividades em uma iteração (Lindsjörn *et al.*, 2016). Assim, acompanhar a autodesignação das atividades para um time ágil pode ser vista como uma atividade árdua e suscetível a falhas para os gerentes de projeto ágeis (Penha *et al.*, 2020).

Jorgensen e Molokken-Ostvold (2004) destacam que entre 30% e 40% dos projetos apresentam algum tipo de falha associado no processo de estimativa de esforços para o planejamento e realização de atividades. Nesse contexto, Vieira, Antonioli e Argoud (2017) apontam que o refinamento na especificação dos requerimentos das partes interessadas deve ser utilizado pelas organizações como medida de redução de tais falhas.

Assim, pode-se apontar que existe uma relação entre a incerteza contida na especificação de requerimentos e o desempenho dos projetos. Para Jiang (2009), uma das formas para minimizar as incertezas é atuando de maneira efetiva desde a coleta das ideias iniciais e do desdobramento das ideias em atividades, no momento em que o requerimento foi coletado e definido pelos responsáveis pela gestão da atividade. Ainda sobre a relação com o desempenho dos projetos, Doherty, Ashurst e Peppard (2012) indicam que os processos envolvidos na gestão das atividades podem variar entre as empresas devido às relações de cada empresa com suas partes interessadas.

Para Cervone (2006) cada grupo de partes interessadas pode ter um nível de abstração e entendimento em relação aos requerimentos do projeto, tornando assim a necessidade de um alinhamento constante ao gerenciamento de atividades. Esse cenário pode se tornar benéfico diante das práticas de GPA, devido ao fato de as práticas ágeis possuírem a capacidade de lidar com a mudança dos requerimentos das partes interessadas (Dingsoyr *et al.*, 2012), tornando assim o gerenciamento de atividades com métodos ágeis uma atividade desafiadora à previsibilidade em questões orçamentárias (Attarzadeh & Ow, 2010; Bilgaiyan *et al.*, 2017).

Nesse ponto, as técnicas para estabelecer estimativas de atividades passam a ganhar papel de destaque nos processos ágeis. As métricas utilizadas para realizar estimativas e gerenciamento das atividades nas práticas ágeis são diferentes das estimativas tradicionais já consolidadas (Aranha & Cardoso, 2017). Em relação às ferramentas, Usman *et al.* (2014) destacam que a ferramenta mais utilizada para estabelecer as estimativas de atividades nas práticas ágeis é o *Planning poker*. Para os autores, o *Planning poker* é mais preciso quando utilizado por um time com profissionais experientes.

O *Planning Poker* é uma técnica baseada no consenso do time de desenvolvimento para estimar os requisitos de negócio do projeto, ou histórias de usuários em ambientes ágeis (Alhamed & Storer, 2020). Em relação à dinâmica do time, no *Planning Poker* cada integrante do time recebe um conjunto de cartas correspondente à sequência de *Fibonacci*². No passo seguinte, uma história de usuário é selecionada para análise pelo time de desenvolvimento e cada integrante do time apresenta uma carta virada para baixo. O valor da carta corresponderá aos pontos que o integrante considera viável para que a história do usuário seja realizada. Caso ocorra uma diferença substancial entre as cartas do time, os integrantes deverão explicar o valor apontado por cada um em relação à complexidade para a realização da atividade em questão e as cartas são apresentadas novamente até que haja consenso entre o time e uma estimativa seja adotada (Mahnič & Hovelja, 2012).

Embora o *Planning Poker* seja largamente utilizado, algumas inconsistências podem ocorrer durante o processo de estimativas realizadas. Alguma imprecisão pode ocorrer devido ao fato do uso da sequência de *Fibonacci* para unidades de medida de tempo (Gamba & Barbosa, 2010). Outras inconsistências podem ocorrer em relação ao consenso do time. Para López-Martínez *et al.* (2018) dois cenários podem ocorrer: (i) a pessoa encarregada pela gestão das histórias dos usuários pode notar que não houve um claro entendimento da complexidade da realização de uma atividade por todo o time de desenvolvimento; e (ii) o time de desenvolvimento pode possuir baixo nível de competências técnicas e estarem pensando em soluções diferentes para a execução de uma determinada história. Todos esses cenários apresentam falhas que podem ocorrer ao utilizar o *Planning Poker*, o que impacta no gerenciamento de atividades de projetos ágeis.

Para procurar resolver esse problema, algumas soluções computacionais viáveis são apresentadas. Kaushik Tayal e Tayal (2020) sugerem a utilização de redes neurais e gráficos de velocidade. Colomo-Palacios *et al.* (2012) destacam o uso de lógica *fuzzy* e análise semântica. Para fase final da iteração, Lwakatare *et al.* (2019) sugerem que o DevOps é uma ferramenta que auxilia no aumento de qualidade da entrega de uma ou mais iterações, servindo ainda como uma ferramenta estratégica para maior entrega de valor ao cliente. Vale esclarecer que DevOps é o acrônimo de *Development and Operations*. De acordo com Lwakatare *et al.* (2019) e Žužek *et al.* (2020), o objetivo do uso do DevOps nas organizações está condicionado à busca de entrega de valor ao cliente por meio da conclusão das atividades de uma iteração.

² Fibonacci é uma sequência matemática em que cada número seguinte corresponde à soma dos dois anteriores.

Diante do exposto, esta pesquisa se justifica e se apresenta relevante para as áreas acadêmica e profissional, pois visa contribuir com prescrições que auxiliem na gestão de atividades pela perspectiva das competências das pessoas envolvidas em projetos ágeis. Para tanto, foi observado o ciclo de vida de uma atividade desde a sua fase inicial de concepção pelas partes interessadas, seu processo de desmembramento e estimativas na fase de planejamento, procurando contribuir em um planejamento eficaz entre as atividades planejadas e as competências técnicas e interpessoais do time de desenvolvimento. Os resultados apresentados nesta pesquisa contribuirão nas iterações, possibilitando a melhor autodesignação de uma atividade por uma pessoa do time, levando em consideração a complexidade para a realização das atividades e a sua respectiva relação com a criticidade durante a fase de autogerenciamento.

1.4. ESTRUTURA DO TRABALHO

Esta pesquisa está organizada em sete capítulos. O capítulo 1 aborda a introdução ao tema de discussão da pesquisa, a problematização, o objetivo geral e os específicos, seguida da justificativa. O capítulo 2, por sua vez, apresenta a fundamentação teórica abordando as áreas de gestão de projetos, modelos ágeis de gestão de projetos, *hard skills* e *soft skills* dos integrantes do time de desenvolvimento. Na sequência, o capítulo 3 contempla a metodologia usada para se alcançar o objetivo da pesquisa. No capítulo 4 são apresentados os resultados da RSL e é realizada a apresentação do modelo conceitual. No capítulo 5 é apresentada a construção da ferramenta computacional gerada a partir do modelo proposto. O capítulo 6 aborda os resultados e discussão dos testes realizados. Por fim, no capítulo 7 são realizadas as conclusões, e na sequência, são apresentadas as referências utilizadas na realização da pesquisa.

2 REFERENCIAL TEÓRICO

A seguir é apresentada uma breve revisão da literatura sobre as competências relacionadas à gestão de projetos, sobre as práticas para o gerenciamento de atividades em projetos e no contexto das práticas ágeis.

2.1 COMPETÊNCIAS PARA GESTÃO DE PROJETOS

No contexto gerencial, a abordagem das competências está vinculada às áreas da Administração Estratégica e da Economia, com início na década de 1970, em que a velocidade das mudanças nas organizações fazia com que cargos baseados em descrição de função se tornassem rapidamente obsoletos (Baumotte *et al.*, 2013). Como resposta, Hollenbeck, Mccall e Silzer (2006) destacaram que a solução encontrada foi desenvolver processos de seleção baseados no requerimento de conjuntos de competências, ao invés de uma descrição de funções de onde surgiu a abordagem focada nas competências das pessoas.

Em relação aos estudos de cunho gerencial, McClelland (1973) é considerado o pioneiro neste assunto. Sua pesquisa aponta o impacto direto das competências do gerente em relação ao desempenho dos projetos. Para Ahsan, Ho e Khan (2013), os pressupostos de McClelland destacam as competências apresentadas no gerente do projeto que, de certo modo, podem influenciar outras pessoas. Em um contexto mais atual, as competências estão associadas às dimensões de Conhecimento, Habilidade e Atitude, conceito na literatura denominado CHA (Boyatzis, 1982; Crawford, 2005; Müller & Turner, 2010). Portanto, as competências estão relacionadas à combinação de Conhecimento (resultado de desenvolvimento pessoal), da Habilidade (desempenho de atividade) e Atitude (relacionada à motivação e proatividade) das pessoas (Partington, 2007; Turner & Müller, 2005).

Com a evolução dos estudos sobre competências, novas categorizações foram criadas para compreendê-las. Neste sentido, Dutra, Hipólito e Silva (2000) salientam a necessidade de tratar das competências interpessoais e técnicas das pessoas, denominadas *soft skills* e *hard skills*. Os autores ainda sugerem que a partir desse contexto frente às aplicações, as competências se relacionam ao ambiente de gerenciamento de projetos.

Em relação às pessoas envolvidas no gerenciamento de projetos, Boterf (2003) define que as competências são condizentes ao ato de uma pessoa em assumir responsabilidades frente a diversos cenários. Nesse contexto, Mills *et al.* (2002) destacam que as pessoas sob um ambiente

de gerenciamento de projetos podem combinar determinadas competências pessoais com a presença de conhecimentos, habilidades e atitudes, com relação direta nos resultados dos projetos.

No contexto de gerenciamento de projetos, as competências *hard skills* e *soft skills* são empregadas de maneira ampla. Crawford e Pollack (2004) desenvolveram um modelo de interação que auxilia a avaliação do projeto com a análise da influência de aspectos das competências *hard skills* e *soft skills*, apresentado na Tabela 1. Como destaque, Atkinson, Crawford e Ward (2006) salientam que o modelo de Crawford e Pollack (2004) apresenta diversas abordagens para lidar com as incertezas e as expectativas dos *stakeholders* dependendo do ambiente do projeto. O modelo destaca as competências técnicas *hard skills* e *soft skills* existentes nos *stakeholders* durante o gerenciamento de projetos preditivos.

Tabela 1. Modelo para as dimensões *hard skills* e *soft skills*

Dimensões	<i>Hard Skills</i>	<i>Soft Skills</i>
Clareza de objetivos e metas	Claros	Ambíguos
Tangibilidade de objetivos e metas	Tangíveis	Abstratos
Medidas de sucesso	Quantitativas	Qualitativas
Permeabilidade do projeto	Sem influência externa	Alta influência externa
Quantidade de opções de solução	Refinamento de solução única	Exploração de soluções alternativas
Grau de participação e papel do praticante	Praticante especialista / Sem participação dos <i>stakeholders</i>	Praticante facilitador / Alto envolvimento dos <i>stakeholders</i>
Expectativa dos <i>stakeholders</i>	Valorização do desempenho técnico e eficiência / Gestão por monitoramento e controle	Valorização do relacionamento, cultura e significado / Gestão por meio de negociação e discussão

Fonte: Adaptado de Crawford e Pollack (2004)

Segundo Crawford e Pollack (2004) o uso prático do modelo permite analisar e discutir as competências *hard skills* e *soft skills* no ambiente de projetos, obtendo como resultado a capacidade de estabelecer um contexto em relação à aplicação de abordagens rígidas rotineiramente adotadas e utilizadas no gerenciamento de projetos preditivos. Por fim, os autores destacam que as competências supracitadas podem variar de acordo com o contexto de gerenciamento de projetos. Os autores salientam ainda que abordagens mais tradicionais tendem a ter o controle para lidar com a incerteza e gestão das expectativas. Por outro lado, em outros ambientes, características mais relacionadas às *soft skills* podem ser encontradas de forma mais espontânea nas organizações.

Frente ao apresentado nessa seção, pode-se notar, tanto em ambientes preditivos ou ágeis, a necessidade do nivelamento das competências *hard skills* e *soft skills* nas pessoas envolvidas nas práticas de gerenciamento de projetos. O nivelamento das competências, como destacado por Engwall e Jerbrant (2003), contribui para que as pessoas em posição de gestão de projetos possam efetuar o gerenciamento das atividades, identificando uma pessoa do time que possua as competências necessárias para a realização da atividade. Atienza (2017) ressalta que a realização de um gerenciamento de atividades de maneira eficaz durante todo o ciclo de desenvolvimento das iterações, pode contribuir para aumentar a eficiência dos projetos em relação a prazo e custos. Os aspectos relacionados a gerenciamento das atividades dos projetos são explorados na próxima seção.

2.2 GERENCIAMENTO DE ATIVIDADES EM GESTÃO DE PROJETOS

As práticas de gestão de projetos são utilizadas para promover a atribuição de pessoas em atividades, isso sendo feito em consonância ao planejamento e controle de um conjunto limitado de pessoas (Meredith & Mantel Jr., 1985; Kezner, 2006). Desse modo, as pessoas envolvidas na gestão de projetos devem empregar técnicas, ferramentas, habilidades e conhecimentos no controle e monitoramento das atividades em um ambiente com presença de riscos e incertezas (PMI, 2017; Pinto & Kharbanda, 1996).

Tratando-se de gestão das atividades, dentro dos pressupostos preditivos, o cronograma é o artefato responsável pelo planejamento das pessoas distribuídas nas atividades de um projeto (PMI, 2017). Com o objetivo de aumentar o desempenho do uso das pessoas em atividades, Glenwright (2007) aponta que a pessoa responsável pelo gerenciamento do projeto é responsável pela elaboração do cronograma. Nesse cenário, Ichihara (2002) destaca que a elaboração do cronograma deve garantir uma alocação eficaz das pessoas procurando estabelecer a melhor relação “pessoa x atividade” mais próxima da combinação ótima possível. Para a elaboração do cronograma, Hartmann (1998) destaca que essa é uma atribuição realizada pela pessoa responsável pelo projeto e com o papel de minimizar o tempo total do projeto sem violar a relação de dependência entre as atividades e a restrição das pessoas disponíveis.

A relação de precedência é um fator crítico para as organizações. Mingozi *et al.* (1998) apontam que a relação de precedência determina se uma atividade vai ser iniciada após a conclusão das atividades predecessoras. Como resultado, uma atividade não pode ser interrompida até a finalização de sua duração. Desse modo, os autores elucidam que o número de pessoas a serem alocados em um determinado projeto é limitado durante o ciclo de vida do

projeto. Como resultado, as pessoas são limitadas a executarem as atividades que foram previamente alocadas.

O processo de elaboração do cronograma, segundo o PMI (2017), segue a seguinte ordem: (i) definição das atividades; (ii) sequenciamento das atividades; e (iii) estimativa da duração das atividades. A definição das atividades dos projetos corresponde ao processo de subdividir as distintas atividades a serem executadas e permitir que a pessoa responsável pelo cronograma estabeleça as atividades prioritárias e as que impactam na realização de outras atividades. O sequenciamento das atividades determina a sequência lógica de realização das atividades. Nesse ponto, as pessoas que desenvolverão as atividades devem ter ciência de tais relações, pois o atraso em atividades interdependentes causa impacto no tempo total de um projeto. Um ponto de atenção nessa etapa é que pessoas do time de desenvolvimento podem ficar ociosas ou sobrecarregadas, podendo impactar no aumento de custos nos projetos (Hartmann & Kolisch, 2000; Agarwal, Colak & Erenguc, 2011; Schutt *et al.*, 2013; Fang *et al.*, 2015). Por fim, a estimativa da duração das atividades tem o papel de estimar a quantidade de unidade de tempo para a realização de cada atividade.

O processo de mensuração de estimativas apresentadas sobre os métodos preditivos é diferente dos processos existentes nos métodos ágeis (PMI, 2017). A Tabela 2 apresenta as principais diferenças entre os fatores associados no processo de estimativa de atividades nos métodos ágeis e, conseqüentemente, a elaboração dos cronogramas nos métodos preditivos.

Em relação à elaboração de cronograma e ao exposto na Tabela 2, pode-se destacar que os métodos preditivos são aplicados quando os requerimentos e suas dependências das partes interessadas são conhecidos e passíveis de estimativas pela pessoa responsável pela gestão de projetos (dos Santos Soares, 2004; Schwaber, 2004; PMI, 2017). Em contrapartida, nos métodos ágeis, as mudanças ocorrem constantemente. Os requerimentos no contexto dos métodos ágeis sofrem alterações frequentes e devido ao ciclo de iterações frequentes, garantem a validação do requerimento durante o ciclo de vida de um requerimento (Eder *et al.*, 2015; PMI Agile Alliance, 2017; Shekhar & Kumar, 2016).

A partir desse ponto, podemos compreender a distinção da relação da atividade com os métodos ágeis e os métodos preditivos (Rahman & Rutz, 2015; Neto *et al.*, 2019; Mohanarangam, 2020). O destaque fica para o escopo das atividades. Bosch e Bosch-Sijtsema (2011) apontam que os métodos tradicionais possuem a característica de planejamento e controle para a realização do total das atividades, para um cenário onde o escopo será decomposto até que ele entregue valor ao negócio do cliente e seja possível a realização em uma iteração nos métodos ágeis. Outro ponto importante nessa visão, é que os métodos ágeis,

por tratarem escopos menores para a realização das atividades, possuem maior maleabilidade em relação às mudanças relacionadas às atividades do que os métodos tradicionais, reduzindo as incertezas e os possíveis impedimentos para a realização de uma determinada atividade (McIver, Lengnick-Hall, & Lengnick-Hall, 2018).

Tabela 2. Principais diferenças no processo de estimativa de atividades nos métodos preditivos e ágeis

Fatores	Métodos Preditivos	Métodos Ágeis	Referência
Cronograma	Deve ser planejado, definido, estimado, desenvolvido e controlado, garantindo que o projeto cumpra o cronograma acordado.	Não possui uma delimitação de tempo, sendo composto de eventos que geram melhorias incrementais no produto desenvolvido.	Eder et al. (2015); PMI (2017); Schwaber (2004); Shekhar e Kumar (2016); dos Santos Soares (2004).
Time	Não existe uma definição de papéis propriamente dita. A equipe é composta pelo gerente de projeto e o time de desenvolvimento.	Composta por papéis bem definidos conforme a metodologia utilizada. Entre eles podem existir <i>Product owner</i> , <i>Scrum master</i> , Time de Desenvolvimento, entre outros.	Eder et al. (2015); PMI (2017); Schwaber (2004); dos Santos Soares (2004).
Escopo	Definido com o maior nível de detalhamento possível, buscando previsibilidade e estabelecendo as atividades a serem desempenhadas. Existe uma lista de requisitos do projeto e uma ordem cronológica para que sejam realizados.	O escopo é descrito de forma a dar liberdade para a equipe criar o produto, sem grande detalhamento nas fases iniciais do projeto. Este documento é vivo, podendo ser alterado durante o seu desenvolvimento a qualquer momento.	PMI Agile Alliance (2017); Eder et al. (2015); Shekhar e Kumar (2016).
Custos	Precisam ser planejados, estimados e controlados, garantindo que o projeto não exceda o orçamento disponível.	Não possui uma diretriz clara neste aspecto.	Eder et al. (2015); PMI (2017); dos Santos Soares (2004).

Fonte: elaborado pelo autor, 2021.

A seleção e a priorização das atividades possuem relação com o planejamento e a execução de uma iteração (Golfarelli, Rizzi & Turricchia 2013). Segundo os autores, o time de desenvolvimento deve possuir alinhamento com a real necessidade do cliente, onde o objetivo é maximizar, dentro do possível, a maior entrega de valor ao cliente a cada iteração. Como nos métodos ágeis, o time tem o papel de ser auto-organizado, nesse ponto será importante que o time possua um papel colaborativo e efetivo, onde o conhecimento e as competências de todos os integrantes do time estejam claros para todos que irão participar de uma próxima iteração. O resultado dessa equação deve ser um *backlog* de atividades balanceadas, levando em consideração as competências do time e a complexidade para a realização das atividades (Trkman, Mendling, & Krisper, 2016; Sarpiri & Gandomani, 2017)

O desafio nesse contexto de práticas de gerenciamento ágeis está em identificar a relação de dependências entre as atividades em uma mesma iteração com o objetivo de garantir a maior entrega de valor ao cliente (Gjoystdal & Karunaratne, 2020). A partir do momento que o ciclo de desenvolvimento nas práticas ágeis é realizado por iterações, Belsis, Koutoumanos, Sgouropoulou (2014) destacam que as pessoas envolvidas na gestão de projetos ágeis precisam possuir competências essenciais para o gerenciamento de atividades durante o ciclo de desenvolvimento das iterações. Portanto, a relação entre competências e gerenciamento de atividades se torna um desafio para estudiosos e praticantes, o que é explorado na próxima seção.

2.3 COMPETÊNCIAS PARA O GERENCIAMENTO ÁGIL DE ATIVIDADES

O papel de cada pessoa em um time ágil deve possuir competências distintas, tanto *hard skills* quanto *soft skills* no que tange tanto ao processo de gerenciamento, como para a execução de atividades (Busse, Weidner, 2020; Kalenda, Hyna, Rossi, 2018). Nesse contexto, Alqudah e Razali (2017) destacam que as pessoas com papéis ligados às partes interessadas devem possuir competências interpessoais mais desenvolvidas que as competências *hard skills*. Em contrapartida, Grapenthin *et al.* (2015) apontam que as pessoas do time de desenvolvimento, por serem auto-organizadas, devem possuir as competências *hard skills* e *soft skills* em um nível elevado quando o tema é a autodesignação e execução de atividades. Para o planejamento de atividades antecedente ao início de uma iteração, Borrego *et al.* (2019) e Salaou *et al.* (2021) destacam que o time de desenvolvimento deve possuir as competências *hard skills* suficientes para resolverem problemas envolvendo a complexidade, como também devem usá-las para a realização da atividade e lidar com o impacto da criticidade da atividade para entrega de valor ao cliente.

O alto nível de competências *hard skills* para o time de desenvolvimento é destacado em ambientes ágeis. Nesse sentido, Kalenda, Hyna e Rossi (2018) e Busse e Weidner (2020) sugerem que as pessoas do time de desenvolvimento devem possuir conhecimentos técnicos para conduzirem a entrega de atividades em um ambiente confiável para o cliente. Para os autores, tais conhecimentos devem garantir a qualidade final do *software* levando em consideração a existência da complexidade embarcada na realização da atividade. Além disso, Mitrovic *et al.* (2020) apontam que o time deve possuir as competências *soft skills* para determinar como o time absorve os conhecimentos e experiências adquiridas após a finalização de uma iteração, podendo aumentar seu nível de auto-organização para iterações futuras.

Além disso, Ganesh e Thangasamy (2012) asseveram a discussão sobre o tema, definindo que o time de desenvolvimento passa a ter maior envolvimento não só com a atividade a ser desenvolvida, mas também desde a sua fase de ideação pelo cliente, no processo de planejamento para o início de iteração, na seleção de uma atividade para desenvolvimento, até seu critério de aceite pelo cliente. Em uma comparação, nos métodos preditivos, Eder *et al.* (2015) sustentam que as pessoas são alocadas de acordo com suas competências pelo responsável do projeto, com um custo relacionado mediante uma relação de precedência entre as atividades. As atividades em execução são controladas e monitoradas pela pessoa responsável do projeto.

Com relação à atribuição de atividades nos métodos ágeis, Shekhar e Kumar (2016) apontam que o time de desenvolvimento passa a ser o responsável pela designação da atividade, exercendo um papel consultivo, auto-organizado e possuindo todas as competências necessárias para selecionar e controlar uma atividade para execução. O time deve estar ciente da complexidade para a realização da atividade, desde o seu planejamento e dos possíveis impactos que a criticidade de uma atividade não concluída possa causar nos resultados de uma iteração. Vale salientar que essa postura pode causar impactos no resultado de uma iteração, como por exemplo, em relação ao sequenciamento e dependências de atividades (Serrador & Pinto, 2015; Špundak, 2014). Como o time passa a ser auto-organizado e, de certo modo, autogerenciado, se o time não possuir uma comunicação eficaz para compreender já no momento do planejamento da iteração a relação de dependências de execução das atividades, pessoas do time poderão ter problemas para realizar suas atividades, sem que uma atividade precedente tenha sido finalizada ou que esteja com algum tipo de impedimento (Heikkila *et al.*, 2017). Esse cenário é asseverado pelo fato de os gestores passarem a organizar suas atividades em *boards* iterativos, ao invés de cronogramas, o que dificulta visualizar tais restrições.

Para um efetivo gerenciamento das atividades, Eder *et al.* (2015) alertam que é evidente que a adoção de metodologias ágeis requer não somente habilidades técnicas e pessoais, mas também necessita que a integração entre os integrantes do time seja assertiva e ativa. Como meio de solucionar a comunicação e estabelecer uma esteira de trabalho, diversos times adotam algumas ferramentas que abrangem todas as etapas do desenvolvimento ágil (Mahmoud *et al.*, 2020; Yin *et al.*, 2020).

Neste contexto algumas competências devem estar presentes no time ágil para o gerenciamento das atividades. A partir do uso de ferramentas, os times ágeis tornam o andamento das atividades do *backlog* transparente a todos os demais membros do time e outras partes interessadas (Mahmoud *et al.*, 2020; Yin *et al.*, 2020). As ferramentas possibilitam que

sejam computadas e extraídas métricas sobre a performance do time em relação à execução das atividades, permitindo ao time compreender o esforço estimado e gasto em relação a complexidade de cada atividade, gerando conhecimento entre as iterações (Wohlrab *et al.*, 2019; Casola *et al.*, 2020). Para Terje, Hagman e Pedersen (2011), este conhecimento pode ser consolidado e documentado na última etapa de *review*, sendo incorporado como aprendizado do time e possibilitando estimativas mais precisas em iterações futuras para o aprimoramento de competências do time.

Por fim, algumas competências técnicas precisam estar presentes no time de desenvolvimento para a entrega das atividades (Majchrzak & Stilger, 2017; Ribeiro, 2020; Pereira Junior, Schroeder, & Dolci, 2019; Seth & Khare, 2015). Nesse sentido, Lwakatere *et al.* (2019) e Raj e Sinha (2020) apontam que o time deve possuir as competências para adaptar o processo de desenvolvimento e entrega de atividades com as práticas DevOps. Com a adoção do DevOps, o objetivo é ampliar a comunicação entre times, tornar os processos mais homogêneos, automatizados e auditáveis, evitando esforços repetitivos e permitindo aos times, tanto de desenvolvimento quanto de operações, que foquem nas atividades que geram valor ao cliente.

A identificação das competências essenciais para as pessoas em papel de gerenciamento de projetos permitiu identificar as limitações existentes no processo de gerenciamento de atividades. É possível destacar as limitações na abordagem do time de desenvolvimento em relação às atividades a serem realizadas em projetos ágeis, desde a sua concepção no desejo do cliente (fase de ideação), no planejamento para uma iteração, até sua entrega final, onde o objetivo é garantir a entrega de maior valor ao cliente.

Outro fator relevante nesse sentido é a pouca existência de ferramentas para apoio no processo de autodesignação de uma atividade para uma pessoa do time, levando em consideração a complexidade para a realização das atividades e a sua respectiva relação com a criticidade durante a fase de gerenciamento. Tais limitações envolvendo as competências das pessoas em papel de gestão e o gerenciamento de atividades por meio de práticas ágeis supracitadas são motivadoras para a realização desta pesquisa. Na próxima seção são apresentados os procedimentos metodológicos que apoiam esta pesquisa.

3 MÉTODO E TÉCNICAS DE PESQUISA

Este capítulo tem como objetivo apresentar os métodos e técnicas de pesquisa que balizam esta dissertação. Com a realização desta pesquisa foi desenvolvido um modelo baseado em competências para auxiliar a autodesignação de atividades de um *backlog* para as pessoas em uma iteração. Para tal, foi efetuada uma Revisão Sistemática da Literatura (RSL) que foi seguida de pesquisa de campo de caráter qualitativo. Os procedimentos estão apresentados a seguir.

3.1 DELINEAMENTO DA PESQUISA

Para a realização desta pesquisa na primeira fase foi realizada uma RSL, onde o objetivo foi identificar o estado atual do conhecimento sobre o contexto envolvendo a criticidade e complexidade para a autodesignação de atividades na gestão de projetos ágeis. Na sequência, com base nos resultados da RSL, foi construído o modelo e sua aplicação se deu em um ambiente laboral conforme procedimentos apresentados no subitem 3.1.2.

3.1.1 Procedimentos da Revisão Sistemática de Literatura

A pesquisa aqui apresentada adotou uma RSL como método a fim de compreender a convergência de duas temáticas relevantes. A primeira temática está relacionada às práticas ágeis de gestão de projetos e a segunda é a gestão de atividades pelo time de desenvolvimento. A RSL se difere das tradicionais revisões narrativas por adotar um processo científico sistemático, que é replicável e transparente. Neste sentido justifica-se também a sua utilização por ela 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 de acordo com as prescrições de 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. O objetivo das fases e atividades apresentadas é garantir o rigor e a robustez que se objetivam neste tipo de pesquisa.

A primeira etapa foi motivada pela questão que norteia esta pesquisa, “como os estudos sobre as práticas ágeis de gestão de projetos abordam a gestão de atividades para uma iteração?” Para tanto, 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 à pesquisas publicadas na área de ciências sociais.

