

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO E**  
**SISTEMAS**

**ALINE BAMPI**

**MODELO MULTICRITÉRIO PARA PRIORIZAÇÃO DE COMPLEXIDADE DE**  
**REQUISITOS DE *SOFTWARE***

**PATO BRANCO**

**2022**

**ALINE BAMPI**

**MODELO MULTICRITÉRIO PARA PRIORIZAÇÃO DE COMPLEXIDADE DE  
REQUISITOS DE *SOFTWARE***

**Multi-criteria model for complexity prioritization of software requirements**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas como requisito para obtenção do título de Mestre em Engenharia de Produção e Sistemas da Universidade Tecnológica Federal do Paraná (UTFPR).  
– Área de Concentração: Modelos e Métodos de Suporte à Tomada de Decisão.  
Orientador: Prof. Dr. Dalmarino Setti

**PATO BRANCO**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



---

**ALINE BAMPI**

**MODELO MULTICRITÉRIO PARA PRIORIZAÇÃO DE COMPLEXIDADE DE  
REQUISITOS DE SOFTWARE**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Engenharia De Produção E Sistemas da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Gestão Dos Sistemas Produtivos.

Data de aprovação: 12 de Novembro de 2021

Prof. Dalmarino Setti, Doutorado - Universidade Tecnológica Federal do Paraná

Prof. Jose Donizetti De Lima, Doutorado - Universidade Tecnológica Federal do Paraná

Prof. Marcelo Goncalves Trentin, Doutorado - Universidade Tecnológica Federal do Paraná

Prof.<sup>a</sup> Sheila Reinehr, Doutorado - Pontifícia Universidade Católica do Paraná (Pucpr)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 18/02/2022.

“Permanência, perseverança e persistência, apesar de todos os obstáculos, desânimos e impossibilidades: é isso que, em todas as coisas, distingue a alma forte da fraca.” – Thomas Carlyle

## AGRADECIMENTOS

A Deus por me permitir sonhar e me possibilitar um mundo de oportunidades e maravilhas, por me amparar nos momentos difíceis, e me dar forças e sabedoria para superar as dificuldades.

À Universidade Tecnológica Federal do Paraná (UTFPR) pelo papel marcante e imprescindível no meu desenvolvimento não só acadêmico como de transformação em minha vida nestes 15 anos.

Ao Programa de Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS) da UTFPR - Pato Branco pela oportunidade de ingressar no mestrado.

Aos meus pais Deonildo e Maria Daluz que não puderam estudar e foram meus maiores incentivadores ao estudo, pelo apoio e suporte incondicional durante todos os momentos da minha vida, não sendo diferente durante o mestrado e também a pandemia que vivenciamos.

Ao meu orientador Prof. Dr. Dalmarino Setti por ter acreditado em minha capacidade mesmo quando nem eu mesma acreditava, pela dedicação, paciência, respeito e praticidade durante o processo da elaboração deste trabalho. Dificilmente conseguirei expressar o quanto isso foi fundamental.

Aos professores da Banca que, com seus questionamentos, durante o período de avaliação contribuíram para a evolução e melhoria deste trabalho.

Aos demais professores do mestrado que foram primordiais nesta caminhada, embora, muitas vezes eu tenha falhado em demonstrar tal reconhecimento nesse conturbado início de vivência científica.

A empresa que trabalho, em especial a Diretora de Operações e ao Diretor de Tecnologia, que permitiram minha aplicação do modelo desenvolvido neste trabalho, assim como, à equipe que realizou a aplicação no projeto.

Aos meus amigos que ouviram meus lamentos por tantas vezes e incentivaram a continuar.

Aos colegas de mestrado pela troca de conhecimento e experiências.

A todas as pessoas mesmo não citadas diretamente aqui, que contribuíram direta ou indiretamente para a realização deste trabalho.

## RESUMO

O uso crescente da tecnologia da informação no cenário mundial, com elevado investimento na área e com prioridade para *softwares*, faz com que para desenvolvê-los, tenham ainda maior responsabilidade de entregar um produto de qualidade, no prazo e orçamento planejados. Neste contexto, este trabalho apresentou como objetivo desenvolver um modelo multicritério de priorização de requisitos de *software* baseado na sua complexidade sob o ponto de vista da equipe técnica. O modelo proposto foi desenvolvido com base em uma revisão sistemática de literatura sobre priorização de requisitos e estimativas de *software* para obter os fatores de impacto. Tais fatores foram agrupados e posteriormente passaram por uma avaliação de importância pelos especialistas, para o contexto de desenvolvimento de *software* de forma geral. Fez-se o uso do método multicritério AHP para gerar o peso de cada fator de impacto avaliado. O método *Fuzzy* TOPSIS foi utilizado para o processamento das avaliações qualitativas dos fatores de impacto quanto a sua complexidade em relação aos requisitos. Foi realizada uma aplicação prática do modelo, no gerenciamento do projeto de um *software web*, em uma empresa na região sudoeste do Paraná. Na aplicação do modelo, foi realizada a avaliação de complexidade de cada fator de impacto para cada um dos dez requisitos do projeto, por três integrantes com diferentes perfis de especialidade, e se utilizou o método multicritério *Fuzzy* TOPSIS para gerar a ordenação dos requisitos pela complexidade avaliada. O requisito 5 (RQ5) (0,631) ficou em primeira posição da ordenação, seguido do RQ8 (0,622) na segunda e do RQ9 (0,609) na terceira, na quarta ficou o RQ7 (0,602), na quinta posição ficou o RQ1 (0,581) e posteriormente o RQ2(0,499), RQ10 (0,478), RQ6 (0,104) e RQ3 (0,101) na sexta, sétima, oitava e nona posições respectivamente, assim o RQ4 (0,051) ficou na última posição quanto a sua complexidade em relação aos fatores de impacto. Os fatores de impacto que se destacaram em relação a complexidade dos dez requisitos de *software* avaliados foram: Restrição de tempo de execução (FI1), Confiabilidade de *software* exigida (FI5), Complexidade do produto (FI7), Documentação compatível com as necessidades do ciclo de vida (FI9), Experiência de aplicações (FI13), Linguagem e experiência com ferramentas (FI15), Uso de ferramentas de *software* (FI17) e Cronograma de Desenvolvimento Requerido (FI19). Constatou-se por meio dos construtos do Modelo de Aceitação de Tecnologia (TAM) de Davis (1989), que os envolvidos na aplicação do modelo elaborado, possuem uma boa aceitação por perceberem a Utilidade Percebida e Facilidade de Uso, o que contribui efetivamente pela Atitude e Intenção Comportamental para o Uso Real, embora tenha recebido sugestões de ajustes quanto a possibilidade de escolha e quantidade de fatores.

**Palavras-chave:** fatores de impacto, priorização de requisitos, AHP, FUZZY TOPSIS.

## ABSTRACT

*The increasing use of information technology in the world scenario, with high investment in the area and with priority for software, means that to develop them, they have an even greater responsibility to deliver a quality product on time and budget. In this context, this work presented an objective to develop a multicriteria model to prioritize software requirements based on its complexity from the technical team's point of view. The proposed model was developed based on a systematic literature review on requirements prioritization and software estimation to obtain the impact factors. These factors were grouped and then evaluated for importance by the experts for the context of software development in general. The AHP multicriteria method was used to generate the weight of each impact factor evaluated. The Fuzzy TOPSIS method was used to process the qualitative evaluations of the impact factors regarding their complexity with the requirements. A practical application was performed in the project management of web software in a company in the southwestern region of Paraná. The complexity of each impact factor was evaluated for each of the ten requirements of the project by three members with different specialty profiles. The Fuzzy TOPSIS multicriteria method was used to generate the ordering of the requirements by the complexity evaluated. The requirement 5 RQ5) (0.631) ranked first, followed by RQ8 (0.622) in second and RQ9 (0.609) in third, RQ7 (0.602) in fourth, RQ1 (0.581) in fifth. Then RQ2 (0.499), RQ10 (0.478), RQ6 (0.104), and RQ3 (0.101) in the sixth, seventh, eighth, and ninth positions, respectively, thus RQ4 (0.051) was in the last position regarding its complexity with the impact factors. The impact factors that highlighted the complexity of the ten software requirements evaluated were: runtime restriction (FI1), required software reliability (FI5), product complexity (FI7), documentation compatible with life cycle needs (FI9), application experience (FI13), language and tool experience (FI15), use of software tools (FI17) and required development schedule (FI19). It was found through the constructs of the Technology Acceptance Model (TAM) of Davis (1989) that those involved in the application of the model developed to have a good acceptance for perceiving the Perceived Usefulness and Ease of Use, which effectively contributes by the Attitude and Behavioral Intention for the Actual Use. However, it has received suggestions for adjustments regarding the possibility of choice and several factors.*

**Keywords:** *impact factors, prioritizing requirements, AHP, FUZZY TOPSIS*

## LISTA DE FIGURAS

Figura 1 - Protocolo de Revisão .....	20
Figura 2- Processo de busca - resultados .....	22
Figura 3 - ER – Processo de elicitação e análise .....	29
Figura 4 - O processo da contagem de pontos de função .....	32
Figura 5 - Representação do Modelo Proposto .....	45
Figura 6 – Fases da construção e aplicação do modelo .....	46
Figura 7 - Etapas da construção do modelo .....	48
Figura 8 - Amostra da ferramenta de coleta de avaliação de importância.....	49
Figura 9 - Protótipo de ferramenta computacional sugerida para implementação .....	53
Figura 10 - Etapas da aplicação do modelo .....	54
Figura 11 - Processo de avaliação de complexidade dos requisitos que serão desenvolvidos no projeto .....	55
Figura 12 - Ferramenta de avaliação de complexidade dos requisitos de software .....	55
Figura 13 - Modelo de Aceitação Tecnológica - TAM .....	59
Figura 14 - Requisitos ordenados com os respectivos valores dos principais fatores ordenados .....	72
Figura 15 - Média dos valores de $CCi$ por dimensão .....	73
Figura 16 - Média dos valores de $CCi$ dos fatores de impacto mais complexos .....	74
Figura 17 - Requisitos <i>versus</i> fatores de impacto e suas pontuações .....	74
Figura 18 - Funções que devem participar da avaliação na percepção dos respondentes. ....	75



## LISTA DE QUADROS

Quadro 1 – <i>Strings</i> de busca .....	21
Quadro 2 - Periódicos do portfólio por número de artigos e indicadores .....	23
Quadro 3 – Principais Técnicas de PR do Portfólio .....	24
Quadro 4 - Fatores de Ajuste da APF .....	33
Quadro 5 - Direcionadores de Custo – COCOMO II .....	35
Quadro 6 - Fatores de Ajuste – Análise de Pontos por Função .....	37
Quadro 7 - Atributos e fatores de impacto definidos para o modelo .....	38
Quadro 8 - Escala Fundamental método AHP .....	40
Quadro 9 - Perguntas da avaliação da aplicação do modelo .....	58
Quadro 10 - Escala de Likert Utilizada no TAM .....	60
Quadro 11 - Constructos e variáveis - TAM .....	60
Quadro 12 - Lista dos artigos do portfólio bibliográfico .....	92
Quadro 13 - Avaliação de Complexidade dos Requisitos do Integrante I .....	96
Quadro 14 - Avaliação de Complexidade dos Requisitos do Integrante II .....	96
Quadro 15 - Avaliação de Complexidade dos Requisitos do Integrante III .....	96
Quadro 16 - Ranking de Requisitos de Software e Fatores de Impacto .....	98
Quadro 17 - Instrumento para coleta da percepção da aplicação do modelo elaborado. ....	108
Quadro 18 - Instrumento para avaliação da aceitação do modelo elaborado - TAM. ....	109
Quadro 19 - Avaliações de Percepção - Respondidas .....	112

## LISTA DE TABELAS

Tabela 1 - Índice Randômico (RI) para valores de Matrizes (n). .....	50
Tabela 2 - Variáveis linguísticas utilizadas na avaliação da complexidade dos requisitos do software em relação aos fatores de impacto .....	51
Tabela 3 - Pesos das Dimensões (método AHP) conforme opinião dos especialistas .....	62
Tabela 4- Pesos dos Fatores de Impacto da D1 (método AHP) conforme opinião dos especialistas .....	63
Tabela 5 - Pesos dos Fatores de Impacto da Dimensão 2 (método AHP) conforme opinião dos especialistas .....	63
Tabela 6 - Pesos dos Fatores de Impacto da D3 (método AHP) conforme opinião dos especialistas .....	64
Tabela 7 - Pesos dos Fatores de Impacto da Dimensão 4 (método AHP) conforme opinião dos especialistas .....	64
Tabela 8 - Peso global dos fatores de impacto obtidos pelo método AHP especialistas .....	65
Tabela 9 - Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C1-C5) em números <i>Fuzzy</i> Trapezoidais .....	67
Tabela 10 - Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C6-C10) em números <i>Fuzzy</i> Trapezoidais .....	67
Tabela 11 - Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C11-C15) em números <i>Fuzzy</i> Trapezoidais .....	68
Tabela 12 - Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C16-C20) em números <i>Fuzzy</i> Trapezoidais .....	68
Tabela 13 - Ordenação dos Requisitos para Desenvolvimento .....	69
Tabela 14 - Requisitos de <i>Software</i> ordenados e com os principais fatores de impactos também ordenados .....	70
Tabela 15 - Resultado da Avaliação de Complexidade dos Requisitos de Software com a Respectiva Posição .....	97

## LISTA DE ABREVIATURAS E SIGLAS

AHP	<i>Analytic Hierarchy Process</i>
DS	<i>Desenvolvimento de Software</i>
ER	<i>Engenharia de Requisitos</i>
IDC	<i>International Data Corporation</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
LSD	<i>Lean Software Development</i>
MCDA	<i>Multiple Criteria Decision Analysis</i>
PMBOK	<i>Project Management Body of Knowledge</i>
PR	<i>Priorização de Requisitos</i>
PROKNOW-C	<i>Knowledge Development Process – Constructivist</i>
RSL	<i>Revisão Sistemática da Literatura</i>
TAM	<i>Modelo de Aceitação de Tecnologia</i>
TI	<i>Tecnologia da Informação</i>
TOPSIS	<i>Technique for Order Performance by Similarity to Ideal Solution</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>13</b>
1.1	Objetivos .....	15
1.2	Justificativa .....	15
1.3	Delimitação da pesquisa .....	17
1.4	Classificação da pesquisa .....	17
1.5	Estrutura do trabalho .....	18
<b>2</b>	<b>REVISÃO DE LITERATURA</b> .....	<b>20</b>
2.1	Procedimentos para a revisão da literatura .....	20
2.2	Processo de desenvolvimento de <i>software</i> .....	26
2.3	Engenharia de requisitos e a priorização de requisitos .....	28
2.4	Estimativas de <i>software</i> .....	30
2.4.1	Direcionadores fatores de custo ou ajuste .....	34
2.5	Métodos multicritério de apoio à tomada de decisão .....	39
2.5.1	Método AHP .....	40
2.5.2	Conjuntos Difusos ( <i>Fuzzy Sets</i> ) e números <i>Fuzzy</i> .....	41
2.5.3	Fuzzy TOPSIS .....	43
<b>3</b>	<b>METODOLOGIA</b> .....	<b>45</b>
3.1	Modelo multicritério de priorização de requisitos de <i>software</i> .....	45
3.2	Estruturação do modelo .....	46
3.2.1	Determinação dos especialistas envolvido nas avaliações de importância .....	47
3.2.2	Determinação da importância dos fatores de impacto .....	47
3.2.3	Escala para avaliação linguística dos requisitos do <i>software</i> em análise.....	51
3.2.4	Protótipo para coleta das avaliações dos requisitos .....	52
3.3	Aplicação do modelo .....	53
3.3.1	Seleção e avaliação dos requisitos para desenvolvimento .....	54
3.3.2	Agregação das avaliações de complexidade dos requisitos .....	56
3.3.3	Ordenação dos requisitos em relação aos fatores de impacto com o <i>fuzzy topsis</i> .....	56
3.3.4	Avaliação da percepção dos envolvidos na aplicação do modelo .....	58
<b>4</b>	<b>RESULTADOS</b> .....	<b>62</b>

<b>4.1</b>	<b>Avaliação da importância dos fatores de impacto .....</b>	<b>62</b>
<b>4.2</b>	<b>Seleção e avaliação dos requisitos para desenvolvimento .....</b>	<b>66</b>
<b>4.3</b>	<b>Ordenação dos requisitos para desenvolvimento .....</b>	<b>69</b>
4.3.1	Apresentação dos requisitos ordenados com os principais fatores de impacto .....	70
4.3.2	Avaliação da percepção da aplicação e aceitação do modelo elaborado .....	75
4.3.3	Comparativo com trabalhos relacionados .....	77
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>78</b>
	<b>REFERÊNCIAS .....</b>	<b>81</b>
	<b>APÊNDICE A .....</b>	<b>92</b>
	<b>APÊNDICE B .....</b>	<b>96</b>
	<b>APÊNDICE C .....</b>	<b>97</b>
	<b>APÊNDICE D .....</b>	<b>98</b>
	<b>APÊNDICE E .....</b>	<b>108</b>
	<b>APÊNDICE F .....</b>	<b>112</b>
	<b>APÊNDICE G .....</b>	<b>114</b>

## 1 INTRODUÇÃO

A Tecnologia da Informação (TI) tem cada vez mais relevância no cenário mundial, de aplicativos a *hardwares*, inseridos no dia a dia da medicina, comunicações e serviços, tornando-se cada vez mais imprescindível para a realização de qualquer trabalho moderno (CANTANHEDE, 2014). Segundo a Associação Brasileira de Indústrias de *Software* (ABES), em estudo realizado pela *International Data Corporation* (IDC) (2021), em 2020 o Brasil ficou em 9º lugar no *ranking* mundial de investimento em TI, destes 26% em software, 20% em serviços e 54% em hardware. Além disso, o setor de software apresentou crescimento mais acentuado do que o setor de serviços pela primeira vez, com crescimento de 28,7% em 2020, em relação a uma redução de 4,5% no mercado de serviços. Segundo o mesmo estudo, no mercado de *software* o crescimento em 2019 foi de 16% comparado ao ano anteriores para 2021 devem atingir US\$ 1,4 bilhões, o que representa um crescimento de 21,3% em relação a 2020. Das empresas no segmento de TI 58,3% delas têm como atividade principal o desenvolvimento e produção de *software* ou prestação de serviços. Nesse cenário, é possível observar que a demanda por tecnologia continua evoluindo, e assim, continuam gerando novas demandas por *softwares*.