A *string* utilizada para realização das buscas foram ((*"agil*" or "sprint" or "scrum" or "kanban"*) and (*"task*" or "human resource" or "willing*" or "team*"*)). A pesquisa foi realizada em 04 de fevereiro de 2021. A utilização dos operadores booleanos “and” e “or”, além do uso do símbolo “*”, que 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, conforme 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 as quatro etapas para a elaboração do *corpus* de análise. 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. Na terceira etapa foram aplicados os critérios de elegibilidade, onde os artigos foram avaliados segundo os critérios de inclusão e exclusão. E por fim, a quarta etapa constituiu o *corpus* de análise, resultando na base de dados amostral de artigos para a pesquisa.

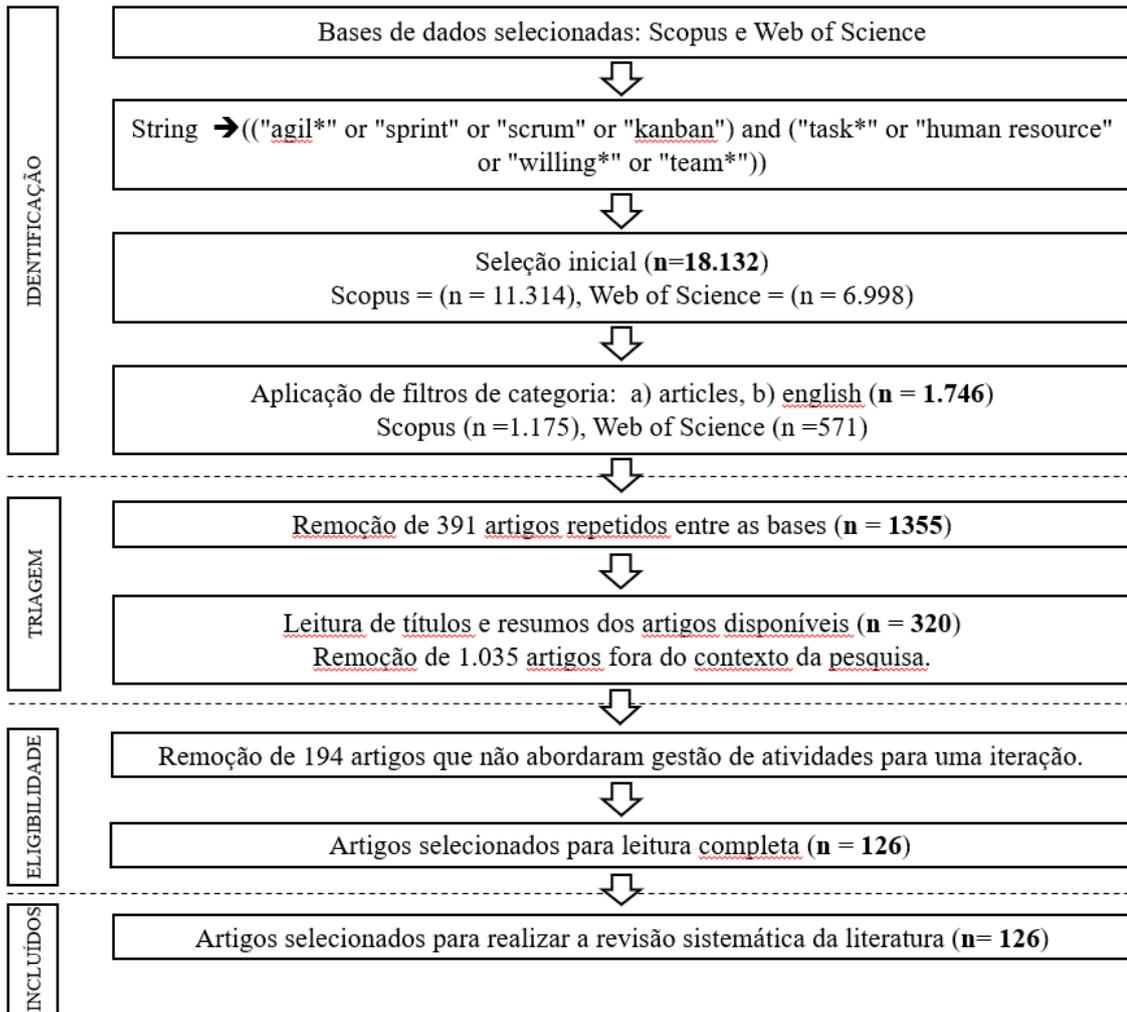


Figura 1. Resultados da pesquisa nas fontes de dados

Fonte: adaptado de Pollock e Berge (2018).

Para a composição da base de dados, alguns 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. As áreas delimitadas para esta pesquisa, representadas por categorias da *Web of Science*, foram: “*Computer Science Software Engineering*”; “*Computer Science Information Systems*”; “*Management*” e “*Computer Science Theory Methods*”. Para a base *Scopus* foram utilizados os filtros: “*Business*” e “*Computer Science*”.

Após a constituição da base de dados, foram removidos os artigos existentes nas duas bases de dados com o objetivo de garantir a homogeneidade do *corpus* de análise. Na sequência, a base de dados foi tratada com auxílio de planilhas eletrônicas *Excel*. Este software 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 sobre o estudo realizado.

O próximo passo foi realizar a leitura atenta dos 126 artigos contidos no *corpus* de análise, categorizando os conteúdos em planilhas *Excel* 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, nesta pesquisa priorizou-se a análise qualitativa dos artigos a fim de constituir uma matriz que pudesse representar os achados desta pesquisa.

Após a obtenção das categorias de análise sobre as práticas ágeis de gestão de projetos que abordam a gestão de atividades para uma iteração com a realização da RSL, de posse dos resultados, foi elaborado um modelo conceitual levando em consideração as particularidades dos processos, das ferramentas e das cerimônias dos frameworks ágeis encontrados no corpus de pesquisa. A seguir, a pesquisa seguiu para a fase da pesquisa empírica.

3.1.2 Procedimentos da pesquisa de campo

Esta pesquisa é de abordagem qualitativa, caracterizada por um conjunto de procedimentos para a compreensão de um problema social (Thiollent, 2009). Para alcance dos objetivos foram adotadas entrevistas em grupo e entrevistas em formato de *focus group*. Neste sentido, a pesquisa seguiu a recomendação de Dilley (2000), que diz que o processo de entrevista deve ser de uma conversa dirigida. Cabe explicitar que se distinguem aqui as entrevistas em grupo das sessões de *focus group* pela sua estrutura e forma.

Morgan e Krueger (1993) definem *focus group* como a técnica de análise e validação de dados de forma interativa controlada, que ajudam a compreender como e o porquê de as pessoas apresentarem determinados comportamentos e atitudes em determinadas situações. Assim, foi entendida como entrevistas em grupo a conversa dirigida nas reuniões sem a finalidade de chegar à validação de algum ponto na discussão, mas sim para explorar ou compreender melhor alguns aspectos da pesquisa ou processo construído.

Esta etapa da pesquisa foi realizada em duas fases. A primeira fase foi composta por entrevistas em grupo com integrantes de times de desenvolvimento de uma empresa do segmento de serviços. Para a realização das entrevistas foram abordados profissionais que

atuam diretamente com a realidade estudada. O objetivo deste processo é explorar seus conhecimentos, observando as experiências dos entrevistados em cada constructo desta pesquisa, buscando assim, uma maior proximidade com o fenômeno estudado (Creswell, 2017). As entrevistas também proporcionaram a oportunidade de adequar as necessidades dos envolvidos ao modelo a ser aplicado.

A segunda fase foi composta por entrevistas em formato de *focus group*. Adotou-se como padrão uma abordagem mista, onde foram mescladas perguntas abertas com perguntas em que as respostas respeitaram a lógica adotada em modelos que utilizam algoritmos de classificação utilizando a técnica *Gradiente Boosting*, um algoritmo iterativo que combina funções parametrizadas simples com desempenho “ruim” (alto erro de predição) para produzir uma regra de predição altamente precisa (Guelman, 2012).

As entrevistas foram realizadas com um grupo multidisciplinar onde haviam pessoas com perfil técnico, como desenvolvedores e especialistas de software, pessoas com perfil voltado às áreas de negócios como *Product Owner* (PO), Coordenadores de Tecnologia e Gerente Senior de Tecnologia, estes últimos, responsáveis por gerenciar projetos voltados à transformação digital e inovação dentro da empresa.

Desse modo, os procedimentos metodológicos da fase de campo foram adequados a uma pesquisa aplicada. O processo de coleta de dados se deu pela introdução do modelo no ambiente laboral. Ressalta-se aqui a importância que a pesquisa qualitativa requer quanto ao rigor da coleta de dados e aos procedimentos metodológicos que devem ser aplicados ao longo do estudo (Bowen, 2009).

Para as coletas de dados que abasteceriam o modelo foram aplicados dois *focus groups* para avaliar e validar os processos. Então, foram desenvolvidos protocolos com base nas competências pessoais e interpessoais (formulários), encontradas por meio da realização da RSL e demonstradas na Tabela 4. Participaram das entrevistas pessoas com papéis diversos dentro do time, como, desenvolvedores, *product owner*, especialistas, coordenadores e gerente. O protocolo foi aplicado em profissionais atuantes no gerenciamento de atividades sob as práticas ágeis, com as pessoas de negócio atuantes desde o processo de ideação das necessidades do cliente até a entrega final da iteração e de pessoas de times de desenvolvimento. O resultado das entrevistas em grupo e *focus group* serviu como base de informação para a elaboração do modelo proposto nesta pesquisa em relação à identificação das competências técnicas e interpessoais necessárias, visando mensurar a complexidade da execução de dadas atividades e sua criticidade em relação ao fim de uma iteração.

As entrevistas, o *focus group* e o processo de coleta de informações foram realizados de forma virtual, por meio de reuniões que ocorreram via ferramenta de comunicação Microsoft Teams, que é utilizada internamente pela empresa, mediante consentimento dos líderes da empresa e dos entrevistados. As entrevistas foram realizadas com Gerente Sênior de TI, responsável pelos integrantes dos dois times que compuseram o grupo, além de coordenadores de TI, especialistas em software, desenvolvedores e *Product Owners*, conforme perfil descrito na Tabela 3.

Tabela 3. Participantes entrevistas e *focus group*

Cargo	Descrição
Gerente Sênior de TI	Está na posição atual há 2,5 anos; líder de 36 pessoas, sendo 3 coordenadores, 3 especialistas, 4 <i>product owners</i> , 18 desenvolvedores e 6 analistas de negócios.
Coordenador de TI	Possui mais de 14 anos de experiência atuando na área de tecnologia em diversos seguimentos, como serviços, aviação e varejo. Possui experiência em projetos ágeis com times da Alemanha e Índia. Está na empresa há 2 anos e ocupava o cargo há 8 meses.
Especialista em TI	Possui ampla formação em tecnologia, servindo como referência técnica dentro da empresa. Ocupa o cargo atual há 2 anos, mesmo tempo em que está na empresa.
Desenvolvedor	Possui mestrado em física, possuía 6 meses na empresa, atuando com projetos estratégicos voltados à análise de dados.
<i>Product Owner</i>	Possui mais de trinta anos de experiência em projetos de tecnologia de empresas nacionais e internacionais de diversos seguimentos. Possui ampla formação acadêmica voltada à gestão de projetos ágeis. Atua na empresa há 3 anos.

Fonte: elaborado pelo autor, 2021.

Vale destacar que o principal objetivo da entrevista no formato *focus group* era compreender e buscar um consenso sobre as definições de Criticidade (CR) e Complexidade (CO) que foram encontrados previamente na RSL. Os integrantes do grupo focal foram convidados a realizarem uma reflexão sobre os temas recorrentes no seu dia a dia em que eles observavam aspectos de criticidade ou complexidade. A partir dessa reflexão, cada um compartilhou os principais itens em um quadro digital. Na sequência, os participantes foram

convidados a votar nos itens que julgavam mais importantes dentro do contexto discutido, como resultado, obtivemos os itens apresentados na Figura 2.

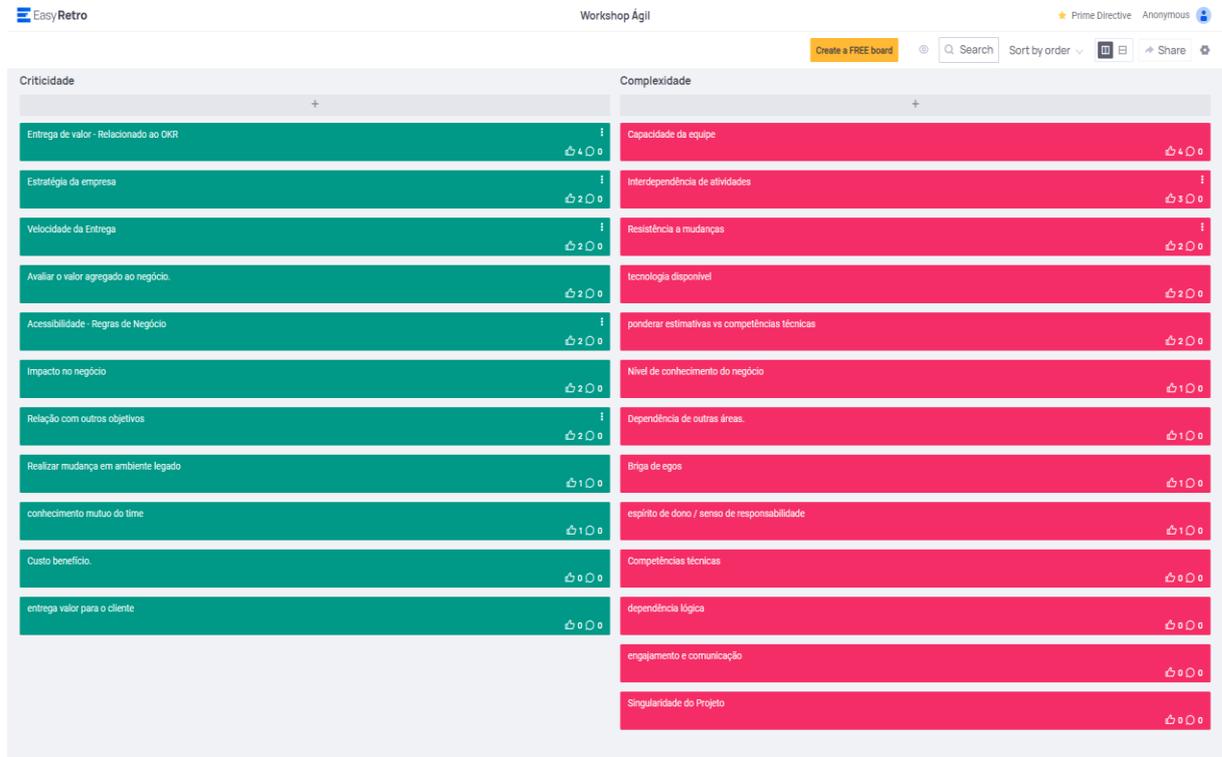


Figura 2. Características de Complexidade e Criticidade – *focus group*

Fonte: Dados de pesquisa, 2021.

Portanto, a primeira sessão de *focus group*, teve como resultado o mapeamento das principais características de Criticidade (CR) e Complexidade (CO) em um ambiente ágil de desenvolvimento de software. A partir dessas características, foi possível iniciar a elaboração do modelo baseado em competências para o gerenciamento de atividades em projetos ágeis. Em uma segunda sessão de *focus group*, o modelo foi apresentado para o grupo da Tabela 3. Nesta ocasião, foram propostas algumas modificações semânticas com o objetivo de tornar o modelo mais aderente para aplicação prática.

Para a análise, foi realizada a categorização dos dados que expõem as diferentes visões relacionadas a cada aspecto da pesquisa (Thiollent, 2009). O processo de análise empregou a classificação por meio da atribuição de um código, neste caso oriundo da RSL, sob a perspectiva “*Theory Driven*”, que permite a comparação entre outros códigos e frequência nas falas dos entrevistados. Os resultados ajudaram a fazer ajustes no modelo construído e aplicado no ambiente laboral.

Portanto, o modelo proposto nesta pesquisa foi aplicado em uma empresa que atua no segmento de serviços, com desenvolvimento de seus produtos por meio do uso de práticas ágeis

de gestão de projetos. A empresa foi selecionada pela acessibilidade no que se refere à pesquisa (Yin, 2015). O modelo foi aplicado em times de desenvolvimentos ágeis, durante o processo de desdobramento de atividades até a entrega final realizada em uma iteração.

Para aumentar a confiabilidade da pesquisa, foram empregados dados oriundos do processo laboral dos entrevistados (Bowen, 2009; da Silva, Russo, & de Oliveira, 2018). O *focus group* avaliou as contribuições do modelo desta pesquisa no processo de autodesignação de atividades, levando em consideração as competências pessoais e interpessoais do time de desenvolvimento em relação à criticidade e complexidade das atividades de uma iteração.

Na próxima seção são apresentados os resultados da RSL que balizaram a proposta do modelo supracitado e sua descrição. Nota-se que a construção do modelo partiu de uma busca pelas pesquisas que trataram da problemática aqui proposta.

4 RESULTADOS E APRESENTAÇÃO DO MODELO CONCEITUAL

Neste capítulo serão apresentados os resultados da RSL e os resultados esperados da pesquisa de campo.

4.1 RESULTADOS DA REVISÃO SISTEMÁTICA DA LITERATURA

O *corpus* da pesquisa foi composto por artigos entre as bases de dados *Web of Science* e *Scopus*. Para garantir a integridade da pesquisa, após a aplicação dos filtros de pesquisa, os resultados das duas bases foram unificados e os documentos duplicados foram retirados. Ao final, o *corpus* da pesquisa resultou em 126 documentos. A evolução da produção acadêmica sobre o tema até a primeira quinzena de fevereiro de 2021 é demonstrada na Figura 3.

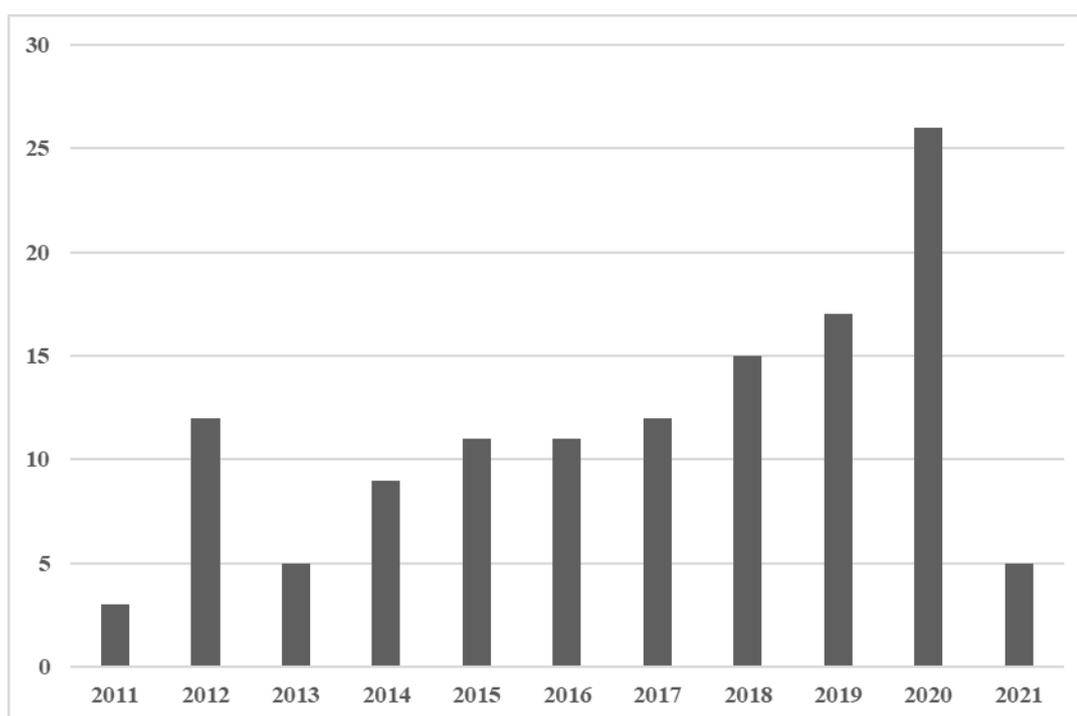


Figura 3. Evolução da produção literária sobre o tema

Fonte: Dados de pesquisa, 2021.

Ao observar a Figura 3, pode-se notar a evolução da produção acadêmica sobre o tema, constando três artigos em 2011 e chegando a 26 artigos publicados em 2020. Esse cenário apresenta a procura de autores para realizar pesquisas, além de destacar a relevância do tema.

Apresentado o panorama da produção científica, a Figura 4 apresenta os 15 principais periódicos com maior número de publicações.

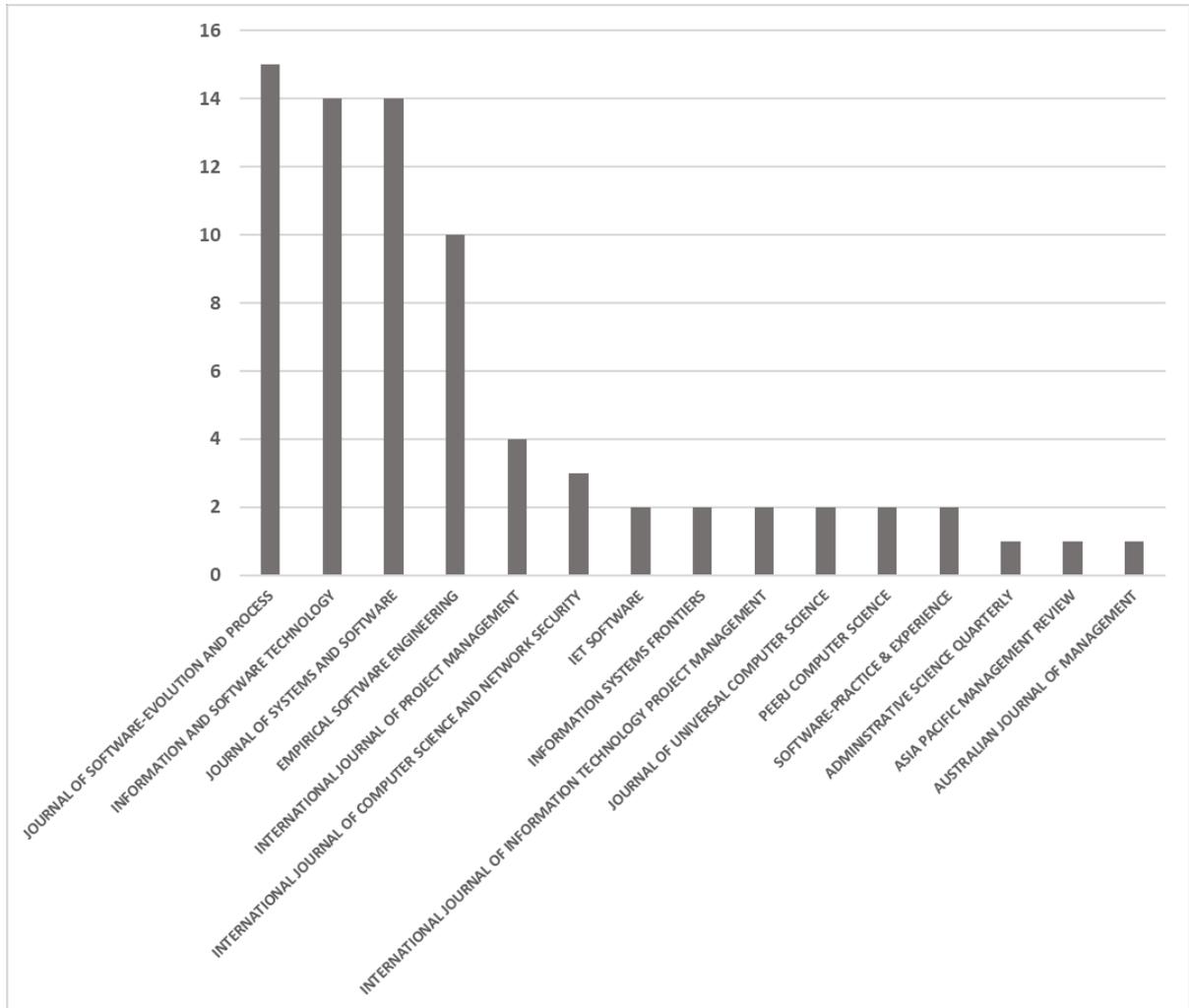


Figura 4. Quantidade de artigos científicos publicados por periódico

Fonte: Dados de pesquisa, 2021.

No passo seguinte, após mapear os artigos e executar uma leitura criteriosa de cada um deles, foi possível identificar as conexões entre os conteúdos e os autores, resultando em quatro categorias constituídas por: competências, ferramentas, relação com a atividade e características do time de desenvolvimento apresentado na Tabela 4.

Tabela 4 Categoria de análise

Categoria	Qtde	Descrição	Autores
Competências	25	Competências são enquadradas em <i>hard skills</i> e <i>soft skills</i> para compreensão e execução das atividades.	Martini, Pareto & Bosch (2016); Shameem <i>et al.</i> (2018); Cetin & Durdu (2019); Gren, Goldman & Jacobsson (2020); Ferreira, Sharp, Robinson & (2011); Magistretti, Dell'Era & Doppio (2020); Hoda, Noble & Marshall (2012); Martinez, Cachero & Melia (2014); Al-Baik & Miller (2015); Lehtinen, Itkonen & Lassenius (2017); Dingsoyr <i>et al.</i> (2018); Hemon <i>et al.</i> (2020); Liu <i>et al.</i> (2019); Naim <i>et al.</i> (2019); Serrador & Pinto (2015); Jorgensen, Mohagheghi & Grimstad (2017); Mishra, Mishra & Ostrovska (2012); Lehtinen <i>et al.</i> (2014); Trkman, Mendling & Krisper (2016); Sarpiri & Gandomani (2017); Kakar (2017); Majchrzak & Stilger (2017); Schloegel (2016); Tripp, Riemenschneider & Thatcher (2016); Vlietland, van Solingen & van Vliet (2016); Hoda & Murugesan (2016); Misirli <i>et al.</i> (2015); Drury-Grogan (2021); Afshari & Gandomani (2021); MoshrefRazav <i>et al.</i> (2019); De Carvalho & Mello (2012); Kaushik, Tayal & Yadav (2020); Cram (2012); Hajjdiab, Taleb & Ali (2012); Cavaleri, Firestone & Reed (2012); Mishra & Abdalhamid (2018); Elghariani <i>et al.</i> (2018); Beaumont <i>et al.</i> (2017);
Ferramentas	30	Ferramentas que apoiam o planejamento e acompanhamento das atividades da iteração.	Concas <i>et al.</i> (2013); Wohlrab <i>et al.</i> (2019); Kremser & Blagoev (2020); Senapathi & Drury-Grogan (2020); Mahmoud <i>et al.</i> (2020); Lopez-Martinez <i>et al.</i> (2018); Malgonde & Chari (2019); Masood, Hoda & Blincoe (2020); Yin <i>et al.</i> (2020); Conforto <i>et al.</i> (2016); Malik, Sarwar & Orr (2021); Tolfo <i>et al.</i> (2018); Cram & Marabelli (2018); Perkusich <i>et al.</i> (2017); Lwakatare <i>et al.</i> (2019); Korkala & Maurer (2014); Jalali, Wohlin & Angelis (2014); Batra, Xia & Rathor (2016); Magdaleno <i>et al.</i> (2015); Nicholls, Lewis & Eschenbach (2015); Casola <i>et al.</i> (2020); Astromskis <i>et al.</i> (2014); Muntés-Mulero <i>et al.</i> (2019); Hemon-Hildgen, Rowe & Monnier-Senicourt (2020); Žužek, Kušar, Rihar & Berlec (2020); Ravi & Narayana (2013); Raj & Sinha (2020); Ganesh & Thangasamy (2012); Rolón & Martínez (2012); Terje, Hagman & Pedersen (2011).
Relação com a atividade	33	Diagnóstico da criticidade e complexidade das atividades.	Shahsavari-Pour <i>et al.</i> (2021); Espinosa-Curiel <i>et al.</i> (2018); Zielske & Held (2020); Akarsu & Yilmaz (2020); Bosch & Bosch-Sijtsema (2011); Belsis, Koutoumanos & Sgouropoulou (2014); Ioannou <i>et al.</i> (2020); Rico <i>et al.</i> (2020); Santos, Goldman & de Souza (2015); Heikkila <i>et al.</i> (2017); Marques <i>et al.</i> (2020); Mitrovic <i>et al.</i> (2020); Luong, Sivarajah & Weerakkody (2019); Tenorio <i>et al.</i> (2020); Zaitsev, Gal & Tan (2020); Varela-Vaca & Gasca (2013); Erdogan, Pekkaya & Gok (2018); Tsai, Chen & Chen (2018); Lavallee & Robillard (2018); Borrego <i>et al.</i> (2019); Fahad <i>et al.</i> (2017); Salaou <i>et al.</i> (2021); Strode, Huff, Hope & Link (2012); Golfarelli, Rizzi & Turricchia (2013); Rahman & Rutz (2015); Rosenkranz, Charaf & Holten (2013); Neto <i>et al.</i> (2019); Kasurinen & Smolander (2017); Mohanarangam (2020); McIver, Lengnick-Hall & Lengnick-Hall (2018); Barke & Prechelt (2019); Destefanis <i>et al.</i> (2016); Arora, Verma & Kavita (2018);
Características do time de desenvolvimento	38	Papéis e responsabilidades do time na autodesignação, acompanhamento e controle das atividades.	Martini, Pareto & Bosch (2016); Shameem <i>et al.</i> (2018); Cetin & Durdu (2019); Gren, Goldman & Jacobsson (2020); Ferreira, Sharp & Robinson (2011); Magistretti, Dell'Era & Doppio (2020); Hoda, Noble & Marshall (2012); Martinez, Cachero & Melia (2014); Al-Baik & Miller (2015); Lehtinen, Itkonen & Lassenius (2017); Dingsoyr <i>et al.</i> (2018); Hemon <i>et al.</i> (2020); Liu <i>et al.</i> (2019); Naim <i>et al.</i> (2019); Serrador & Pinto (2015); Jorgensen, Mohagheghi & Grimstad (2017); Mishra, Mishra & Ostrovska (2012); Lehtinen <i>et al.</i> (2014); Trkman, Mendling & Krisper (2016); Sarpiri & Gandomani (2017); Kakar (2017); Majchrzak & Stilger (2017); Schloegel (2016); Tripp, Riemenschneider & Thatcher (2016); Vlietland, van Solingen & van Vliet (2016); Hoda & Murugesan (2016); Misirli <i>et al.</i> (2015); Drury-Grogan (2021); Afshari & Gandomani (2021); MoshrefRazav <i>et al.</i> (2019); De Carvalho & Mello (2012); Kaushik, Tayal & Yadav (2020); Cram (2012); Hajjdiab, Taleb & Ali (2012); Cavaleri, Firestone & Reed (2012); Mishra & Abdalhamid (2018); Elghariani <i>et al.</i> (2018); Beaumont <i>et al.</i> (2017);
Total artigos	126		

Fonte: Dados da Pesquisa, 2021.

As quatro categorias encontradas compreendem uma abstração observada nas leituras dos artigos. O processo de análise inicial dos artigos foi individual e, depois, para seleção das categorias passou-se pela análise e agrupamento. O processo de categorização foi realizado levando em consideração a aderência entre os artigos e autores em cada uma das categorias constituídas.

Cabe destacar que as competências supracitadas são categorizadas em *soft skills* e *hard skills*. No entanto, como aponta Luiz Neto (2020), há certa complexidade em distinguir algumas competências das pessoas porque existe um *hard side* e um *soft side* em muitas delas. Não obstante esse aspecto complexo das competências, algumas competências *hard skills* ligadas às competências técnicas são facilmente identificáveis como, por exemplo, elaboração de uma planilha eletrônica. Outras, como a comunicação, são multifacetadas. A comunicação pode ser avaliada pela capacidade das pessoas em serem empáticas, mas também pela capacidade de redigirem um texto objetivo sem erros gramaticais.

Outro aspecto relevante para tratar das competências é com relação ao nível a ser observado. As competências podem ser observadas no nível individual, muito tratada pela perspectiva do CHA (conhecimento, habilidade e atitude), mas também de forma coletiva e organizacional (Michaux, 2011). Ruas, Antonello e Boff (2005) salientam que a competência no nível coletivo é pouco observada empiricamente, mas que pode ser compreendida na junção de competências manifestadas em equipes, departamentos ou mesmo nas relações entre pessoas em empresas distintas. A competência coletiva é obtida pela sinergia de diversas atividades, como por exemplo, um sistema logístico. No nível organizacional, Prahalad e Hamel (1995) citam a competência organizacional como uma capacidade que diferencia uma empresa em um ambiente de negócios.

4.1.1 Competências

Em um ambiente de projetos de *software*, um time de desenvolvimento deve possuir um conjunto de competências para o desenvolvimento de uma iteração (Busse & Weidner, 2020; Kalenda, Hyna, & Rossi, 2018). As competências, tanto técnicas como interpessoais, estão associadas à cultura (Alqudah & Razali, 2017), às estratégias da organização (Lindsjorn *et al.*, 2016; Jovanović *et al.*, 2016) e às partes interessadas (Manekar & Pradeepini, 2019). As principais competências, sejam elas técnicas ou interpessoais, suas descrições e autores foram mapeadas a partir da RSL e são apresentadas na Tabela 5.

Tabela 5. Lista de competências e autores

Tipo de Competência	Competência	Descrição	Autores
Técnicas	Planejamento de Atividades	Trata-se da capacidade de planejar as atividades necessárias assim como identificar a precedência entre elas necessária para completar a iteração e entregar valor ao cliente.	Trkman <i>et al.</i> (2019); Jia <i>et al.</i> (2019)
	Compreensão de Atividades	Trata-se da capacidade de compreender o contexto e o trabalho necessário para a execução da atividade de forma correta, de acordo com o que foi solicitado.	Grapenthin <i>et al.</i> (2015)
	Mensuração das atividades	Trate-se da capacidade de mensurar o esforço necessário para executar uma atividade. Esta capacidade é desenvolvida durante as iterações.	Ramirez-Mora, Oktaba & Perez (2020)
	Capacidade de Abstração	A partir da compreensão das atividades é possível criar generalizações e abstrações dos requisitos, tornando tangível o desenvolvimento de requerimentos abstratos em funcionalidades sistêmicas.	Gren, Knauss & Stettina (2018)
	Análise dos Requerimentos	Através do conhecimento sobre o contexto em que se atua, essa competência torna possível o desmembramento de requerimentos em atividades que se tornaram funcionalidades do software.	Medeiros <i>et al.</i> (2020)
	Monitoramento das Atividades	Diz respeito ao conhecimento de análise das ferramentas de acompanhamento das atividades do <i>backlog</i> de uma iteração.	Fontana <i>et al.</i> (2015); Licorish & MacDonell (2015); Hsieh & Chen (2015)
	Conhecimento em Tecnologia	Esta competência trata-se do conhecimento de linguagens de programação, assim como padrões de projeto e padrões arquiteturais voltadas ao desenvolvimento de soluções.	Pascoa, Telha & Santos (2019)
Decomposição de atividades	Trata-se da competência de desmembrar e agrupar atividades provenientes dos requerimentos. O desenvolvimento desta competência está ligado ao desenvolvimento da competência de abstração.	Tan & Vicente (2019)	
Interpessoais	Negociação	Capacidade de negociar a adição, extinção ou modificações de requerimentos e atividades no <i>backlog</i>	Adolph, Kruchten & Hall (2012);
	Influência	Habilidade em influenciar pessoas do time para alcançar objetivos pessoais ou organizacionais	Busse & Weidner (2020)
	Comunicação Efetiva	Capacidade de transmitir uma mensagem de forma clara e objetiva sem perda de informações	Gjoystdal & Karunaratne (2020)
	Senso de Auto-organização	Esta competência diz respeito à capacidade dos membros do time de se auto-organizarem de acordo com a sua rotina de trabalho, de forma que seja possível cumprir com a execução das atividades que o time se comprometeu.	Gjoystdal & Karunaratne (2020)
	Adaptabilidade a Mudanças	Habilidade de se adaptar de forma rápida as mudanças que ocorrem durante uma iteração. Ao desenvolver essa habilidade, o integrante deixa de ver as mudanças de forma negativa e passa a enxergar a mudança como ação estratégica para alcançar resultados e entregar valor ao final da iteração.	Colomo-Palacios <i>et al.</i> (2012)
	Desenvolvimento de Atividades em Equipe	Trata-se da capacidade de trabalhar em equipe, aceitar eventuais críticas construtivas ao trabalho desenvolvido e, dessa forma, realizar a troca de conhecimentos e experiências, tanto técnicas quanto relacionadas ao contexto de negócio.	Paasivaara & Lassenius (2014); Yilmaz <i>et al.</i> (2017)
	Senso de Importância	Esta competência diz respeito à habilidade de reconhecer a importância de eventos, atividades e comportamentos do time que podem afetar a iteração.	Grapenthin <i>et al.</i> (2015)

Fonte: Dados da Pesquisa, 2021.

Em relação aos indivíduos, as competências técnicas estão presentes no planejamento das atividades (Trkman *et al.*, 2019; Jia *et al.*, 2019), na compreensão (Grapenthin *et al.*, 2015) e no processo de mensuração das atividades (Ramirez-Mora, Oktaba & Perez, 2020). Para o processo de mensuração, o time pode empregar competências interpessoais (Fontana *et al.*, 2014), negociação (Adolph, Kruchten & Hall, 2012) e influência (Busse & Weidner, 2020).

Em alguns casos, as competências transcendem o planejamento até a compreensão das atividades (Moe, Aurum & Dyba, 2012). Nesse ponto, destacam-se as competências técnicas relacionadas aos processos de desenvolvimento de *softwares*, como a abstração (Gren, Knauss & Stettina, 2018) e a capacidade de análise dos requerimentos (Medeiros *et al.*, 2020). Em relação ao monitoramento das atividades, o time deve empregar competências técnicas para interagir com o quadro de *backlog* (Fontana *et al.*, 2015), como no monitoramento das atividades em execução (Licorish & MacDonell, 2015; Hsieh & Chen, 2015).

Para o processo de execução das atividades em cada iteração, o desenvolvimento poderá ser efetuado por técnicas de desenvolvimento por pares, conhecidas como *Work in Pairs* (Paasivaara & Lassenius, 2014). Para a realização de tais técnicas, o time de desenvolvimento deverá possuir as competências interpessoais necessárias para que o desenvolvimento seja realizado de maneira eficaz (Yilmaz *et al.*, 2017). Em relação ao processo de desenvolvimento de *software*, o time deverá possuir alto conhecimento em tecnologia (linguagem de computação e arquitetura computacional) (Pascoa, Telha, & Santos, 2019) e em técnicas de decomposição das atividades (Tan & Vicente, 2019).

Por fim, no processo de finalização de uma atividade, o time deverá possuir um conjunto de competências coletivas como a comunicação efetiva, senso de auto-organização (Gjoystdal & Karunaratne, 2020), adaptabilidade às constantes mudanças (Colomo-Palacios *et al.*, 2012) e senso de importância (Grapenthin *et al.*, 2015). As relações das competências técnicas e interpessoais existentes em times ágeis são demonstradas na Figura 5.

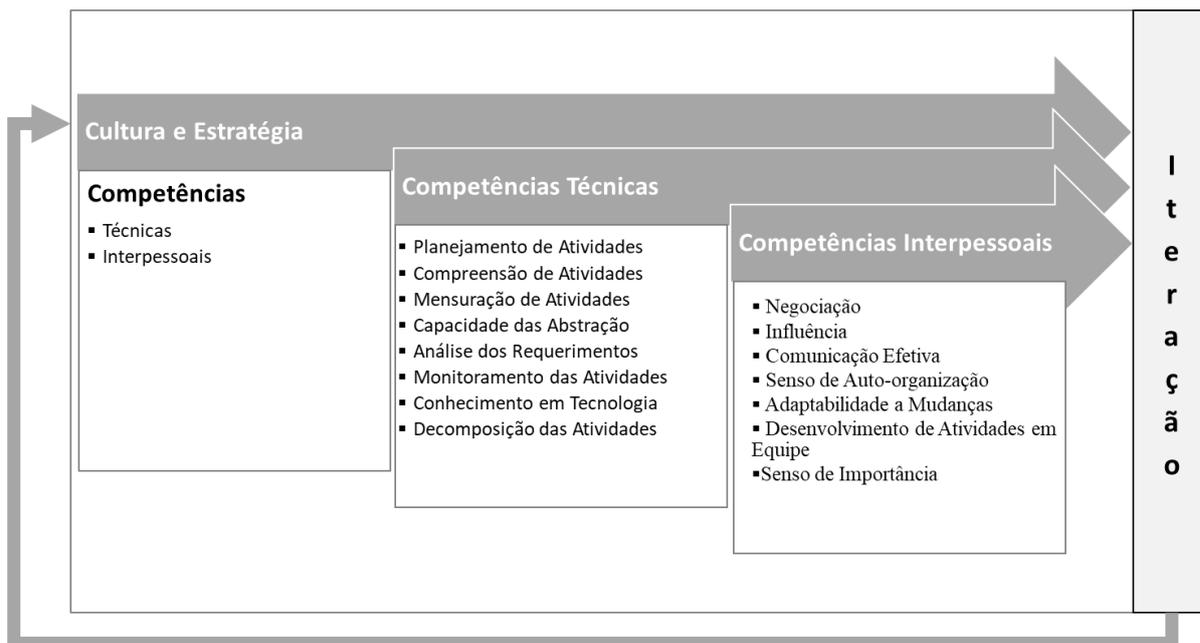


Figura 5. Relação das competências para times ágeis

Fonte: Dados de pesquisa, 2021.

As relações entre as competências demonstradas na Figura 5 destacam que após a execução de diversas iterações, o time de desenvolvimento e seus integrantes desenvolvem competências técnicas e interpessoais que se tornarão parte da cultura tanto do time (Alqudah & Razali, 2017), quanto da organização. Busse e Weidner (2020) afirmam que durante as etapas das iterações, os indivíduos do time exercem influência sobre os demais, por meio do uso de técnicas como a negociação que é utilizada para a designação de atividades (Adolph, Kruchten & Hall, 2012).

Para que o time ágil desenvolva competências técnicas para lidar com a análise e decomposição das atividades, os integrantes do time deverão possuir outras competências individuais, como conhecimentos em planejamento e execução de atividades por meio de técnicas como o *Work in Pairs* (Paasivaara, Lassenius, 2014). Para alcançar este cenário, o time deverá se engajar desde a concepção da ideia do cliente em atividade até o término da iteração (Yilmaz *et al.*, 2017).

Em relação às competências interpessoais, um time para ser auto-organizado deverá possuir competências necessárias para a negociação e gerenciamento da execução das atividades, deixando claro a todos os membros do time a relação de cada atividade e o término da iteração mediante as constantes mudanças que visam agregar valor à entrega (Al-Baik, Miller, 2015). Para alcançar este cenário, o time deverá promover efetividade na comunicação, além de confiança mútua entre os membros do time. As competências

que foram desenvolvidas pelo time serão absorvidas e melhoradas nas iterações seguintes, assim como novas competências poderão surgir e serão incorporadas por indivíduos e times, tornando-se parte integrante da cultura (Liu *et al.*, 2019).

A partir desse cenário, o time de desenvolvimento passa a ter um papel de auto-organização durante uma iteração (Ganesh & Thangasamy, 2012), a ponto de identificarem a complexidade e a criticidade de cada atividade. Neste contexto, Mitrovic *et al.* (2020) destaca que as pesquisas não adotaram as competências (*hard skills* e *soft skills*) de todo o time de desenvolvimento, focando apenas nas competências do indivíduo.

Com o objetivo de explicitar tais competências na efetividade da iteração, Colomo-Palacios *et al.* (2012) destacam que em relação à iteração, o time deve possuir adaptabilidade às constantes mudanças. Grapenthin *et al.* (2015) complementam que o senso de importância do time em relação à criticidade das atividades em uma iteração também devem ser levado em conta neste processo de adaptação. Por fim, Zaitsev, Gal, Tan (2020) asseveram que neste ponto é necessário que exista ampla comunicação entre o time com o objetivo da iteração. Neste processo o segundo integrante do time não pode deixar de executar nenhuma atividade que impacte o desenvolvimento de atividades futuras contidas no *backlog* (Zaitsev, Gal, Tan, 2020).

4.1.2 Ferramentas

No decorrer das iterações, a gestão da execução das atividades é realizada por meio de diversas ferramentas (Mahmoud *et al.*, 2020; Yin *et al.*, 2020). As ferramentas devem estar em consonância com as práticas ágeis (Wohlrab *et al.*, 2019; Casola *et al.*, 2020) e com as políticas de desenvolvimento de *softwares* adotada pela organização (Kremser & Blagoev, 2020). Ainda em relação ao processo de desenvolvimento de *softwares*, algumas ferramentas abordam as questões de planejamento (Masood, Hoda & Blincoe, 2020), de associação ao risco para a execução da atividade (Malgonde & Chari, 2019; Korkala & Maurer, 2014), chegando até o processo de entrega da atividade para o cliente (Lopez-Martinez *et al.*, 2018; Concas *et al.* 2013; Senapathi & Drury-Grogan, 2020).

Para Malik, Sarwar e Orr, (2021) a categorização das atividades deve ser feita pelo time diante da análise do contexto para o desenvolvimento de cada atividade (Cram &

Marabelli, 2018), apresentando o risco associado à realização e monitoramento e prevenção da alteração da atividade (Rolón & Martínez, 2012). O processo de categorização das atividades se torna crítico quando o time se encontra em transição para as práticas ágeis (Batra, Xia & Rathor, 2016), inclusive em cenários de projetos de transformação digital com adoção de *cloud computing* (Muntes-Mulero *et al.*, 2019).

Alguns desafios podem ocorrer quanto ao uso de ferramentas para o processo de ideação e priorização de atividades pelo time de desenvolvimento (Ganesh & Thangasamy, 2012). Como o time de desenvolvimento possui características de auto-organização (Grapenthin *et al.*, 2015), as ferramentas como o *User Story*, *Story Telling* e *Story Board*, caso não sejam manuseadas corretamente, podem acarretar desvios nas métricas e na atualização do desempenho das atividades (Ganesh & Thangasamy, 2012).

As ferramentas ganham papel de destaque quando alinhadas ao perfil do time de desenvolvimento (Conforto *et al.* 2016; Tolfo *et al.*, 2018), obtendo como resultado um conjunto de métricas (Perkusich *et al.*, 2017; Ravi & Narayana, 2013) que servirão de apoio ao processo de disseminação de conhecimento do time (Terje, Hagman & Pedersen, 2011). Além disso, a escolha de uma ferramenta adequada à organização deve estar relacionada às melhores práticas de desenvolvimento de *softwares* (Nicholls, Lewis & Eschenbach, 2015), com a otimização do uso de uma pessoa do time de desenvolvimento (Magdaleno *et al.*, 2015; Jalali, Wohlin & Angelis, 2014) e com a política de boas práticas de desenvolvimento e testes de *softwares* (Astromskis *et al.*, 2014).

Nesse sentido, algumas ferramentas como o *Kanban* (Majchrzak e Stilger, 2017), o *Jira* (Ribeiro, 2020), *Trello* (Pereira Junior, Schroeder e Dolci, 2019) e *Jenkins* (Seth & Khare, 2015) podem contribuir para demonstrar o fluxo de iterações de atividades entre os times, além de servirem de ferramentas de acompanhamento de métricas ágeis. A relação entre as etapas de uma iteração, assim como os processos, ferramentas e métricas produzidas foram consolidados e demonstrados na Figura 6.



Figura 6. Ferramentas para times ágeis

Fonte: Dados de pesquisa, 2021.

Em relação às ferramentas ágeis, Lwakatare *et al.* (2019) afirmam que a entrega das atividades busca também entregar valor ao cliente. Neste processo, ferramentas como o DevOps (*Development and Operations*) garantem a maior qualidade na produção do *software* (Raj & Sinha, 2020), além de aproximar pessoas, processos e times de acordo com as práticas ágeis adotadas pela organização (Hemon-Hildgen, Rowe & Monnier-Senicourt, 2020). Além disso, o DevOps ganha cada vez mais importância nos times de desenvolvimento, por permitir que os processos, desde a concepção da ideia até a produção do incremento de *software*, sejam entregues ao cliente com maior valor ao seu negócio (Žužek *et al.*, 2020).

4.1.3 Relação com a atividade

As pesquisas de Rahman e Rutz, (2015), Neto *et al.* (2019) e Mohanarangam (2020) corroboram que as práticas ágeis trazem uma relação distinta às atividades frente a metodologia tradicional (preditiva). Enquanto na metodologia tradicional as atividades possuem escopo, restrições, riscos, mudanças e planejamentos definidos previamente (Golfarelli, Rizzi & Turrichia 2013), nas metodologias ágeis, a definição das atividades ocorre por meio das cerimônias que marcam o início de cada iteração (Ioannou *et al.*, 2020; Luong, Sivarajah & Weerakkody, 2019), onde os requisitos são apresentados ao time de desenvolvimento. Bosch e Bosch-Sijtsema (2011) afirmam que o *Scrum Master*

não influencia na formação do *backlog* e, a equipe, deve selecionar as atividades priorizadas que mais geram valor ao cliente, sem existir uma divisão padrão das atividades.

Na cerimônia que inicia cada iteração, ocorre a análise e a classificação do esforço para a execução das atividades (Barke & Prechelt, 2019). Essa etapa contempla não apenas descobrir, diagnosticar e entender os principais problemas, mas também a capacidade de projetar evidências baseadas em soluções que são ajustadas às necessidades de mudança dinâmicas do ambiente e requisitos de mudança (McIver, Lengnick-Hall & Lengnick-Hall, 2018). De acordo com Rosenkranz, Charaf e Holten (2013), os requisitos apresentados nesta etapa podem ser qualificados como: incompletos; sem sentido; redundante e ambíguos, o que pode levar uma atividade a não ser compreendida (Ioannou *et al.*, 2020). Belsis, Koutoumanos e Sgouropoulou (2014) sugerem que seja realizada uma validação dos itens que compõe o detalhamento dos requisitos para mitigar que os requisitos cheguem com deficiência a esta etapa.

Ainda nesta etapa, é necessário garantir que não existam mais incertezas quanto ao que precisa ser executado em cada atividade (Strode *et al.* 2012). A partir deste momento, o time de desenvolvimento utiliza técnicas, como *story points*, para obter uma estimativa do esforço para a execução da atividade (Varela-Vaca & Gasca, 2013). Nesta etapa, cada indivíduo pontua o esforço estimado para execução da atividade, baseando-se na complexidade e dependências da atividade (Borrego *et al.*, 2019). O artefato desta cerimônia é o *backlog*, com todas as atividades priorizadas de acordo com a criticidade, complexidade e dependência identificada pelos integrantes do time (Salaou *et al.*, 2021).

A partir deste momento, inicia-se a execução das atividades do *backlog*, onde os membros do time possuem a autonomia de selecionar a atividade que irão executar (Akarsu & Yilmaz, 2020). Caso o membro do time não possua a competência necessária para assinar a próxima atividade (Barke & Prechelt, 2019), ou ainda seja uma atividade de alta criticidade, (Espinosa-Curiel *et al.*, 2018; Erdogan) Pekkaya & Gok (2018) sugere que a atividade seja executada em colaboração com um sênior, proporcionando a qualidade na entrega e a passagem de conhecimento entre os membros dos times (Luong, Sivarajah & Weerakkody, 2019). Neste ponto, é necessário que exista ampla comunicação entre o time, para que o segundo integrante do time não deixe de executar nenhuma atividade que impacte no desenvolvimento de atividades futuras contidas no *backlog* (Zaitsev, Gal & Tan, 2020).

De acordo com Heikkila *et al* (2017) é importante que os papéis estejam claros e a metodologia bem definida, para que o time de desenvolvimento saiba quem acionar, caso surja um impedimento durante a execução de alguma atividade. Lavallee e Robillard (2018) afirmam que para participar da resolução de impedimentos é preciso ter habilidades interpessoais. Negligenciar os impedimentos pode, de certo modo, impactar negativamente a iteração, como a baixa qualidade de entrega ou gerar o cancelamento de uma determinada atividade (Salaou *et al.*, 2021).

4.1.4 Características do time de desenvolvimento

O time de desenvolvimento possui em certos momentos da iteração, diversos papéis e responsabilidades (Trkman, Mendling & Krisper, 2016; Sarpiri & Gandomani, 2017). Na parte de planejamento, o time deve assumir papel consultivo nas cerimônias intermediárias de acompanhamento da evolução do *backlog* (Cetin & Durdu, 2019; Martini, Pareto, & Bosch, 2016), para minimizar os problemas de entrada de novas pessoas no time (Al-Baik & Miller, 2015), ou algum tipo de barreira cultural associadas às práticas ágeis (Liu *et al.*, 2019). Ainda no planejamento, o time se torna responsável pela análise dos documentos de requerimentos do cliente (Hemon *et al.*, 2020), independente do esforço necessário para a realização de uma ou mais atividades (Dingsoyr *et al.*, 2018; Elghariani *et al.*, 2018). Nesse ponto, Serrador e Pinto (2015) e Hoda e Murugesan (2016) afirmam que a responsabilidade do time de desenvolvimento deverá abranger a visão e as necessidades dos *stakeholders*.

De posse do planejamento, o time passa a ter um papel focado na preparação das atividades para o início da iteração (Lehtinen *et al.*, 2014; Majchrzak & Stilger, 2017). Nesse momento, o time passa a ter a responsabilidade de efetuar a decomposição das atividades (Schloegel, 2016), levando em consideração a complexidade (Kaushik, Tayal & Yadav, 2020) e a criticidade (Pascoa, Telha & Santos, 2019) para a realização de cada atividade.

O planejamento traz consigo barreiras para o controle e gerenciamento de atividades (Magistretti, Dell'Era & Doppio, 2020). Como as atividades são sequenciadas por ordem de necessidade de negócio aos olhos do cliente (Vlietland, van Solingen, & van Vliet, 2016), por pessoas com papéis que podem não participar do processo de desenvolvimento das iterações (Hajjdiab, Taleb & Ali, 2012; Beaumont *et al.*, 2017), as

pessoas envolvidas nesse contexto possuem responsabilidade direta com o desempenho da iteração e a satisfação do cliente (Mishra & Abdalhamid, 2018; Cram, 2012). O planejamento pode ser afetado ainda pelas mudanças das atividades dos projetos (Shameem *et al.*, 2018; Hoda, Noble & Marshall, 2012) e pelo nível de competência técnica de um time (Gren, Goldman & Jacobsson, 2020; Lehtinen, Itkonen & Lassenius, 2017), o que pode ocasionar em impactos negativos nos resultados de uma iteração (Drury-Grogan, 2021; Afshari & Gandomani, 2021; MoshrefRazav *et al.*, 2019).

Para aproximar o resultado das iterações com o cliente, papéis como o *User Experience* (UX), estão relacionados à obtenção de maior conhecimento sobre a necessidade do negócio (Ferreira, Sharp & Robinson, 2011; Naim *et al.*, 2019). Assim, compreender o mais breve possível a necessidade do cliente passa a ser responsabilidade do time de desenvolvimento também (Jorgensen, Mohagheghi, & Grimstad, 2017).

A partir do início da iteração, o time assume o papel e a responsabilidade pelo controle e monitoramento das atividades (Misirli *et al.*, 2015; Tripp, Riemenschneider & Thatcher, 2016). Segundo De Carvalho e Mello (2012), o time deve promover a transferência do conhecimento obtido na realização de iterações anteriores, promovendo o aumento de desempenho do time de desenvolvimento (Kakar, 2017).

4.1.5 Proposta do modelo conceitual

Com base no exposto até aqui, foi possível desenhar um panorama geral em que competências e atividades pudessem estar alinhadas para promover o melhor processo de autodesignação em projetos ágeis. Assim, após a análise individual do *corpus* de pesquisa, foi possível, a partir das conexões entre os conteúdos e os autores, apresentar quatro categorias relacionadas ao gerenciamento de atividades em times ágeis. As categorias levaram em consideração *hard skills* e *soft skills* presentes nos times ágeis para uma iteração, desde a fase de planejamento até a fase de *release* (entrega ao cliente).

Nesse ponto, vale salientar que tais competências foram abstraídas, levando em consideração as características de cada uma delas em relação ao objetivo desta pesquisa. Como resultado, as competências foram divididas em técnicas e interpessoais, previamente apresentadas na Tabela 5. As competências técnicas e interpessoais foram subdivididas destacando as competências e as ferramentas presentes no time de

desenvolvimento para a realização de atividades, destacadas previamente nas figuras 5 e 6 respectivamente.

Na fase de planejamento, pode-se destacar que as competências técnicas e interpessoais, presentes no time de desenvolvimento, podem ser utilizadas junto às pessoas de negócio para a análise e divisão dos requerimentos da visão do produto em atividades mensuráveis e capazes de serem entregues em uma iteração. Esta fase deve respeitar as particularidades dos *frameworks* ágeis em relação ao processo de planejamento. Por exemplo, a cerimônia de planejamento (*Sprint Planning Meeting*) do *Scrum* e a de reabastecimento (*Replenishment Meeting*) do *KanBan*. O ponto de atenção aqui está relacionado à capacidade do time de desenvolvimento de identificar a prioridade de execução das atividades para atendimento a uma determinada história do usuário, levando em consideração a criticidade de cada atividade, independente das ferramentas escolhidas pelo time de desenvolvimento para essa fase.

A partir da finalização da fase de planejamento, o time de desenvolvimento passa a possuir um papel auto-organizado, onde cada integrante do time se torna responsável pela realização, execução e monitoramento de uma atividade, frente a uma ordem de priorização de atividades. Nessa fase de autodesignação, cada integrante deve ter o engajamento necessário para identificar se possui as competências, tanto técnicas quanto interpessoais, necessárias para a realização de uma determinada atividade, frente à complexidade e criticidade de cada uma das atividades em uma determinada iteração.

Na fase final da iteração, o time de desenvolvimento deve finalizar as atividades que compõem uma determinada história de usuário, respeitando a ordem de priorização entre elas, previamente acordadas na fase de planejamento. Independente do *framework* ágil, esse processo deve estar conforme as definições acordadas com o cliente para a qualidade de uma atividade pronta. Além disso, o time de desenvolvimento deve possuir as competências técnicas necessárias para as etapas de entrega da história do usuário, por meio do uso de ferramentas ágeis, como por exemplo, o DevOps.

Esse processo deve ocorrer sem levar em consideração o *framework* ágil adotado pela empresa, com a cerimônia de revisão (*Sprint Review Meeting*) do *Scrum* e a de planejamento de entrega (*Delivery Planning*) do *KanBan*. Por fim, o time deve possuir as competências necessárias para identificar os pontos de melhoria do time na finalização de uma iteração para as próximas iterações, buscando tornar o time o mais auto-organizado possível. Aqui, esse processo está presente nas cerimônias de *retrospectiva*

(*Sprint Retrospective Meeting*) e as cadências de revisão de operações (*Operations Review*) e revisão da estratégia (*Strategy Review*) do Kanban.