A cada dois anos, o *Standish Group* publica um novo CHAOS Report - (*Comprehensive Human Appraisal for Originating Software*) que avalia e oferece excelentes informações sobre as causas do fracasso ou sucesso do projeto de desenvolvimento de *software*. Ao analisar os fatores avaliados pelo *Standish Group* (2018), dos itens considerados que contribuem para o sucesso de um projeto de desenvolvimento de *software*, dentre entrega dentro de prazo, orçamento e valor satisfatório, supõe-se que a entrega do sistema no prazo definido é um dos mais críticos. Se o *software* demorar muito tempo para ser concluído, maiores serão os esforços e os recursos financeiros que ele demandará tanto dos desenvolvedores quanto da empresa. Comparado com a análise do *Standish Group* (2015), entre os anos de 2011 e 2015, mostrou que a faixa anual de projetos bem-sucedidos, que ficou entre 27% e 31%, e que 49% a 56% embora considerados satisfatórios, foram contestados pelo orçamento e cronograma excedentes ao aceitável. Os dados para o ano de 2017 são: bem-sucedido: 14%, desafiados: 67%, reprovado em 19%. Nesse contexto, para que empresas desenvolvedoras de *software*, concluir com sucesso seus projetos é cada vez mais valioso para sua cadeia de valor.

Além disso, a construção de um *software* é considerada uma atividade complexa (PRESSMAN 2011), e de acordo com o CHAOS Report de 2018, o primeiro fator de sucesso para os projetos, é a latência de decisão. A teoria da latência de decisão afirma: “O valor do

intervalo é maior do que a qualidade da decisão.” (Standish Group, 2018). Em outras palavras, se a empresa quiser melhorar o sucesso do projeto, deve acelerar sua tomada de decisão. Decisões incorretas, gestão inadequada daquilo que está sendo desenvolvido, além de gerar um grave desperdício no processo de produção de *software*, podem levar ao fracasso do projeto (POPPENDIECK, 2003; MISHRA, 2011; PERRY, 2001).

Pressman (2011), afirma que os sistemas de software vão se deteriorando devido a necessidade de evolução, seja ela para inclusão ou para alteração de requisitos, passando por vários ciclos de desenvolvimento e assim em contínuo gerenciamento de projeto. De acordo com o PMBOK (2017) o gerenciamento de projetos, inclui entre outras atividades, a necessidade de equilibrar as restrições conflitantes do projeto como Escopo, Cronograma, Custo, Qualidade, Recursos e Riscos. Tais restrições que no contexto de desenvolvimento de *software*, estão relacionadas às estimativas de esforço e custo, que por sua vez, são dependentes dos requisitos e seu gerenciamento, pois são realizadas apoiadas nas suas especificações.

Os requisitos de software são de responsabilidade da Engenharia de Requisitos (ER), área que trata da análise das necessidades, das especificações e da documentação dos requisitos e fornece mecanismos para facilitar sua verificação (THAYER; DORFMAN, 2000).

Conforme Wang (2014), durante a ER, a priorização é realizada para ranquear ou classificar os requisitos em sua ordem de importância e liberações de implementação subsequentes, sendo usada nas fases iniciais do desenvolvimento de software para determinar a ordem na qual os requisitos devem ser implementados

Em uma pesquisa realizada com gerentes de projeto, a ER foi classificada como uma área de maior risco a ser gerenciado em um projeto (BEECHAM et al.,2005) colocando-a como um processo centrado em decisão (SVENSSON et al., 2011). Frequentemente, atender e garantir os requisitos essenciais das partes interessadas é considerada uma das principais razões para produzir um sistema de software de boa qualidade (KESHTA et al., 2017) e priorizar requisitos é uma parte importante da negociação (SVENSSON et al.,2012).

Priorizar requisitos de software é uma tarefa importante e difícil durante a fase de gerenciamento de requisitos da ER (YASEEN et al.,2020) pois quando um problema que nasceu na fase de requisitos é encontrado em fases avançadas de desenvolvimento, mais recursos deverão ser gastos para corrigir este problema. Numa situação como esta, o melhor que se pode esperar é que os requisitos que não poderão mais ser desenvolvidos sejam de funcionalidades não essenciais para o *software* (SAWYER et al., 1999). Afraz (2021) afirma que um dos principais fatores no insucesso de projetos de software é a falta de atenção na PR.

Hujainah et al.(2018) conclui que ainda há vários desafios nos processos das técnicas de PR existentes, como escalabilidade, falta de processo de SQP, necessidade de envolvimento de analistas profissionais em PR e SQP, dependências de requisitos, complexidade, falta de nível de automação e fatores de tempo.

Nesse contexto, a proposta deste trabalho se insere na necessidade de priorizar-se os requisitos conforme a complexidade sob a ótica da equipe técnica. Neste sentido, este trabalho busca desenvolver um modelo de priorização de requisitos de *software* com ênfase na complexidade e deste modo, permitir aos gestores, tomar decisões mais assertivas e rápidas, desde o planejamento e durante o acompanhamento da execução.

## **1.1 Objetivos**

Nesta seção serão apresentados o objetivo geral e os específicos da pesquisa.

### **1.1.1 Objetivo Geral**

Este trabalho tem como objetivo desenvolver um modelo multicritério de priorização de requisitos de *software* baseado na sua complexidade sob o ponto de vista da equipe técnica.

### **1.1.2 Objetivos Específicos**

Os objetivos específicos definidos para a pesquisa são: (i) identificar na literatura existente, as principais técnicas usadas para priorizar requisitos e limitações; (ii) selecionar e aplicar os fatores de impacto como critérios de avaliação de complexidade dos requisitos sob o ponto de vista da equipe de técnica; (iii) desenvolver um modelo multicritérios para priorizar requisitos de software e (iv) aplicar o modelo desenvolvido, na gestão de um projeto de desenvolvimento de software, para observar a percepção dos envolvidos na implementação, bem como obter sugestões de melhorias no modelo.

## **1.2 Justificativa**

ARRUDA *et. al.* (2014) diz que o sucesso no desenvolvimento de um software pode ser medido pela forma com que ele realiza as atividades para qual foi proposto. Os requisitos



impõem restrições tanto ao software a ser desenvolvido quanto ao processo usado para o desenvolver. É uma tarefa complexa e requer muito esforço na tomada de decisão em vários pontos, como redundância, não redundância dos requisitos, preferências dos requisitos, seleção de requisitos corretos e valiosos. Uma solução para lidar com essas decisões complexas é a priorização de requisitos (GUPTA *et. al.* 2018).

A priorização de requisitos (PR) é um processo de decisão, onde alguma coisa será escolhida em detrimento de outra, no contexto de projetos de tecnologia da informação, serão escolhidos requisitos de um produto que se espera alcançar com um projeto em questão e assim ajudar a equipe do projeto a entender quais requisitos são mais importantes e mais urgentes (AASEM *et. al.* 2010; DANEVA *et al.*, 2013). A PR apresenta um papel significativo durante a implementação efetiva dos requisitos (YASEEN *et. al.* 2019), sendo uma das atividades centrais da ER (YASEEN *et. al.* 2015; YASEEN *et. al.* 2019).

Em estudo sobre o estado da arte em PR, Achimugu *et al.* (2014) mapearam 49 técnicas de priorização em 73 trabalhos selecionados, das quais mais da metade (26) foram utilizadas em no máximo 3 estudos, demonstrando que várias tentativas de aperfeiçoamento para essa tarefa estão sendo testadas. Entre as várias técnicas com diferentes abordagens para executar o processo de PR encontram-se o Método Hierárquico Analítico (AHP) é a que mais aparece com 48 citações em Achimugu *et al.*(2014), Priorização Orientada a Valor (VOP), Técnica de Atribuição Numérica (NAT), Pesquisa Binária (BS), Votação Cumulativa (CV), Compreensão e Associação de Chance de Engenharia Multimídia (SERUM), Ranking, Arranjo de Jogo (PG), EVOLUIR entre outras.

Porém, tais técnicas deixam ainda muitas lacunas a serem trabalhadas, como classificações simples sem valor de prioridade, ou dificuldades para atender a requisitos grandes, necessidade de testes para eficácia da complexidade computacional, dificuldade em obter consenso entre as partes, problemas de escalabilidade, complexidade entre outras (SHAO *et al.* 2017; HUIJAINAH *et al.* 2018; SUFIAN *et al.* 2018). Além disso, existem um conjunto de fatores que dificultam uma correta priorização dos requisitos, esses fatores podem ser entendidos como obstáculos a tomada de decisões sobre priorização de requisitos de software, e afetam de forma significativa o sucesso de um projeto de tecnologia da informação (DANEVA *et al.* 2013; YASEEN *et. al.* 2019).

Sendo assim, considerando o contexto e a importância da problemática apresentados, tendo então, como contribuição prática, o desenvolvimento de uma nova ferramenta de apoio no processo de tomada de decisões. Possibilita aos gerentes de projetos, decisões de maneira acelerada, além de apresentar uma perspectiva de ordenação de requisitos por complexidade

sob o ponto de vista da equipe técnica, e valorizar o conhecimento dos analistas e programadores do projeto, por meio da obtenção de informações qualitativas. Nesse contexto, este trabalho faz uso de termos linguísticos para avaliar fatores de impacto já estabelecidos e validados em modelos de estimativas de esforço e custo de *software*, para gerar a ordenação de requisitos conforme sua complexidade e também apresentar quais os fatores que mais impactam em cada requisito avaliado.

Nesse sentido, como contribuição teórica, este trabalho amplia e enriquece as opções de modelos de priorização de requisitos de projetos de desenvolvimento de *software* (DS) disponíveis na literatura, e permite uma abordagem com potencial de aplicação na gestão de projetos nas mais diversas escalas de DS.

### **1.3 Delimitação da pesquisa**

O modelo proposto nesta dissertação foi elaborado a partir de conceitos da revisão de literatura referente a ER, mais especificamente a priorização de requisitos de software. Além disso, foram utilizados da literatura clássica da área de ER, os fatores de impacto, identificados a partir dos modelos das estimativas de esforço e de custo COCOMO II (BOEHM, 2000; BOEHM, 2017) e Análise de Pontos por Função (ALBRECHT, 1979).

A construção do modelo é limitada a avaliações com profissionais de desenvolvimento de software do Distrito Federal e do Paraná, especificamente para o contexto de priorização de *software* por meio de avaliações de complexidade. A aplicação do modelo desenvolvido, ocorreu em apenas uma empresa de médio porte, que desenvolve *softwares* para várias áreas de negócio e está localizada na região sudoeste do estado do Paraná e ocorreu no ano de 2021.

Os resultados obtidos são restritos, portanto, a este contexto..

### **1.4 Classificação da pesquisa**

Quanto a sua natureza, esta pesquisa se caracteriza como aplicada, a qual segundo Gil (2017) é voltada para a aquisição de conhecimento e aplicação em situações específicas. Neste sentido, por se tratar do desenvolvimento de um modelo de apoio ao gerenciamento de projetos de desenvolvimento de *software*, proporcionando aos gestores, um modelo para identificar

quais são os requisitos que apresentam maior complexidade e com isso contribuir para um melhor planejamento, monitoramento e controle de ações durante a execução do projeto.

Em relação à abordagem, os métodos podem ser classificados em qualitativos e quantitativos. A abordagem qualitativa, permite à compreensão de opiniões e avaliações subjetivas e linguísticas para obter teorias conforme define Hakansson (2013) e de acordo com Cauchick (2012), busca coletar dados sem a necessidade de medições numéricas, enquanto a quantitativa utiliza-se da coleta de dados para testar hipóteses por meio de medições numéricas e análises estatísticas. Desta maneira, a pesquisa desenvolvida apresenta uma abordagem mista.

Foram adotados procedimentos caracterizados como bibliográfico e documental, com base em referências como livros e artigos para a busca de conceitos relevantes nas bases bibliográficas *Elsevier*, IEEE e *Web of Science* (WOS) e *Scopus*. Além disso, a pesquisa adota a modelagem por meio de método de decisão multicritério linguístico (*fuzzy*), pois segundo Morabito *et al.* (2018) o uso de modelos permite melhor compreensão do ambiente, identificar problemas, formular estratégias e oportunidades e apoiar e sistematizar o processo de tomada de decisões.

A pesquisa é caracterizada como exploratória, quanto aos objetivos, com a finalidade de buscar maiores informações sobre o assunto. Nesse sentido, realizou-se o levantamento bibliográfico com o intuito de ampliar o conhecimento acerca do problema e assim, obter melhor compreensão sobre o tema, e, por meio deste, adquirir o embasamento teórico para a proposição do modelo apresentado nesta dissertação.

## **1.5 Estrutura do trabalho**

No capítulo um, são apresentados: introdução, objetivos, justificativa, delimitação e a classificação da pesquisa e a estrutura da dissertação.

No capítulo dois é abordado o referencial teórico relativo às bases e definições que amparam as áreas abordadas neste trabalho: estratégia para revisão de literatura, processo de desenvolvimento de *software*, engenharia de requisitos, mais especificamente priorização de requisitos, bem como, os métodos de decisão multicritérios empregados no modelo proposto.

No capítulo três apresenta-se o desenvolvimento do Modelo Multicritério para Priorização De Requisitos De Software com o detalhamento dos procedimentos empregados na aplicação do modelo proposto.

No capítulo quatro apresentam-se os resultados para a construção do Modelo Multicritério para Priorização De Requisitos De Software e os resultados de sua aplicação

E por fim, no capítulo cinco são apresentadas as conclusões obtidas com o desenvolvimento do modelo e sugestões de trabalhos futuros

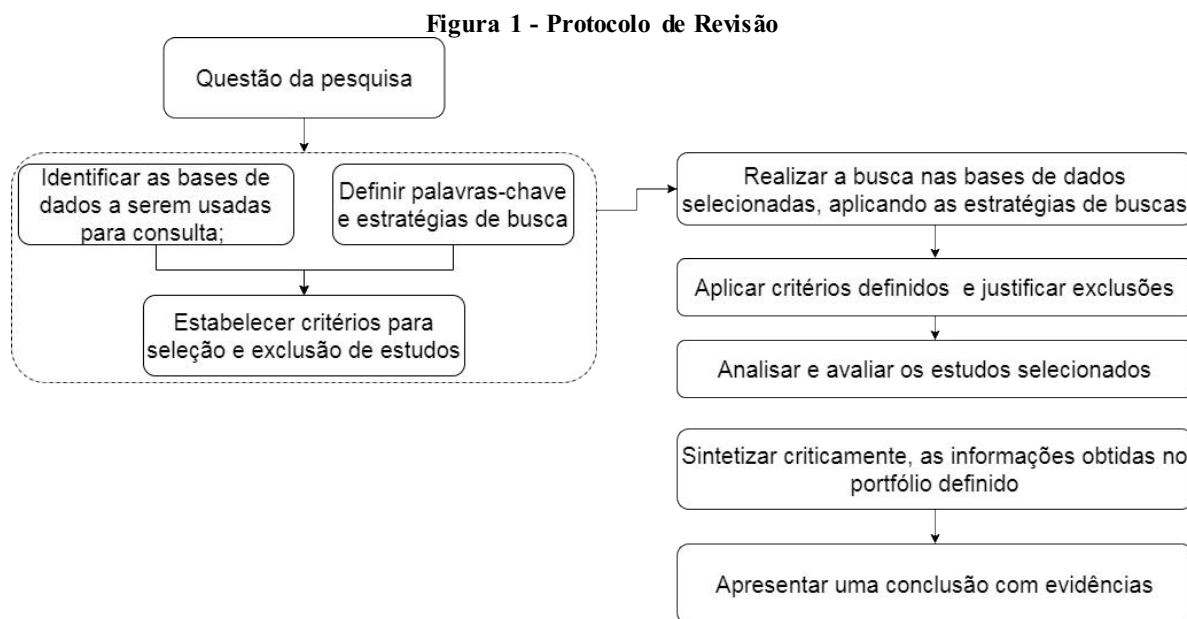
## 2 REVISÃO DE LITERATURA

Este capítulo tem o objetivo de fundamentar teoricamente o processo de desenvolvimento de *software*, os modelos de estimativas e os métodos multicritérios utilizados, os quais procuram oferecer as bases da literatura para o tema em questão.

### 2.1 Procedimentos para a revisão da literatura

Para a realização deste estudo, foi utilizado o método de revisão de literatura de acordo com as diretrizes de Kitchenham e Charters (2007), para encontrar e analisar estudos publicados anteriormente, objetivando a construção de conhecimento acerca dos temas priorização de requisitos de software o qual é apresentado na etapa 2.1.

O processo de revisão de literatura como primeiro passo a definição de um protocolo de revisão. O protocolo de revisão de pesquisa definido, inclui quatro atividades: desenvolvimento de questões de pesquisa, implementação de estratégias de busca e seleção de estudos (critérios de inclusão e exclusão e critérios de avaliação de qualidade) e coleta de dados juntamente com a síntese de dados. O protocolo é apresentado na Figura 1.



Fonte: Autoria própria (2021)

Após a seleção de estudos relevantes foi definido os critérios para inclusão e exclusão. O critério de inclusão estipulado foi estudo que apresenta evidências da abordagem do assunto de PR; E os critérios de exclusão: (i) estudos repetidos; (ii) estudos que não tratam de PR; (iii) estudos cujo texto completo não foi acessível por restrições autorais; (iv) estudos apresentados em idiomas diferentes do idioma inglês ou português.

Com os resultados atingidos a partir da execução dessa pesquisa, pretende-se responder às seguintes questões de pesquisa (Q): Q1: Quais são as técnicas existentes usadas para priorizar requisitos e lacunas para novos estudos? Q2: Como avaliar a complexidade dos requisitos selecionados para desenvolvimento, sob o ponto de vista da equipe de técnica? Q3: Os métodos AHP e Fuzzy TOPSIS poderiam ser aplicados na priorização de Requisitos de software?