Frente ao apresentado, um time de desenvolvimento deverá possuir as competências técnicas e interpessoais para atuar com as pessoas de negócio na fase de planejamento até o desdobramento das histórias de usuário em atividades. Durante a iteração, o time deve aplicar tais competências na autodesignação das atividades e no uso de ferramentas ágeis na análise e controle das atividades em andamento. Frente aos processos relacionados ao gerenciamento de atividades até aqui destacados, a Figura 7 apresenta um modelo conceitual destacando a relação das competências técnicas e interpessoais presentes em um time de desenvolvimento para a autogestão de atividades em uma iteração.

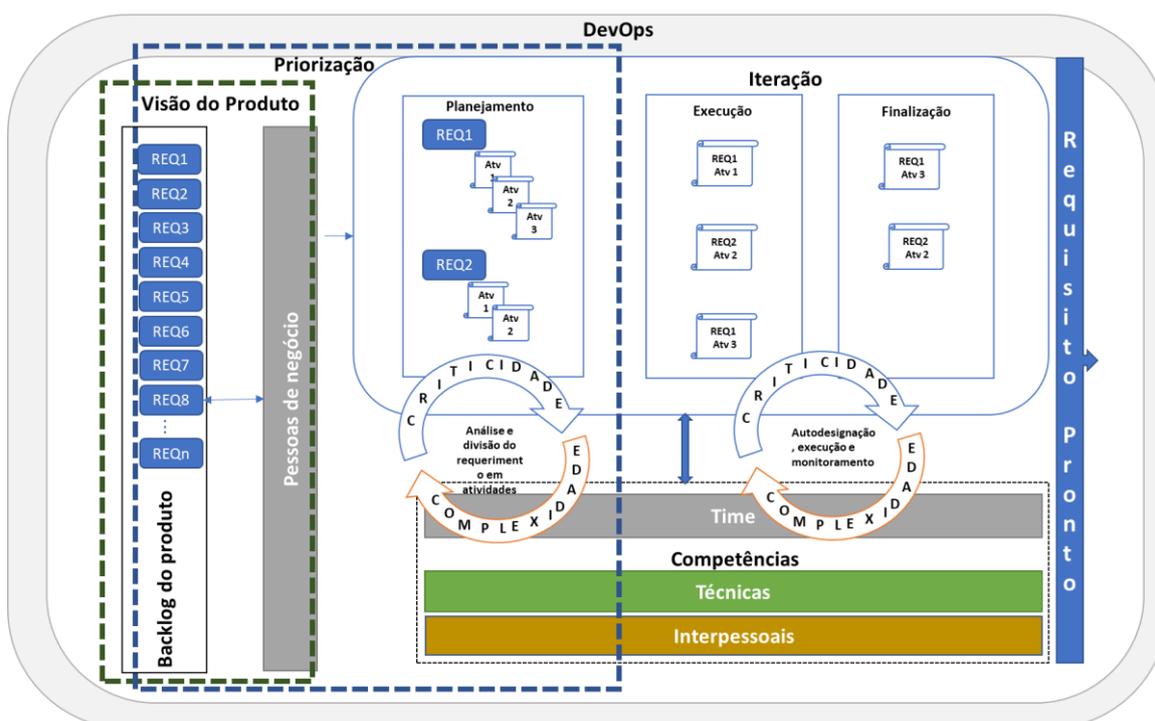


Figura 7. Modelo conceitual das competências técnicas e interpessoais dos times ágeis para a autogestão de atividades de uma iteração

Fonte: Dados de pesquisa, 2021.

O modelo conceitual apresentado na Figura 7 foi elaborado levando em consideração as particularidades dos processos, das ferramentas e das cerimônias dos frameworks ágeis encontrados no corpus de pesquisa. Para cada fase, são apresentadas a relação das competências técnicas e interpessoais presentes no time de desenvolvimento durante a fase de planejamento, de autodesignação e do monitoramento das atividades.

Após a fase de finalização, o resultado do processo deverá fornecer um conjunto de dados da fase de desenvolvimento de uma iteração como subsídio às métricas ágeis pela organização.

Com base no que foi apresentado na fase de levantamento e análise da literatura, aqui compreendida como RSL, foi possível chegar ao modelo conceitual apresentado na Figura 7. Na próxima seção é apresentada a construção e aplicação da ferramenta computacional.

5 CONSTRUÇÃO E APLICAÇÃO DA FERRAMENTA COMPUTACIONAL

Com os resultados desta pesquisa, espera-se destacar as contribuições do modelo proposto a ser utilizados por times de desenvolvimentos ágeis. O modelo foi utilizado como uma ferramenta computacional para apoio no planejamento para a autodesignação de atividades durante uma iteração, mediante a complexidade existente em cada atividade e a relação de criticidade de cada atividade com a fase de finalização de uma iteração.

Por se tratar de um processo abstrato e complexo, foi elaborado o infográfico apresentado na Figura 8, onde é possível relacionar as etapas de implementação do modelo e aplicação do modelo para realizar a classificação dos analistas para cada atividade.

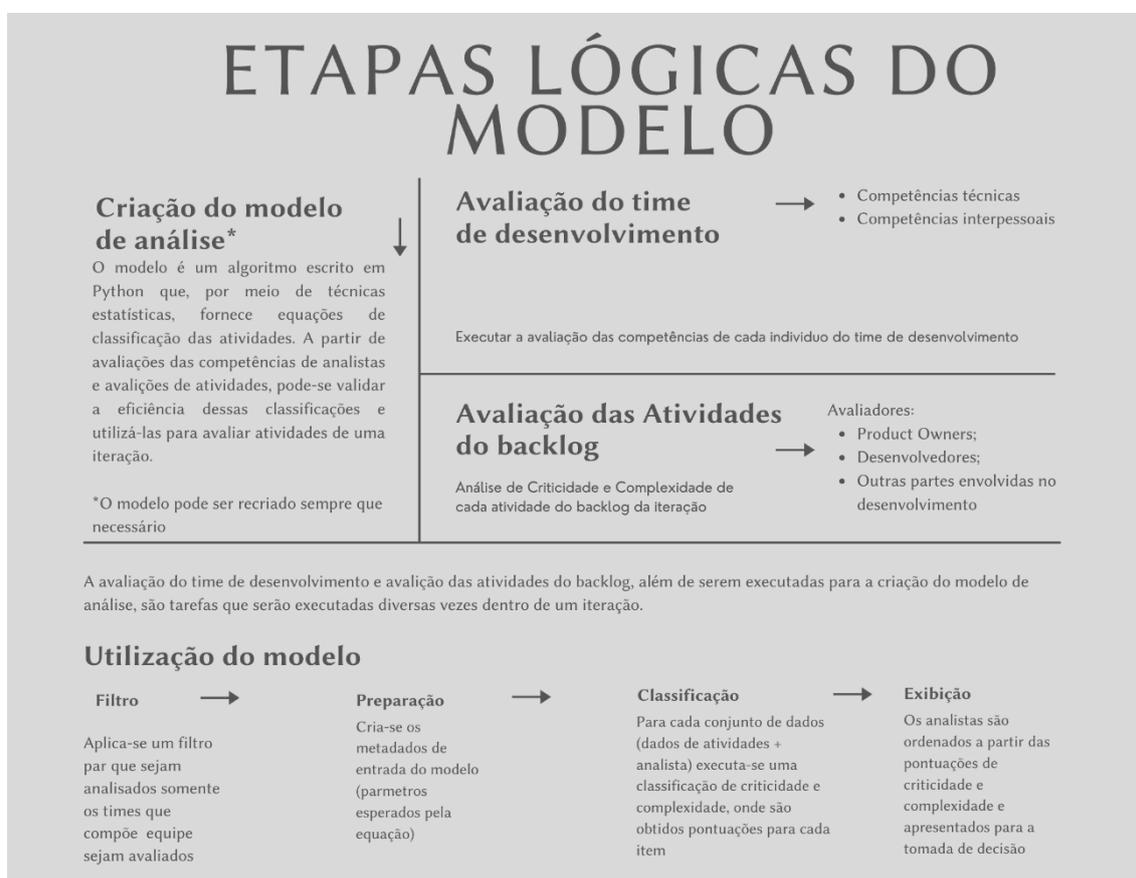


Figura 8. Infográfico: Construção e aplicação do modelo

Fonte: Dados de pesquisa, 2021.

O modelo proposto foi aplicado em uma empresa que atua no segmento de serviços, com desenvolvimento de seus produtos por meio do uso de práticas ágeis de

gestão de projetos. O modelo foi aplicado em 3 times de desenvolvimento ágeis, durante o planejamento de 4 iterações de cada time.

5.1 CONTEXTO DE APLICAÇÃO DO MODELO PROPOSTO

A pesquisa foi realizada em uma empresa brasileira do segmento de serviços. Foram analisados dados de dois times de desenvolvimento que atuam utilizando *frameworks* ágeis para realizar a gestão de atividades. Os times trabalham com iterações de 15 dias úteis, com as entregas priorizadas de acordo com o maior valor para o negócio da empresa.

5.1.1 Caracterização da organização

A organização é uma empresa líder de terceirização de veículos no Brasil, contando com uma frota de 85 mil veículos disponibilizados para clientes corporativos e 208 lojas físicas espalhadas por todo o território nacional. A empresa possui um quadro de 3.200 funcionários e estrutura organizacional projetizada, obtendo em 2019 o faturamento de 4,6 bilhões de reais.

Descrivendo a base de clientes, a empresa possui aproximadamente 3 milhões de clientes, entre pessoas físicas e jurídicas, no serviço de aluguel de veículos. Esse volume gera em média 65 mil atendimentos de suporte ao cliente por mês realizados pela equipe de SAC.

5.1.2 Descrição dos times de desenvolvimento

Como preparação para a etapa de ambiente laboral, foram selecionados dois times de desenvolvimento de projetos de software da empresa objeto de estudo, denominados nesta pesquisa como Time A e Time B. O Time A possui papel de exclusividade na atuação de projetos de autoatendimento da empresa, enquanto o time B é responsável pelo desenvolvimento de novos sistemas e integrações com o objetivo de entregar mais produtividade e competitividade para a empresa por meio do uso de novos *softwares*. O

papel de cada integrante de cada time possui uma relação direta para a execução do modelo conceitual proposto, em relação a competência técnica e interpessoal de cada integrante em relação a complexidade e a criticidade de cada atividade de uma iteração.

Time A

O Time A é um time novo na empresa, com criação em 2019, porém é o time mais estruturado dentre os times de desenvolvimento de *softwares* atuais. O time foi criado em setembro de 2019 e, na época da coleta de dados, possuía dois anos de existência. Atualmente, o time possui 7 integrantes, sendo 5 dedicados exclusivamente as atividades dos projetos de autoatendimento aos clientes da empresa. O papel de cada integrante está descrito na Tabela 6.

Tabela 6 Integrantes do Time A

Integrante	Função	Tempo na Empresa	Atuação
A-01	Coordenador	2 anos	Compartilhado
A-02	Coordenador	2 anos	Compartilhado
A-03	Programador Sênior	6 meses	Exclusivo
A-04	Programador Sênior	6 meses	Exclusivo
A-05	Coordenador	1 ano e 11 meses	Exclusivo
A-06	Especialista	3 meses	Exclusivo
A-07	Trainee	7 meses	Exclusivo

Fonte: Desenvolvido pelo autor, 2021.

Ao observar a Tabela 6, nota-se que o Time A possui 3 coordenadores, identificados como integrantes 1 e 2, que atuam no apoio a execução das atividades do integrante 5. Em relação ao integrante 5, vale destacar que ele trabalha exclusivamente na liderança do time, possuindo a responsabilidade de coordenar as atividades de todos os demais integrantes do Time A.

Papéis – Time A

A composição do Time A possui papéis bem definidos. Os integrantes 1, 2 e 5 são responsáveis por receber os dos clientes. Os requisitos são documentados, negociados e priorizados em ordem de valor para o cliente e, para a segunda fase, para que os requisitos possam entrar no *backlog* do produto do Time A. O integrante 5 é responsável por fazer a gestão das atividades e auxiliar os demais integrantes do time a executarem as

atividades, em relação à complexidade técnica para a execução de cada atividade e à criticidade para o projeto.

O integrante 1 atua na execução de atividades relacionada à arquitetura de *software* e em atividades que requerem interações com outros times técnicos da empresa, como por exemplo, o time de infraestrutura e do time de banco de dados, ambos que suportam tecnicamente a realização das atividades do Time A. O integrante 2 atua em atividades de arquitetura de *software* e em atividades de definição de regras de negócios complexas, relacionadas às necessidades de negócios do cliente. Os integrantes 3, 4, 6 e 7 que atuam como *Trainee*, programadores seniores e especialista respectivamente, com pouca interação com os clientes, mas possuindo como principal papel realizar a execução das atividades em busca garantir a melhor entrega de valor ao cliente por meio da finalização das atividades de uma iteração.

Time B

O time B é responsável pelo desenvolvimento de novos sistemas e integrações que visam entregar mais produtividade e competitividade empresa por meio da execução de projetos para atendimento interno da empresas objeto de estudo. O Time B foi constituído em julho de 2021. O time possui 7 integrantes, sendo 3 exclusivos do projeto. O papel de cada integrante está descrito na Tabela 7

Tabela 7. Integrantes do Time B

Integrante	Função	Tempo na Empresa	Atuação
B-01	Programador Frontend	3 meses	Exclusivo
B-02	Coordenador	2 anos	Compartilhado
B-03	Coordenador	2 anos	Compartilhado
B-04	Especialista	6 meses	Compartilhado
B-05	Programador Backend	4 meses	Exclusivo
B-06	Product owner	9 meses	Compartilhado
B-07	Programador Backend	1 ano	Exclusivo

Fonte: Desenvolvido pelo autor, 2021.

Ao se observar a Tabela 7, nota-se que o Time B possui 2 coordenadores, 1 *Product Owner*, 1 especialista e 3 programadores. Como destaque, o time possui uma pessoa com o papel e responsabilidade de alinhamento estratégico do *backlog* do produto, diferentemente do Time A.

Papéis – Time B

O time B possui papéis bem definidos, com abrangência de papéis de análise de requisitos de negócios (*Product Owner*). Apesar de possuir 2 integrantes que atuam exclusivamente no projeto, os papéis são claros e o time segue os ritos estabelecidos na metodologia ágil determinada pela área de governança da empresa para o gerenciamento de atividades. O integrante 6 é responsável por negociar com os clientes os requisitos do produto que serão desenvolvidos e já, previamente, priorizados com o cliente. O integrante também é responsável por gerenciar as atividades que entram no *backlog* e negociar os itens de cada iteração com o cliente.

Os integrantes 2, 3 e 4 são responsáveis por auxiliar na definição do *backlog* do produto e do *backlog* da iteração (quando o time se compromete a desenvolver as atividades em uma iteração), definir padrões técnicos e melhores práticas, realizar a revisão de código das atividades concluídas, disponibilizar a infraestrutura necessária para o funcionamento do sistema e realizar a mentoria e treinamento do time. Os integrantes 1, 5 e 7 trabalham de forma exclusiva no projeto, atuando no desenvolvimento das atividades de programação de *front-end* do *software*. O *front-end* corresponde a parte visual do sistema e o *backend* é a programação que realiza os processamentos no servidor. Esses integrantes não possuem interações com os clientes.

5.1.3 Descrição da ferramenta computacional

Em posse do modelo conceitual previamente apresentado na Figura 7, foi elaborada uma ferramenta computacional para a etapa de testes laborais na empresa foco desta pesquisa. A ferramenta foi desenvolvida em uma arquitetura distribuída, com a adoção das linguagens de programação *Python* 3.9, *HTML* 5 e *CSS* 3. A linguagem *Python* foi escolhida pelo fato de existir diversas bibliotecas para análise e classificação e ser amplamente utilizada por cientista de dados e acadêmicos. Já o *HTML* 5 e *CSS* 3 foram escolhidos para o desenvolvimento da interface com o usuário, por meio dessas linguagens, é possível criar interfaces amigáveis e prover interação com sistemas de processamento de dados via *API*, *Application Interface Programming*, utilizando a linguagem de programação *Python*.

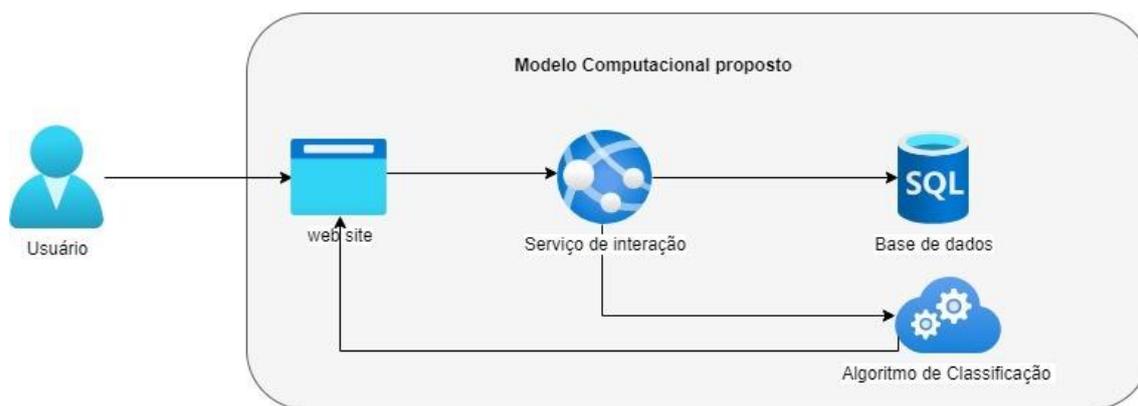


Figura 9. Modelo conceitual das competências técnicas e interpessoais dos times ágeis para a autogestão de atividades de uma iteração

Fonte: Dados de pesquisa, 2021.

Ao observar a Figura 9, fica explícito que o usuário corresponde a pessoa que irá interagir com o modelo computacional proposto, sendo responsável pela inclusão dos dados referente as atividade de uma iteração, por meio de um *web site*. O *web site* é a camada de *software* responsável por encapsular os dados imputados e por efetuar a interação com o algoritmo de classificação. De posse dos dados, o algoritmo efetua o processamento das equações para o gerenciamento de atividades, retornando ao *web site* um conjunto de tabelas e gráficos com a complexidade das atividades frente as competências do time de desenvolvimento. Em questões técnicas, o modelo computacional proposto foi desenvolvido sob a plataforma *Microsoft Visual Studio 2019*, utilizando as linguagens ASP NET MVC e *Microsoft C#*, escolhidas por conveniência e por serem linguagens utilizadas em multiplataformas de acesso ao usuário, como *laptops*, *tablets* e *smartphones*. O algoritmo de classificação foi desenvolvido na linguagem de programação Python, selecionada por ser um linguagem voltada especificamente para cálculos matemáticos.

A ferramenta computacional possui quatro etapas de funcionamento. As etapas estão encadeadas logicamente, com a origem na identificação e nivelamento das competências técnicas e interpessoais dos integrantes dos times, passando pela análise dos requisitos do *backlog* em relação a complexidade e criticidade até a análise do perfil de cada integrante por cada atividade. As etapas para a implementação da ferramenta computacional são destacadas na Figura 10.

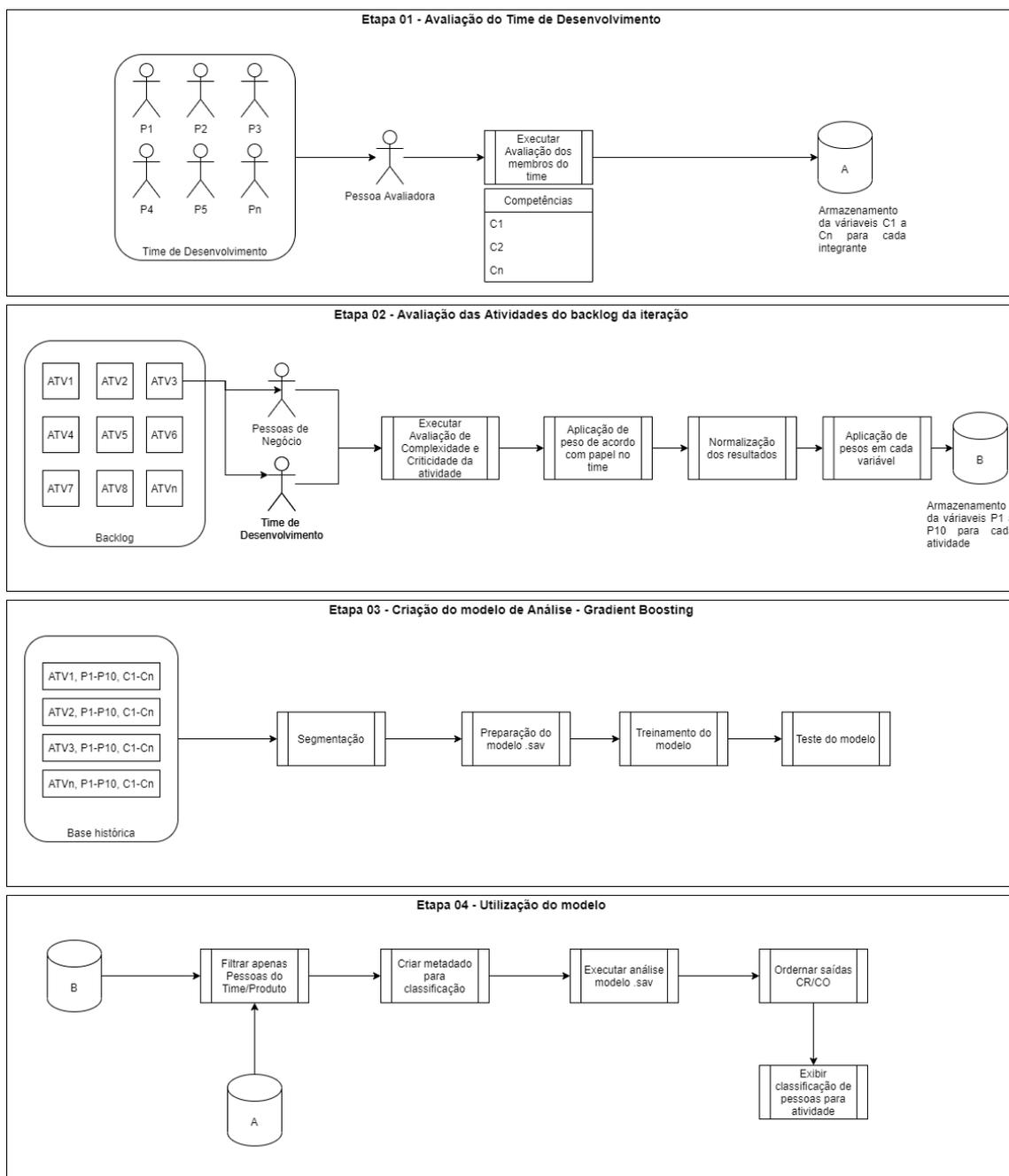


Figura 10. Etapas de implementação da ferramenta computacional

Fonte: Dados de pesquisa, 2021.

A seguir são detalhados todos os processos de cada etapa de implementação da ferramenta computacional proposta.

5.1.3.1 Etapa 01 – Avaliação do Time de Desenvolvimento

Na primeira etapa da implementação do modelo, foi realizada uma avaliação dos integrantes de cada equipe pelo gestor ou líder do time, objetivando a avaliação da realização de um mapeamento das Competências Técnicas e Competências Interpessoais de cada indivíduo que compõe o time de desenvolvimento para uso posterior no modelo de classificação. Inicialmente foram mapeadas quinze competências que se fazem necessárias para que um time seja auto-organizado e tenha a capacidade de realizar a autodesignação de atividades. A Tabela 8 apresenta as competências que são agrupadas em duas dimensões: **Técnicas** (C1-C8) e **Interpessoais** (C9-C15).

Tabela 8. Competências dos integrantes do Time de Desenvolvimento

Dimensão	Tipo	Competência
Técnicas	C1	Planejamento de Atividades
	C2	Compreensão de Atividades
	C3	Mensuração das atividades
	C4	Capacidade de Abstração
	C5	Análise dos Requerimentos
	C6	Monitoramento das Atividades
	C7	Conhecimento em Tecnologia
	C8	Decomposição de atividades
Interpessoais	C9	Negociação
	C10	Influência
	C11	Comunicação Efetiva
	C12	Senso de Auto-organização
	C13	Adaptabilidade a Mudanças
	C14	Desenvolvimento de Atividades em Equipe
	C15	Senso de Importância

Fonte: Dados da Pesquisa, 2021.

A avaliação das competências deverá ser realizada de forma individual, nominal e utilizar uma escala *Likert* de dez pontos. É importante ressaltar que, de acordo com característica evolutiva de times ágeis, recomenda-se refazer a avaliação de competências a cada uma ou duas iterações completas, a fim de ter os dados de novos participantes do time e/ou registrar o desenvolvimento de competências pelos membros que já compõem o time. A atualização constante da avaliação de competências do time permitirá que o modelo seja capaz de realizar a classificação e ranqueamento de acordo com as competências mais atuais do time. A Fase de avaliação de competências do time de desenvolvimento é apresentada na Figura 11.

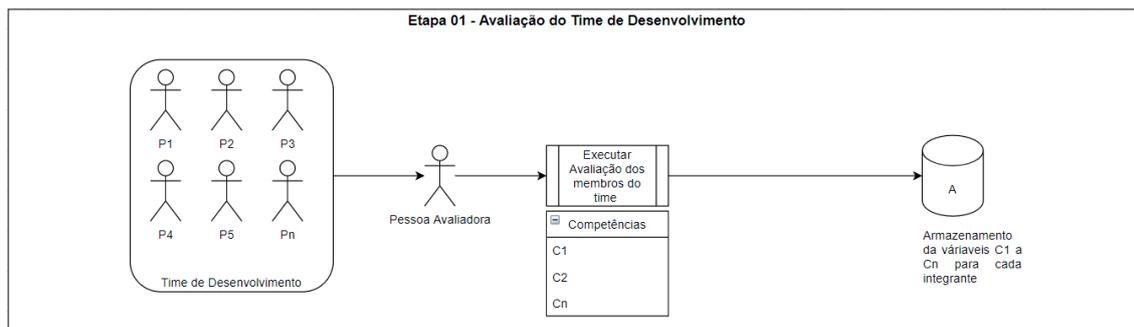


Figura 11. Processo de avaliação dos integrantes do time de desenvolvimento pelo gestor da equipe.

Fonte: Desenvolvido pelo autor, 2021.

Como visto anteriormente, cabe destacar que as competências técnicas possuem relação estreita com a parte de Complexidade de execução das atividades, enquanto as competências interpessoais possuem relação próxima com a dimensão Criticidade. Entretanto, como as atividades podem ser agrupadas por categorias lógicas para seu desenvolvimento, eventualmente, algum tipo de atividade pode ter uma correlação diferente entre as variáveis. Cada uma das competências listadas acima será analisada como uma variável dentro do modelo. A Tabela 9 demonstra o resultado de um processo de avaliação de competências.

Tabela 9. Avaliação de competências de integrantes do Time de Desenvolvimento

Pessoas	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
Pessoa 1	10	9	6	9	5	5	4	6	9	2	3	2	4	4	8
Pessoa 2	10	2	3	8	8	9	1	9	8	7	7	7	4	2	4
Pessoa 3	8	5	10	10	7	7	4	7	5	5	7	7	8	3	4
Pessoa 4	5	2	8	6	9	5	1	1	1	9	7	7	6	1	4
Pessoa 5	7	10	9	10	2	6	4	10	5	7	3	9	4	6	9
Pessoa 6	2	1	2	2	3	3	5	2	6	8	6	4	2	2	2

Fonte: Dados da Pesquisa, 2021.

Após a realização da etapa 01, a organização poderá avançar para segunda etapa de implementação da ferramenta computacional, detalhada no próximo tópico.

5.1.3.2 Etapa 02 - Avaliação das Atividades do *backlog* da iteração

A segunda etapa realizada no início de cada iteração, com seu início após os ritos de definição de backlog. A sugestão de realizar esta etapa após o planejamento da iteração dá-se exclusivamente pela necessidade de entendimento necessário dos requisitos de

negócio, da clareza dos objetivos, da definição das restrições e das dependências de cada atividade por possíveis integrantes do time, tanto por desenvolvedores quanto por pessoas de negócio.

Esta etapa tem como objetivo coletar a avaliação empírica do time de desenvolvimento e pessoas de negócio sob a **Complexidade (CO)** e **Criticidade (CR)** de cada uma das atividades que compõe o backlog da iteração. A Figura 12 representa o processo de avaliação da atividade do backlog da iteração.

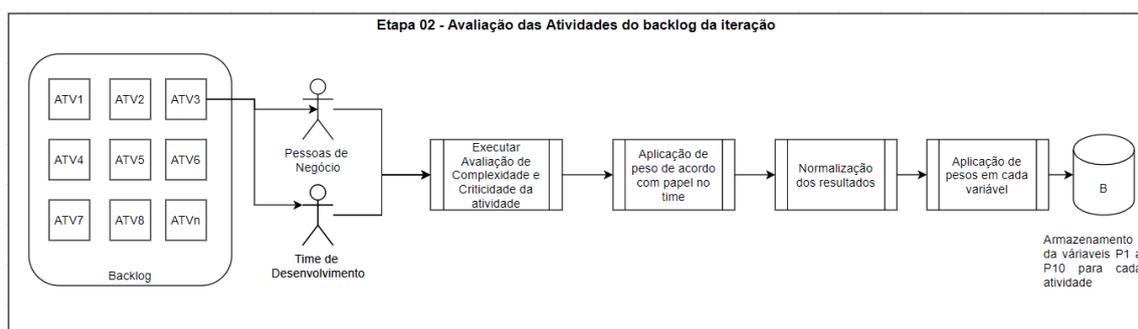


Figura 12. Processo de avaliação de Criticidade e Complexidade de uma atividade do *backlog*

Fonte: Desenvolvido pelo autor, 2021.