### 2.1.1 Seleção do portfólio bibliográfico

Para o desenvolvimento deste trabalho utilizou-se as bases de dados *Scopus*, *ScienceDirect* e *Web of Science*, *IEEE Xplore* sendo as 4 acessíveis pelo portal CAPES (Coordenação de Aperfeiçoamento de Ensino Superior), justificando a escolha das bases por: (i) bases de dados das áreas de conhecimento Ciências Exatas e da Terra – Subárea: Ciência da Computação seguindo a classificação dos critérios do portal Sucupira, e portanto na área de ER; (ii) bases de dados que contém resumos e textos completos; (iii) bases de dados que disponibilizam ferramentas de busca com expressões booleanas; (iv) bases de dados que permitem a pesquisa das combinações dos eixos no Título, Palavras-chave e Resumo simultaneamente; e (v) base de dados que apresentam a disponibilidade de exportar os artigos para um *software* de gerenciamento bibliográfico;

Para buscar os artigos nas bases de dados definidas, foram definidas as palavras-chave relacionadas, e aplicado o protocolo de busca conforme Figura 2 apresentada anteriormente.

Após a definição dos parâmetros da pesquisa, a busca foi iniciada aplicando as *strings* de busca conforme Quadro 1. Utilizou-se a expressão booleana “and”, para fazer a ligação entre as combinações de palavras e OR para palavras sinônimas.

**Quadro 1 – Strings de busca**

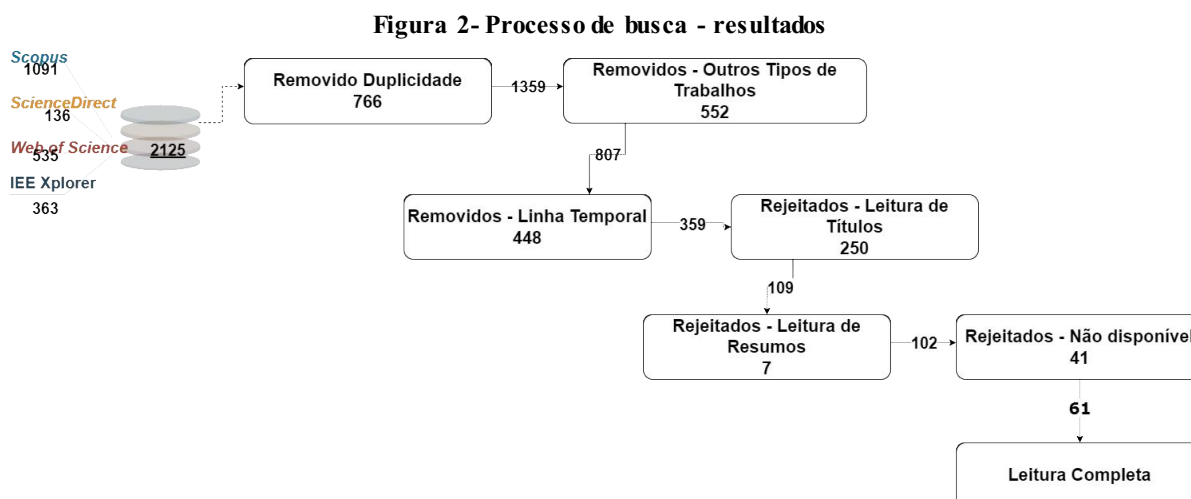
<b>Base</b>	<b>String de Busca</b>
<i>Science direct</i>	<i>Title, abstract, keywords: ( requirements AND ( "prioritization criteria " OR "prioritization" OR "technique of prioritization" OR "ordenation methods" ) AND ("software development management" OR "Requirement engineering" OR software OR "software development" ) )</i>

Scopus	TITLE-ABS-KEY ( requirements AND ( "prioritization criteria" OR "prioritization" OR "technique of prioritization" OR "ordenation methods" ) AND ( "software development management" OR "Requirement engineering" OR software OR "software development" ) )
IEEE Xplore	(( "All Metadata": requirements AND ( "All Metadata": "prioritization criteria" OR "prioritization" OR "technique of prioritization" OR "ordenation methods" ) AND ( "All Metadata": "software development management" OR "Requirement engineering" OR software OR "software development" ) ))
Web of Science	ts=( requirements AND ( "prioritization criteria" OR "prioritization" OR "technique of prioritization" OR "ordenation methods" ) AND ( "software development management" OR "Requirement engineering" OR software OR "software development" ) )

Fonte: Autoria Própria (2021).

Retornaram 2125 artigos para análise inicial. O gerenciamento bibliográfico dos artigos coletados foi realizado com o uso do EndNote X7.2. Desse total, foram excluídos 766 artigos duplicados e 552 itens por não serem *article*, *review* nem *Conference Paper*, restando 807 itens que passaram pela linha de corte temporal para somente após 2016 restando 359 estudos.

Na etapa seguinte foi realizada a leitura dos títulos e removido os que não estavam alinhados ao tema, restando 109 para análise, dos quais foram eliminados 3 na leitura dos resumos e 41 por não estarem disponíveis por direito autoral. A inclusão dos estudos foi determinada pela relevância, pela análise do título, resumo, introdução, conclusão e leitura completa do estudo em diferentes estágios do processo de revisão restando assim 65 estudos conforme Figura 2.



Fonte: Elaboração própria(2021)

Estes artigos são relevantes para o estudo do tema, os quais formam o portfólio de artigos da pesquisa e serão utilizados para embasar na literatura alguns conceitos abordados nesta pesquisa.

Na próxima etapa foi realizado a análise das publicações presentes no portfólio de suas referências.

Na primeira análise, buscaram-se identificar quais são os periódicos com maior número de publicações relacionadas ao tema, e assim, servindo de referência para futuras pesquisas relacionadas ao assunto. Conforme apresentado no Quadro 2, os periódicos, *Institute of Electrical and Electronics Engineers* com 21, seguido por *International Journal of Advanced Computer Science and Applications* com 5 artigos selecionados, e depois *CEUR Workshop Proceedings*, *Knowledge Systems Institute Graduate School*, *IEEE Access*, *IEEE Computer Society* com três cada.

**Quadro 2 - Periódicos do portfólio por número de artigos e indicadores**

<b>Journal</b>	<b>Número de Artigos</b>	<b>CiteScore 2020</b>	<b>SNIP 2018</b>
<i>Advances in Systems Science and Applications</i>	1	0,9	130
<i>Algorithms Open Access</i>	1	2,9	2.615
<i>Applied Sciences</i>	1	0,9	130
<i>CEUR Workshop Proceedings</i>	3	0,8	15.553
<i>Complexity Open Access</i>	1	3,3	10.227
<i>Concurrent Engineering Research and Applications</i>	1	2,2	228
<i>Faculty of Mechanical Engineering and Naval Architecture</i>	1	-	-
<i>IEEE Access</i>	3	4,8	201.619
<i>IEEE Computer Society</i>	3	4,8	201.619
<i>IET Software</i>	1	2,7	620
<i>Information and Software Technology</i>	1	8,6	4.461
<i>Institute of Electrical and Electronics Engineers</i>	21	-	-
<i>Institute of Physics Publishing</i>	2	-	-
<i>International Journal of Advanced Computer Science and Applications</i>	5	1,1	3.285
<i>International Journal of Innovative Technology and Exploring Engineering</i>	1	N/A	N/A
<i>International Journal of Scientific and Technology Research</i>	1	N/A	N/A
<i>Journal of Information Technology Research</i>	2	1,2	60
<i>Journal of Telecommunication, Electronic and Computer Engineering</i>	1	N/A	N/A
<i>Journal of Theoretical and Applied Information Technology</i>	1	1,3	3.003
<i>Knowledge Systems Institute Graduate School</i>	3	-	-
<i>Neutrosophic Sets and Systems</i>	2	2,7	823

Fonte: Dados da pesquisa.



Para responder a Q1, as principais técnicas após leitura e análise dos artigos do portfólio foram listadas no Quadro 3.

**Quadro 3 – Principais Técnicas de PR citadas no Portfólio**

<b>Técnica</b>	<b>Autor(es) Citando</b>
Abordagem Custo-Valor	Tufail <i>et. al.</i> (2019), Upadhyay e Sharma (2020)
<i>AHP-GORE-PSR</i>	Sadiq <i>et. al.</i> (2017),
<i>ANN Fuzzy Model</i>	Cavalcanti <i>et. al.</i> (2018)
Arranjo de Jogo (PG)	Ahuja <i>et. al.</i> (2016), Hudaib <i>et. al.</i> (2018), Cavalcanti <i>et. al.</i> (2018)
Arvore de Decisão Binária (BST)	Yaseen <i>et. al.</i> (2018), Ibriwesh <i>et. al.</i> (2019), Hudaib <i>et. al.</i> Hudaib <i>et. al.</i> (2018), Aziz <i>et. al.</i> (2019) Cavalcanti <i>et. al.</i> (2018) , AbdElazim, <i>et. al.</i> (2020), Hujainah <i>et. al.</i> (2021), Galarza <i>et. al.</i> (2021)
Compreensão e Associação de Chance de Engenharia Multimídia (SERUM)	Ahuja <i>et. al.</i> (2016), Ahmad <i>et. al.</i> (2017)
Classificação Baseada em Casos	Ibriwesh <i>et. al.</i> (2019), Yaseen Lee <i>et. al.</i> (2019), Yaseen <i>et. al.</i> (2020)
Desdobramento de Função Qualidade (QFD)	Ahmad <i>et. al.</i> (2017), Ibriwesh <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018), Abbas <i>et. al.</i> (2019), Yaseen <i>et. al.</i> (2020), Govil e Sharma(2021)
EVOLVE	Ahuja <i>et. al.</i> (2016), Misaghian <i>et. al.</i> (2019), AbdElazim, <i>et. al.</i> (2020),
<i>Fuzzy-AHP</i>	Cavalcanti <i>et. al.</i> (2018), Win. <i>et. al.</i> (2020), Upadhyay e Sharma (2020), Amelia e Mohamed(2021)
<i>Fuzzy Based-Moscow</i>	Ahmad <i>et. al.</i> (2017)
<i>Hierarchy AHP</i>	Hudaib <i>et. al.</i> (2018), Alzaqebah <i>et. al.</i> (2018), Lee <i>et. al.</i> (2019), Aziz <i>et. al.</i> (2019), Ijaz <i>et. al.</i> (2019), Devadas e Srinivasan (2019)Win. <i>et. al.</i> (2020), Hujainah <i>et. al.</i> (2021)
Lógica Fuzzy	Gulzar <i>et. al.</i> (2017), Abbas <i>et. al.</i> (2019), Devadas e Srinivasan (2019), Cabrera <i>et. al.</i> (2020), AbdElazim, <i>et. al.</i> (2020),

Modelo Kano	Hudaib <i>et.al</i> (2018) , Cavalcanti <i>et. al.</i> (2018), Aziz <i>et. al.</i> (2019) Amelia e Mohamed(2021)
Moscow	Ahmad <i>et.al.</i> (2017), Hudaib <i>et.al</i> (2018) Lee <i>et. al</i> (2019), Tufail <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018), Alzaqebah <i>et. al.</i> (2018), Aziz <i>et. al.</i> (2019) Zacarias <i>et.al.</i> (2020) , Govil e Sharma(2021)
Multi-criteria RP	Ahmad <i>et.al.</i> (2017), Kifetew <i>et.al.</i> (2017), Cavalcanti <i>et. al.</i> (2018), Amelia e Mohamed(2021).
Pesquisa Binária (BS)	Ahuja <i>et.al.</i> (2016), Hudaib <i>et.al</i> (2018) Lee <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018)
Jogo de planejamento	Hudaib <i>et.al.</i> (2018) Lee <i>et. al.</i> (2019), Tufail <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018), AbdElazim, <i>et.al.</i> (2020),
Priorização Orientada a Valor (VOP)	Ahuja <i>et.al.</i> (2016), Ahuja <i>et.al.</i> (2016), Lee <i>et. al.</i> (2019), Tufail <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018), Galarza <i>et.al.</i> (2021)
Processo Hierárquico Analítico (AHP)	Ahuja <i>et.al.</i> (2016), Kifetew <i>et.al.</i> (2017) Sadiq <i>et.al.</i> (2017), Hudaib <i>et.al.</i> (2018) Lee <i>et. al.</i> (2019), Misaghian <i>et.al.</i> (2019), Tufail <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018), Alzaqebah <i>et. al.</i> (2018), Aziz <i>et. al.</i> (2019), Abbas <i>et.al.</i> (2019), Win. <i>et. al.</i> (2020), Yaseen <i>et.al.</i> (2020), AbdElazim, <i>et.al.</i> (2020), Zacarias <i>et.al.</i> (2020) , Upadhyay e Sharma (2020), Ali <i>et.al.</i> (2020), Hujainah <i>et.al.</i> (2021), Amelia e Mohamed(2021), Galarza <i>et.al.</i> (2021), Govil e Sharma (2021)
Ranking	Ahmad <i>et.al.</i> (2017), Tufail <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018), AbdElazim, <i>et.al.</i> (2020), Upadhyay e Sharma (2020), Ali <i>et.al.</i> (2020)
Técnica de Atribuição Numérica (NAT)	Ahuja <i>et.al.</i> (2016), Ahmad <i>et.al.</i> (2017), Hudaib <i>et.al.</i> (2018), Lee <i>et. al.</i> (2019), Yaseen <i>et.al.</i> (2020), Tufail <i>et. al.</i> (2019), Aziz <i>et. al.</i> (2019) Cavalcanti <i>et. al.</i> (2018), Zacarias <i>et.al.</i> (2020), Upadhyay e Sharma (2020), Ali <i>et.al.</i> (2020), Galarza <i>et.al.</i> (2021)
TOPSIS	Ahmad <i>et.al.</i> (2017), Misaghian <i>et.al.</i> (2019), Galarza <i>et.al.</i> (2021)
Valued Based	Hudaib <i>et.al.</i> (2018)
VDA	Barbosa <i>et.al.</i> (2018), Barbosa <i>et.al.</i> (2019), Barbosa <i>et.al.</i> (2020)

Votação (CV)	Cumulativa	Ahuja <i>et.al.</i> (2016), Ahmad <i>et.al.</i> (2017), Hudaib <i>et.al.</i> (2018), Lee <i>et. al.</i> (2019), Tufail <i>et. al.</i> (2019), Cavalcanti <i>et. al.</i> (2018), Alzaqebah <i>et. al.</i> (2018), Ali <i>et.al.</i> (2020)
-----------------	------------	---

Fonte: Dados da pesquisa (2021).

Existem muitas técnicas com diferentes abordagens para executar o processo de PR, e como demonstra o Quadro 5, sendo que o Método Hierárquico Analítico (AHP) foi o que mais foi citado, usado individualmente ou em conjunto com outras técnicas. As demais técnicas também são muito utilizadas, como Priorização Orientada a Valor (VOP), Técnica de Atribuição Numérica (NAT), Pesquisa Binária (BS), Votação Cumulativa (CV), Compreensão e Associação de Chance de Engenharia Multimídia (SERUM), Ranking, Arranjo de Jogo (PG), EVOLUIR, PHandler, e DRank.

É comum nos trabalhos do portfólio as priorizações sob a perspectiva de outras partes interessadas e não da equipe técnica. Além disso, há a necessidade de mais estudos com abordagens híbridas, pois de acordo com os pesquisadores frequentemente pesquisam várias técnicas, no entanto, há menos esforço em mesclar essas técnicas (IJAZ *et. al.* 2019)

## 2.2 Processo de desenvolvimento de *software*

A ABNT NBR ISO/IEC 12207:2009 define processo como “conjunto de atividades que se relacionam ou interagem e que transformam entradas em saídas” (ABNT, 2009). O processo é denominado ciclo de vida, quando se refere às fases específicas do *software*, como, levantamento, análise e validação de requisitos, desenvolvimento e implantação (FUGGETTA, 2000). Na produção de *softwares*, uma ação envolve um conjunto de tarefas que resulta em um artefato de software, enquanto a tarefa é voltada para um objetivo pequeno, que resulta em produzir um resultado tangível (SOMMERVILLE, 2011).

A qualidade do *software* está diretamente relacionada ao seu processo de desenvolvimento. São os processos que definem a base para o controle gerencial de projetos, estabelecendo como a qualidade é garantida e as modificações são gerenciadas e o cronograma seja cumprido. Além disso, também representam o contexto de aplicação dos métodos, técnicas, uso das ferramentas e de obtenção dos resultados (PRESSMAN, 2011).

O processo de desenvolvimento de *software* é uma atividade complexa que demanda planejamento, decisão, fatores humanos e estratégicos para desenvolver um produto final de

qualidade, que atenda às necessidades do cliente. Os projetos de *software* envolvem uma série de variáveis determinantes para o seu sucesso, que devem ser gerenciadas e estimadas para não comprometer o que foi previamente estabelecido (SOMMERVILLE, 2011; PRESSMAN, 2011; BASSI FILHO, 2008).

Pressman (2011) define a composição do processo de desenvolvimento de *software* em quatro atividades básicas: especificação, desenvolvimento, validação e evolução, sendo que organização e comunicação entre elas dependem do modelo de desenvolvimento adotado. Por outro lado, Sommerville (2011); considera que embora haja variação conforme os processos; adotados, algumas atividades são comuns: especificação, projeto e implementação, validação e evolução do *software*. É na fase de especificação que está inserida a ER, em que ocorre o processo de compreensão e definição dos requisitos e suas restrições demandados pelos sistemas.

Requisitos são as descrições do que o sistema deve fazer, os serviços oferecidos e as restrições de funcionamento. São capacidades e condições de um sistema, atributos necessários em um sistema para que ele tenha valor e utilidade para os clientes e usuários e fornecem a base para todo o trabalho de desenvolvimento de *software* subsequente. (LARMAN, 2004; YOUNG, 2004).

Em uma classificação técnica, os requisitos são frequentemente definidos como (SOMMERVILLE, 2011): (i) funcionais que determinam as funcionalidades, serviços, que o sistema deve realizar; e (ii) não funcionais que se tratam de restrições aos serviços e funcionalidades oferecidos pelo sistema e ao sistema como um todo, à organização e externas.

Embora possa haver variação dos processos, como afirma Sommerville (2011), geralmente entre as atividades de especificação e desenvolvimento acontece a priorização dos requisitos a serem implementados. Quando há mudanças dos requisitos, pode-se criar riscos adicionais (PMBOK, 2017) e a equipe precisa ser capaz de avaliar a situação, equilibrar e priorizar as demandas, além de manter uma comunicação proativa entre as partes interessadas, visando um projeto bem-sucedido.