A avaliação de **Criticidade (CR)** e **Complexidade (CO)** foi realizada pelo time de desenvolvimento em conjunto com as pessoas de negócio. O formulário de avaliação possui 12 perguntas e está dividido em três dimensões. A primeira dimensão aborda a identificação da atividade e a qualificação do respondente. A identificação da atividade frente a ferramenta de gestão é importante para que seja possível realizar futuras consultas sobre o desenvolvimento da atividade, agindo assim como fonte de alimentação do modelo. Em contrapartida, a qualificação do respondente é importante para poder atribuir o peso correto para cada subconjunto de respostas. A Tabela 10 apresenta o questionário para avaliação de Criticidade e de Complexidade.

As dimensões dois e três são compostas por questões assertivas com escala de dez pontos que visa avaliar a atividade. A segunda dimensão possui cinco perguntas (P1-P5) que estão relacionadas a criticidade da atividade e buscam compreender a importância da execução da atividade frente à estratégia e entrega de valor para o produto/organização. A terceira dimensão possui cinco perguntas (P6-P10) que estão relacionadas a complexidade de execução da atividade, geralmente mais próximo de questões técnicas.

Tabela 10. Questionário de avaliação de Criticidade e Complexidade

Dimensão	Item Avaliação
Criticidade	Essa atividade contribui para a entrega de valor relacionado aos OKRs da empresa/produto Essa atividade está alinhada com a estratégia/negócio da empresa Essa atividade impacta positivamente no negócio/produto A prioridade de execução dessa atividade na iteração A objetividade das regras de negócio para a execução dessa atividade
Complexidade	A experiência do time em relação a esse tipo de atividade A execução dessa atividade depende de outras atividades O nível de aceitação do time frente a mudanças no gerenciamento da atividade A tecnologia que o time utiliza é o suficiente para desenvolver a atividade As competências técnicas do time estão alinhadas com a estimativa de execução da atividade

Fonte: Dados da Pesquisa, 2021.

Após cada integrante do time realizar a avaliação da atividade, o modelo irá ponderar sobre as respostas, de acordo com o conhecimento do respondente face a dimensão avaliada. Tratando-se de respostas de pessoas de negócio relacionadas a dimensão de Criticidade (impacto direto na entrega de valor ao cliente), acredita-se que essas respostas deveriam ter um peso maior uma vez que pessoas com esse papel deveriam ter mais propriedades sob este aspecto. Por outro lado, as respostas das pessoas com papel técnico dentro do time deverão possuir maior peso nas respostas enquadradas na dimensão de Complexidade (impacto direto no desenvolvimento de cada atividade), por se tratar de itens relacionados a execução técnica da atividade. Para a elaboração da ferramenta computacional, foi desenvolvido um formulário na ferramenta *Google Forms*, disponível no Apêndice D.

A notação utilizada para realizar a ponderação está descrita na Equação 1:

$$\text{Equação 1: } p_n = R_n \cdot x$$

Onde:

p_n representa o valor final do item avaliado após a aplicação do peso.
 R_n representa o valor selecionado pelo respondente para cada resposta.
 x representa o peso aplicado.

A Tabela 11 apresenta os pesos adotados para realizar a avaliação da atividade em relação aos integrantes do time.

Tabela 11. Definição de pesos para papéis

Papel/Dimensão	Criticidade	Complexidade
Pessoa de Negócio	1,2	1
Time de desenvolvimento	1	1,2

Fonte: Dados da Pesquisa, 2021.

Após a etapa de ponderação, a ferramenta efetua a normalização das respostas com o objetivo de ter uma avaliação única para atividade. A normalização está representada na Equação 2:

$$\text{Equação 2: } P_x = \frac{\sum P_x p}{n}$$

Onde:

P_x : Média ponderada da assertiva respondida (P1-P10)

p : Representa o peso aplicado aquela variável, recomenda-se utilizar 0,2 como valor inicial

n : Número de respostas obtidas para cada variável

As Tabelas de 12 à 14 representam o passo a passo da aplicação das fórmulas, desde o momento em que o formulário foi respondido pelas pessoas de negócio e time de desenvolvimento.

Tabela 12. Avaliação inicial da atividade

ATV	Papel	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1524	PN	7	6	9	7	9	8	9	8	10	8
1524	PD	3	7	5	1	3	2	2	3	10	9
1524	PN	4	10	7	9	8	3	4	1	5	4
1524	PD	7	7	3	8	2	5	4	8	3	2
1524	PD	8	3	4	5	4	9	5	7	7	1
1524	PD	7	9	10	1	10	10	7	1	5	8
1524	PD	8	9	5	5	6	10	2	8	9	8
1524	PD	2	6	6	2	10	8	2	4	4	6

Fonte: Dados da Pesquisa, 2021.

A coluna papel indica o perfil do respondente. PN (Pessoa de Negócio) representa que o questionário foi respondido por uma pessoa de negócio, já PD (Pessoa de desenvolvimento) representa que a resposta foi originada de uma pessoa do time de desenvolvimento. Como próxima etapa, são aplicados os pesos de acordo com o perfil do respondente, obtendo-se a Tabela 10.

Tabela 13. Avaliação com pesos de papéis aplicados

ATV	Papel	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1524	PN	8,4	7,2	10,8	8,4	9	8	9	8	10	8
1524	PD	3	7	5	1	3,6	2,4	2,4	3,6	12	10,8
1524	PN	4,8	12	8,4	10,8	8	3	4	1	5	4
1524	PD	7	7	3	8	2,4	6	4,8	9,6	3,6	2,4
1524	PD	8	3	4	5	4,8	10,8	6	8,4	8,4	1,2
1524	PD	7	9	10	1	12	12	8,4	1,2	6	9,6
1524	PD	8	9	5	5	7,2	12	2,4	9,6	10,8	9,6
1524	PD	2	6	6	2	12	9,6	2,4	4,8	4,8	7,2

Fonte: Dados da Pesquisa, 2021.

A terceira e última operação desta etapa é responsável por consolidar todas as avaliações de uma atividade aplicando a média ponderada, demonstrada na Tabela 11.

Tabela 14. Avaliação ponderada da atividade

ATV	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1524	7,23	9,03	7,83	6,18	8,85	9,57	5,91	6,93	9,09	7,92

Fonte: Dados da Pesquisa, 2021.

Estes dados foram armazenados pela ferramenta para servirem de entrada para o modelo de classificação na etapa 03.

5.1.3.3 Etapa 03 - Criação do modelo de Análise - *Gradient Boosting*

A partir do um histórico de atividades executadas em conjunto com as competências avaliadas dos executores das atividades, são criados um conjunto de modelos de classificação utilizando a técnica *Gradiente Boosting*. Os modelos são criados pela ferramenta aos pares.

O primeiro par define a classificação de Complexidade (CO) e o segundo par define a classificação de Criticidade para um conjunto de dados de atividades existentes em um *backlog* de uma iteração. Para essa etapa, a ferramenta pode criar diversos modelos para atender as especificidades de um conjunto de atividades do *backlog*, por exemplo, o modelo de análise de um time de desenvolvimento *mobile* difere em diversos aspectos se comparado ao time de desenvolvimento de aplicações embarcadas (por exemplo: IoT). Desse modo, torna-se necessária a customização da complexidade e criticidade das atividades pelos integrantes do time.

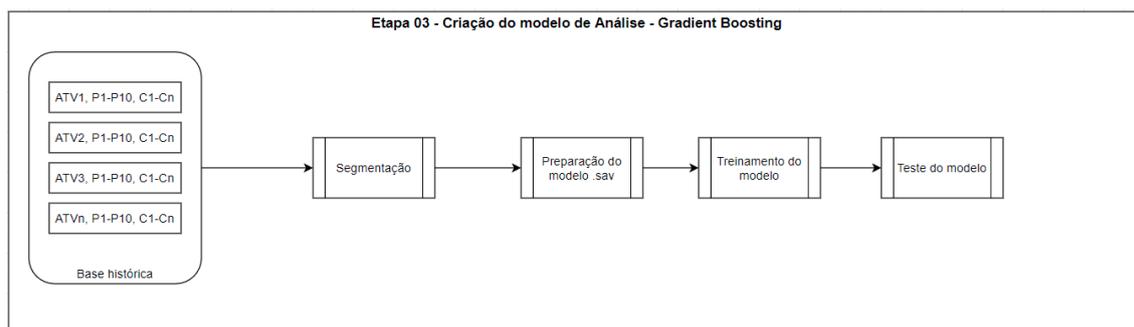


Figura 13. Desenvolvimento do modelo de classificação de Criticidade e Complexidade

Fonte: Desenvolvido pelo autor, 2021.

Para a criação do modelo foi realizado o processo de *feature engineering*, que se trata da análise, seleção e tratamento das variáveis que irão alimentar o modelo atendendo os requisitos pré-estabelecidos pela área de negócios (Turner *et al.* 1999).

As variáveis podem ser analisadas em um primeiro momento como colunas, apresentadas na Tabela 15.

Tabela 15. Processo *feature engineering*

ID	Identificador da Atividade
P1	Variáveis de análise de Criticidade
P2	
P3	
P4	
P5	
P6	Variáveis de análise de Complexidade
P7	
P8	
P9	
P10	
CR	Criticidade Estimada
CO	Complexidade Estimada
IdPessoa	Identificador do indivíduo que executou a atividade
C1	Avaliação das competências técnicas do indivíduo na execução da atividade
C2	
C3	
C4	
C5	
C6	
C7	
C8	
C9	Avaliação das competências interpessoais do indivíduo na execução da atividade
C10	
C11	
C12	
C13	
C14	
C15	
T_ATV	Tipo de Atividade
AREA_SOLICITANTE	Área solicitante
Cargo	Cargo da pessoa que executou a atividade
Data_Nascimento	Data de Nascimento da pessoa que executou a atividade

Fonte: Dados da Pesquisa, 2021.

As *features* (variáveis) demonstradas na Tabela 15 constituem a base para o modelo. Além dessas variáveis, poderão ser incluídas no modelo variáveis de controle em modelos específicos, que tem por objetivo, atribuir qualificações requeridas para a execução da atividade. Assim, são criados dois modelos específicos para cada tipo de atividade, essa abstração possibilita a inclusão de variáveis de controle e a manipulação

correta dos pesos para variáveis de acordo com o tipo de atividade, tornando a predição mais assertiva.

5.1.3.4 Etapa 04 - Utilização do modelo

O modelo utiliza técnicas de clusterização e classificação fazendo uso do algoritmo *GradientBoostingClassifier*. O algoritmo permite atribuir pesos graduais para variáveis, permitindo assim criar uma variação de análise frente aos tipos de atividades que acontecem dentro de iterações de times de desenvolvimento.

Cada tipo de atividade irá conter dois modelos, o primeiro será responsável por estimar as competências técnicas de cada integrante do time frente a complexidade da atividade. Já o segundo modelo é responsável por analisar a competências interpessoais face a criticidade estimada. O modelo espera como entrada os dados no formato exibido na Figura 14.

Análise de Criticidade/Competências Interpessoais

ID	P1	P2	P3	P4	P5	C9	C10	C11	C12	C13	C14	C15	T_ATV	AREA_SOLICITANTE	Cargo	Data_Nascimento
----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-------	------------------	-------	-----------------

Análise de Complexidade/Competências Técnicas

ID	P6	P7	P9	P10	C1	C2	C3	C4	C5	C6	C7	C8	T_ATV	AREA_SOLICITANTE	Cargo	Data_Nascimento
----	----	----	----	-----	----	----	----	----	----	----	----	----	-------	------------------	-------	-----------------

Figura 14. Entrada de dados de Criticidade e Complexidade.

Fonte: Desenvolvido pelo autor, 2021.

Na Figura 14 é descrito os campos necessários para que o modelo possa realizar a avaliação de complexidade e criticidade referentes a uma atividade. Os dados contemplam informações referentes à atividade e às competências técnicas e interpessoais dos analistas.

Cada especialização do modelo pode também requerer variáveis adicionais para que a análise seja mais assertiva. Para cada “atividade X indivíduo” analisado os **modelos retornam um score decimal de 0 a 1** pontuando a **aderência dos indivíduos** face a análise da atividade. De posse dos dados, pode ser plotado um gráfico de radar para análise visual da classificação dos indivíduos como apresentado na Figura 15.

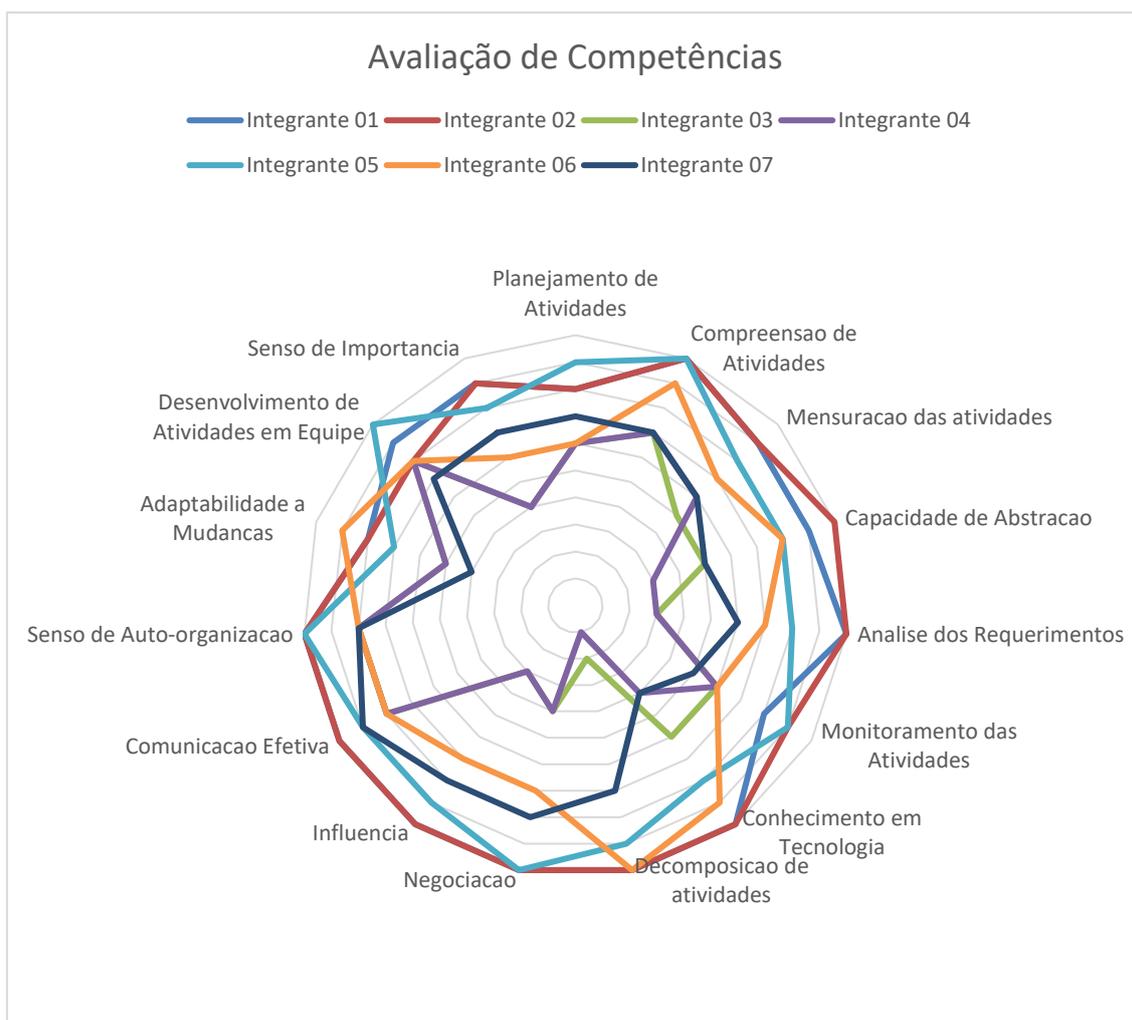


Figura 15. Análise de classificação de Criticidade e Complexidade.

Fonte: Desenvolvido pelo autor, 2021.

Ao observar a Figura 15 é possível identificar o nível de desenvolvimento das competências técnicas e interpessoais dos indivíduos dos times de desenvolvimento analisados. O gráfico apresentado será empregado no processo de classificação dos indivíduos frente as atividades. O gestor poderá utilizar o gráfico para elaborar planos de evolução das competências técnicas e interpessoais com objetivo de construir um time auto-organizado e auto gerenciável.

Avaliação do time de desenvolvimento

Antes da aplicação do modelo, foi realizada uma avaliação de cada um dos integrantes dos times A e B. Essa avaliação aconteceu por meio de um comitê, onde participaram o Gerente Sênior, além dos coordenadores dos times A e B. Para que a avaliação ocorresse de forma assertiva e os vieses fossem mitigados, foram estabelecidos critérios de avaliação para cada competência, detalhados na Tabela 15.

Tabela 16. Critérios de avaliação de competências

Tipo de Competência	Competência	Descrição	Critério de avaliação
Técnicas	Planejamento de Atividades	Trata-se da capacidade de planejar as atividades necessárias, assim como identificar a precedência entre elas necessária para completar a iteração e entregar valor ao cliente.	É capaz de analisar o contexto como um todo ou apenas a unidade de trabalho? Consegue visualizar a precedência das atividades?
	Compreensão de Atividades	Trata-se da capacidade de compreender o contexto e o trabalho necessário para a execução da atividade de forma correta, de acordo com o que foi solicitado.	Compreende bem o que deve ser entregue? É necessário explicar diversas vezes o escopo de uma mesma atividade?
	Mensuração das atividades	Trata-se da capacidade de mensurar o esforço necessário para executar uma atividade. Esta capacidade é desenvolvida durante as iterações.	Os prazos informados são tangíveis? Há coerência entre a compreensão e a métrica informada?
	Abstração	A partir da compreensão das atividades é possível criar generalizações e abstrações dos requisitos, tornando tangível o desenvolvimento de requerimentos abstratos em funcionalidades sistêmicas.	É capaz de identificar itens semelhantes e abstrair?
	Análise dos requerimentos	Através do conhecimento sobre o contexto em que se atua, essa competência torna possível o desmembramento de requerimentos em atividades que se tornaram funcionalidades do software.	Analisa de forma sistêmica cada um dos requerimentos?
	Monitoramento das atividades	Diz respeito ao conhecimento em análise em ferramentas de acompanhamento das atividades do <i>backlog</i> de uma iteração.	Realiza acompanhamento e atualização constante do status da atividade que está executando?
	Conhecimento em Tecnologia	Esta competência trata-se do conhecimento em linguagens de programação, assim como padrões de projeto e padrões arquiteturais voltadas ao desenvolvimento de soluções.	Realiza atualizações e treinamentos periódicos na tecnologia em que atua?
	Decomposição de atividades	Trata-se da competência de desmembrar e agrupar atividades provenientes dos requerimentos. O desenvolvimento dessa competência está ligado ao desenvolvimento da competência de abstração.	É capaz de decompor uma atividade complexa em atividades mais simples para agilizar as entregas?
Interpessoais	Negociação	Capacidade de negociar a adição, extinção ou modificações de requerimentos e atividades no <i>backlog</i> .	Possui a capacidade de argumentar eventuais mudanças para adequação do <i>backlog</i> ?
	Influência	Habilidade de influenciar pessoas do time para alcançar objetivos pessoais ou organizacionais.	É capaz de alterar ou influenciar demais membros do time?
	Comunicação efetiva	Capacidade de transmitir uma mensagem de forma clara e objetiva sem perda de informações.	É claro nas informações que transmite? A comunicação é clara e precisa?

	Senso de auto-organização	Esta competência diz respeito a capacidade dos membros do time de organizarem sua rotina de trabalho, de forma que seja possível cumprir com a execução das atividades com as quais o time se comprometeu.	Administra as suas tarefas diárias? Entrega as tarefas no prazo acordado? Avisa previamente quando irá atrasar uma tarefa?
	Adaptabilidade a mudanças	Habilidade de se adaptar de forma rápida as mudanças que ocorrem durante uma iteração. Ao desenvolver essa habilidade, o integrante deixa de ver as mudanças de forma negativa e passar a enxergar a mudança como ação estratégica para alcançar resultados e entregar valor ao final da iteração.	Compreende as mudanças e se adapta a ela?
	Work in Pairs	Trata-se da capacidade de trabalhar em equipe, aceitar eventuais críticas construtivas ao trabalho desenvolvido e, dessa forma, realizar a troca de conhecimentos e experiências tanto técnicas quanto de contexto de negócio	Aceita críticas construtivas? É resistente a alterações propostas por colegas?
	Pair Programming	Trata-se da capacidade de trabalhar em equipe, aceitar eventuais críticas construtivas ao trabalho desenvolvido, e dessa forma, realizar a troca de conhecimentos e experiências, tanto as técnicas quanto as relacionadas ao contexto do negócio.	Aceita críticas construtivas? É capaz de compartilhar conhecimento com pares?
	Senso de importância	Esta competência diz respeito à habilidade de reconhecer a importância de eventos, atividades e comportamentos do time que podem afetar a iteração	Compreende a importância da atividade e prioriza de forma adequada?

A Tabela 16 foi construída para que fosse possível obter e definir critérios claros e concisos de avaliação e tornar a avaliação de competência mais justa para os diversos membros do time de desenvolvimento. Os critérios de avaliação foram criados a partir da percepção dos coordenadores e gerentes participantes dos projetos.

6 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os resultados obtidos após a implementação e validação do modelo em dois projetos.

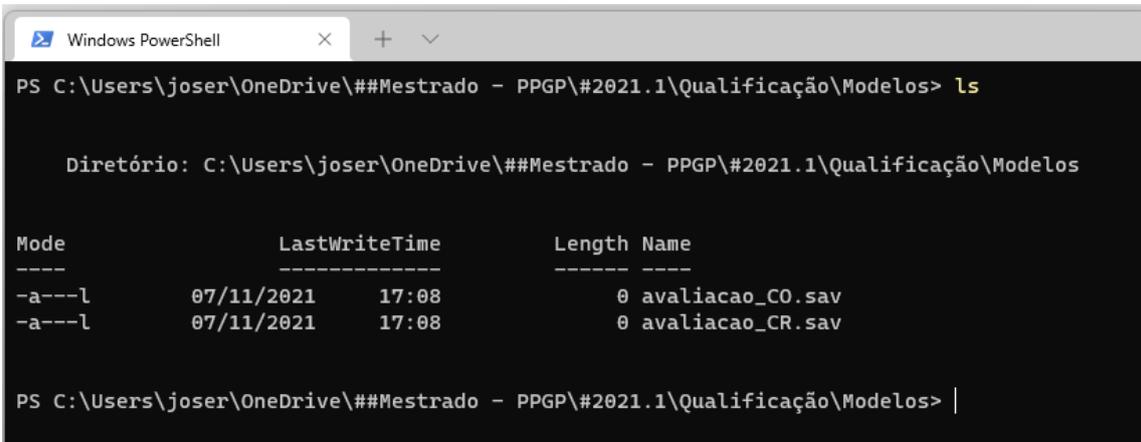
6.1 CRIAÇÃO DOS MODELOS DE ANÁLISE

Para que seja possível realizar o processamento dos dados, é necessário realizar a criação do modelo de classificação conforme descrito na Etapa 03 - Criação do modelo de Análise - *Gradient Boosting*. Para isso, é necessário que seja realizado o treinamento de um modelo a partir de um conjunto massivo de dados. Para alcançar esse objetivo, foram extraídos da ferramenta *Azure DevOps* dados referentes ao período de janeiro de 2021 à setembro de 2021 do Time A. Esse período representa a execução de 12 iterações de 15 dias úteis e continha 668 atividades executadas. Referente ao time B, foram extraídos dados do período de julho a setembro de 2021, onde haviam sido executadas 3 iterações completas contendo 56 atividades concluídas.

As atividades de ambos os times foram consolidadas e, ao final, obteve-se um total de 724 atividades para a criação do modelo. Inicialmente, foram analisadas dez atividades, onde o pesquisador, com o auxílio de especialistas em software e desenvolvedores, definiu manualmente a Criticidade (CR) e Complexidade (CO) das atividades. A realização da avaliação de CR e CO mostrou-se um processo moroso e demorado, além de incorrer na inserção de viés na avaliação, conforme observado por Salaou *et al.* (2021) no uso de outras ferramentas como o *Planning Poker*.

Com o objetivo de tornar o tratamento de toda a massa de dados mais sistêmico e evitar problemas como os apontados no uso de ferramentas, por exemplo o *Planning Poker*, apresentados por Salaou *et al.* (2021), foram adotadas as soluções apresentadas por Kaushik Tayal e Tayal (2020), que sugerem a utilização de ferramentas computacionais baseadas em análise de dados, como redes neurais e gráficos de velocidade. Para realizar essa fase os dados foram submetidos à biblioteca *Keras* do Python. A biblioteca *Keras* possibilita a inferência de comportamento de uma população a partir de uma amostra, possibilitando, dessa forma, inferir os níveis de CR e CO para toda a base de atividades coletada.

Após a execução do procedimento utilizando a biblioteca *Keras*, os dados foram tratados e utilizados para a criação e treinamento do modelo de classificação conforme apresentado na Figura 16. Para a criação do modelo, foram selecionadas de forma aleatória cerca de 506 atividades que representam 70% das atividades disponíveis. As demais 217, que representam 30% da massa de dados foram utilizadas a título de teste de acurácia do modelo. Essa prática de modelo e treino está relacionada as boas práticas do uso de *Machine Learning* no tratamento de dados para algoritmos de classificação (Chicco, 2017).



```

Windows PowerShell
PS C:\Users\joser\OneDrive\##Mestrado - PPGP\#2021.1\Qualificação\Modelos> ls

Diretório: C:\Users\joser\OneDrive\##Mestrado - PPGP\#2021.1\Qualificação\Modelos

Mode                LastWriteTime         Length Name
----                -
-a---l             07/11/2021   17:08           0 avaliacao_CO.sav
-a---l             07/11/2021   17:08           0 avaliacao_CR.sav

PS C:\Users\joser\OneDrive\##Mestrado - PPGP\#2021.1\Qualificação\Modelos> |

```

Figura 16. Modelos de classificação de Criticidade e Complexidade implementados em Python

Fonte: Dados de pesquisa, 2021.

Após essa etapa, os modelos encontravam-se criados e disponíveis para a execução de classificação de Criticidade (CR) e Complexidade (CO) das atividades da iteração. No entanto, os modelos de classificação que foram criados não possuíam uma interface amigável para o processamento dos dados. Com o objetivo de facilitar a utilização do modelo que foi codificado utilizando a linguagem de programação Python, foi desenvolvida uma API – *Application Programming Interface* – que possuía a responsabilidade de expor as funcionalidades de classificação do modelo para protocolos de comunicação que possibilitam a integração com aplicações que rodam em navegadores como o *Google Chrome*. Também foi desenvolvida uma aplicação *front end* utilizando a linguagem Angular.

Com os componentes desenvolvidos e integrados, o cenário proposto na Figura 9 estava completo e, dessa forma, tornou-se possível a utilização dos modelos demonstrados na Figura 10 de forma amigável, para realizar a coleta dos dados conforme

questionário descrito na Tabela 10 e enviar as informações para o processamento no modelo criado previamente.

6.2 ANÁLISE DAS AVALIAÇÕES DO TIME DE DESENVOLVIMENTO

De acordo com a prescrição do modelo, é necessário executar a Etapa 01 – Avaliação do Time de Desenvolvimento, para identificar as competências e poder fornecer insumos para que o modelo classifique o melhor indivíduo para a execução de determinada atividade, conforme descrito por Medeiros *et al.* (2020). Para isso, foram realizadas avaliações de competências dos integrantes dos times A e B. As avaliações de competências contemplam aspectos de Complexidade conforme proposto por López-Martínez *et al.* (2018) e Criticidade conforme destacado por Pascoa, Telha e Santos (2019). Podemos ressaltar também que o nível de desenvolvimento das competências do indivíduo pode impactar comportamentos e percepções frente às atividades. Para melhor visualização, a tabela com as avaliações foi dividida em duas, contendo primeiramente as competências técnicas e em seguida as competências interpessoais.

Tabela 17. Modelo para as dimensões *hard skills*

Analista	Time	Planejamento de Atividades	Compreensão de Atividades	Mensuração das atividades	Capacidade de Abstração	Análise dos Requerimentos	Monitoramento das Atividades	Conhecimento em Tecnologia	Decomposição de atividades
Integrante 08	B	5	7	4	4	7	3	7	6
Integrante 09	B	4	7	4	5	7	3	6	5
Integrante 01	A/B	8	10	9	9	10	8	10	10
Integrante 02	A/B	8	10	9	10	10	9	10	10
Integrante 10	B	8	10	9	9	10	8	10	10
Integrante 11	B	6	7	7	4	5	5	8	3
Integrante 03	A	6	7	5	5	3	6	6	2
Integrante 04	A	6	7	6	3	3	6	4	1
Integrante 05	A	9	10	8	8	8	9	8	9
Integrante 06	A	6	9	7	8	7	6	9	10
Integrante 07	A	7	7	6	5	6	5	4	7

Fonte: Desenvolvido pelo autor, 2021.