A área de conhecimento gerenciamento de tempo do PMBOK (2017), possui entre as muitas responsabilidades, as quais encontram-se a de estimar o tempo de trabalho, o cronograma, os recursos dentro de um prazo definido, além de controlar e monitorar o cronograma: e, se necessário, promover mudanças sem afetar o rendimento geral do projeto. Nesse sentido, quando observado o contexto de empresas desenvolvedoras de software, é comum perceber que repetidas vezes a quantidade de requisitos supera a quantidade de tempo, recurso e orçamentos disponíveis para a construção dos mesmos. Por isso, existe a necessidade

de priorizar-se os requisitos, selecionando os que tendem atingir a maior satisfação dos clientes, ao ser implementado (MULLA, 2012).

As empresas cada vez mais, enfrentam a dificuldade de alcançar equilíbrio entre requisitos e recursos, e conforme Davis (2003), devido ao mercado competitivo, elas tentam adicionar características e reduzir os cronogramas para a entrega de cada produto. Características específicas do projeto de desenvolvimento ou da área de negócio podem impossibilitar que o desenvolvimento ocorra exatamente conforme definido pela lista de requisitos priorizados. O resultado pode levar ao desacordo entre recursos e requisitos, gerando produtos que não são entregues no prazo ou que não conseguem atender as necessidades dos clientes. Portanto, o gerenciamento do projeto serve para oferecer suporte na tomada de decisões desde a formação de equipe, elaboração do orçamento de projeto e também precisa definir mecanismos para monitorar o projeto (PRESSMAN, 2011; SOMMERVILLE, 2011).

### **2.3 Engenharia de requisitos e a priorização de requisitos**

O ciclo de vida de um software é composto por diversas fases, independente da metodologia utilizada. A primeira delas é a coleta e análise dos requisitos que o sistema deve atender. Esta é uma área tão ampla e importante, que se transformou em um campo de estudo próprio, denominado de Engenharia de Requisitos (ER) (GUPTA *et. al.* 2018; YASEEN *et. al.* 2019).

De acordo com Paetsch (2003), a ER está preocupada em identificar, modelar, comunicar e documentar e priorizar os requisitos de um sistema e os contextos em que o sistema será usado. A ER é definida como a área da Engenharia de Software (ES) que se preocupa em compreender os objetivos reais das funcionalidades do software. Assim, a ER age como uma ponte entre as reais necessidades de usuários e os demais *Stakeholders* fornecendo os insumos necessários para a solução final (NUSEIBEH e EASTERBROOK, 2000; PRESSMAN, 2011).

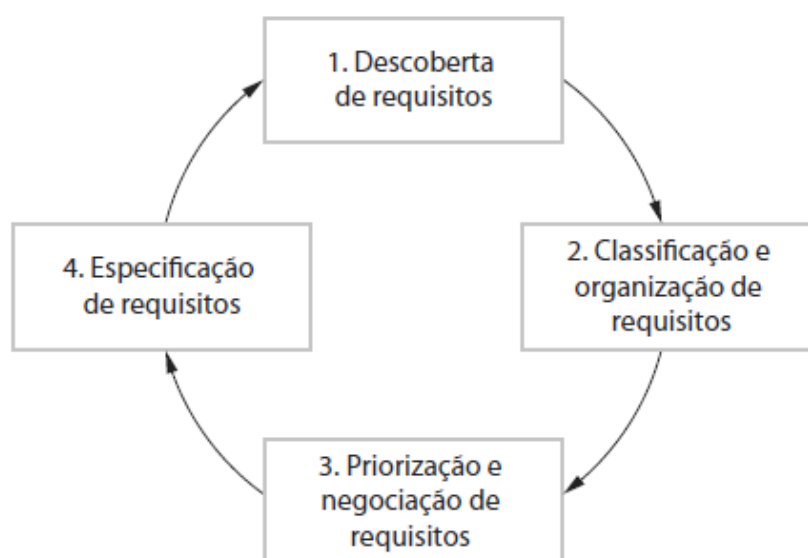
Sommerville (2011), afirma que não faz sentido falar em processo ideal sabendo que o processo de ER pode variar de uma organização para outra, pois o processo pode ser adaptado atendendo as necessidades reais de cada organização, e que de maneira generalizada a maioria dos processos de ER podem ser descritos em um modelo de atividades de alta granularidade. Além disso, o autor cita que o processo de ER possa ter suas particularidades, de maneira geral ela é subdividida em quatro fases principais, segundo Sommerville (2011), essas fases podem

ser descritas como: (i) Estudo da viabilidade; (ii) Elicitação de requisitos; (iii) Especificação de requisitos; (iv) Validação.

Sommerville (2016) apresenta um modelo de processo de elicitação de requisitos exibido na exibido na

Figura 3, em que existem quatro fases, sendo elas (i) Descoberta de requisitos; (ii) Classificação e organização de requisitos; (iii) Priorização e negociação de requisitos: (iv)Especificação de requisitos.

**Figura 3 - ER – Processo de elicitação e análise**



**Fonte: Sommerville (2016)**

Observando a Figura 3 , é possível entender que o processo é espiral e iterativo o que permite constantes feedbacks nas atividades. Além disso, possibilita que os *stakeholders* envolvidos mostrem opiniões diferentes sobre a importância e a prioridade dos requisitos o que pode ser produtivo ou gerar conflitos constantes. Portanto, conforme o desenvolvimento de software se torna mais complexo, mais cresce a importância da ER (YASEEN *et. al.* 2015), pois todo o sucesso do software depende da exatidão dos requisitos coletados nessa fase (WANG,2014, OCHODEK E KOPCZYŃSKA, 2018).

Requisitos de software podem ser definidos como uma propriedade que deve ser exibida por algo, para resolver um problema no mundo real (BOURQUE; FAIRLEY, 2014), e possuem uma função primordial no processo de software. Por apresentar a função de identificar as necessidades das partes envolvidas, definirem as funcionalidades, restrições é considerado

um fator decisivo para o sucesso ou fracasso de um projeto (ARRUDA *et al.*, 2014; YASEEN *et al.* 2015; YASEEN *et al.* 2019; WANG, 2014, OCHODEK E KOPCZYŃSKA, 2018).

Bourque e Fairley (2014) afirmam que a priorização de requisitos é necessária, como um meio de filtrar requisitos importantes, mas vai além disso, pois se faz primordial também para resolver conflitos e planejar entregas em etapas. Fato bastante comum nos dias de hoje, pois com a adoção de métodos ágeis de desenvolvimento que buscam entregar software funcionando com frequência, as entregas ocorrem em intervalos mais curtos. Nesse sentido, as técnicas de PR visam balancear o benefício do desenvolvimento de determinado requisito para o sistema com seu respectivo custo de implementação (WIEGERS, 1999).

A priorização da lista de requisitos não é uma tarefa espontânea, pois envolve vários atributos como importância dos requisitos, complexidade, custo e tempo de conclusão (ALAWNEH, 2018). Várias metodologias e abordagens têm sido propostas na literatura para apoiar a priorização de requisitos em projetos de desenvolvimento de software, fato que pode ser observado, por exemplo, em Achimugu *et al.* (2014) e Pitangueira *et al.* (2015), como: Atribuição numérica, MoScow, Top 10, AHP (*Analytic Hierarchy Process*), Modelo de Kano, Matriz GUT, Priorização Orientada a Valor (VOP), Técnica de Atribuição Numérica (NAT), Pesquisa Binária (BS), Votação Cumulativa (CV) .

Afraz (2021) afirma que um dos principais fatores no insucesso de projetos de software é a falta de atenção na priorização de seus requisitos. Nesse sentido, segundo Sufian *et al.* (2018), há lacunas como classificações simples sem valor de prioridade, ou dificuldades para atender a requisitos grandes, necessidade de testes para eficácia, complexidade computacional, dificuldade em obter consenso entre as partes, problemas de escalabilidade, possibilidade de manipulação por parte dos *Stakeholders* para atingir seus próprios objetivos, entre outras.

## 2.4 Estimativas de *software*

Estimativa de *software*, segundo Fenton e Pfleeger (1997), é uma avaliação de probabilidade e não uma certeza. Refere-se a uma previsão aproximada de tempo, recursos consequentemente de custos relacionados a um projeto.

De acordo com Rezende (1999) e Hazan (2001) as principais razões para se medir *software* são: Gerar uma *baseline* para estimativas; Verificar se as metas de produtividade e qualidade estão sendo atingidas; Avaliar as vantagens do uso de novos métodos e ferramentas de engenharia de *software*; Melhorar o relacionamento com o cliente; Auxiliar na justificativa

de solicitações de treinamento e aquisição de ferramentas; Melhorar a gerência de contratos de *software*; Reduzir o risco da empresa de gerar um cronograma inviável; e Melhorar a gerência de projetos de desenvolvimento de *software*.

Diversos estudos na engenharia de *software*, exploram a importância da precisão das estimativas como: Kazemifard (2011), Hamdy (2014), Amazal et al. (2014), entre outros. Estudos que exploram continuamente técnicas de estimativa de *software*, seja individualmente ou combinadas, sempre em busca de elevar os níveis de precisão. Dentre as questões que são levantadas durante o projeto de desenvolvimento de um *software*, Salinas (2014), destaca as seguintes: Quanto vai custar? Quando ficará pronto? Quanto esforço será necessário investir? Quais funcionalidades ou requisitos devem ser desenvolvidos? Respostas às perguntas acima elencadas, são objetos para a concepção de um produto de *software* de sucesso, contudo, a omissão à elas, ou à algumas delas na gestão do projeto de desenvolvimento, costumam gerar impactos negativos para o produto final. Além disso, tudo que esteja no caminho entre entregar um produto que satisfaça o cliente e a sua entrega propriamente dita, é então considerada uma atividade de desperdício (POPPENDIECK, 2003). Estimativas precisas são essenciais para a empresa e os clientes porque podem ajudar a empresa a classificar, priorizar e determinar recursos a serem comprometidos com o projeto (NISAR *et al.* 2008).

Dada a relevância da realização de estimativas, muitos modelos foram propostos pelos pesquisadores. (JØRGENSEN e SHEPPERD 2007; TRENDOWICZ *et al.* 2008). Esses modelos são classificados em duas principais categorias: Modelos baseados em algoritmos e modelos não baseados em algoritmos (SALINAS, 2014). Modelos baseados em algoritmos utilizam abordagens matemáticas para calcular o esforço e, consecutivamente, realizar a priorização das demandas.

Estas soluções envolvem modelos algorítmicos como, por exemplo, COCOMO (BOEHM 1984, BOEHM e CLARK 1995) e SLIM (PUTNAM,1978), Julgamento de Especialistas (TSUNODA,2012) (JØRGENSEN, SHEPPERD 2007) (JØRGENSEN, 2004), Regressão Estatística (NGUYEN, STEECE, BOEHM 2008) Estimativa por Analogia (SHEPPERD, SCHOFIELD,1997) dentre outras.

Conforme afirma o CMUSEI (2006) (*Carnegie Mellon University - Software Engineering Institute*) no CMMI (*Capability Maturity Model Integration*), a estimativa de custo de um projeto deve ser baseada no esforço necessário para sua construção, sendo por sua vez, o esforço calculado em função do tamanho estimado para o *software* a ser desenvolvido. De maneira que tudo se baseia na acurácia da estimativa de tamanho. A estimativa de custo é um



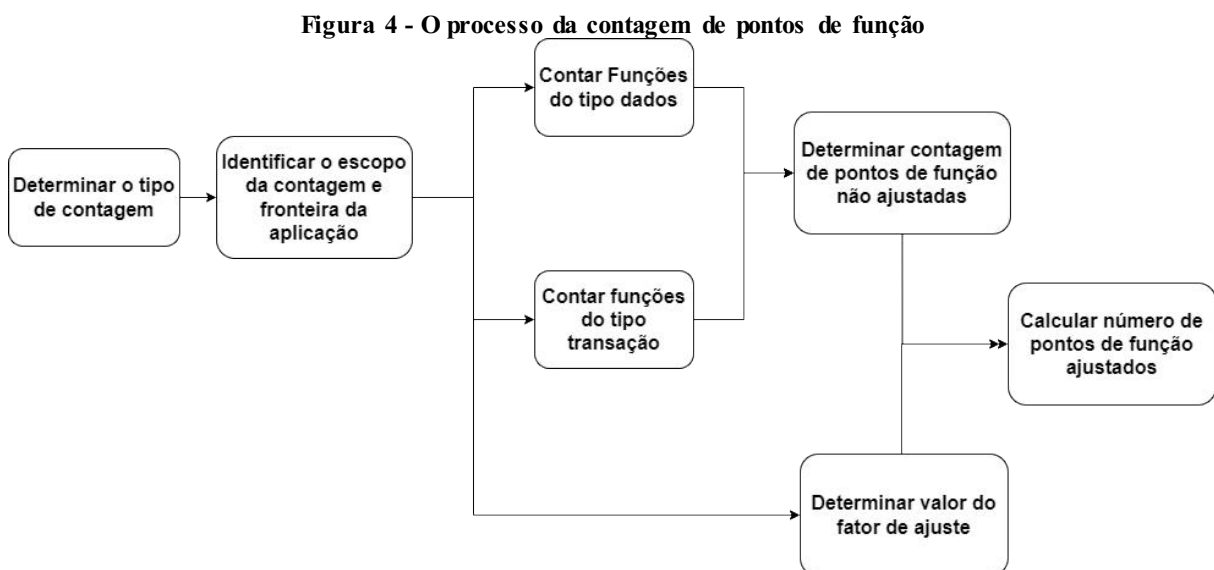
processo ou uma aproximação do custo provável de um produto, programa ou projeto, calculado com base nas informações disponíveis (SEHRA, 2018).

O Modelo de Custo Construtivo (COCOMO) é um modelo bastante conhecido no cenário da engenharia de *software* (PRESSMAN, 2011; SOMMERVILLE, 2011). O modelo COCOMO, foi introduzido por Boehm (1981). Ele se tornou um dos modelos de estimativa de custo de *software* mais amplamente usados na indústria de *software*. Para oferecer suporte a novos ciclos de vida e capacidade, evoluiu para um modelo de estimativa mais abrangente, denominado COCOMO II (BOEHM, 2000; BOEHM, 2017)

Pontos de função foi proposta em 1979 por Alan Albrecht da IBM (ALBRECHT, 1979) sendo atualmente regulamentada pelo *International Function Point Users Group* (IFPUG) e foi oficializado através do padrão internacional ISO/IEC 20926 de 2002. O IFPUG publica o *Counting Practices Manual* (Manual de Práticas de Contagem) que estabelece os padrões para o cálculo dos pontos de função.

Segundo Vazquez *et al.* (2011), a Análise de Pontos de Função (APF) é um método para a medição do desenvolvimento de *software*, buscando estabelecer uma medida de tamanho do *software* em Pontos de Função, com base na quantificação da funcionalidade solicitada e fornecida, sob o ponto de vista do usuário. Dessa forma, a APF tem como objetivo medir o que o *software* faz, por meio de uma avaliação padronizada dos requisitos de negócio do sistema.

O processo para contagem de pontos de função abrange sete etapas relacionadas entre si (VAZQUEZ, SIMÕES, ALBERT, 2008), como mostra o diagrama da Figura 4.



Fonte: Adaptado de Vazquez, Simões e Albert (2011).

Contudo, Pressman (2008) define de forma genérica, que o processo de cálculo de APFs é executado em três etapas: (i) determinação dos pontos de função não ajustados ou pontos de função brutos; (ii) cálculo do valor de ajuste; e (iii) cálculo final dos pontos de função.

Na etapa um são avaliados aspectos do sistema e, de acordo com Jones (2008), Albrecht considerou 5 itens que segundo ele poderiam ser enumerados com precisão, são aspectos externos visíveis de *software*, Esses componentes definem os pontos de função não ajustados e seriam os seguintes: (i) as entradas para a aplicação; (ii) as saídas fornecidas pela aplicação; (iii) as consultas dos usuários; (iv) os arquivos de dados a serem atualizados pela aplicação; e (v) as interfaces para outras aplicações.

Já na segunda etapa são calculados os fatores de ajuste, usados para adequar a contagem. E na terceira é aplicado os fatores de ajustes juntamente com os pontos de função não ajustados para chegar aos pontos de função ajustados.

Para Braga (1996) para o cálculo do fator de ajuste deve-se: (i) Atribuir um nível de influência que pode variar de 0 (nenhuma influência) até 5 (influência alta) para cada uma das catorze características gerais do sistema na aplicação que está sendo contada. O fator de ajuste, quando aplicado aos pontos de função não ajustados, pode produzir uma variação de cerca de 35%. (ii) Para cada característica é atribuído um peso de 0 (zero) a 5 (cinco), de acordo com o nível de influência na aplicação. O nível de influência geral é obtido pelo somatório do nível de influência de cada uma do catorze características (iii) Calcular o FA (Fator de Ajuste para permitir o cálculo dos pontos de função ajustados através da fórmula: Fator de ajuste =  $0,65 + (\text{Nível de Influência Geral} * 0,01)$ . (iv) Calcular os PFA (Pontos de Função Ajustados) através do seguinte cálculo: PF ajustados = PF não ajustados \* Fator de ajuste.

As catorze características do sistema definidas pela IFPUG que determinam o fator de ajuste são apresentadas no Quadro 4.

**Quadro 4 - Fatores de Ajuste da APF**

<b>Fator de Ajuste</b>	<b>Pergunta de orientação</b>
Comunicação de dados	É requerida comunicação de dados?
Funções distribuídas	Existem funções de processamento distribuído?
Desempenho	O desempenho é crítico?
Configuração do equipamento	O sistema funcionará num sistema operacional existente e intensamente utilizado?
Volume de transações	Qual o volume de transações realizadas?
Entrada de dados online	São requeridas entrada de dados on-line?
Eficiência com o usuário	As entradas on-line requerem que as transações de entrada sejam construídas com várias telas e operações?
Atualização online	Os arquivos são atualizados on-line?

Processamento complexo	Entradas, saídas, arquivos e consultas são complexos?
Reusabilidade	O código é projetado para ser reusável?
Facilidade de implementação	O processamento interno é complexo?
Facilidade de operação	A conversão e a instalação estão incluídas no projeto?
Múltiplos locais	O sistema é projetado para múltiplas instalações em diferentes organizações?
Facilidade de mudanças	A aplicação é projetada de forma a facilitar mudanças e o uso pelo usuário?

Fonte: Adaptado de VAZQUEZ et al. (2011)

De acordo com VAZQUEZ *et al.* (2011) embora o padrão IFPUG tenha sido aprovado pela ISO, as duas últimas etapas não aderem ao padrão de medição funcional, pois contém requisitos tecnológicos e de qualidade.

#### 2.4.1 Direcionadores fatores de custo ou ajuste

Modelos de estimativa de custo de *software* utilizam o termo *cost driver* para identificar fatores que influenciam os custos. Esse termo veio das definições associadas à abordagem da atribuição de custos, no contexto das práticas mais sofisticadas de custeio, particularmente o ABC – *Activity-Based Costing*, em que o rastreamento de custos atribuídos aos objetos de custos é elemento chave do processo de gestão. Nesse cenário, *cost driver* são geralmente traduzidos para português como direcionadores de custos (BRIMSON, 1991). Porém, ao analisar as literaturas referente aos modelos de estimativas de custo (BOEHM 1984, BOEHM e CLARK 1995; TSUNODA,2012; JØRGENSEN, SHEPPERD 2007), são encontradas variações para o termo, como multiplicadores e fatores de ajuste.

Alguns modelos de estimativa de custos convencionais têm parâmetros direcionadores de custos que cobrem os níveis de capacidade do pessoal; níveis de experiência de domínio, tecnologia e ferramentas; níveis de continuidade de pessoal e coesão da equipe (BOEHM, 2017). Assim como estimativas de tamanho como a Pontos por Função, que utiliza fatores de ajuste para melhor estimar. O modelo COCOMO intermediário (BOEHM, FAIRLEY, 2000) considera como fatores de ajuste, um conjunto de atributos direcionadores do custo agrupados em quatro categorias: produto, hardware, pessoal e projeto. Cada um desses atributos deve ser classificado de acordo com uma escala que varia de “muito baixo” a “extremamente elevado” (em importância e valor).

Para este trabalho, para a elaboração do modelo proposto, foram selecionados o Modelo COCOMO II e a técnica de análise de pontos de função como base para obtenção dos fatores de impacto (direcionadores de custo e fatores de ajuste) pelo maior suporte teórico encontrado na literatura, pelas opiniões favoráveis encontradas em publicações e pela abrangência fornecida, além da equivalência de fatores entre os modelos. A seguir são apresentados no Quadro 5 os direcionadores de custo do modelo COCOMO II.

**Quadro 5 - Direcionadores de Custo – COCOMO II**

<b>Categorias</b>	<b>Direcionadores</b>	<b>Descrição</b>
Fatores de Produtos	Confiabilidade de <i>software</i> exigida	Refere-se a medida de até que ponto o <i>software</i> deve executar sua função pretendida durante um período de tempo. Se o efeito de uma falha de <i>software</i> for apenas um pequeno inconveniente, a confiabilidade é baixa. Se uma falha arriscar a vida humana, então a confiabilidade é muito alta.
	Dimensão da base de dados	Essa medida tenta capturar o efeito que os requisitos de grandes volumes de dados têm no desenvolvimento do produto. A classificação é determinada calculando D / P ( <i>Data Base Size, Program size</i> ). O motivo do tamanho do banco de dados é importante considerá-lo devido ao esforço necessário para gerar os dados de teste que serão usados para exercitar o programa.
	Complexidade do produto	A complexidade é dividida em cinco áreas: operações de controle, operações computacionais, operações dependentes de dispositivos, operações de gerenciamento de dados e operações de gerenciamento de interface do usuário. Selecione a área ou combinação de áreas que caracterizam o produto ou um subsistema do produto. A classificação de complexidade é a média subjetiva ponderada dessas áreas.
	Reutilização necessária	Esse item é responsável pelo esforço adicional necessário para construir componentes destinados à reutilização em projetos atuais ou futuros. Esse esforço é consumido com a criação de um design mais genérico de <i>software</i> , documentação mais elaborada e testes mais extensos para garantir que os componentes estejam prontos para uso em outros aplicativos.
	Documentação compatível com as necessidades do ciclo de vida	Escala de classificação é avaliada em termos da adequação da documentação do projeto às suas necessidades do seu ciclo de vida.
Fatores de plataforma	Restrição de tempo de execução	Esta é uma medida da restrição de tempo de execução imposta a um sistema de <i>software</i> . A classificação é expressa em termos da porcentagem de tempo de execução disponível que se espera que seja usado pelo sistema ou subsistema que consome o recurso de tempo de execução.
	Restrição de armazenamento principal	Essa classificação representa o grau de restrição de armazenamento principal imposto a um sistema ou subsistema de <i>software</i> .
	Volatilidade da plataforma	Plataforma é usado aqui para significar o complexo de <i>hardware</i> e <i>software</i> que o produto de <i>software</i> irá executar em suas tarefas.

		Se o <i>software</i> a ser desenvolvido for um sistema operacional, a plataforma é o <i>hardware</i> do computador. Se um sistema de gerenciamento de banco de dados for desenvolvido, a plataforma é o <i>hardware</i> e o sistema operacional. A plataforma inclui quaisquer compiladores ou montadores de suporte ao desenvolvimento do sistema de <i>software</i> .
Fatores Pessoal	de Capacidade analista do	Analistas são pessoas que trabalham com requisitos, design de alto nível e design detalhado. Os principais atributos que devem ser considerados nesta classificação são a capacidade de Análise e Design, eficiência e meticulosidade, e a capacidade de comunicar e cooperar. A classificação não deve considerar o nível de experiência do analista; que é classificado com Experiência de aplicações.
	de Capacidade programador do	A avaliação deve ser baseada na capacidade dos programadores como uma equipe, e não como indivíduos. Os principais fatores que devem ser considerados na classificação são habilidade, eficiência e meticulosidade, e a habilidade de comunicar e cooperar. A experiência do programador não deve ser considerada aqui; é classificado com AEXP.
	de Experiência aplicações de	Essa classificação depende do nível de experiência em aplicações da equipe do projeto que desenvolve o sistema ou subsistema de <i>software</i> . As classificações são definidas em termos do nível equivalente de experiência da equipe do projeto com este tipo de aplicações.
	de Experiência plataforma de	Refere-se ao uso de plataformas mais poderosas, incluindo interface de usuário mais gráfica, banco de dados, rede e <i>middleware</i> distribuído
	de e com Linguagem experiência ferramentas	Esta é uma medida do nível de experiência em linguagem de programação e ferramenta de <i>software</i> da equipe do projeto que desenvolve o sistema ou subsistema de <i>software</i> . O desenvolvimento de <i>software</i> inclui o uso de ferramentas que executam requisitos e representação e análise de design, gerenciamento de configuração, extração de documentos, gerenciamento de biblioteca, estilo e formatação do programa, verificação de consistência, etc.
	de Continuidade pessoal de	A escala de classificação de pessoal é em termos de rotatividade anual de pessoal do projeto: de 3%, muito alto, a 48%, muito baixo.
Fatores Projeto	de Uso de ferramentas de desenvolvimento de <i>software</i>	A classificação da ferramenta varia de edição e código simples, muito baixo, a ferramentas de gerenciamento de ciclo de vida integrado, muito alto.
	do Desenvolvimento Multisite	A determinação de sua classificação envolve a avaliação e a média de dois fatores: colocação de site (de totalmente colocado a distribuição internacional) e suporte de comunicação (de correio de superfície e algum acesso portelefone a multimídia totalmente interativa).
	de Cronograma Desenvolvimento Requerido	Mede a restrição de cronograma imposta à equipe do projeto que desenvolve o <i>software</i> . As classificações são definidas em termos de porcentagem de alongamento ou aceleração do cronograma em relação a um cronograma nominal para um projeto que exige um determinado esforço.

Fonte: adaptado do Manual de definição do modelo COCOMO II (2000).

No Quadro 6, os fatores de ajuste do modelo de estimativa de tamanho Análise de Ponto por Função.

**Quadro 6 - Fatores de Ajuste – Análise de Pontos por Função**

<b>Fatores de Ajuste</b>	<b>Descrição</b>
Comunicações de dados	Quantas facilidades de comunicação existem para auxiliar na transferência ou troca de informações com o aplicativo ou sistema? Comunicação de dados descreve o grau em que a aplicação se comunica diretamente com o processador. Se o aplicativo é processamento em lote ou se possui processamento remoto, se suporta mais que um tipo de protocolo de comunicação.
Processamento de dados distribuído	Diz respeito a como os dados distribuídos e as funções de processamento são tratados. Dados distribuídos ou funções de processamento são uma característica do sistema dentro dos limites do sistema.
Desempenho	Desempenho descreve o grau em que as considerações de desempenho de rendimento e tempo de resposta influenciaram o desenvolvimento do aplicativo. O usuário necessita tempo de resposta ou taxa de transferência, seja na resposta ou na taxa de transferência, influenciam (ou irão influenciar) o design, o desenvolvimento, a instalação e o suporte do aplicativo. Se o tempo de resposta ou transferência é crítico em determinados horários, é necessário uso de tarefas de análise de desempenho em alguma fase na construção do sistema. O tempo de resposta normalmente está relacionado ao processamento interativo; Taxa de transferência relaciona-se ao processamento em lote
Configuração muito usada. (Uso do sistema)	Avalia o grau em que o sistema necessita ser projetado para compartilhar recursos de processamento. Descreve o grau em que as restrições de recursos do computador influenciam o desenvolvimento do sistema, quão fortemente utilizada é a plataforma de <i>hardware</i> atual onde o sistema será executado.
Taxa de transação	Avalia a quantidade de transações simultâneas esperada. Os usuários podem necessitar o que consideram um tempo de resposta normal, mesmo durante horários de pico de volume.
Entrada de dados <i>online</i>	Descreve o grau em que os dados são inseridos ou recuperados por meio de transações interativas. Interface do usuário on-line para entrada de dados, funções de controle, relatórios e consultas são fornecidos no sistema. Qual porcentagem das informações é inserida <i>On-Line</i> ?
Eficiência do usuário final	Descreve o grau de consideração pelos fatores humanos e a facilidade de uso pelo usuário do sistema. Se faz uso de menus, documentação <i>on-line</i> , se é disponível em várias línguas.
Atualização Online	Descreve o grau em que os arquivos lógicos internos (tabelas) são atualizados <i>on-line</i>
Processamento Complexo	Descreve o grau de processamento lógico, de segurança ou matemático complexo.
Reutilização	Descreve o grau em que o aplicativo e o código no aplicativo foram especificamente projetados, desenvolvidos e com suporte para serem usados em outros ou módulos ou sistemas
Facilidade de instalação	A facilidade de instalação descreve o grau em que a conversão de ambientes anteriores influenciou o desenvolvimento do sistema. Um plano de conversão e instalação e / ou ferramentas de conversão serão fornecidos e testados durante a fase de teste do sistema?
Facilidade operacional	Descreve o grau em que o aplicativo atende aos aspectos operacionais, como processos de inicialização, backup e recuperação. Diz respeito a quão eficazes e / ou automatizados são os procedimentos de inicialização, backup e recuperação.

Vários sites	Descreve o grau que a aplicação foi projetada, desenvolvida e com suporte especificamente para ser instalada em vários locais para várias organizações e diferentes ambientes de <i>hardware</i> e <i>software</i> .
Facilite mudança	a Descreve o grau em que a aplicação foi desenvolvida para facilitar a modificação da lógica de processamento ou da estrutura de dados.

Fonte: adaptado do Manual de Práticas de Contagem de Pontos de Função (IFPUG,2005).

A Revisão Sistemática de Literatura (RSL) mostrou que cada modelo de estimativa de custo de desenvolvimento de *software* possui termos distintos para os fatores envolvidos, embora muitas vezes o objetivo seja semelhante. Além disso, geralmente possuem seus critérios para melhorar a assertividade.

As nomenclaturas utilizadas para os direcionadores de custo variam conforme o modelo e conforme autores, porém, neste trabalho, para facilitar o entendimento, os termos serão chamados de Dimensão e Fatores de Impacto. Sendo a Dimensão os grupos e os fatores de impacto ficam relacionados aos grupos.

A seleção dos Fatores de Impacto é relevante tendo em vista que há uma quantidade considerável deles. Além disso, conforme levantamento bibliográfico, embora os fatores de impacto variem de um modelo de estimativa para outro, eles possuem várias similaridades. Sendo assim, para a utilização no método, foram avaliados os fatores de impacto de cada dimensão, e na sequência foram agrupados conforme seu objetivo para então realizar a remoção das duplicações.

Dos 31 fatores de impactos obtidos dos Modelos COCOMO II (Quadro 5) e Análise de Pontos por Função (Quadro 6), após a avaliação, restaram 20 conforme Quadro 7.

**Quadro 7 - Atributos e fatores de impacto definidos para o modelo**

Dimensão	Fatores de Impacto	Modelo Original
Técnica	Restrição de tempo de execução	COCOMO II
	Restrição de armazenamento principal	COCOMO II
	Volatilidade da plataforma	APF e COCOMO II
	Taxa de transação	APF
De Produto	Confiabilidade de <i>software</i> exigida	COCOMO II
	Dimensão da base de dados	COCOMO II
	Complexidade do produto	APF e COCOMO II
	Reutilização necessária	APF e COCOMO II
	Documentação compatível com as necessidades do ciclo de vida	COCOMO II
De Pessoal	Eficiência do usuário final	APF
	Capacidade do analista	COCOMO II
	Capacidade do programador	COCOMO II
	Experiência de aplicações	COCOMO II
	Experiência de plataforma	COCOMO II

	Linguagem e experiência com ferramentas	COCOMO II
	Continuidade de pessoal	COCOMO II
De Projeto	Uso de ferramentas de <i>software</i>	COCOMO II
	Desenvolvimento Multisite	APF e COCOMO II
	Cronograma de Desenvolvimento Requerido	COCOMO II
	Processamento de dados distribuído	APF

Fonte: Autoria Própria (2020).

Os vinte fatores de impacto apresentados no Quadro 7 serão utilizados na avaliação dos requisitos para o desenvolvimento do modelo.

## 2.5 Métodos multicritério de apoio à tomada de decisão

Os métodos multicritério de apoio à tomada de decisão são técnicas para auxiliar o agente decisor (que vai de um indivíduo, até a comitês de especialistas ou lideranças) para a tomada de decisão sobre problemas complexos, selecionando, avaliando ou ordenando alternativas conforme diferentes pontos de vista e dentro de critérios pré-estabelecidos. Constituindo assim, um conjunto de ferramentas para abordar difíceis decisões auxiliando os tomadores de decisão em situações de incerteza e complexidade, que por diversas vezes são conflitantes (GOMES; GOMES, 2019; GONÇALVES; MACHADO, 2015; WANG, 2010).

Conforme Barfod (2012), a tomada de decisão multicritérios ou *MultiCriteria Decision Making* (MCDM) é uma maneira de lidar com problemas complexos, dividindo-os em partes menores, em seguida ponderando os procedimentos e julgamentos dos componentes menores, depois as peças são remontadas e apresentando um quadro geral aos decisores. Tchembra (2009), afirma que os métodos multicritérios podem ser diferenciados pela capacidade de considerar tanto critérios quantitativos como qualitativos e, permitem a análise da decisão permitindo ainda testar sua confiabilidade.

Já na metodologia de apoio à decisão multicritério ou *MultiCriteria Decision Aiding* (MCDA), a problemática refere-se ao modo como o apoio à decisão é considerado. São quatro os grupos de problemas de referência para MCDA:

- (i) Escolha - problemática P.α: ajuda a escolher e desenvolve um processo de seleção;
- (ii) Classificação - problemática P.β: ajuda a ordenar as ações de acordo com as normas ou com a compilação de um procedimento de atribuição;



(iii) Ordenação - problemática P.γ: auxilia as ações de classificação em ordem decrescente de preferência; e

(iv) Descrição - problemática P.δ: ajuda a descrever as ações e suas consequências de maneira formalizada e sistemática ou desenvolver um procedimento cognitivo. (ROY,1996; CARIGNANO, 2005).

Os métodos que se utilizam da problemática de ranking P.γ, serão abordados neste trabalho. Se enquadram nessa problemática, métodos como o TOPSIS, AHP, VIKOR, entre outros.

Na engenharia de *software*, a metodologia multicritérios tem sido aplicada em várias áreas conforme mostram alguns estudos: na priorização de requisitos em Karlsson (1996), através do método AHP; para selecionar técnicas de testes através do uso dos métodos AHP e TOPSIS (VICTOR, 2011); Para avaliação de qualidade utilizando a metodologia multicritério *Fuzzy*, por Challa (2011). Para a seleção de um modelo de ciclo de vida de *software* adequado, por Hicdurmaz (2012); Para seleção de desenvolvedores para tipos específicos de *softwares*, por Hota (2012), fez uso do *Fuzzy-AHP*; Para a seleção da metodologia de desenvolvimento ágil de *software*, mais adequada para pequenas e médias empresas com base no método multicritério SMARTER, por Silva (2016); Para estimativa de esforço na escassez de dados por Sehra (2017), com o uso do *Fuzzy-AHP*.

### 2.5.1 Método AHP

De acordo com Saaty (2006), a análise multicritério deveria ser uma ciência de medição baseada em matemática, psicologia e filosofia. Inspirado por essas ideias, ele desenvolveu em 1977 o método de tomada de decisão ao Processo de Análise Hierárquica ou *Analytic Hierarchy Process* (AHP) – por meio de comparações entre cada alternativa e cada uma das demais, usando uma escala de medidas capaz de refletir o grau da preferência dos decisores por uma das duas alternativas.

Com a Escala Fundamental de Saaty (1977), conforme Quadro 8 é possível fornecer valores para os critérios de acordo com a preferência do decisor.