Na Tabela 17 são apresentadas as avaliações das competências técnicas de cada integrante, bem como os times em que os integrantes atuavam durante o estudo.

Tabela 18. Modelo para as dimensões *soft skills*

Analista	Time	Negociação	Influência	Comunicação Efetiva	Senso de Auto-organização	Adaptabilidade a Mudanças	Desenvolvimento de Atividades em Equipe	Senso de Importância
Integrante 08	B	8	4	10	8	10	6	3
Integrante 09	B	3	4	7	6	6	6	3
Integrante 01	A/B	10	10	10	10	8	9	9
Integrante 02	A/B	10	10	10	10	8	8	9
Integrante 10	B	10	9	10	10	10	7	9
Integrante 11	B	8	3	8	3	2	2	4

Integrante 03	A	4	3	8	8	5	8	4
Integrante 04	A	4	3	8	8	5	8	4
Integrante 05	A	10	9	9	10	7	10	8
Integrante 06	A	7	7	8	8	9	8	6
Integrante 07	A	8	8	9	8	4	7	7

Fonte: Desenvolvido pelo autor, 2021.

Na Tabela 18 são apresentadas as avaliações das competências interpessoais de cada integrante, bem como os times em que os integrantes atuavam durante o estudo.

Podemos notar ainda que os indivíduos com competências mais desenvolvidas, sejam elas técnicas ou interpessoais, participam de mais de um time, fato que corrobora com o cenário apresentado por Gandomani (2017) em que times ágeis tem o papel de serem auto-organizados, colaborativos e possuírem um papel efetivo, onde o conhecimento e as competências de todos os integrantes do time possam ser empregados de maneira balanceada para a execução das atividades.

6.3 COLETA E AVALIAÇÃO DOS DADOS DA ITERAÇÃO

Os times A e B utilizam um *framework* ágil baseado no *Scrum*. A configuração da metodologia de trabalho dos times analisados vai ao encontro da metodologia descrita por Bosch e Bosch-Sijtsema (2011). As iterações possuem duração de 15 dias úteis e são precedidas por uma reunião denominada *planning*, em que a pessoa responsável pela gestão do *backlog* apresenta as *user stories* priorizadas ao time, explica detalhadamente o escopo de cada uma e tira eventuais dúvidas. A partir dessa explicação, o time técnico puxa para o *backlog* da iteração as *user stories* que possuem os pontos suportados pela capacidade de entrega do time para uma iteração.

O processo de puxar a atividade havia sido identificado na literatura e apontado por Colomo-Palacios *et al.* (2012) como características de um grupo multifuncional e auto-organizado com a responsabilidade de gestão das atividades de uma iteração. Nesta etapa ocorre o desmembramento de atividades mais específicas que devem ter duração máxima de 8 horas. Ambos os times utilizam a ferramenta Azure DevOps para realizar a gestão de iterações e a gestão de atividades. Lwakatare *et al.* (2019), sugere o uso de ferramentas que suportam processos de DevOps auxiliam no aumento da qualidade das entregas de uma ou mais iterações.

A ferramenta permite que sejam criadas visões personalizadas para acompanhamento das atividades de cada time/iteração, ampliando assim a comunicação dos integrantes do time, permitindo que ocorram entregas de valor ao cliente por meio da conclusão das atividades de uma iteração Žužek *et al.* (2020). Para validação do modelo, foram selecionadas e extraídas atividades de 4 iterações, sendo duas iterações do Time A e outras duas iterações do Time B. Durante a reunião de *planning*, foi disponibilizado aos participantes um formulário on-line para a execução da segunda etapa proposta pelo modelo, a Etapa 02 - Avaliação das Atividades do backlog da iteração, onde o principal objetivo é coletar a percepção de complexidade e criticidade da execução de cada atividade que foi incluída no backlog. A dinâmica aplicada foi

de cada integrante do time responder o formulário para cada atividade logo após ela ser criada no Azure DevOps. Dessa maneira todos os respondentes teriam conhecimentos necessários sobre o que deveria ser executado na atividade em questão e possuíam a identificação da atividade dentro da ferramenta de gestão.

6.4 PROCESSAMENTO DOS DADOS E ANÁLISE DOS RESULTADOS

Após a realização da Etapa 02 – Avaliação das Atividades do backlog da iteração por todos os integrantes do time de desenvolvimento, por meio do preenchimento do formulário para cada atividade, o software criado para a manipulação do modelo realiza o tratamento dos dados conforme descrito nos passos “Aplicação de peso de acordo com o papel no time” e “Normalização dos resultados” da Etapa 02 do modelo. A partir deste ponto, o software de manipulação faz o envio dos dados via API para que o modelo realize o processamento. O modelo retorna para cada atividade os integrantes do time em questão e a sua respectiva pontuação. Importante ressaltar que a pontuação é dividida em Pontuação de Criticidade, Pontuação de Complexidade e uma Pontuação média para que seja possível realizar a ordenação dos analistas.

Com o objetivo de avaliar a acurácia dos modelos de classificação de Criticidade (CR) e Complexidade (CO) descritos na Etapa 03 do modelo e detalhados na seção 6.1, foram selecionadas 4 atividades, sendo duas do Time A e duas do Time B. As atividades selecionadas possuem características distintas no que diz respeito à Criticidade e Complexidade. Na tabela 19 podemos verificar os detalhes das atividades selecionadas para análise.

Tabela 19. Atividades avaliadas

Time	Id	Descrição	Esforço estimado (horas)	Característica dominante
A	1702	Criar protótipo de looping de fornecedor	8	Complexa
A	1662	Verificar disponibilidade do veículo na data.	6	Critica
B	1583	Criar API para o <i>Endpoint</i> de Vilões	6	Complexa
B	1719	Incluir Coluna Tempo Médio nos <i>Endpoints</i> do Tubo de Produção	3	Critica

Fonte: Desenvolvido pelo autor, 2021.

A atividade 1702 tratava-se da criação de um protótipo de digitalização de um processo de atendimento de fornecedores em um novo canal de comunicação. Conforme proposto por

Busse e Weidner (2020), a atividade foi analisada pelos integrantes do time A e foi classificada como complexa devido à sua característica inovativa. A atividade também possuía diversas dependências de execução de outras atividades e necessitava de uma comunicação ativa e assertiva para que pudessem ser incluídos todos os aspectos do processo no protótipo, o que requer pessoas do time de desenvolvimento com conhecimentos técnicos desenvolvidos em um nível maior Kalenda, Hyna e Rossi (2018). Na Tabela 20 podemos verificar as respostas de análise de complexidade e criticidade dadas pelos integrantes do time A.

Tabela 20. Avaliação de Criticidade e Complexidade

Avaliador	Papel	Criticidade					Complexidade				
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Integrante 01	NEG	10,8	12	9,6	10,8	4,8	6	4	10	7	8
Integrante 02	DEV	10	8	4	8	4	8,4	6	10,8	8,4	8,4
Integrante 03	DEV	8	9	5	3	8	9,6	9,6	3,6	6	9,6
Integrante 04	DEV	7	6	8	3	9	4,8	3,6	8,4	8,4	12
Integrante 05	NEG	12	9,6	7,2	9,6	3,6	4	3	8	6	6
Integrante 06	DEV	6	4	4	6	5	3,6	9,6	10,8	3,6	9,6
Integrante 07	DEV	7	6	6	7	10	9,6	10,8	9,6	8,4	3,6
Médias		8,69	7,80	6,26	6,77	6,34	6,57	6,66	8,74	6,83	8,17
		7,17					7,39				
Mediana		7,50	6,90	6,13	6,39	7,17	5,69	8,13	8,57	6,41	8,89
		6,90					8,13				

Fonte: Desenvolvido pelo autor, 2021.

A partir da análise das respostas iniciais é possível verificar que os integrantes 1 e 5, que atuaram no papel de pessoas de negócio (NEG), atribuíram pontuações média maiores nas questões de P1 a P5, que são relacionadas à Criticidade. As pessoas com papéis relacionados à negócios tiveram média de respostas de 7,5, enquanto o time de desenvolvimento (DEV) teve uma média geral de 6,74. Podemos relacionar essa diferença ao tempo de atuação na empresa e à perspectiva de atuação. A primeira é influenciada pelo tempo médio dos integrantes do time que é de 13,2 meses, enquanto os integrantes 1 e 5 possuem cerca de 24 meses na empresa.

A diferença de tempo, para um maior entendimento da estratégia da empresa, bem como a adaptação e conhecimento da cultura da organização, podem determinar o nível de criticidade de uma atividade. As pessoas de negócio possuem uma perspectiva voltada aos negócios, o que influencia na avaliação de Criticidade. O aspecto supracitado demonstra a relevância de aplicar um modelo racional de avaliação de Criticidade e Complexidade a fim de mitigar problemas de viés e subjetividade durante a análise.

Ao realizar a análise voltada as questões de Complexidade, é possível verificar que a média de todos os integrantes foi de 6,46 enquanto a média dos integrantes ligados ao desenvolvimento foi de 6,50. O desvio padrão da população foi de 2,18, enquanto o desvio padrão das respostas dos desenvolvedores foi 2,14. A comparação entre a dispersão dos dados referentes a avaliação de complexidade demonstra que existe um padrão de comportamento semelhante, dado do qual o índice de desvio-padrão está próximo. A Tabela 21 foi gerada a partir da aplicação dos pesos.

Tabela 21. Normalização das avaliações de Criticidade e Complexidade

Avaliador	Papel	Criticidade					Complexidade				
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Integrante 01	NEG	10,8	12	9,6	10,8	4,8	6	4	10	7	8
Integrante 02	DEV	10	8	4	8	4	8,4	6	10,8	8,4	8,4
Integrante 03	DEV	8	9	5	3	8	9,6	9,6	3,6	6	9,6
Integrante 04	DEV	7	6	8	3	9	4,8	3,6	8,4	8,4	12
Integrante 05	NEG	12	9,6	7,2	9,6	3,6	4	3	8	6	6
Integrante 06	DEV	6	4	4	6	5	3,6	9,6	10,8	3,6	9,6
Integrante 07	DEV	7	6	6	7	10	9,6	10,8	9,6	8,4	3,6
Médias		8,69	7,80	6,26	6,77	6,34	6,57	6,66	8,74	6,83	8,17
		7,17					7,39				

Fonte: Desenvolvido pelo autor, 2021.

Após a aplicação dos pesos é possível gerar uma média ponderada da avaliação do grupo para cada dimensão avaliada. A interpretação do modelo é de que a média ponderada se trata de um consenso sobre os aspectos de criticidade e complexidade da atividade a ser executada. Para a atividade analisada, podemos perceber que a média ponderada de Criticidade é 7,17 enquanto a média ponderada dos itens de Complexidade é 7,39, ratificando a proposição inicial do time de desenvolvimento sobre a atividade possuir um aspecto maior de complexidade.

A identificação da característica da atividade é importante para a auto-organização do time. A partir da identificação da atividade, o time pode se organizar estrategicamente de duas maneiras: a) deixar o recurso mais adequado disponível para a execução de determinada atividade; b) criar um plano de desenvolvimento de competências para ter mais recursos capazes de executar a atividade com a mesma qualidade e primazia. Tratando-se de atividades mais críticas, espera-se que a pessoa responsável por executar a atividade tenha características como negociação e comunicação efetiva, sendo a de auto-organização mais desenvolvida, pois, devido à criticidade da atividade, ele poderá ter interações maiores com pessoas externas ao time e, provavelmente, as relações não serão tão amistosas. Eder *et al.* (2015) sustentam que as

peças são alocadas de acordo com suas competências por aquele que é responsável pelo projeto, com um custo relacionado à precedência das atividades. As atividades em execução são controladas e monitoradas pela pessoa responsável pelo projeto.

6.5 OUTPUT DO MODELO: CLASSIFICAÇÃO DOS ANALISTAS

Como etapa final, os dados da atividade e dos analistas foram submetidos ao modelo. A partir da combinação dos dados da tarefa e das características do analista, o modelo realizou a classificação de cada um dos analistas e gerou uma pontuação para as dimensões de Complexidade (CO) e Criticidade (CR). Na tabela abaixo 22 podemos verificar a classificação na íntegra:

Tabela 22. Classificação de Complexidade e Criticidade

Integrante	CO	CR	Score
Integrante 02	9,52	9,21	9,36
Integrante 01	9,27	8,99	9,13
Integrante 05	8,64	8,43	8,54
Integrante 06	7,76	7,66	7,71
Integrante 07	5,89	5,99	5,94
Integrante 03	5,01	5,21	5,11
Integrante 04	4,51	4,77	4,64

Fonte: Desenvolvido pelo autor, 2021.

Como análise inicial, podemos notar que o modelo classificou os integrantes 1, 2 e 5 com uma pontuação maior, tanto para Criticidade quanto para Complexidade. Esse comportamento era esperado, pois esses integrantes possuem características que sugerem maior senioridade e, portanto, capacidade analítica mais apurada. Na Figura 17 podemos comparar o nível de competência dos integrantes.

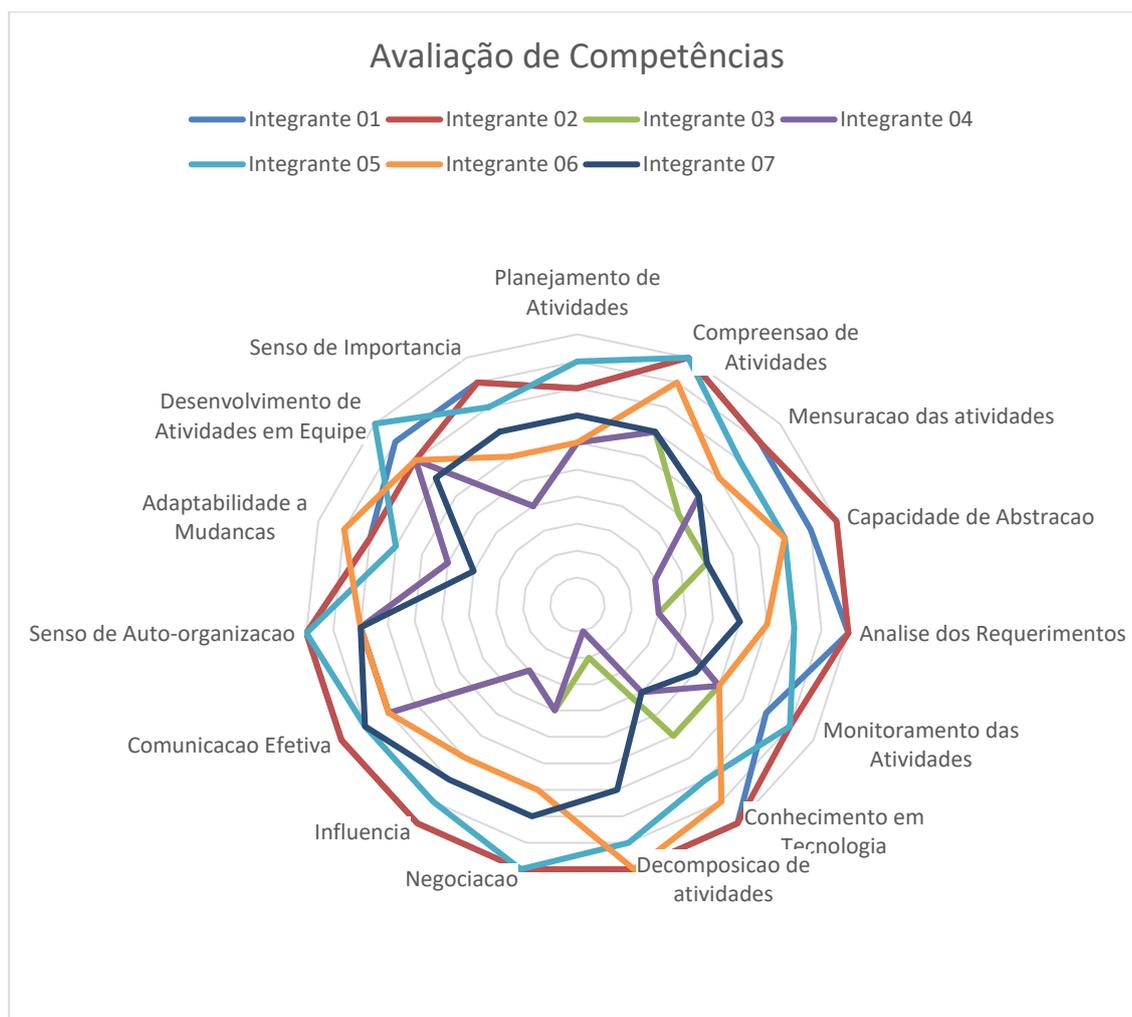


Figura 17. Análise de competências do time

Fonte: Desenvolvido pelo autor, 2021.

Tratando-se de competências, ao analisar a Figura 17, fica evidente que os integrantes 1, 2 e 5 possuem competências técnicas e interpessoais mais desenvolvidas que os demais integrantes. O integrante 2 destaca-se na avaliação por ter as competências técnicas em um grau de desenvolvimento maior.

Apesar do integrante 2 ser apontado pelo modelo como o recurso mais preparado para a execução da atividade, vale lembrar que no cenário analisado, ele é um recurso compartilhado do projeto e pode não estar disponível durante toda a iteração. Vale salientar que essa postura pode causar impactos no resultado de uma iteração, como por exemplo, em relação ao sequenciamento e dependências de atividades (Serrador & Pinto, 2015; Špundak, 2014)

O segundo lugar trata-se do integrante 1, que também é um recurso compartilhado e por isso também pode não estar disponível para a execução da atividade. A terceira opção é o recurso 5, mas como exposto na tabela 6, além de atuar como coordenador do time, ele é o

responsável pela interação com os clientes e elaboração das regras de negócio, isso significa que talvez ele não tenha tempo disponível para executar a atividade, assim como os integrantes 1 e 2.

Podemos concluir que escolhas que a princípio seriam óbvias, muitas vezes não se concretizam por interferências do próprio ambiente do projeto. Dessa forma, podemos constatar que o recurso mais adequado para executar a atividade, não é necessariamente o recurso que possui maior pontuação. O integrante 6 está classificado em quarto lugar, mas, ao analisar a sua avaliação de competências proposta pelo modelo, podemos verificar que as pontuações referentes às competências técnicas não possuem grandes variações quando comparadas aos integrantes 1, 2 e 5.

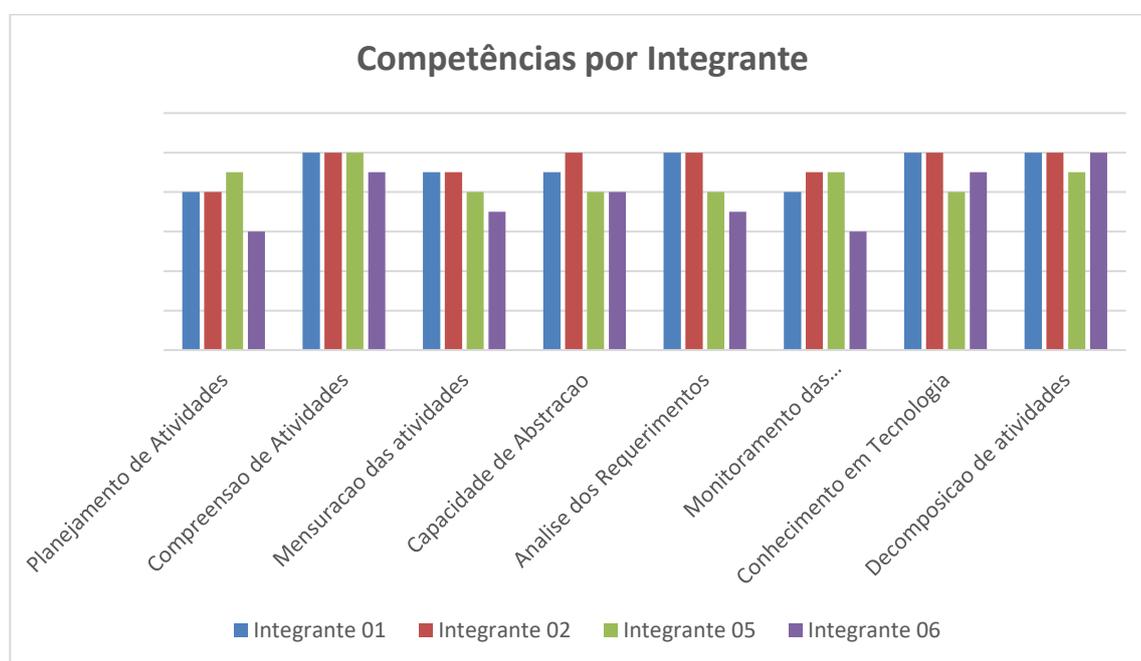


Figura 18. Análise de competências técnicas

Fonte: Desenvolvido pelo autor, 2021.

Ao analisar a Figura 18, podemos perceber que não existem grandes diferenças quanto ao nível de competências. Raj e Sinha (2020), apontam que os integrantes do time devem possuir as competências para adaptar-se ao processo de desenvolvimento e entrega de atividades, para validar a diferença real, foi computado na tabela abaixo, a média das competências técnicas dos três primeiros classificados e comparado com a do integrante 6. Com isso, podemos verificar que a maior diferença é de 2,67 e trata-se da competência de monitoramento de atividades. A competência “Decomposição de atividades” aponta que o integrante 6 possui melhor avaliação de competência se comparado com a média dos primeiros colocados.

Tabela 23. Ranking de Competências

Competência	Média TOP 3	Integrante 06	Diferença
Planejamento de Atividades	8,33	6,00	2,33
Compreensão de Atividades	10,00	9,00	1,00
Mensuração das atividades	8,67	7,00	1,67
Capacidade de Abstração	9,00	8,00	1,00
Análise dos Requerimentos	9,33	7,00	2,33
Monitoramento das Atividades	8,67	6,00	2,67
Conhecimento em Tecnologia	9,33	9,00	0,33
Decomposição de atividades	9,67	10,00	-0,33

Fonte: Desenvolvido pelo autor, 2021.

Conforme a Tabela 23, podemos notar que as diferenças são pequenas. É válido que o gestor do projeto analise cada caso de forma individual e sugira que o recurso 6 realize a autodesignação da atividade. O coordenador do time pode traçar um plano de desenvolvimento das competências: Monitoramento das Atividades, Análise dos Requerimentos, Planejamento de Atividades e Mensuração das atividades para que o integrante 6 se desenvolva profissionalmente e esteja apto a executar atividades que tenham maior grau de complexidade em interações futuras.

7 CONCLUSÃO

Nesta seção são apresentados os resultados da pesquisa que contribuem para as práticas relacionadas ao gerenciamento de atividades em projetos ágeis nas organizações. O objetivo geral e os objetivos específicos são revisitados e é demonstrado como eles foram alcançados. Além disso, esse capítulo relembra a questão de pesquisa e descreve como ela foi respondida, bem como são apresentados as limitações e propostas de estudos futuros.

7.1 CONSIDERAÇÕES FINAIS

Ao final desta pesquisa, após a construção e validação do modelo proposto, pode-se dizer que o objetivo geral foi alcançado. Destaca-se que o objetivo geral estabelecido foi desenvolver um modelo baseado em competências que auxiliem na autodesignação de atividades de um time ágil em uma iteração. Para alcançar esse objetivo, foi desenvolvida uma ferramenta computacional que encapsulou o modelo conceitual aplicado em ambiente laboral em uma empresa sob práticas ágeis de gerenciamento de projetos, atuante no segmento de serviços.

Nesse sentido, também foram alcançados os objetivos específicos. O primeiro objetivo específico foi descrever o processo de gerenciamento de atividades de um time ágil durante as iterações, que foi alcançado por meio da realização de uma RSL relacionada às práticas ágeis de gestão de projetos e à gestão de atividades pelo time de desenvolvimento. Os resultados da RSL foram categorizados e, com eles, foi possível identificar como ocorre o processo de gerenciamento de atividades de um time ágil durante as iterações, desde a fase de planejamento do *backlog* da iteração até o gerenciamento da entrega da atividade ao final de uma iteração.

O segundo objetivo foi identificar as competências necessárias às pessoas de um time ágil para a autodesignação das atividades em uma iteração. De posse das categorias resultantes da RSL, foi possível analisar e apresentar as competências técnicas e interpessoais relacionadas ao processo de gerenciamento e autodesignação de atividades em uma iteração. As principais competências técnicas encontradas foram: planejamento de atividades, compreensão de atividades, mensuração das atividades, capacidade de abstração, análise dos requerimentos, monitoramento das atividades, conhecimento em tecnologia e decomposição de atividades. As competências interpessoais foram: negociação, influência, comunicação efetiva, senso de auto-

organização, adaptabilidade às mudanças, desenvolvimento de atividades em equipe e senso de importância.

O terceiro objetivo específico foi verificar a adequação das competências das pessoas para a autodesignação das atividades de um time ágil em uma iteração. Após a realização de entrevistas em grupo e duas sessões de *focus group* resultou na identificação das competências técnicas e interpessoais necessárias para mensuração da complexidade para a execução de atividades e a criticidade de cada atividade em relação ao fim de uma iteração. Tais competências, tanto técnicas como interpessoais, serviram para a composição do modelo conceitual previamente apresentado na Figura 7.

O quarto objetivo específico foi construir um modelo baseado em competências que auxilie na autodesignação de atividades de um time ágil em uma iteração. As sessões de *focus group* resultaram em um questionário que compreende as competências técnicas e interpessoais presentes nos indivíduos de um time e a complexidade e a criticidade existentes nas atividades de uma iteração. Os resultados servem de parâmetros de entrada para um processo *Gradiente Boosting* de *Machine Learning*. Esse processo foi encapsulado em um código *Python* e disponibilizado para interação com os usuários por meio de um *web site*.

O quinto objetivo foi validar o modelo baseado em competências que auxilie na autodesignação de atividades de um time ágil em uma iteração. A ferramenta apresentou dois gráficos para análise visual da classificação dos indivíduos com as competências técnicas e interpessoais aderentes a cada atividade analisada, um gráfico de radar e um gráfico de colunas agrupado por competência de cada indivíduo em relação a atividade analisada.

Destacados o objetivo geral e os objetivos específicos, vale lembrar que a questão de pesquisa é: **Como o uso de um modelo baseado em competências pode contribuir na autodesignação de atividades de um time ágil em uma iteração?** Para responder essa questão, destaca-se que o principal benefício do uso da ferramenta foi reduzir o problema de mensuração e autodesignação de atividades e a subjetividade das avaliações quanto à criticidade e complexidade de uma atividade por meio da utilização do algoritmo de classificação *Gradiente Boosting*, promovendo ferramentas visuais para análise das competências técnicas e interpessoais dos indivíduos de um time frente as atividades a serem executadas em um ciclo de desenvolvimento de projetos ágeis. Além disso, o uso da ferramenta reduziu vieses de escolha, bem como a subjetividade das avaliações quanto à criticidade e complexidade de uma atividade frente aos indivíduos de um time em uma iteração.

Portanto, pode-se dizer que a questão da pesquisa e os objetivos foram atendidos, demonstrando assim, a finalização da pesquisa. Além disso, a construção e validação do modelo proposto confirmam a sua aplicabilidade, o que remete às contribuições nos campos teóricos e práticos apresentados a seguir.

7.2 CONTRIBUIÇÕES PARA PRÁTICA

A solução apresentada atua no problema de gerenciamento e auto designação de atividades dentro do modelo de desenvolvimento utilizando gestão através de metodologias ágeis. O modelo apresentado busca reduzir vieses de escolha, bem como a subjetividade das avaliações quanto à criticidade e complexidade de uma atividade por meio da utilização do algoritmo de classificação *Gradiente Boosting*, podendo ser utilizado com informações prévias da empresa, ou por meio de um processo de avaliação inicial.

Outra contribuição prática desta pesquisa é o desenvolvimento de uma ferramenta computacional, onde é possível realizar a avaliação das competências técnicas e interpessoais dos indivíduos de um time em relação à criticidade e complexidade para a execução de uma atividade. A ferramenta computacional foi registrada como registro de programa de computador e o modelo conceitual apresentado na Figura 7 foi realizado o pedido nacional de invenção de modelo de utilidade, ambos depositados no Instituto Nacional de Propriedade Industrial (INPI).

A solução está disponível e pode ser utilizada na forma de um *website* (<https://dissertacao.joseromualdo.com.br>) por meio de um navegador com acesso à *internet* e conta com um cadastro para acesso bem como uma tela de acesso para usuários já cadastrados conforme Figura 19.

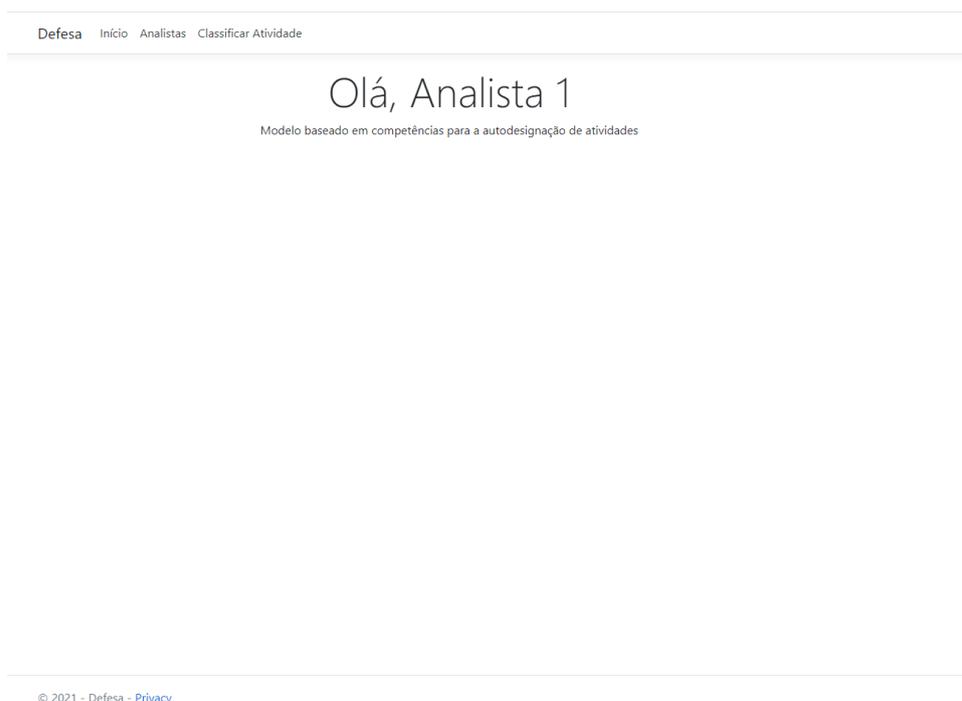


Figura 19 – Tela inicial do software

Fonte: Elaborado pelo autor, 2021.