**Quadro 8 - Escala Fundamental método AHP**

<b>Intensidade de Importância</b>	<b>Definição</b>	<b>Explicação</b>
1	Mesma importância	As duas atividades contribuem igualmente para o objetivo

3	Importância pequena de uma sobre a outra	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Importância grande ou essencial	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra.
7	Importância muito grande ou demonstrada	Uma atividade é muito fortemente favorecida em relação à outra; sua dominação de importância é demonstrada na prática
9	Importância absoluta	A evidência favorece uma atividade em relação à outra com o mais alto grau de certeza.
2, 4, 6, 8	Valores intermediários entre os valores adjacentes	Quando se procura uma condição de compromisso entre duas definições
Recíprocos dos valores acima de zero	Se a atividade $i$ recebe uma das designações diferentes acima de zero, quando comparada com a atividade $j$ , então $j$ tem o valor recíproco quando comparada com $i$ .	Uma designação razoável
Racionais	Razões resultantes da escala	Se a consistência tiver de ser forçada para obter valores numéricos $n$ , somente para completar a matriz

Fonte: Traduzido de Saaty (1977)

Primeiramente são realizadas as comparações segundo cada critério. Posteriormente, esses resultados são agregados a fim de se obter uma única ordenação das alternativas. O ferramental utilizado são vetores e matrizes especiais, cálculo de inversa das matrizes, autovalores e auto vetores, entre outros

### 2.5.2 Conjuntos Difusos (*Fuzzy Sets*) e números *Fuzzy*

O termo *fuzzy* ao ser traduzido, tem o significado de vago, indefinido, incerto, difuso. A lógica *fuzzy* é um raciocínio que classifica em números um determinado cenário, trabalhando com variáveis incertas, para permitir o trabalho computacional (SHAW; SIMÕES,1999). Assim, a lógica *fuzzy* generalizou a lógica clássica, de modo que a sua implantação possa reduzir o projeto a um ponto entre  $[0, 1]$ , passando os problemas intratáveis a serem factíveis de solução.

A noção dos conjuntos *fuzzy* provê um ponto de partida para a construção de um quadro conceitual que se assemelha à estrutura usada nos conjuntos comuns, porém mais robusto com maior alcance e aplicabilidade, especialmente nos campos de classificação padrão e processamento de informações. Esse quadro oferece maneiras de lidar com os problemas em que a fonte de imprecisão é a ausência de critérios bem definidos em uma classe e a ausência de variáveis aleatórias (ZADEH, 1965).

**Definição 1:** Um conjunto *fuzzy*  $\tilde{A}$  em um universo de discurso  $X$  é caracterizado por uma função de pertinência  $\mu_{\tilde{A}}(x)$  que associa a cada elemento  $x$  em  $X$  um número real no intervalo  $[0, 1]$ . O valor da função  $\mu_{\tilde{A}}(x)$  é denominado o grau de pertinência de  $x$  em  $\tilde{A}$  (KAUFMANN; GUPTA, 1991).

**Definição 2:** Um número *fuzzy* trapezoidal positivo (PTFN)  $\tilde{n}$  pode ser definido como  $(n_1, n_2, n_3, n_4)$  mostrado na Figura 4. A função de pertinência  $\mu_{\tilde{A}}(x)$  é definida como (Kaufmann e Gupta, 1991).

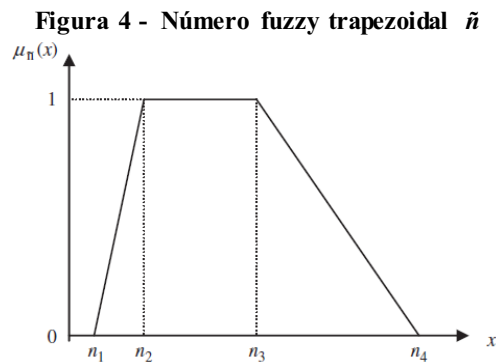
Para um número *fuzzy* trapezoidal  $\tilde{n} = (n_1, n_2, n_3, n_4)$ , se  $n_2 = n_3$ , então  $\tilde{n}$  é chamado de número *fuzzy* triangular. Um número não *fuzzy*  $r$  pode ser expresso como  $(r, r, r, r)$ . Pelo princípio de extensão (DUBOIS; PRADE, 1980), a soma difusa  $\oplus$  e a subtração difusa  $\ominus$  de quaisquer dois números difusos trapezoidais também são números difusos trapezoidais; mas a multiplicação  $\otimes$  de quaisquer dois números difusos trapezoidais é apenas um número difuso trapezoidal aproximado. Dados quaisquer dois números nebulosos trapezoidais positivos,  $\tilde{m} = (m_1, m_2, m_3, m_4)$  e  $\tilde{n} = (n_1, n_2, n_3, n_4)$  e um número real positivo  $r$ , algumas operações principais de números *fuzzy*  $\tilde{m}$  e  $\tilde{n}$  podem ser expressas do seguinte modo:

$$\tilde{m} \oplus \tilde{n} = [m_1 + n_1, m_2 + n_2, m_3 + n_3, m_4 + n_4], \quad (1)$$

$$\tilde{m} \ominus \tilde{n} = [m_1 - n_4, m_2 - n_3, m_3 - n_2, m_4 - n_1], \quad (2)$$

$$\tilde{m} \otimes r = [m_1 r, m_2 r, m_3 r, m_4 r], \quad r \geq 0, \quad (3)$$

$$\tilde{m} \otimes \tilde{n} \cong [m_1 n_1, m_2 n_2, m_3 n_3, m_4 n_4], \quad (4)$$



**Definição 3:** Uma variável linguística é uma variável cujos valores são expressos em termos linguísticos. O conceito de variável linguística é útil para lidar com situações que são

muito complexas ou não bem definidas para serem razoavelmente descritas em expressões quantitativas convencionais (Zimmermann, 1991; GLIGORIC et al., 2010). Por exemplo, “peso” é uma variável linguística cujos valores são muito baixo, baixo, médio, alto, muito alto, etc. Os números difusos também podem representar esses valores linguísticos.

**Definição 4:** Se  $\tilde{m} = (m_1, m_2, m_3, m_4)$  e  $\tilde{n} = (n_1, n_2, n_3, n_4)$  são dois números *fuzzy* trapezoidais. Assim, a distância entre eles pode ser calculada usando o método do vértice como (Chen, 2000):

$$d_v(\tilde{m}, \tilde{n}) = \sqrt{\frac{1}{4}[(m_1 - n_1)^2 + (m_2 - n_2)^2 + (m_3 - n_3)^2 + (m_4 - n_4)^2]} \quad (5)$$

O método do vértice é um método eficaz e simples para calcular a distância entre dois números difusos trapezoidais ou triangulares. De acordo com o método do vértice, dois números difusos trapezoidais  $\tilde{m}$  e  $\tilde{n}$  são idênticos se e somente se  $d_v(\tilde{m}, \tilde{n}) = 0$  em que sejam  $\tilde{m}$ ,  $\tilde{n}$  e  $\tilde{p}$ , três números difusos. O número  $\tilde{n}$ , está mais próximo do número  $\tilde{m}$  do que o número  $\tilde{p}$ , se e somente se  $d_v(\tilde{m}, \tilde{n}) < d_v(\tilde{m}, \tilde{p})$  (Chen, 2000).

### 2.5.3 Fuzzy TOPSIS

O método *Fuzzy* TOPSIS, é uma versão do método TOPSIS para tomada de decisão em cenários de incerteza na qual valores numéricos apresentam limitações. Neste método, as pontuações das alternativas e o peso dos critérios de decisão são definidos como variáveis linguísticas. Existem diferentes versões do método *Fuzzy* TOPSIS. Zadeh (1973) define variável linguística uma variável que os valores são sentenças em linguagem natural ou artificial que permite distinguir as qualificações por meio de faixas de graduação.

O *Fuzzy* TOPSIS, baseia se na escolha das melhores alternativas em relação a sua aproximação com a solução ideal positiva e maior distanciamento da solução ideal negativa. O que diferencia, nesse método em relação ao TOPSIS, é a lógica que permeia seus procedimentos. Há uma diversidade de métodos e abordagens utilizadas na adaptação do TOPSIS a lógica difusa, com normalização, sem normalização, com pesos na forma de números *fuzzy* triangular ou trapezoidal ou real ou ainda pesos na forma de número real, com número *fuzzy* trapezoidal.

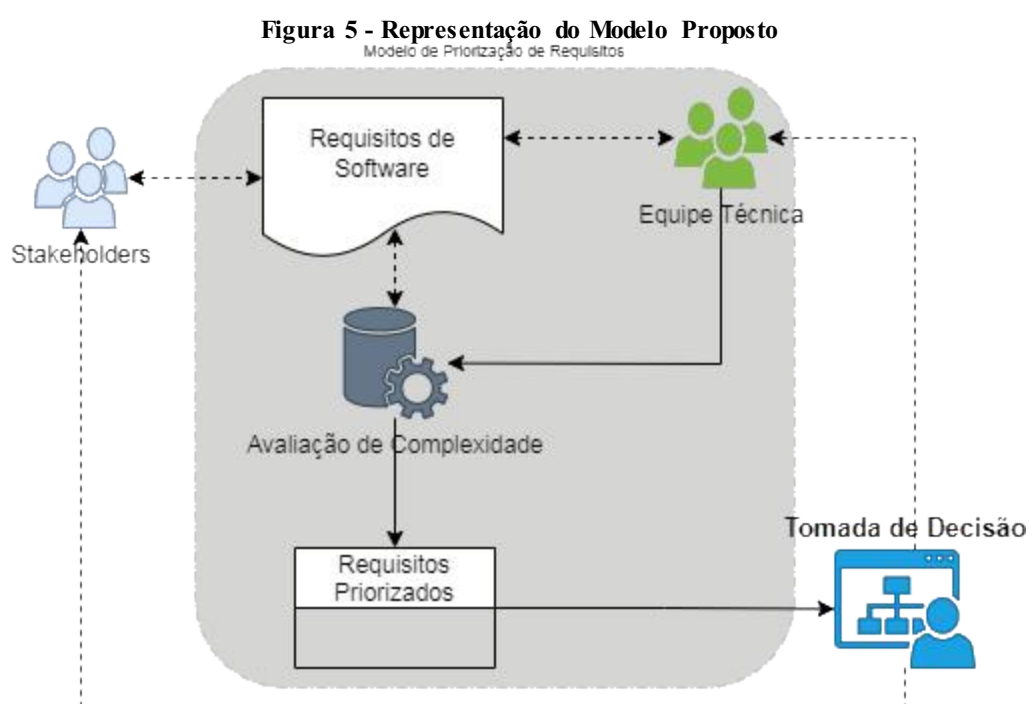
Em relação a outros modelos de análise multicritério, o método *Fuzzy* TOPSIS se torna o mais atrativo na realização para a aplicação da avaliação de complexidade dos fatores em relação aos requisitos de *software*, pois possibilita um número não limitado de alternativas e de critérios, além de ser recomendado para dar apoio a situações de decisão diante incertezas e por auxiliar o monitoramento do desempenho dos critérios (JUNIOR E CARPINETTI; 2015).

### 3 METODOLOGIA

Neste capítulo é apresentada a metodologia utilizada para o desenvolvimento do Modelo de Apoio ao Gerenciamento de Projetos de Desenvolvimento de *Software Web*. Composto pelas seguintes fases: (i) descrição do Modelo de Apoio ao Gerenciamento de Projetos de Desenvolvimento de *Software*, e suas etapas; e (ii) descrição da aplicação do modelo proposto.

#### 3.1 Modelo multicritério de priorização de requisitos de *software*

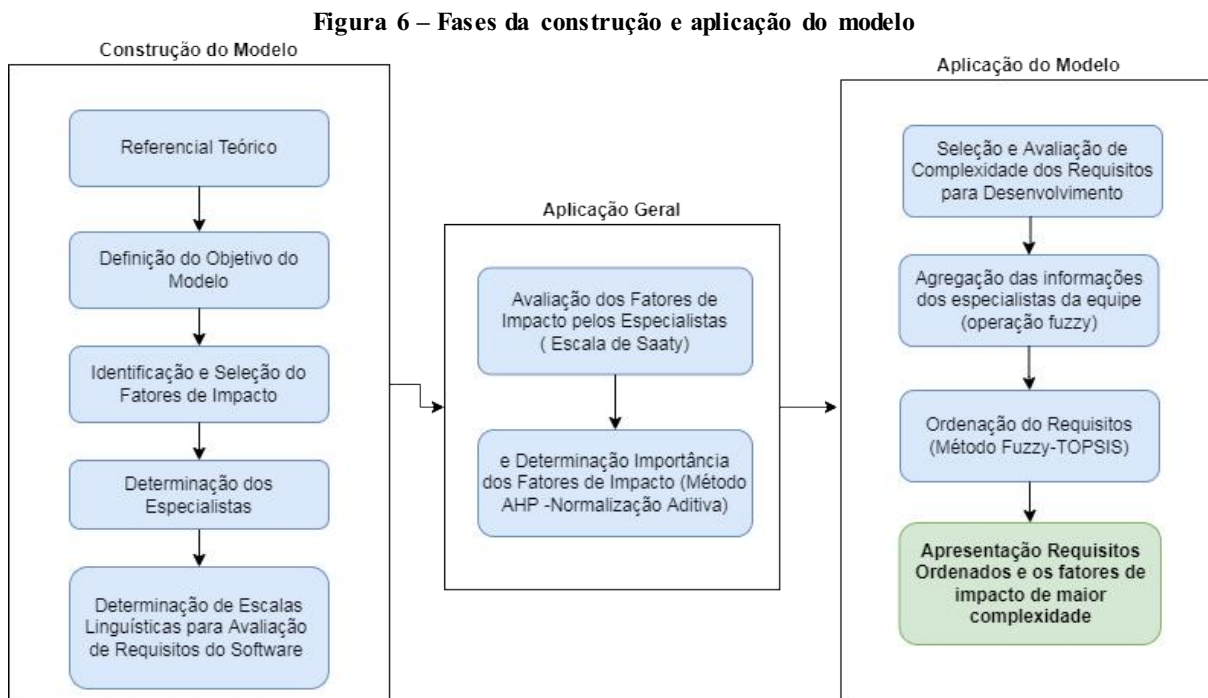
O Modelo Multicritério de Priorização de Requisitos de *Software* proposto nesta dissertação, tem como objetivo avaliar qualitativamente os fatores de impacto em relação a sua complexidade para os requisitos de acordo com a visão da equipe técnica sendo apresentado de forma simplificada na Figura 5.



Fonte: Autoria própria (2021)

O modelo apresentado na Figura 5 envolve a equipe técnica, os requisitos de *software* que representam as necessidades dos *Stakeholders* e ao final fornece insumos para os gestores tomarem decisões de forma mais assertiva. A elaboração do modelo apresentou as seguintes

fases: Construção do Modelo; Aplicação Geral e Aplicação do Modelo que são apresentadas na Figura 6.



Fonte: Autoria própria (2021).

Conforme é possível observar na Figura 6, houve 3 fases até a finalização do modelo proposto. As fases são explicadas nas seções a seguir.

### 3.2 Estruturação do modelo

Na fase de construção do modelo, houveram as seguintes etapas: (i) referencial teórico; (ii) definição do objetivo do modelo; (iii) identificação e seleção dos fatores de impacto; (iv) determinação dos especialistas; (v) determinação das escalas linguísticas para uso na aplicação da avaliação dos requisitos.

As primeiras etapas foram a construção do referencial teórico, a qual contou com a metodologia de Revisão Sistemática de Literatura para formar o portfólio e a definição do objetivo do modelo, conforme já apresentado na seção 2.1.1.

A identificação dos fatores de impacto, ocorreu a partir do referencial teórico, do qual foi realizado um levantamento de fatores utilizados por modelos de estimativas de *software*, bem como o objetivo de cada fator, para então realizar o agrupamento conforme afinidade entre

o objetivo deles, além de selecionar quais seriam utilizados. Considerou-se que a revisão da literatura do portfólio exibiu características similares em relação aos conceitos teóricos e operacionais. Foram selecionados vinte fatores, os quais aqui são denominados de fatores de impacto, e agrupados em quatro dimensões (Técnica, Produto, Pessoal e de Projeto), conforme já explanado na seção 2.3.1 no Quadro 7 cujo agrupamento foi realizado conforme direcionamento do objetivo identificado nas descrições de cada fator.

As etapas seguintes, determinação dos especialistas, e determinação das escalas linguísticas para uso na aplicação da avaliação dos requisitos, são descritas nas seções 3.2.1 e 3.2.2 apresentadas a seguir.

### 3.2.1 Determinação dos especialistas envolvido nas avaliações de importância

Com o objetivo do modelo já definido e os fatores de impactos identificados e selecionados, faz-se necessário determinar os especialistas que irão realizar a avaliação de importância dos fatores de impacto para o desenvolvimento de *software web*.

Os três especialistas selecionados para a avaliação de importância dos fatores de impacto, são profissionais formados na área de desenvolvimento de *software*, que trabalham no Distrito Federal, Paraná e Santa Catarina. Os especialistas foram selecionados por serem atuantes nesse mercado entre 6 e 15 anos, e que trabalham atualmente com desenvolvimento de Aplicações *Web*, nos papéis de desenvolvedor sênior, analista de requisitos e arquiteto de *softwares*.

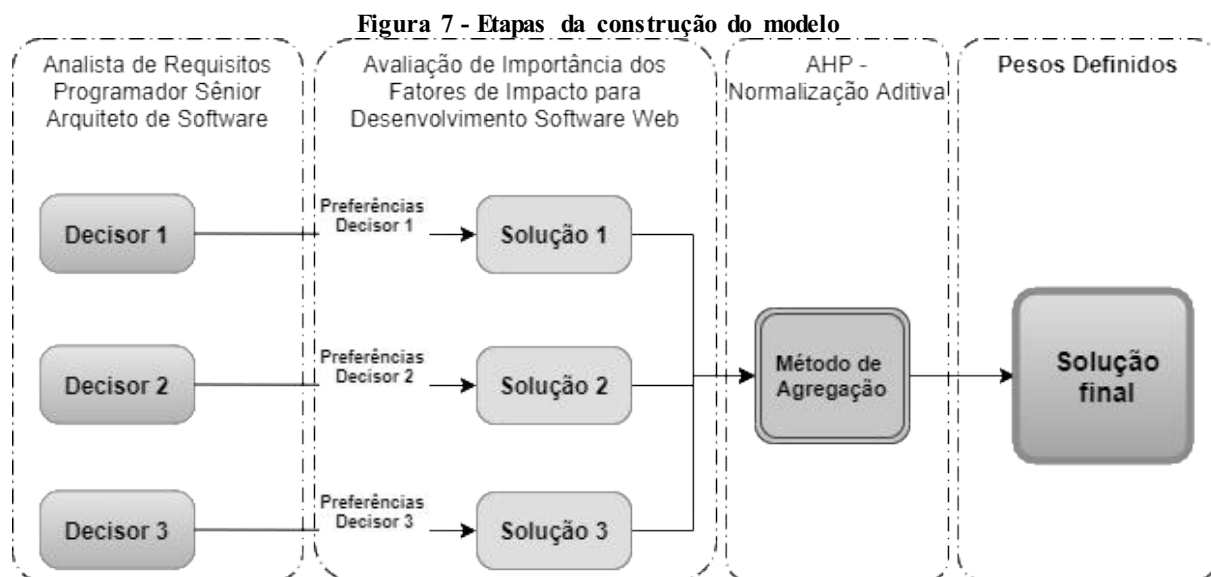
### 3.2.2 Determinação da importância dos fatores de impacto

Após a determinação dos especialistas, fez-se necessário a determinação da importância dos fatores de impacto e, portanto, também da realização das avaliações de importância das dimensões e os fatores relacionados. As dimensões e seus fatores de impacto estão relacionados no Quadro 7 na sessão 2.3.1.

A determinação da importância (peso) de cada fator de impacto é uma etapa fundamental na aplicação dos métodos multicritério. Nesse trabalho, a determinação de importância é necessária para a aplicação do modelo proposto no momento da agregação das avaliações de complexidade dos requisitos de *software*.



As avaliações de importância foram realizadas primeiramente para as dimensões, e posteriormente para os fatores de impacto. A Figura 7 apresenta de maneira ilustrativa o processo de avaliação para obtenção dos pesos.



Fonte: Autoria própria (2021).

A avaliação dos especialistas foi estruturada de acordo com o método AHP, conhecido como um método estruturado para auxílio à tomada de decisões complexas. O AHP é um método de decisão multicritério discreto e hierárquico, desenvolvido por Saaty (1977), na qual a quantificação do julgamento se deu pela utilização da escala de valores proposta por Saaty (1977) apresentada no Quadro 8 da Seção 2.4.1.

Nesse sentido, para a construção do modelo, os fatores de impacto são critérios a serem avaliados pelos especialistas, na comparação par a par quanto à importância ser maior, igual ou menor e em seguida qual o grau dessa importância. As informações foram obtidas no instrumento de coleta de avaliação conforme exemplo na Figura 8.

**Figura 8 - Amostra da ferramenta de coleta de avaliação de importância**

Critérios		Importância	Grau de Importância
	Sabendo que a dimensão técnica (d_01) é caracterizada pelo agrupamento dos fatores:	e que a dimensão de produto (d_02) é caracterizada pelo agrupamento dos fatores:	De acordo com sua opinião, você considera que a dimensão técnica, tem importância maior, igual ou menor em relação
d_01	d_02	<ul style="list-style-type: none"> <li>-Restrição de tempo de execução</li> <li>-Restrição de armazenamento principal</li> <li>-Volatilidade da plataforma</li> <li>-Taxa de transação</li> </ul>	<ul style="list-style-type: none"> <li>-Confiabilidade de software exigida</li> <li>-Tamanho da base de dados</li> <li>-Complexidade do produto</li> <li>-Reutilização necessária</li> <li>-Documentação compatível com as necessidades do ciclo de vida</li> </ul>
		<b>Menor</b>	<b>Importância fraca</b>

Fonte: Autoria própria (2021).

Conforme apresentado na Figura 8 para a avaliação de importância, não foram usados os números mas sim as definições, da escala de importância definida por Saaty (1977) e expostas no Quadro 8, conforme citado anteriormente, objetivando facilitar o julgamento dos especialistas.

Nessa dissertação, para se determinar a importância dos critérios foi adotado o método de Normalização Aditiva, o qual apresenta pouca complexidade computacional e bons resultados conforme relatado por Srdjevic (2005). A Normalização Aditiva é um método algébrico e consiste em obter a soma dos elementos da matriz de decisão, gerar uma nova matriz a partir da normalização dos elementos da matriz pela sua respectiva soma das colunas e nessa matriz obter o vetor pela média dos elementos da linha normalizados, conforme as equações (6) e (7). O método de Normalização Aditiva foi implementado em planilhas no Microsoft Excel.

$$a'_{ij} = \frac{a_{ij}}{\sum_{i=1}^m a_{ij}}, j = 1, 2, \dots, m. \quad (6)$$

$$P_i = \left(\frac{1}{m}\right) \sum_{j=1}^m a'_{ij}, j = 1, 2, \dots, m. \quad (7)$$

Em que:

$a_{ij}$ : é a importância relativa do elemento decisão;

$a'_{ij}$ : é a importância relativa normalizada;

$P_i$ : é o vetor prioridade normalizado;

Após se determinar o vetor de prioridades é necessário avaliar a consistência dos julgamentos. No trabalho de Saaty (1977), a consistência era determinada pela análise do autovalor da matriz de decisão. Porém, quando se utiliza o método de Normalização Aditiva, se recomenda utilizar a Razão da Consistência Harmônica (RCH) proposta por Stein e Mizzi (2007).

O cálculo da Razão da Consistência Harmônica (RCH) é obtido em (8) dada por:

$$RCH = \frac{HCI}{HRI} \quad (8)$$

O Índice de Consistência Harmônica (HCI), é calculado pela equação (9).

$$HCI = \frac{[HM(s) - n](n + 1)}{n(n - 1)} \quad (9)$$

Em que: HM é a média harmônica dos elementos soma de cada coluna da Matriz de Decisão e  $n$  é o número de critérios considerados na matriz.

O Índice Harmônico Randômico (HRI) é apresentado na **Tabela 1**, equivalente ao tamanho da matriz.

**Tabela 1 - Índice Randômico (RI) para valores de Matrizes (n).**

$n$	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10	15x15	20x20	25x25
RI	0,550	0,859	1,061	1,205	1,310	1,381	1,437	1,484	1,599	1,650	1,675

Fonte: Stein e Mizzi (2007).

A regra geral utilizada é que se uma matriz tiver um RCH até 0,10 (0,05 para  $n = 3$  e 0,08 para  $n=4$ ), então o vetor prioridade obtido está suficientemente próximo do auto vetor da matriz de decisão para ser consistente (STEIN E MIZZI, 2007).

Uma etapa de agregação dos pesos é necessária após a determinação dos pesos pelos especialistas. A agregação dos pesos obtidos pela média geométrica, tal como descrita na equação (10), não compromete a propriedade da reciprocidade (ISHIZAKA e LABIB, 2011). A normalização da média geométrica dos julgamentos pode ser definida de acordo com a equação (11).

$$\bar{w}_j = \prod_{k=1}^l (w_{jk})^{\frac{1}{l}} \quad (10)$$

$$\bar{\bar{w}}_j = \frac{\bar{w}_j}{\sum_{j=1}^m \bar{w}_j} \quad (11)$$

Em que:

$\bar{w}_j$  = peso agregado do critério  $j$  para os  $l$  especialistas.

$\bar{\bar{w}}_j$  = peso agregado e normalizado do critério  $j$  para os  $l$  especialistas.

### 3.2.3 Escala para avaliação linguística dos requisitos do *software* em análise

Para que ocorra a fase de aplicação do modelo foi necessário a determinação da escala linguística a ser usada.

De acordo com Herrera e Herrera-Viedma (2000), a escolha do conjunto de termos linguísticos possui como objetivo estabelecer a descrição linguística dos resultados de modo a fornecer ao avaliador algumas palavras pelas quais ele pode naturalmente expressar sua informação. De acordo com essa perspectiva, é necessário que o conjunto de termos seja pequeno o suficiente para não impor precisão aos especialistas, e ao mesmo tempo, deve ser rico o suficiente para permitir aos avaliadores discriminarem suas avaliações conforme suas percepções em um número limitado de graus. A Tabela 2 apresenta as variáveis linguísticas utilizadas pelos especialistas na aplicação do modelo no projeto de desenvolvimento, mais especificamente na avaliação de complexidade dos requisitos do projeto de *software* em desenvolvimento, em relação aos fatores de impacto.

**Tabela 2 - Variáveis linguísticas utilizadas na avaliação da complexidade dos requisitos do software em relação aos fatores de impacto**

<b>Termo Linguístico</b>	<b>Número <i>Fuzzy</i> Trapezoidal</b>
Muito Baixa (VL)	(0.0, 0.0, 0.1, 0.2)
Baixa (L)	(0.1, 0.2, 0.2, 0.3)
Média Baixa (ML)	(0.2, 0.3, 0.4, 0.5)
Média (M)	(0.4, 0.5, 0.5, 0.6)
Média alta (MH)	(0.5, 0.6, 0.7, 0.8)
Alta (H)	(0.7, 0.8, 0.8, 0.9)
Muito Alta (VH)	(0.8, 0.9, 1.0, 1.0)

Fonte: Autoria própria (2021).

A Tabela 2 apresenta uma escala linguística regular com equivalência com números *fuzzy* trapezoidais normalizados. Assim, obtém a determinação das variáveis linguísticas, para avaliação individual de cada fator de impacto para aplicação em um projeto de desenvolvimento de *software web*, conforme definido para este trabalho.

#### 3.2.4 Protótipo para coleta das avaliações dos requisitos

Foi desenvolvido um protótipo de ferramenta computacional para a aplicação do modelo com a finalidade de reduzir o tempo da coleta de informações, além de apresentar uma interface mais agradável de uso. A ferramenta deve apresentar o requisito selecionado e mecanismos para acessar a especificação do mesmo, que se encontra em ferramenta externa, além de apresentar os fatores de impacto a serem avaliados com sua descrição para melhor clareza no momento da análise. Será necessário informar a escala para cada um dos fatores para o requisito selecionado. Ao final, a avaliação deve ser salva para que as informações possam ser armazenadas para serem utilizadas no desenvolvimento do método *Fuzzy* TOPSIS em etapa posterior. A Figura 9 apresenta o protótipo principal da ferramenta de avaliação.

Figura 9 - Protótipo de ferramenta computacional sugerida para implementação

Moqzilla

← → ↻

## Avaliação

Avaliador:

Requisito 1  Link do Requisito:

Fatores de Impacto	Complexidade
Restrição de tempo de execução	Muito Alta <input type="range" value="50"/> Muito Baixa
Restrição de armazenamento principal	Muito Alta <input type="range" value="50"/> Muito Baixa
Volatilidade da plataforma	Muito Alta <input type="range" value="50"/> Muito Baixa
Taxa de transação	Muito Alta <input type="range" value="50"/> Muito Baixa
Confiabilidade de software exigida	Muito Alta <input type="range" value="50"/> Muito Baixa
Tamanho da base de dados	Muito Alta <input type="range" value="50"/> Muito Baixa
Complexidade do produto	Muito Alta <input type="range" value="50"/> Muito Baixa
Reutilização necessária	Muito Alta <input type="range" value="50"/> Muito Baixa

demais fatores

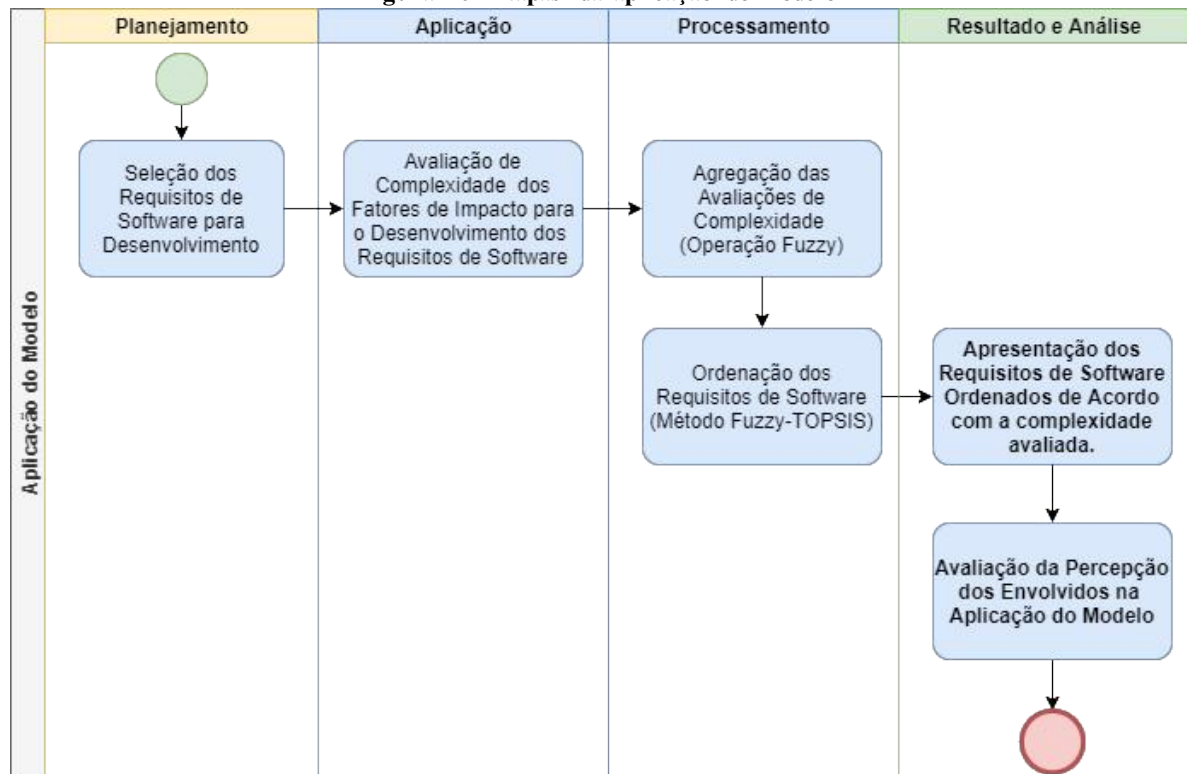
Fonte: Autoria própria (2020)

### 3.3 Aplicação do modelo

De acordo com a classificação de categorias de tipos de *software* realizada por Pressman (2014), são sete as categorias: *softwares* de sistema, de aplicativo, de Engenharia, embutido em *hardware*, para linha de produtos, aplicações web e de inteligência artificial. Para este trabalho, foi utilizado o tipo *software web*, por ser popular, possuindo assim, maior opção para seleção de especialistas no processo de avaliação. *Software/Aplicações Web*, de acordo com Pressman (2014), são programas focados em rede, que são acessados por meio de um navegador a partir de computadores e dispositivos móveis, geralmente o acesso acontece via internet.

A aplicação do modelo ocorreu em três fases: o (i) Planejamento, (ii) Aplicação e (iii) Resultado e Análise conforme apresenta a Figura 10.

Figura 10 - Etapas da aplicação do modelo



Fonte: Autoria própria (2021).

O modelo foi aplicado no projeto de desenvolvimento, de um ciclo de determinado *software web*, fornecido no formato *Software como Serviço* (SAAS - *Software as a Service*) sendo uma plataforma de recursos humanos, para gestão de desempenho e engajamento. Tal *software* é desenvolvido por uma empresa situada no Sudoeste do Paraná, com 30 anos de experiência na área.

### 3.3.1 Seleção e avaliação dos requisitos para desenvolvimento

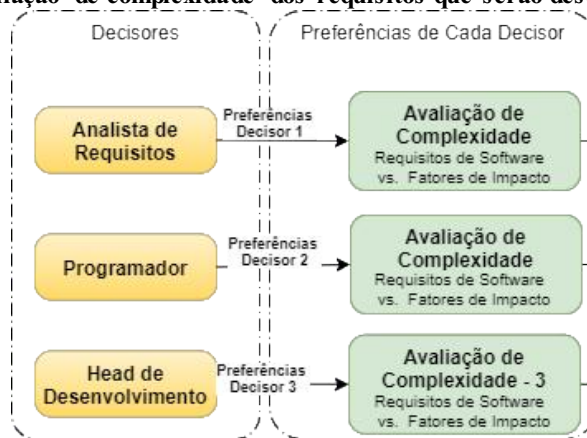
Na etapa de planejamento, ocorreu a seleção dos requisitos de *software* que serão implementados no projeto de desenvolvimento, conforme normalmente ocorre. Com os itens para desenvolvimento já selecionados, a próxima etapa foi realizar a avaliação de complexidade dos fatores de impacto para o desenvolvimento de cada requisito.

Antes de realizar as avaliações dos requisitos de *software* foram realizadas reuniões com alguns Diretores da organização, bem como com o responsável pelo projeto, a fim de apresentar o modelo proposto, seus objetivos e explicar como as etapas que ocorreriam. Após a aprovação para aplicação do modelo, foi realizada a escolha dos avaliadores. A escolha

ocorreu pelo responsável do projeto, de forma a atender aos critérios de serem de perfis distintos (Análise de Requisitos, Programação e Responsável pelo Projeto).

Com os avaliadores já definidos, ocorreu uma oficina virtual de orientação, para explicar o uso da ferramenta de avaliação e tirar dúvidas do processo. Cada profissional realizou a avaliação de maneira individual. As avaliações foram realizadas por três profissionais, sendo um analista de requisitos, um desenvolvedor pleno e um *head* de desenvolvimento conforme exemplifica a Figura 11.

**Figura 11 - Processo de avaliação de complexidade dos requisitos que serão desenvolvidos no projeto**



Fonte: Autoria própria (2020)

A avaliação ocorreu por meio da ferramenta computacional *Google Sheets*, a qual foi estruturada para apresentar as informações necessárias para que a avaliação pudesse ocorrer de forma correta conforme apresenta a Figura 12.

**Figura 12 - Ferramenta de avaliação de complexidade dos requisitos de software**

	Fatores de Impacto
<b>Requisito</b>	Restrição de tempo de execução
<a href="#">Implementar</a> <a href="#">compartilhar</a> <a href="#">publicações</a>	Em relação ao fator "Restrição de tempo de execução", na sua opinião, qual a complexidade pra o desenvolvimento do requisito "Implementar compartilhar publicações"?
	Baixa

Fonte: Autoria própria (2020)



A Figura 12, apresenta um exemplo de como foi estruturada a ferramenta de avaliação. Cada requisito de *software* possui o link externo para o *software* organizacional de gestão de projetos da empresa, de forma que, seria possível fazer a leitura da especificação, se necessário. Também fica visível a descrição do fator de impacto, ao posicionar o cursor do *mouse* sobre ele. Além disso, uma pergunta, na qual foi direcionado o sentido da avaliação de complexidade. Os resultados das avaliações são apresentados capítulo 4.