Ao acessar a ferramenta, o *website* apresentará 2 opções, conforme demonstrado na Figura 17, onde será necessário realizar a avaliação de competência dos analistas para posteriormente realizar a avaliação das atividades. Esta etapa será necessária para atribuição das variáveis adequadas ao sistema.

Novo Analista			
Nome	Cargo	UltimaAvaliacao	
Analista 01	Desenvolvedor Pleno	06/08/2021	Editar Detalhes Excluir
Analista 02	Coordenador	06/08/2021	Editar Detalhes Excluir
Analista 03	Coordenador	06/08/2021	Editar Detalhes Excluir
Analista 04	Especialista	06/08/2021	Editar Detalhes Excluir
Analista 05	Programador Backend	06/08/2021	Editar Detalhes Excluir
Analista 06	Programador Backend	06/08/2021	Editar Detalhes Excluir
Analista 07	Desenvolvedor Pleno	06/08/2021	Editar Detalhes Excluir
Analista 01	Desenvolvedor Senior	06/08/2021	Editar Detalhes Excluir

Figura 20 – Lista de Analistas avaliados

Fonte: Elaborado pelo autor, 2021.

A avaliação de competências deverá ser realizada periodicamente. É possível também adicionar avaliação de competência de novos analistas ou realizar a exclusão de competência de um analista previamente cadastrado. Para a realização da avaliação da atividade no sistema será necessário que o cadastro de competências dos analistas seja finalizado, pois além de levá-lo em consideração, será útil no momento da auto designação dos analistas nas atividades, como demonstrado na Figura 18.

Avaliação de competências

Analista

Analista 01

<p>Planejamento de Atividades 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Compreensao de Atividades 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Mensuracao das atividades 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Capacidade de Abstracao 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Análise dos Requerimentos 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Monitoramento das Atividades 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Conhecimento em Tecnologia 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Decomposicao de atividades 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p>	<p>Negociação Influencia 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Comunicacao Efetiva 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Senso de Auto-organizacao 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Adaptabilidade a Mudancas 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Desenvolvimento de Atividades em Equipe 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p> <p>Senso de Importancia 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0</p>
---	--

Figura 21 - Cadastro de competências dos Analistas

Fonte: Elaborado pelo autor, 2021.

Uma vez finalizado o cadastro e a avaliação das competências dos analistas seguimos para a etapa de avaliação de criticidade e complexidade das atividades, como apresentados na Figura 19. Esta avaliação deverá ser realizada por todos os integrantes do time para cada atividade.

Avaliação da Atividade

Identificador da Atividade

Após concluída, qual a entrega de valor dessa atividade?

1 2 3 4 5 6 7 8 9 10

Quanto a execução dessa atividade contribui com a estratégia da empresa?

1 2 3 4 5 6 7 8 9 10

Quanto a conclusão dessa atividade irá causar impacto no negócio?

1 2 3 4 5 6 7 8 9 10

Quão rápida deverá ser a entrega dessa atividade?

1 2 3 4 5 6 7 8 9 10

Quão clara são as regras de negócio para a execução dessa atividade?

1 2 3 4 5 6 7 8 9 10

Qual a experiência do time em relação a esse tipo de atividade?

1 2 3 4 5 6 7 8 9 10

Quanto a execução dessa atividade depende de outras atividades?

1 2 3 4 5 6 7 8 9 10

Caso seja necessário realizar mudanças, quanto o time está disposto a aceitá-las?

1 2 3 4 5 6 7 8 9 10

Avaliar Atividade

Figura 22 – Avaliação de complexidade e criticidade das atividades

Fonte: Elaborado pelo autor, 2020.

A ferramenta permite que cada integrante do time avalie características de Complexidade e Criticidade da atividade, conforme a Figura 22.

Avaliação da atividade 783

[Avaliar nova atividade](#)

Classificação dos analistas - Gradient Boosting

Analista	CR	CO	Score
Integrante 02	9,52	9,21	9,36
Integrante 01	9,27	8,99	9,13
Integrante 05	8,64	8,43	8,54
Integrante 06	7,76	7,66	7,71
Integrante 07	5,89	5,99	5,94
Integrante 03	5,01	5,21	5,11
Integrante 04	4,51	4,77	4,64

Figura 23 – Apresentação da classificação dos analistas

Fonte: Elaborado pelo autor, 2021.

Ao fim do processo, o sistema estará apto para sugerir uma lista de classificação para a auto designação da atividade, o sistema permite inclusive que o usuário possa visualizar o *score* de criticidade, *score* de complexidade e *score* geral do analista, conforme Figura 23, permitindo a avaliação por parte dos gestores de quais aspectos deveriam ser melhorados nas competências de cada integrante do time.

Desta forma, o processo de avaliação de criticidade (CR) e complexidade (CO) é finalizado para a atividade, classificando os analistas de acordo com as suas competências e possibilitando que os gestores e o próprio time tenham um panorama das competências que precisam ser aperfeiçoadas, para que o time possa alcançar alta performance e alto desempenho.

Por fim, o modelo computacional proposto pode ser utilizado como apoio às ferramentas que auxiliam os times a realizarem a mensuração de atividades, como o *Planning Poker*, com o objetivo de eliminar limitações em relação à subjetividade da análise de criticidade e complexidade da atividade (Salaou *et al.*, 2021), além de permitir que o modelo contribua para que os gestores possam identificar competências a serem desenvolvidas e elaborar planos para que o time alcance níveis elevados frente às competências requeridas para a execução das atividades.

7.3 CONTRIBUIÇÕES PARA TEORIA

Em relação ao aspecto teórico, a principal contribuição desta dissertação é oferecer um modelo conceitual, que permita a compreensão da análise de criticidade e complexidade alinhada às competências interpessoais e técnicas para gestão de atividades em uma iteração. Esta pode ser compreendida como uma concepção inédita, uma vez que não foi encontrada a partir da RSL tal conteúdo nos artigos. Além disso, o modelo proposto serviu de base para construção da ferramenta de auxílio no processo de autodesignação de atividades de um time ágil em uma iteração, uma vez que as pesquisas efetuadas nas bases de dados acadêmicas não apresentaram ferramentas computacionais compatíveis a ferramenta computacional apresentada nesta pesquisa.

Em complemento, a contribuição no campo teórico está relacionada à identificação da complexidade para a realização das atividades e a sua respectiva relação com a criticidade de uma atividade não ser concluída em uma iteração em ambientes sob as práticas ágeis de gerenciamento de projetos. Aqui salienta-se que o time de desenvolvimento do projeto deve ser auto-organizado e responsável por selecionarem a atividade de acordo com suas competências. Para a realização dessa tarefa, o time deve apresentar competências técnicas e interpessoais com a finalidade de identificar a complexidade relacionada à execução de cada atividade e o seu nível de criticidade em relação ao término de uma iteração, o que está diretamente relacionado à entrega de valor ao cliente. Todos esses elementos foram compilados a partir da literatura estudada, formando assim os resultados da pesquisa.

Neste cenário, o modelo conceitual proposto identifica as competências técnicas e interpessoais de cada pessoa do time de desenvolvimento, minimizando o cenário de abstração existente na constatação da complexidade para a realização e a criticidade relacionada a não conclusão da atividade em uma iteração (Pascoa, Telha & Santos, 2019; Kaushik, Tayal & Yadav, 2020). O modelo conceitual proposto apoia as organizações independente do *framework* ágil adotado (Beck *et al.*, 2001; Grapenthin *et al.*, 2015; Hoda & Murugesan, 2016), contribuindo para que o time de desenvolvimento do projeto selecione as atividades priorizadas que mais geram valor ao cliente, mediante a melhor combinação de competências técnicas e interpessoais de cada pessoa, frente à complexidade e criticidade da atividade em uma iteração.

7.4 LIMITAÇÕES E PROPOSTA DE ESTUDOS FUTUROS

A ferramenta computacional proposta nesta pesquisa foi construída sob uma arquitetura de *software* flexível e escalável para implementação em outras organizações. Porém, a realização desta pesquisa limitou-se a uma única organização e dois times de desenvolvimento de projetos ágeis. Assim, os resultados desta pesquisa são considerados apropriados particularmente a empresa desta pesquisa. Desse modo, os resultados e as contribuições apresentados na pesquisa podem ser aplicados à demais empresas ágeis, para um número maior de projetos e um número maior de times de desenvolvimento.

Outra limitação está direcionada a adoção dos *frameworks* ágeis. A pesquisa foi realizada em uma empresa que adota o *Scrum* e o *Kanban* como *frameworks* ágeis para o desenvolvimento de seus projetos. Nesse sentido, tais limitações geram oportunidades de realizações de estudos futuros, como a validação da ferramenta computacional em organizações que adotam *frameworks* ágeis diferentes aos adotados nesta pesquisa.

Estudos futuros podem ser realizados para análise do uso da ferramenta computacional em organizações com práticas ágeis em escala. Sugere-se estudos em times de desenvolvimento de projetos ágeis escalados, como o *Scaled Agile Framework* (SAFe), *Scrum of Scrums*, *Disciplined Agile Delivery* (DAD) e *Large Scale Scrum* (LeSS). Estudos podem ser realizados para aperfeiçoar a ferramenta computacional em relação a identificação das competências técnicas e interpessoais do time de desenvolvimento, armazenando os dados de uma iteração para calibrar a ferramenta computacional para novas iterações, contribuindo assim para identificar as competências técnicas e interpessoais, por exemplo, da entrada de uma nova pessoa no time, ou na expansão do time para as práticas ágeis escaladas. Uma sugestão aqui é a realização de pesquisa para teste de práticas computacionais para apoio de análise de dados, como Inteligência Artificial e *Machine Learning*.

REFERÊNCIAS

- Adolph, S., Kruchten, P., & Hall, W. (2012). Reconciling perspectives: A grounded theory of how people manage the process of software development. *Journal of Systems and Software*, 85(6), 1269-1286.
<https://doi.org/10.1016/j.jss.2012.01.059>
- Alhamed, M., & Storer, T. (2020). Playing Planning Poker in Crowds: Human Computation of Software Effort Estimates.
- Akarsu, Z., & Yilmaz, M. (2020). Managing the social aspects of software development ecosystems: An industrial case study on personality. *Journal of Software: Evolution and Process*, 32(11), e2277.
<https://doi.org/10.1002/smr.2277>
- Al-Baik, O., & Miller, J. (2015). The kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering*, 20(6), 1861-1897.
- Alqudah, M. K., & Razali, R. (2017). Key factors for selecting an Agile method: A systematic literature review. *International Journal on Advanced Science, Engineering and Information Technology*, 7(2), 526-537.
- Altintas, C., & Azizoglu, M. (2020). A Branch and Bound Algorithm for a Multi-Mode Project Scheduling Problem With a Single Non-Renewable Resource. *International Journal of Information Technology Project Management (IJITPM)*, 11(2), 1-14.
<https://doi.org/10.4018/IJITPM.2020040101>
- Aranha, D. P., & Cardoso, M. D. O. (2017). Métodos e métricas para estimativas e planejamento de projetos ágeis de software: estudo comparativo entre pontos de função e story points.
- Astromskis, S., Janes, A., Sillitti, A., & Succi, G. (2014). Continuous CMMI assessment using non-invasive measurement and process mining. *International Journal of Software Engineering and Knowledge Engineering*, 24(09), 1255-1272.
<https://doi.org/10.1142/S0218194014400117>
- Atienza, C. M. R. (2017). Building organizational identity: An insider action research from a founder's viewpoint. *Systemic Practice and Action Research*, 30(6), 569-592.
<https://doi.org/10.1007/s11213-017-9410-2>
- Atkinson, R. (1999). Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International journal of project management*, 17(6), 337-342.
- Attarzadeh, I., & Ow, S. H. (2010). A novel algorithmic cost estimation model based on soft computing technique. *Journal of computer science*, 6(2), 117.
- Babenko, V., Lomovskykh, L., Oriekhova, A., Korchynska, L., Krutko, M., & Konjaieva, Y. (2019). Features of methods and models in risk management of IT projects. *Periodicals of Engineering and Natural Sciences*, 7(2), 629-636.
<http://dx.doi.org/10.21533/pen.v7i2.558.g330>
- Baumotte, A. C. T. (2015). *Gerenciamento de pessoas em projetos*. 3a Edição. Rio de Janeiro: Editora FGV.
- Barke, H., & Prechelt, L. (2019). Role clarity deficiencies can wreck agile teams. *PeerJ Computer Science*, 5, e241.
<https://doi.org/10.7717/peerj-cs.241>
- Batra, D., Xia, W., & Rathor, S. (2016). Agility facilitators for contemporary software development. *Journal of Database Management (JDM)*, 27(1), 1-28.
<https://doi.org/10.4018/JDM.2016010101>

- Beaumont, M., Thuriaux-Alemán, B., Prasad, P., & Hatton, C. (2017). Using agile approaches for breakthrough product innovation. *Strategy & Leadership*.
<https://doi.org/10.1108/SL-08-2017-0076>
- Belsis, P., Koutoumanos, A., & Sgouropoulou, C. (2014). PBURC: a patterns-based, unsupervised requirements clustering framework for distributed agile software development. *Requirements engineering*, 19(2), 213-225.
<https://doi.org/10.1007/s00766-013-0172-9>
- Bilgaiyan, S., Sagnika, S., Mishra, S., & Das, M. (2017). A Systematic Review on Software Cost Estimation in Agile Software Development. *Journal of Engineering Science & Technology Review*, 10(4).
- Bohlander, G., Snell, G., & Sherman, A. (2005). *Human Resources Administration*. São Paulo, Pioneira Thomson Learning.
- Boroujen, I. (2014). A case study research on software cost estimation using experts' estimates, wideband delphi, and planning poker technique. *International Journal of Software Engineering and its applications*, 8(11), 173-182.
- Borrego, G., Morán, A. L., Palacio, R. R., Vizcaíno, A., & García, F. O. (2019). Towards a reduction in architectural knowledge vaporization during agile global software development. *Information and Software Technology*, 112, 68-82.
<https://doi.org/10.1016/j.infsof.2019.04.008>
- Boyatzis, R. E. (1982). *The competent manager: A model for effective performance*. John Wiley & Sons.
- Busse, R., & Weidner, G. (2020). A qualitative investigation on combined effects of distant leadership, organisational agility and digital collaboration on perceived employee engagement. *Leadership & Organization Development Journal*.
<https://doi.org/10.1108/LODJ-05-2019-0224>
- Carvalho, B. V. D., & Mello, C. H. P. (2012). Implementation of scrum agile methodology in software product project in a small technology-based company. *Gestão & Produção*, 19, 557-573.
- Casola, V., De Benedictis, A., Rak, M., & Villano, U. (2020). A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach. *Journal of Systems and Software*, 163, 110537.
<https://doi.org/10.1016/j.jss.2020.110537>
- Cegarra-Navarro, J. G., Soto-Acosta, P., & Wensley, A. K. (2016). Structured knowledge processes and firm performance: The role of organizational agility. *Journal of Business Research*, 69(5), 1544-1549.
<https://doi.org/10.1016/j.jbusres.2015.10.014>
- Cervone, H. F. (2006). *Project risk management*. OCLC Systems & Services: International digital library perspectives.
- Çetin, E., & Onay Durdu, P. (2019). Blended Scrum model for software development organizations. *Journal of Software: Evolution and Process*, 31(2), e2147.
<https://doi.org/10.1002/smr.2147>
- Charmaz, K. (2006), *Constructing Grounded Theory: A Practical Guide through Qualitative Research*, Sage, London.
- Chin, C., & Chia, L. G. (2004). Implementing project work in biology through problem-based learning. *Journal of biological education*, 38(2), 69-75.
<https://doi.org/10.1080/00219266.2004.9655904>
- Colomo-Palacios, R., González-Carrasco, I., López-Cuadrado, J. L., & García-Crespo, Á. (2012). ReSySTER: A hybrid recommender system for Scrum team roles based on fuzzy and rough sets. *International Journal of Applied Mathematics and Computer Science*, 22, 801-816.

- Concas, G., Lunesu, M. I., Marchesi, M., & Zhang, H. (2013). Simulation of software maintenance process, with and without a work-in-process limit. *Journal of Software: Evolution and Process*, 25(12), 1225-1248.
<https://doi.org/10.1002/smr.1599>
- Conforto, E. C., Amaral, D. C., da Silva, S. L., Di Felippo, A., & Kamikawachi, D. S. L. (2016). The agility construct on project management theory. *International Journal of Project Management*, 34(4), 660-674.
<https://doi.org/10.1016/j.ijproman.2016.01.007>
- Cook, D. J., Mulrow, C. D., & Haynes, R. B. (1997). Systematic reviews: synthesis of best evidence for clinical decisions. *Annals of internal medicine*, 126(5), 376-380.
<https://doi.org/10.7326/0003-4819-126-5-199703010-00006>
- Cooper, R. G., Edgett, S. J., & Kleinschmidt, E. J. (2002). Optimizing the stage-gate process: What best-practice companies do—II. *Research-Technology Management*, 45(6), 43-49.
- Cram, W. A. (2012). Aligning organizational values in systems development projects. *Management Research Review*, 35(8), 709-726.
<https://doi.org/10.1108/01409171211247703>
- Crawford, L. (2005). Senior management perceptions of project management competence. *International journal of project management*, 23(1), 7-16.
- Crawford, L., & Pollack, J. (2004). Hard and soft projects: a framework for analysis. *International Journal of Project Management*, 22(8), 645-653.
- Creswell, J. W. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- Damas, M., & De Ketele, J.M. (1985) *Observar para avaliar*. Coimbra: Livraria Almedina.
- da Silva, L. F., Russo, R. D. F. S. M., & de Oliveira, P. S. G. (2018). Quantitativa ou qualitativa? um alinhamento entre pesquisa, pesquisador e achados em pesquisas sociais. *Revista Pretexto*, 30-45.
- Dias, C. M. (2009). Olhar com olhos de ver. *Revista portuguesa de pedagogia*, 175-188.
- Dilley, P. (2000). Conducting successful interviews: Tips for intrepid research. *Theory into practice*, 39(3), 131-137.
- Dingsøy, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development.
- Dingsøy, T., Moe, N. B., Fægri, T. E., & Seim, E. A. (2018). Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23(1), 490-520.
<https://doi.org/10.1007/s10664-017-9524-2>
- Dinsmore, P. C., & Cabanis-Brewin, J. (2009). *Manual de gerenciamento de projetos*. Rio de Janeiro: American Management Association (AMA).
- Doherty, N. F., Ashurst, C., & Peppard, J. (2012). Factors affecting the successful realisation of benefits from systems development projects: findings from three case studies. *Journal of Information technology*, 27(1), 1-16.
- dos Santos Soares, M. (2004). Comparação entre metodologias Ágeis e tradicionais para o desenvolvimento de software. *INFOCOMP Journal of Computer Science*, 3(2), 8-13.
- Drury-Grogan, M. L. (2021). The Changes in Team Cognition and Cognitive Artifact Use During Agile Software Development Project Management. *Project Management Journal*, 52(2), 127-145.
<https://doi.org/10.1177/8756972820960301>
- Eder, S., Conforto, E. C., Amaral, D. C., & Silva, S. L. D. (2015). Diferenciando as abordagens tradicional e ágil de gerenciamento de projetos. *Production*, 25(3), 482-497.

- Elghariani, K., Kama, N., Azmi, N. F. M., & Abu Bakar, N. A. (2018). Implicit Thinking Knowledge Injection Framework for Agile Requirements Engineering. *International journal of advanced computer science and applications*, 9(11), 141-146.
- Erdoğan, O., Pekkaya, M. E., & Gök, H. (2018). More effective sprint retrospective with statistical analysis. *Journal of Software: Evolution and Process*, 30(5), e1933.
<https://doi.org/10.1002/smr.1933>
- Espinosa-Curiel, I. E., Rodríguez-Jacobo, J., Vázquez-Alfaro, E., Fernández-Zepeda, J. A., & Fajardo-Delgado, D. (2018). Analysis of the changes in communication and social interactions during the transformation of a traditional team into an agile team. *Journal of Software: Evolution and Process*, 30(9), e1946.
<https://doi.org/10.1002/smr.1946>
- Ferreira, F. S., & Pocivi, V. C. B. (2017). GESTÃO ÁGIL DE PROJETOS: Método para estimativa em projeto de software para fábrica de software acadêmica. *Revista Ada Lovelace*, 1, 05-17.
- Ferreira, J., Sharp, H., & Robinson, H. (2011). User experience design and agile development: managing cooperation through articulation work. *Software: Practice and Experience*, 41(9), 963-974.
<https://doi.org/10.1002/spe.1012>
- Fontana, R. M., Fontana, I. M., da Rosa Garbuio, P. A., Reinehr, S., & Malucelli, A. (2014). Processes versus people: How should agile software development maturity be defined?. *Journal of Systems and Software*, 97, 140-155.
<https://doi.org/10.1016/j.jss.2014.07.030>
- Fontana, R. M., Meyer Jr, V., Reinehr, S., & Malucelli, A. (2015). Progressive Outcomes: A framework for maturing in agile software development. *Journal of Systems and Software*, 102, 88-108.
<https://doi.org/10.1016/j.jss.2014.12.032>
- Gamba, M. L., & Barbosa, A. C. G. (2010). Engenharia de Software-Aplicação de Métricas de Software com Scrum. *Anais SULCOMP*, 5.
- Ganesh, N., Thangasamy, S. (2012). Lessons learned in transforming from traditional to agile development. *Journal of Computer Science*, 8(3), 389-392.
- Gjøystdal, S., & Karunaratne, T. (2020). Effect of Inadequate Self-Organized Teams in Agile Project Management: A Case Study From the Oil and Gas Industry. *International Journal of Information Technology Project Management (IJITPM)*, 11(3), 95-106.
- Golfarelli, M., Rizzi, S., & Turricchia, E. (2013). Multi-sprint planning and smooth replanning: An optimization model. *Journal of systems and software*, 86(9), 2357-2370.
<https://doi.org/10.1016/j.jss.2013.04.028>
- Grapenthin, S., Poggel, S., Book, M., & Gruhn, V. (2015). Improving task breakdown comprehensiveness in agile projects with an Interaction Room. *Information and Software Technology*, 67, 254–264.
<https://doi.org/10.1016/j.infsof.2015.07.008>
- Gren, L., Goldman, A., & Jacobsson, C. (2020). Agile ways of working: a team maturity perspective. *Journal of Software: Evolution and Process*, 32(6), e2244.
<https://doi.org/10.1002/smr.2244>
- Gren, L., Knauss, A., & Stettina, C. J. (2018). Non-technical individual skills are weakly connected to the maturity of agile practices. *Information and Software Technology*, 99, 11-20.
<https://doi.org/10.1016/j.infsof.2018.02.006>
- Guelman, L. (2012). Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Systems with Applications*, 39(3), 3659-3667.

- Gurd, B. & Ifandoudas, P. (2014). Moving towards agility: the contribution of a modified balanced scorecard system., *Measuring Business Excellence*, Vol. 18 No. 2, pp. 1-13.
<https://doi.org/10.1108/MBE-10-2012-0052>
- Hamel, G., & Prahalad, C. K. (1995). *Competindo pelo futuro*. Rio de Janeiro: Campus.
- Heikkilä, V. T., Paasivaara, M., Lasssenius, C., Damian, D., & Engblom, C. (2017). Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson. *Empirical Software Engineering*, 22(6), 2892-2936.
<https://doi.org/10.1007/s10664-016-9491-z>
- Hemon, A., Lyonnet, B., Rowe, F., & Fitzgerald, B. (2020). From agile to DevOps: Smart skills and collaborations. *Information Systems Frontiers*, 22(4), 927-945.
<https://doi.org/10.1007/s10796-019-09905-1>
- Hemon-Hildgen, A., Rowe, F., & Monnier-Senicourt, L. (2020). Orchestrating automation and sharing in DevOps teams: a revelatory case of job satisfaction factors, risk and work conditions. *European Journal of Information Systems*, 29(5), 474-499.
<https://doi.org/10.1080/0960085X.2020.1782276>
- Highsmith, J. (2009). *Agile project management: creating innovative products*. Pearson education.
- Hoda, R., & Murugesan, L. K. (2016). Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software*, 117, 245-257.
<https://doi.org/10.1016/j.jss.2016.02.049>
- Hoda, R., Noble, J., & Marshall, S. (2012). Developing a grounded theory to explain the practices of self-organizing Agile teams. *Empirical Software Engineering*, 17(6), 609-639.
<https://doi.org/10.1007/s10664-011-9161-0>
- Hollenbeck, G. P., McCall Jr, M. W., & Silzer, R. F. (2006). Leadership competency models. *The Leadership Quarterly*, 17(4), 398-413.
- Hsieh, C. Y., & Chen, C. T. (2015). Patterns for Continuous Integration Builds in Cross-Platform Agile Software Development. *J. Inf. Sci. Eng.*, 31(3), 897-924.
- Ioannou, C., Nurdiani, I., Burattin, A., & Weber, B. (2020). Mining reading patterns from eye-tracking data: method and demonstration. *Software and Systems Modeling*, 19(2), 345-369.
<https://doi.org/10.1007/s10270-019-00759-4>
- Jalali, S., Wohlin, C., & Angelis, L. (2014). Investigating the applicability of agility assessment surveys: A case study. *Journal of Systems and Software*, 98, 172-190.
<https://doi.org/10.1016/j.jss.2014.08.067>
- Jia, J., Yang, X., Zhang, R., & Liu, X. (2019). Understanding software developers' cognition in agile requirements engineering. *Science of Computer Programming*, 178, 1-19.
<https://doi.org/10.1016/j.scico.2019.03.005>
- Jiang, J. J., Klein, G., Wu, S. P., & Liang, T. P. (2009). The relation of requirements uncertainty and stakeholder perception gaps to project management performance. *Journal of Systems and Software*, 82(5), 801-808.
- Jørgensen, M. (2013). The influence of selection bias on effort overruns in software development projects. *Information and Software Technology*, 55(9), 1640-1650.
<https://doi.org/10.1016/j.infsof.2013.03.001>
- Jorgensen, M., & Molokken-Ostfold, K. (2004). Reasons for software effort estimation error: impact of respondent role, information collection approach, and data analysis method. *IEEE Transactions on Software engineering*, 30(12), 993-1007.
<https://doi.org/10.1109/TSE.2004.103>

- Jørgensen, M., Mohagheghi, P., & Grimstad, S. (2017). Direct and indirect connections between type of contract and software project outcome. *International Journal of Project Management*, 35(8), 1573-1586.
<https://doi.org/10.1016/j.ijproman.2017.09.003>
- Jovanović, M., Mesquida, A. L., Radaković, N., & Mas, A. (2016). Agile retrospective games for different team development phases. *Journal of Universal Computer Science*, 22(12), 1489-1508.
- Kakar, A. K. (2017). Investigating the prevalence and performance correlates of vertical versus shared leadership in emergent software development teams. *Information Systems Management*, 34(2), 172-184.
<https://doi.org/10.1080/10580530.2017.1288526>
- Kaushik, A., Tayal, D. K., & Yadav, K. (2020). The Role of Neural Networks and Metaheuristics in Agile Software Development Effort Estimation. *International Journal of Information Technology Project Management (IJITPM)*, 11(2), 50-71.
- Kalenda, M., Hyna, P., & Rossi, B. (2018). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*, 30(10), e1954.
<https://doi.org/10.1002/smr.1954>
- Karlsen, J. T., Hagman, L., & Pedersen, T. (2011). Intra-project transfer of knowledge in information systems development firms. *Journal of Systems and Information Technology*, 13(1), 66-80.
<https://doi.org/10.1108/13287261111118359>
- Kaushik, A., Tayal, D. K., & Yadav, K. (2020). The role of neural networks and metaheuristics in agile software development effort estimation. *International Journal of Information Technology Project Management*, 11(2), 50-71.
- Kerzner, H. (2009). *Project management: A systems approach to planning (Vol. 11). Scheduling and Controlling*.
- Korkala, M., & Maurer, F. (2014). Waste identification as the means for improving communication in globally distributed agile software development. *Journal of Systems and Software*, 95, 122-140.
<https://doi.org/10.1016/j.jss.2014.03.080>
- Kremser, W., & Blagoev, B. (2021). The Dynamics of Prioritizing: How Actors Temporally Pattern Complex Role–Routine Ecologies. *Administrative Science Quarterly*, 66(2), 339-379.
<https://doi.org/10.1177/0001839220948483>
- Laslo, Z. (2010). Project portfolio management: An integrated method for resource planning and scheduling to minimize planning/scheduling-dependent expenses. *International Journal of Project Management*, 28(6), 609-618.
- Lavallée, M., & Robillard, P. N. (2018). Are we working well with others? How the multi team systems impact software quality. *e-Informatica Software Engineering Journal*, 12(1), 117-131.
<https://doi.org/10.5277/e-inf180105>
- Le Boterf, G. (2003). *Desenvolvendo a competência dos profissionais*. trad. Patrícia Chittoni Ramos Reuillard. Porto Alegre: Artmed.
- Lehtinen, T. O., Itkonen, J., & Lassenius, C. (2017). Recurring opinions or productive improvements—what agile teams actually discuss in retrospectives. *Empirical Software Engineering*, 22(5), 2409-2452.
<https://doi.org/10.1007/s10664-016-9464-2>

- Lehtinen, T. O., Virtanen, R., Viljanen, J. O., Mäntylä, M. V., & Lassenius, C. (2014). A tool supporting root cause analysis for synchronous retrospectives in distributed software teams. *Information and Software Technology*, 56(4), 408-437.
<https://doi.org/10.1016/j.infsof.2014.01.004>
- Licorish, S. A., & MacDonell, S. G. (2015). Communication and personality profiles of global software developers. *Information and Software Technology*, 64, 113-131.
<https://doi.org/10.1016/j.infsof.2015.02.004>
- Lindsjörn, Y., Sjöberg, D. I., Dingsøy, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, 122, 274-286.
<https://doi.org/10.1016/j.jss.2016.09.028>
- Liu, J. W., Ho, C. Y., Chang, J. Y., & Tsai, J. C. A. (2019). The role of Sprint planning and feedback in game development projects: Implications for game quality. *Journal of Systems and Software*, 154, 79-91.
<https://doi.org/10.1016/j.jss.2019.04.057>
- López-Martínez, J., Ramírez-Noriega, A., Juárez-Ramírez, R., Licea, G., & Jiménez, S. (2018). User stories complexity estimation using Bayesian networks for inexperienced developers. *Cluster Computing*, 21(1), 715-728.
- Luiz Neto, A. (2020). Framework para propor alocação de agentes de transformação levando em consideração as competências e o tipo de abordagem de gerenciamento de projetos. Dissertação (Programa de Pós-Graduação em Gestão de Projetos) - Universidade Nove de Julho, São Paulo.
<https://doi.org/10.1007/s10586-017-0996-z>
- Luwakatare, L. E., Kilamo, T., Luong, T. T., Sivarajah, U., & Weerakkody, V. (2021). Do agile managed information systems projects fail due to a lack of emotional intelligence?. *Information Systems Frontiers*, 23(2), 415-433.
<https://doi.org/10.1007/s10796-019-09962-6>
- Luwakatare, L. E., Kilamo, T., Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., ... & Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Information and Software Technology*, 114, 217-230.
<https://doi.org/10.1016/j.infsof.2019.06.010>
- Karvonen, T., Sauvola, T., Heikkilä, V., Itkonen, J., ... & Lassenius, C. (2019). DevOps in practice: A multiple case study of five companies. *Information and Software Technology*, 114, 217-230.
- Magdaleno, A. M., de Oliveira Barros, M., Werner, C. M. L., de Araujo, R. M., & Batista, C. F. A. (2015). Collaboration optimization in software process composition. *Journal of Systems and Software*, 103, 452-466.
<https://doi.org/10.1016/j.jss.2014.11.036>
- Magistretti, S., Dell'Era, C., & Doppio, N. (2020). Design sprint for SMEs: an organizational taxonomy based on configuration theory. *Management Decision*, 58(9), 1803-1817.
<https://doi.org/10.1108/MD-10-2019-1501>
- Mahmoud, A. B., Fuxman, L., Mohr, I., Reisel, W. D., & Grigoriou, N. (2020). "We aren't your reincarnation!" workplace motivation across X, Y and Z generations. *International Journal of Manpower*, 42(1), 193-209.
<https://doi.org/10.1108/IJM-09-2019-0448>
- Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086-2095.
- Majchrzak, M., & Stilger, L. (2017). Experience report: Introducing Kanban into automotive software project. *e-Informatica Software Engineering Journal*, 11(1).
<https://doi.org/10.5277/e-Inf170102>

- Malgonde, O., & Chari, K. (2019). An ensemble-based model for predicting agile software development effort. *Empirical Software Engineering*, 24(2), 1017-1055.
<https://doi.org/10.1007/s10664-018-9647-0>
- Malik, M., Sarwar, S., & Orr, S. (2021). Agile practices and performance: Examining the role of psychological empowerment. *International Journal of Project Management*, 39(1), 10-20.
<https://doi.org/10.1016/j.ijproman.2020.09.002>
- Marques, R., Costa, G., da Silva, M. M., Gonçalves, D., & Gonçalves, P. (2020). A gamification solution for improving Scrum adoption. *Empirical Software Engineering*, 25(4), 2583-2629.
<https://doi.org/10.1007/s10664-020-09816-9>
- Martini, A., Pareto, L., & Bosch, J. (2016). A multiple case study on the inter-group interaction speed in large, embedded software companies employing agile. *Journal of Software: Evolution and Process*, 28(1), 4-26.
<https://doi.org/10.1002/smr.1757>
- Mascarenhas, A. O. (2020). *Gestão estratégica de pessoas: evolução, teoria e crítica*. Cengage Learning.
- Masood, Z., Hoda, R., & Blincoe, K. (2020). How agile teams make self-assignment work: a grounded theory study. *Empirical Software Engineering*, 25(6), 4962-5005.
<https://doi.org/10.1002/smr.1757>
- McClelland, D. C. (1973). Testing for competence rather than intelligence. *American Psychologist*, 28(1), 1-40.
- McIver, D., Lengnick-Hall, M. L., & Lengnick-Hall, C. A. (2018). A strategic approach to workforce analytics: Integrating science and agility. *Business Horizons*, 61(3), 397-407.
<https://doi.org/10.1016/j.bushor.2018.01.005>
- Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2020). Requirements specification for developers in agile projects: Evaluation by two industrial case studies. *Information and Software Technology*, 117, 106194.
<https://doi.org/10.1016/j.infsof.2019.106194>
- Michaux, V. (2011). Articular as competências individual, coletiva, organizacional e estratégica: esclarecendo a teoria dos recursos e do capital social. *Competências coletivas: no limiar da estratégia*, 1-21.
- Mills, J., Cooper, D., & Forest, D. (2002). Polarization of interpersonal attraction: The effect of perceived potency. *Basic and applied social psychology*, 24(2), 157-162.
- Mishra, D., & Abdalhamid, S. (2018). Software Quality Issues in SCRUM: A Systematic Mapping. *J. Univers. Comput. Sci.*, 24(12), 1690-1716.
- Misirli, A. T., Verner, J., Markkula, J., & Oivo, M. (2015). Factors Affecting Team Motivation: A Survey of Finnish Software Engineers. *International Journal of Information System Modeling and Design (IJISMD)*, 6(3), 1-26.
<https://doi.org/10.4018/IJISMD.2015070101>
- Miterev, M., Jerbrant, A., & Feldmann, A. (2020). Exploring the alignment between organization designs and value processes over the program lifecycle. *International Journal of Project Management*, 38(2), 112-123.
<https://doi.org/10.1016/j.ijproman.2019.12.003>
- Mitrović, Z. M., Rakićević, A. M., Mihić, M. M., Rakićević, J. Đ., & Jelisić, E. T. (2020). Systems Thinking in Software Projects-an Artificial Neural Network Approach. *IEEE Access*, 8, 213619-213635.
<https://doi.org/10.1109/ACCESS.2020.3040169>

- Moe, N. B., Aurum, A., & Dybå, T. (2012). Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology*, 54(8), 853-865.
<https://doi.org/10.1016/j.infsof.2011.11.006>
- Mohanaragam, K. (2020). Transitioning to Agile—In a Large Organization. *IT Professional*, 22(2), 67-72.
<https://doi.org/10.1109/MITP.2019.2902345>
- Müller, R., & Turner, R. (2010). Leadership competency profiles of successful project managers. *International Journal of project management*, 28(5), 437-448.
- Muntés-Mulero, V., Ripolles, O., Gupta, S., Dominiak, J., Willeke, E., Matthews, P., & Somosköi, B. (2019). Agile risk management for multi-cloud software development. *IET Software*, 13(3), 172-181.
<https://doi.org/10.1049/iet-sen.2018.5295>
- Naim, W., Hussain, S., Hassan, B., Ali, S. A., & Bhatti, P. (2019). How to address Impediments in Scrum?. *IJCSNS*, 19(6), 130.
- Neto, P. S., Medeiros, D. B., Ibiapina, I., & da Costa Castro, O. C. (2019). Case study of the introduction of game design techniques in software development. *IET Software*, 13(2), 129-143.
- Nicholls, G. M., Lewis, N. A., & Eschenbach, T. (2015). Determining when simplified agile project management is right for small teams. *Engineering Management Journal*, 27(1), 3-10.
<https://doi.org/10.1080/10429247.2015.11432031>
- Paasivaara, M., & Lassenius, C. (2014). Agile coaching for global software development. *Journal of software: Evolution and Process*, 26(4), 404-418.
<https://doi.org/10.1002/smr.1577>
- Páscoa, C., Telha, A., & Santos, T. (2019). Online organization: a flexible, agile and adaptable organizational instrument. *International Journal of Human Capital and Information Technology Professionals*, 10(1), 55-75.
<https://doi.org/10.4018/IJHCITP.2019010104>
- Penha, R., da Silva, L. F., & Russo, R. D. F. S. M. (2020). Escalando as práticas ágeis. *Revista de Gestão e Projetos*, 11(2), 1-11.
- Penha, R., Kniess, C. T., Bergman, D. R., & Biancolino, C. A. (2014). Emprego de Técnicas de Gerenciamento de Riscos Técnicos em uma Empresa de Desenvolvimento de Softwares. *Revista Gestão e Tecnologia*, 14, 151-173.
- Pereira Junior, E. F. Z., Schroeder, E. A., & Dolci, D. B. (2019). Limitações digitais, causas e consequências na efetividade do uso do site trello no planejamento estratégico de uma secretaria de educação a distância de uma universidade federal.
- Perkusich, M., Gorgônio, K. C., Almeida, H., & Perkusich, A. (2017). Assisting the continuous improvement of Scrum projects using metrics and Bayesian networks. *Journal of Software: Evolution and Process*, 29(6), 1-17.
<https://doi.org/10.1002/smr.1835>
- PMI - Project Management Institute. (2017). *Guide to the project Management body of knowledge - Sixth Edition* (Project Management Institute, ed.). Pennsylvania USA.
- Pollock, A., & Berge, E. (2018). How to do a systematic review. *International Journal of Stroke*, 13(2), 138-156.
<https://doi.org/10.1177/1747493017743796>
- Project management institute [PMI]. (2017). *Guide to the project Management body of knowledge - Sixth Edition*. Project Management Institute, Pennsylvania USA.

- Pradeepini, G. (2019). Experimenting cloud infrastructure for tomorrows big data analytics. *International Journal of Innovative Technology and Exploring Engineering*, 8(5), 885-890.
- Prahalad, C. K., & Hamel, G. (1990). The Core Competence of the Corporation. *Harvard Business Review*, 1-15.
- Rahman, N., & Rutz, D. (2015). Building data warehouses using automation. *International Journal of Intelligent Information Technologies*, 11(2), 1-22.
<https://doi.org/10.4018/IJIT.201504010>
- Raj, P., & Sinha, P. (2015). Project Management In Era Of Agile And Devops Methodologies. In *International Conference on Applied Sciences*, 9(1), 1024-1033.
- Ramírez-Mora, S. L., Oktaba, H., & Patlan Perez, J. (2020). Group maturity, team efficiency, and team effectiveness in software development: A case study in a CMMI-DEV Level 5 organization. *Journal of Software: Evolution and Process*, 32(4), e2232.
<https://doi.org/10.1002/smr.2232>
- Razavi, A. M., Nasir, M. H. N. M., Yasin, N. B. M., & Ahmad, R. (2019). Cultural Issues in Offshore Teams: A Categorization based on Existing Studies. *KSII Transactions on Internet and Information Systems (TIIS)*, 13(3), 1363-1384.
<https://doi.org/10.3837/tiis.2019.03.014>
- Ribeiro, P. C. (2020). Problemas na implantação do Jira: estudo de caso em uma equipe de desenvolvimento de software. *Tecnologia em Gestão da Tecnologia da Informação- Unisul Virtual*.
- Rolón, M., & Martínez, E. (2012). Agent-based modeling and simulation of an autonomic manufacturing execution system. *Computers in industry*, 63(1), 53-78.
<https://doi.org/10.1016/j.compind.2011.10.005>
- Rosenkranz, C., Charaf, M. C., & Holten, R. (2013). Language quality in requirements development: Tracing communication in the process of information systems development. *Journal of Information Technology*, 28(3), 198-223.
<https://doi.org/10.1057/jit.2012.33>
- Ruas, R., Antonello, C. S., & Boff, L. H. (2005). *Aprendizagem organizacional e competências*. Porto Alegre: Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft press.
- Salaou, A. D., Damian, D., Lassenius, C., Voda, D., & Gançarski, P. (2021). Archetypes of delay: An analysis of online developer conversations on delayed work items in IBM Jazz. *Information and Software Technology*, 129, 106435.
<https://doi.org/10.1016/j.infsof.2020.106435>
- Sampaio, R. F., & Mancini, M. C. (2007). Estudos de revisão sistemática: um guia para síntese criteriosa da evidência científica. *Brazilian Journal of Physical Therapy*, 11, 83-89.
- Sarpiri, M., & Gandomani, T. (2017). How Agile managers affect the process of software development. *International Journal of Computer Science and Network Security*, 17(5), 283-286.
- Schloegel, U., Stegmann, S., Maedche, A., & Van Dick, R. (2016). Reducing age stereotypes in software development: The effects of awareness-and cooperation-based diversity interventions. *Journal of Systems and Software*, 121, 1-15.
<https://doi.org/10.1016/j.jss.2016.07.041>
- Senapathi, M., & Drury-Grogan, M. L. (2021). Systems thinking approach to implementing kanban: A case study. *Journal of Software: Evolution and Process*, 33(4), 1-16.
<https://doi.org/10.1002/smr.2322>
- Serrador, P., & Pinto, J. K. (2015). Does Agile work? A quantitative analysis of agile project success. *International journal of project management*, 33(5), 1040-1051.
<https://doi.org/10.1016/j.ijproman.2015.01.006>

- Seth, N., & Khare, R. (2015, December). ACI (automated Continuous Integration) using Jenkins: Key for successful embedded Software development. In 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS) (pp. 1-6). IEEE.
<https://doi.org/10.1109/RAECS.2015.7453279>
- Shameem, M., Kumar, R. R., Kumar, C., Chandra, B., & Khan, A. A. (2018). Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process. *Journal of Software: Evolution and Process*, 30(11), e1979.
<https://doi.org/10.1002/smr.1979>
- Shankar, P. R., & Narayana, M. G. (2013). Developing agile teams for project execution: A cybernetic approach. *Global Journal of Flexible Systems Management*, 14(1), 33-43.
- Shekhar, S., & Kumar, U. (2016). Review of various software cost estimation techniques. *International Journal of Computer Applications*, 141(11), 31-34.
- Spradley, J. P. (2016). *Participant observation*. Waveland Press.
- Strode, D. E., Huff, S. L., Hope, B., & Link, S. (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software*, 85(6), 1222-1238.
<https://doi.org/10.1016/j.jss.2012.02.017>
- Tan, T. A. G., & Vicente, A. J. (2019). An innovative experiential and collaborative learning approach to an undergraduate marketing management course: A case of the Philippines. *The International Journal of Management Education*, 17(3), 1-18.
<https://doi.org/10.1016/j.ijme.2019.100309>
- Tolfo, C., Wazlawick, R. S., Ferreira, M. G. G., & Forcellini, F. A. (2018). Agile practices and the promotion of entrepreneurial skills in software development. *Journal of Software: Evolution and Process*, 30(9), 1-18.
<https://doi.org/10.1002/smr.1945>
- Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British journal of management*, 14(3), 207-222.
<https://doi.org/10.1111/1467-8551.00375>
- Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job satisfaction in agile development teams: Agile development as work redesign. *Journal of the Association for Information Systems*, 17(4), 1.
<https://doi.org/10.17705/1jais.00426>
- Trkman, M., Mendling, J., & Krisper, M. (2016). Using business process models to better understand the dependencies among user stories. *Information and software technology*, 71, 58-76.
<https://doi.org/10.1016/j.infsof.2015.10.006>
- Trkman, M., Mendling, J., Trkman, P., & Krisper, M. (2019). Impact of the conceptual model's representation format on identifying and understanding user stories. *Information and software technology*, 116, 106169.
<https://doi.org/10.1016/j.infsof.2019.08.001>
- Turner, C. R., Fuggetta, A., Lavazza, L., & Wolf, A. L. (1999). A conceptual basis for feature engineering. *Journal of Systems and Software*, 49(1), 3-15.
- Varela-Vaca, A. J., & Gasca, R. M. (2013). Towards the automatic and optimal selection of risk treatments for business processes using a constraint programming approach. *Information and Software Technology*, 55(11), 1948-1973.
<https://doi.org/10.1016/j.infsof.2013.05.007>
- Vlietland, J., Van Solingen, R., & Van Vliet, H. (2016). Aligning codependent Scrum teams to enable fast business value delivery: A governance framework and set of intervention actions. *Journal of Systems and Software*, 113, 418-429.

<https://doi.org/10.1016/j.jss.2015.11.010>

- Vieira, G. L. S., Antonioli, P., & Argoud, A. R. T. T. (2017). Estimativas de custo e de tempo durante a seleção do fornecedor de um sistema de informação para área jurídica: um estudo de caso comparando abordagens tradicionais e ágeis de projeto no brasil. *Iberoamerican Journal of Project Management*, 8(1), 20-48.
- Wohlrab, R., Pelliccione, P., Knauss, E., & Larsson, M. (2019). Boundary objects and their use in agile systems engineering. *Journal of Software: Evolution and Process*, 31(5), 1-24.
<https://doi.org/10.1002/smr.2166>
- Yilmaz, R. (2017). Exploring the role of e-learning readiness on student satisfaction and motivation in flipped classroom. *Computers in Human Behavior*, 70, 251-260.
<https://doi.org/10.1016/j.chb.2016.12.085>
- Yin, X., Huang, J., He, W., Guo, W., Yu, H., & Cui, L. (2021). Group task allocation approach for heterogeneous software crowdsourcing tasks. *Peer-to-Peer Networking and Applications*, 14(3), 1736-1747.
<https://doi.org/10.1007/s12083-020-01000-6>
- Zaitsev, A., Gal, U., & Tan, B. (2020). Coordination artifacts in agile software development. *Information and Organization*, 30(2), 1-23.
<https://doi.org/10.1016/j.infoandorg.2020.100288>
- Žužek, T., Kušar, J., Rihar, L., & Berlec, T. (2020). Agile-Concurrent hybrid: A framework for concurrent product development using Scrum. *Concurrent Engineering*, 28(4), 255-264.
<https://doi.org/10.1177/1063293X20958541>

APÊNDICE A – PROTOCOLO DE PESQUISA E COLETA DE DADOS

I) Instruções para o entrevistador:

O foco das entrevistas em profundidade será o de obter dos entrevistados informações relevantes, vivenciadas no processo de gerenciamento de atividades em projetos ágeis, relatos de suas experiências, opiniões e pontos de vista (Turner, 2010). Nesse contexto o objetivo da entrevista será descrever quais são as competências (*hard skills* e *soft skill*) necessárias que o time de desenvolvimento deverá possuir no processo de gerenciamento de atividades, levando em consideração a complexidade para a realização das atividades e criticidade de cada atividade em relação a uma iteração.

Pesquisador: José Romualdo da Costa Filho

Professor Orientador: Prof. Dr. Renato Penha

Professor Coorientador: Prof. Dr. Luciano Ferreira da Silva

II) Condições da entrevista

- Quem será entrevistado?

Nome:

Data da entrevista:

Local:

Duração da entrevista:

Cargo:

Experiência em Projetos Ágeis:

- Quando? Entre os meses de xxxx e xxxxxx de 2021.
- Onde?
- Quanto tempo? Entre 40 – 60 minutos.
- Como será conduzida a entrevista? Gravada.

III) Categorias iniciais.

Objetivo Geral: Desenvolvimento de um modelo baseado em competências que auxilie a autodesignação de atividades de um time ágil em uma iteração.			
Objetivos Específicos	Perguntas	Competência	Autor/Fundamentação
Descrever o processo de gerenciamento de atividades de um time ágil durante as iterações	<ul style="list-style-type: none"> Como é realizado o planejamento das atividades para o início de uma iteração? 	Planejamento de Atividades	Trkman <i>et al.</i> (2019); Jia <i>et al.</i> (2019)
	<ul style="list-style-type: none"> Qual é a sua percepção do envolvimento time no planejamento e decomposição dos requerimentos das atividades para o início de uma iteração? 	Compreensão de Atividades; Decomposição de atividades	Trkman <i>et al.</i> (2019); Jia <i>et al.</i> (2019); Tan & Vicente (2019)
	<ul style="list-style-type: none"> Na sua opinião, em se tratando de uma iteração, qual a sua percepção sobre a capacidade do time em interpretar o que o cliente está solicitando em relação a entrega de atividades que se tornaram funcionalidades do software? 	Análise dos Requerimentos; Capacidade de Abstração	Gren, Knauss & Stettina (2018); Medeiros <i>et al.</i> (2020)
	<ul style="list-style-type: none"> Como você pode apresentar a relação do conhecimento em linguagens de programação, assim como padrões de projeto e padrões arquiteturais voltados ao desenvolvimento de soluções pelo time de desenvolvimento e o resultado das iterações? 	Conhecimento em Tecnologia;	Pascoa, Telha & Santos (2019); & Vicente (2019)
	<ul style="list-style-type: none"> Como é o comportamento do time aceitar eventuais críticas construtivas ao trabalho desenvolvido e, dessa forma, realizar a troca de conhecimentos e experiências, tanto técnicas quanto relacionadas ao contexto de negócio? 	Desenvolvimento de Atividades em Equipe	Paasivaara & Lassenius (2014); Yilmaz <i>et al.</i> (2017)
Identificar as competências necessárias às pessoas de um time ágil para a autodesignação das atividades em uma iteração	<ul style="list-style-type: none"> Quais as competências que o time deve possuir na auto-organização das atividades durante uma iteração? 	Senso de Auto-organização	Gjoystdal & Karunaratne (2020)
	<ul style="list-style-type: none"> Quais as competências técnicas que o time deve possuir para determinar a complexidade para a realização de uma atividade? 	Monitoramento das Atividades; Conhecimento em Tecnologia	Fontana <i>et al.</i> (2015); Licorish & MacDonell (2015); Hsieh & Chen (2015)
	<ul style="list-style-type: none"> Quais as competências interpessoais que o time deve possuir para gerenciar a criticidade para a realização de uma atividade em relação ao término de uma iteração? 	Influência; Comunicação Efetiva	Busse & Weidner (2020); Adolph, Kruchten & Hall (2012);
	<ul style="list-style-type: none"> Quais as competências técnicas e interpessoais necessárias para o planejamento das atividades para o início de uma iteração? 	Mensuração das atividades	Ramirez-Mora, Oktaba & Perez (2020)
Verificar a adequação das competências das pessoas para a	<ul style="list-style-type: none"> Qual a relação das competências interpessoais no time de desenvolvimento no gerenciamento de atividades durante uma iteração? 	Desenvolvimento de Atividades em Equipe	Colomo-Palacios <i>et al.</i> (2012)

autodesignação das atividades de um time ágil em uma iteração	<ul style="list-style-type: none"> Quais as competências interpessoais que o time de desenvolvimento emprega, ou deveria empregar, em relação as constantes mudanças que ocorrem durante uma iteração? 	Adaptabilidade a Mudanças	Colomo-Palacios <i>et al.</i> (2012)
	<ul style="list-style-type: none"> Como o time negocia as modificações de requerimentos durante uma iteração? 	Negociação	Busse & Weidner (2020)

APÊNDICE B – ROTEIRO DE ENTREVISTAS

1. Empresa:
2. Entrevistado (s):
3. Entrevistador:
4. Seções da entrevista:
 - () *Background* do entrevistado
 - () Itens que quero verificar
 - () Validação dos itens percebidos
 - () Comentários finais

5. Introdução da entrevista

Você foi selecionado(a) para essa entrevista porque estamos fazendo um estudo que visa identificar as competências necessárias às pessoas de um time ágil para a autodesignação das atividades em uma iteração.

Vale ressaltarmos que a sua participação é voluntária e muito importante para nossa pesquisa. Havendo interesse, os resultados serão posteriormente compartilhados com o(a) senhor(a).

Mediante sua autorização e para auxiliar-nos na análise do conteúdo, desejamos que a entrevista seja gravada. Destacamos que a qualquer momento o(a) senhor(a) poderá solicitar a interrupção da gravação ou da entrevista. Somente os pesquisadores terão acesso à gravação, assim como os nomes e empresas citadas serão mantidos em sigilo e não serão repassados ou publicados em nenhum momento.

A transcrição da entrevista será enviada a(o) senhor(a) para que seja avaliada e validada.

6. Itens a serem tratados na entrevista:
 - a) Descrever o processo de gerenciamento de atividades de um time ágil durante as iterações;
 - b) Identificar as competências necessárias às pessoas de um time ágil para a autodesignação das atividades em uma iteração;
 - c) Verificar a adequação das competências das pessoas para a autodesignação das atividades de um time ágil em uma iteração;

APÊNDICE C - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Convidamos o (a) Sr (a) para participar da Pesquisa sob o título “**Proposta de Modelo Baseado em Competências para Gerenciamento de Atividades em Projetos Ágeis**”, sob a responsabilidade do pesquisador José Romualdo da Costa Filho, o qual pretende elaborar a dissertação com base na análise da literatura relacionada ao tema e sua entrevista, com o intuito de identificar as competências necessárias às pessoas de um time ágil para a autodesignação das atividades em uma iteração.

A dissertação supracitada é requisito para conclusão do curso de Mestrado em Administração, Programa de Pós-graduação em Gestão de Projetos, PPGP da Universidade Nove de Julho – UNINOVE. A sua participação é voluntária e acontecerá por entrevista, presencial ou virtual, com a utilização de perguntas abertas que terão como propósito registrar fatos importantes vivenciados por você no processo de gerenciamento de atividades em projetos ágeis, relatos de suas experiências, opiniões e seus pontos de vista. A entrevista tem previsão de duração entre 30 minutos e 1 hora.

Os riscos decorrentes de sua participação na pesquisa são inexistentes ou de baixíssima probabilidade, uma vez que o seu envolvimento na pesquisa se dará por meio de respostas verbais às perguntas. Além disso, para garantir que não ocorra nenhum constrangimento para com o entrevistado ou sua empresa, ambos serão mantidos em sigilo. É importante destacar que se o (a) senhor (a) participar, contribuirá para um melhor entendimento sobre as decisões a respeito da priorização de projetos.

Se depois de consentir sobre sua participação o senhor (a) desistir de continuar participando, tem o direito e a liberdade de retirar seu consentimento em qualquer fase da pesquisa, seja antes ou depois da coleta dos dados, independente do motivo e sem nenhum prejuízo à sua pessoa. O (a) Sr (a) não terá nenhuma despesa e também não receberá nenhuma remuneração. Os resultados da pesquisa serão analisados e publicados, mas sua identidade não será divulgada, sendo guardada em sigilo. Para qualquer outra informação, o (a) senhor (a) poderá entrar em contato com o pesquisador no telefone (11) 96468-7373.

Consentimento Pós-Informação

Eu, _____, fui informado sobre o que a pesquisadora quer fazer e porquê precisa da minha colaboração, e entendi a explicação. Por isso, eu concordo em participar do projeto, sabendo que não vou ganhar

nada e que posso sair quando quiser. Este documento é emitido em duas vias que serão ambas assinadas por mim e pelo pesquisador, ficando uma via com cada um de nós.

Data: ___/ ___/ ____

Assinatura do participante

Assinatura do Pesquisador Responsável

APÊNDICE D – FORMULÁRIO DE PESQUISA

Avaliação de Criticidade e Complexidade

*Obrigatório

Informe o ID da atividade *

Sua resposta

Após concluída, qual a entrega de valor dessa atividade? *

1 2 3 4 5 6 7 8 9 10

Quanto a execução dessa atividade contribui com a estratégia da empresa? *

1 2 3 4 5 6 7 8 9 10

Quanto a conclusão dessa atividade irá causar impacto no negócio? *

1 2 3 4 5 6 7 8 9 10

Quão rápida deverá ser a entrega dessa atividade? *

1 2 3 4 5 6 7 8 9 10

Quão clara são as regras de negócio para a execução dessa atividade? *

1 2 3 4 5 6 7 8 9 10

Qual a experiência do time em relação à esse tipo de atividade? *

1 2 3 4 5 6 7 8 9 10

Quanto a execução dessa atividade depende de outras atividades? *

1 2 3 4 5 6 7 8 9 10

Caso seja necessário realizar mudanças, quanto o time está disposto a aceita-las? *

1 2 3 4 5 6 7 8 9 10

A tecnologia que o time utiliza é o suficiente para desenvolver a atividade? *

1 2 3 4 5 6 7 8 9 10

Quanto as competências técnicas do time estão alinhadas com a estimativa de execução da atividade? *

1 2 3 4 5 6 7 8 9 10

Enviar

[Limpar formulário](#)

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado fora de seu domínio. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)

Formulários