### 3.3.2 Agregação das avaliações de complexidade dos requisitos

A classificação *fuzzy* do  $k$ -ésimo especialista é dada representada por  $\tilde{x}_{ijk} = (a_{ijk}, b_{ijk}, c_{ijk}, d_{ijk})$ ,  $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ . Para se obter as classificações *fuzzy* agregadas ( $\tilde{x}_{ij}$ ) dos requisitos do *software web* em desenvolvimento com relação a cada critério (Fator de impacto) se utiliza a equação (12).

$$\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij}, d_{ij}) \quad (12)$$

$$\text{Em que: } a_{ij} = \min_k \{a_{ijk}\}, \quad b_{ij} = \frac{1}{k} \sum_{k=1}^k b_{ijk}, \quad c_{ij} = \frac{1}{k} \sum_{k=1}^k c_{ijk}, \quad d_{ij} = \max_k \{d_{ijk}\}$$

Com o desenvolvimento da agregação foi possível obter um único número *fuzzy* trapezoidal para cada um dos requisitos do *software web* em desenvolvimento.

### 3.3.3 Ordenação dos requisitos em relação aos fatores de impacto com o *fuzzy* topsis

Na aplicação do método *Fuzzy* TOPSIS no modelo proposto os fatores de impacto desempenham o papel de critérios e os requisitos do *software* em desenvolvimento desempenham o papel das alternativas, considerando a perspectiva dos métodos multicritério.

Nesta dissertação, as variáveis linguísticas estão associadas a números *fuzzy* trapezoidais positivos (PTFNs) conforme apresentado na Tabela 3. Neste trabalho, a versão de Chen (2000) foi aplicada para desenvolver o método *Fuzzy* TOPSIS.

Utilizando os pesos agregados (importância) dos fatores de impacto determinados pelo método AHP (seção 3.2.2), foi aplicado o método *Fuzzy* TOPSIS, para gerar a ordenação final

e apresentação dos fatores que obtiveram maior pontuação de complexidade para cada requisito do *software* em desenvolvimento.

Como afirmado acima, o problema de decisão pode ser expresso de forma concisa no formato de matriz como segue:

$$\tilde{D} = A_1 A_2 \dots A_m [\tilde{x}_{11} \tilde{x}_{12} \tilde{x}_{21} \tilde{x}_{22} \dots \tilde{x}_{1n} \dots \tilde{x}_{2n} \dots \tilde{x}_{m1} \tilde{x}_{m2} \dots \tilde{x}_{mn}]$$

$$\overline{w}_j = [w_1, w_2 \dots w_n]$$

Em que  $\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij}, d_{ij})$   $i = 1, 2, \dots, m, j = 1, 2, \dots, n$  pode ser aproximado por números *fuzzy* trapezoidais positivos.

Observação: Se considerarmos o algoritmo do método TOPSIS, o passo seguinte seria a normalização da matriz, mas ao utilizar termos linguísticos em associação com números *fuzzy* trapezoidais normalizados para a avaliação das alternativas para todos os critérios, esta etapa é desnecessária. Mais informações sobre esta etapa podem ser encontradas em Chen (2000).

Considerando a importância diferente de cada critério, a matriz de decisão *fuzzy* normalizada ponderada é construída como apresenta equação (13):

$$\tilde{V} = [\tilde{v}_{ij}]_{m \times n}, i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (13)$$

$$\text{Em que: } \tilde{v}_{ij} = \tilde{x}_{ij} \otimes \overline{w}_j.$$

De acordo com a matriz de decisão *fuzzy* normalizada ponderada, os números difusos trapezoidais positivos normalizados também podem aproximar os elementos  $\tilde{v}_{ij}, \forall i, j$ . Então, a solução ideal-positiva *fuzzy* (FPIS,  $A^*$ ) e a solução ideal-negativa *fuzzy* (FNIS,  $A^-$ ) pode ser definida conforme equações (14) e (15):

$$A^* = (\tilde{v}_1^*, \tilde{v}_2^*, \dots, \tilde{v}_n^*) \quad (14)$$

$$A^- = (\tilde{v}_1^-, \tilde{v}_2^-, \dots, \tilde{v}_n^-) \quad (15)$$

$$\text{Em que: } \tilde{v}_j^* = \max_i \{v_{ij}\} \text{ e } \tilde{v}_j^- = \min_i \{v_{ij}\}, i = 1, 2, \dots, m; j = 1, 2, \dots, n$$

A distância de cada alternativa de  $A^*$  e  $A^-$  pode ser calculada conforme equações (16) e (17).

$$d_i^* = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^*), i = 1, 2, \dots, m \quad (16)$$

$$d_i^- = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^-), i = 1, 2, \dots, m \quad (17)$$

Em que:  $d_v$  (equação 5) é a medida de distância entre 2 números *fuzzy*.

Um coeficiente de proximidade é definido para determinar a ordem de classificação de todas as alternativas uma vez  $d_i^*$  e  $d_i^-$  de cada alternativa  $A_i (i = 1, 2, \dots, m)$  foi calculado. O coeficiente de proximidade representa as distâncias para a solução ideal-positiva *fuzzy* ( $A^*$ ) e a solução ideal-negativa *fuzzy* ( $A^-$ ) simultaneamente, levando a proximidade relativa à solução ideal-positiva difusa. O coeficiente de proximidade ( $CC_i$ ) de cada alternativa é calculada conforme equação (18).

$$CC_i = \frac{d_i^-}{d_i^- + d_i^*}, \quad i = 1, 2, \dots, m \quad (18)$$

### 3.3.4 Avaliação da percepção dos envolvidos na aplicação do modelo

O instrumento para a avaliação da percepção dos envolvidos na aplicação do modelo, é apresentado no APÊNDICE D e o envio ocorreu em agosto de 2021 via *Google Forms* com as perguntas apresentadas no Quadro 9.

**Quadro 9 - Perguntas da avaliação da aplicação do modelo**

Em relação ao modelo aplicado e a sua composição, qual a sua percepção quanto a relevância de cada item abaixo mencionado, para o processo de tomada de decisão no planejamento e acompanhamento do projeto?	
Fatores de Impacto utilizados:	Avaliação de Complexidade do Requisitos:
<input type="checkbox"/> irrelevante <input type="checkbox"/> pouco relevante <input type="checkbox"/> moderadamente relevante <input type="checkbox"/> muito relevante <input type="checkbox"/> extremamente relevante	<input type="checkbox"/> irrelevante <input type="checkbox"/> pouco relevante <input type="checkbox"/> moderadamente relevante <input type="checkbox"/> muito relevante <input type="checkbox"/> extremamente relevante
Se desejar, comente como foi a sua experiência e sua percepção sobre o impacto da aplicação do modelo de apoio para o gerenciamento do projeto?	
Na sua opinião, qual desses papéis deve participar da avaliação de complexidade do modelo aplicado?	
<input type="checkbox"/> Gestor de Produto <input type="checkbox"/> Head de Desenvolvimento <input type="checkbox"/> Desenvolvedor <input type="checkbox"/> Analista de Requisitos	<input type="checkbox"/> Analista de Testes <input type="checkbox"/> Head de Qualidade <input type="checkbox"/> Analistas de Qualidade <input type="checkbox"/> Outro: _____

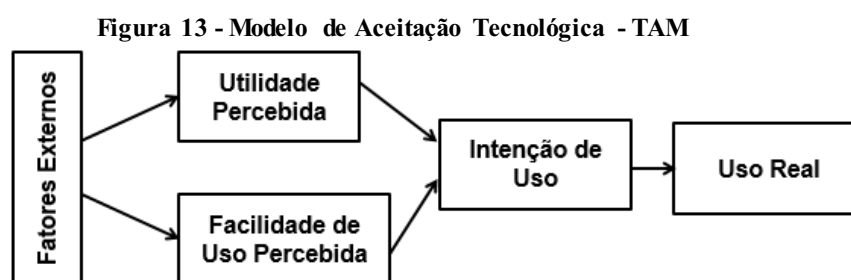
Se desejar, comente qual foi sua percepção sobre a ferramentas utilizada para aplicação do modelo, quanto a sua facilidade de uso e outros pontos que julgar importantes.

Se desejar, deixe sugestões para melhorar a aplicação ou o aproveitamento do modelo.

Fonte: Aatoria própria(2021)

Conforme demonstra o Quadro 9, foram aplicadas questões com o objetivo de captar as primeiras impressões referente a percepção da aplicação do modelo. Depois foi aplicado o *Technology Acceptance Model* (TAM) para avaliar a aceitação e uso de tecnologias quanto a sua facilidade e utilidade percebida ao aplicar o modelo.

TAM foi proposto por Davis (1989), para focar no motivo dos usuários aceitarem ou rejeitarem a tecnologia da informação e como melhorar a aceitação, oferecendo, desse modo, um suporte para prever e explicar a aceitação. Davis (1989) define os dois principais determinantes do modelo TAM da seguinte maneira: Utilidade percebida - Grau em que uma pessoa acredita que o uso de um sistema particular pode melhorar o seu desempenho; Facilidade de uso percebida - É o grau em que uma pessoa acredita que o uso de um sistema de informação será livre de esforço. Os principais determinantes mediam completamente os efeitos das variáveis externas, como características do sistema, processo de desenvolvimento, treinamento, na intenção de uso (DAVIS, 1989). A Figura 13 apresenta o modelo TAM original.



Fonte: Adaptado de Davis *et al.* (1989)

Conforme apresenta a Figura 13, sugere-se que os indivíduos usarão uma determinada tecnologia se acreditarem que este uso acarretará em resultados positivos, focalizando-se na facilidade de uso percebida (*Perceived Ease of Use*) e na utilidade percebida (*Perceived Usefulness*).

No questionário foi utilizado a escala de *Likert*, usada comumente em questionários. Ao responderem a um questionário baseado nesta escala, os respondentes especificam seu nível de concordância plena ou discordância total, a partir de uma afirmação feita em tal instrumento. Nesse estudo foram empregados os níveis de afirmação na escala de *Likert* de 5 pontos apresentada o Quadro 10.

**Quadro 10 - Escala de Likert Utilizada no TAM**

<b>Escala Likert</b>
Discordo totalmente
Discordo
Neutro
Concordo
Concordo Totalmente

**Fonte: Autoria própria (2021)**

Conforme pode-se observar no Quadro 10 as opções variam do discordo totalmente até o concordo totalmente. Uma vantagem do uso desta escala é que ela fornece direções sobre a atitude do respondente em relação a cada afirmação.

Foram definidas variáveis para os construtos Utilidade Percebida, Facilidade de Uso Percebida, Intenção de Uso e Variáveis Externas que são apresentadas no Quadro 11.

**Quadro 11 - Constructos e variáveis – TAM**

<b>Utilidade percebida</b>	<b>Facilidade de uso percebida</b>	<b>Intenção de uso</b>	<b>Variáveis externas</b>
Considero relevante a utilização dos Fatores de Impacto	Realizar a Avaliação de Complexidade do Requisitos é Simples	Recomendo a utilização deste modelo de priorização de requisitos	A quantidade de fatores de impacto é adequada
Considero relevante a Avaliação de Complexidade do Requisitos Aplicada	A Avaliação de Complexidade do Requisitos é Rápida		Os fatores de impacto apresentados para a avaliação de complexidade na aplicação do modelo, são adequados
Considero o modelo relevante quanto a proporcionar benefícios para o processo de tomada de decisão no gerenciamento do projeto	Senti dificuldade em realizar a aplicação do Modelo de Priorização de Requisitos com a Avaliação de Complexidade		Considero suficiente os cargos envolvidos na avaliação de complexidade durante a aplicação do modelo
O resultado da aplicação do modelo auxiliou no processo	Aprender a aplicar o modelo de priorização de requisitos é fácil		Se desejar cite 3 adjetivos que expressam sua opinião sobre

de tomada de decisão no gerenciamento do projeto			o modelo de priorização de requisitos aplicado
A aplicação do modelo permitiu melhorar a assertividade no processo de tomada de decisão	A descrição dos fatores de impacto do modelo, são de fácil entendimento		Se desejar, complete a afirmação a seguir, conforme sua percepção: Para ser mais simples de utilizar, o modelo de priorização de requisitos deve ...
A aplicação do modelo facilitou a justificativa da ordem de entrega de versões para o cliente			Se desejar, complete a afirmação a seguir, conforme sua percepção: Para ser mais assertivo, o modelo de priorização de requisitos deve
A utilização do modelo favorece o alinhamento de prazos			
A utilização do modelo favorece o acompanhamento de impedimentos técnicos			
A utilização do modelo favorece a melhoria de performance da entregas			

Fonte: Elaborada pela autora (2021)

Conforme demonstra o Quadro 11, foram aplicadas questões para verificar a percepção de relevância dos fatores de impactos utilizados, além da importância percebida quanto a realizar a avaliação de complexidade para o processo de desenvolvimento e gestão.

## 4 RESULTADOS

Esta seção apresenta o resultado das etapas envolvidas na elaboração do Modelo Multicritério de Priorização de Requisitos de *Software Web* e os resultados de sua aplicação no projeto de uma empresa.

### 4.1 Avaliação da importância dos fatores de impacto

Avaliação da importância dos fatores de impacto, realizada pelos especialistas na fase de construção do modelo proposto, conforme explicado na sessão 3.2.1. Por meio da comparação par a par de cada critério, os especialistas desenvolveram a avaliação das importâncias relativas das dimensões dos fatores de impacto e dos fatores impacto dentro de cada dimensão, desta maneira se determinou as importâncias de cada uma das dimensões e de cada um dos fatores de impacto.

Conforme exemplificado na Figura 12, as respostas fornecidas no formulário de avaliação de importância foram coletadas e os dados foram estruturados conforme definido pelo método AHP em que os decisores expuseram o seu posicionamento na comparação par a par, indicando as preferências em relação às dimensões e fatores de impacto. O mesmo processo foi realizado com os fatores de impacto de cada dimensão.

Com os dados da matriz de comparação, foi aplicado o método da Normalização Aditiva por meio da divisão de cada valor obtido com a soma dos elementos de cada coluna, indicando o vetor de ponderação de cada critério. O vetor determina o peso relativo de cada critério no resultado total e corresponde de modo aproximado à média aritmética dos valores de cada um dos critérios, deixando evidente qual critério é considerado o mais relevante conforme a preferência do especialista.

Os pesos relativos determinados de cada especialista, foram verificados quanto a sua consistência conforme equações (8) e (9). Os resultados das avaliações de consistência assim como valor normalizado para as Dimensões são apresentados na Tabela 3.

**Tabela 3 - Pesos das Dimensões (método AHP) conforme opinião dos especialistas**

(continua)

Dimensões	Pesos dos Especialistas			Peso Agregado das Dimensões	Ranking
	E1	E2	E3		
D1 -	0,143	0,110	0,240	0,168	3º
D2 -	0,302	0,530	0,400	0,432	1º

D3 -	0,407	0,100	0,280	0,243	2°
D4 -	0,148	0,260	0,080	0,157	4°
RCH< 0,08	0,076	0,066	0,079	-	-

Fonte: A autoria própria (2021).

Os resultados das avaliações de consistência assim como peso dos fatores de impacto da Dimensão Técnica (D1) são apresentados na Tabela 4.

**Tabela 4 - Pesos dos Fatores de Impacto da D1 (método AHP) conforme opinião dos especialistas**

Fatores D1	Pesos dos Especialistas			Peso Agregado
	E1	E2	E3	
FI1	0,590	0,260	0,343	0,419
FI2	0,067	0,060	0,066	0,072
FI3	0,154	0,620	0,296	0,341
FI4	0,189	0,060	0,296	0,168
RCH< 0,08	0,057	0,055	0,008	-

Fonte: A autoria própria (2021).

De acordo com os especialistas, na dimensão técnica (D1), o fator de impacto restrição de tempo de execução (FI1) apresenta maior importância para o desenvolvimento de *software web* de maneira geral, seguido dos fatores de impacto volatilidade da plataforma (FI3), taxa de transação (FI4) e restrição de armazenamento principal (FI2).

Os resultados das avaliações de consistência assim como peso dos fatores de impacto da Dimensão de Produto (D2) são apresentados na Tabela 5.

**Tabela 5 - Pesos dos Fatores de Impacto da Dimensão 2 (método AHP) conforme opinião dos especialistas**

Fatores D2	Pesos dos Especialistas			Peso Agregado	Ranking
	E1	E2	E3		
FI5	0,23	0,326	0,292	0,293	1°
FI6	0,04	0,035	0,032	0,036	6°
FI7	0,21	0,257	0,308	0,267	2°
FI8	0,23	0,076	0,042	0,094	5°
FI9	0,14	0,113	0,098	0,122	4°
FI10	0,13	0,194	0,227	0,188	3°
RCH < 0,100	0,064	0,034	0,066	-	-

Fonte: A autoria própria (2021).



Conforme apresenta a Tabela 5, de acordo com os especialistas, na dimensão de produto (D2), o fator de impacto Confiabilidade de software exigida (FI5), seguido da complexidade do produto (FI7), eficiência do usuário final (FI10), documentação compatível com as necessidades do ciclo de vida (FI9), reutilização necessária (FI8) e por último o fator de impacto tamanho da base de dados (FI6).

Os resultados das avaliações de consistência assim como peso dos fatores de impacto da Dimensão de Pessoal (D3) são apresentados na Tabela 6.

**Tabela 6 - Pesos dos Fatores de Impacto da D3 (método AHP) conforme opinião dos especialistas**

Fatores D3	Pesos dos Especialistas			Peso Agregado	Ranking
	E1	E2	E3		
FI11	0,268	0,179	0,099	0,178	3°
FI12	0,172	0,307	0,322	0,273	1°
FI13	0,127	0,268	0,204	0,202	2°
FI14	0,071	0,090	0,163	0,107	5°
FI15	0,195	0,109	0,132	0,150	4°
FI16	0,167	0,047	0,080	0,091	6°
RCH < 0,100	0,093	0,100	0,090	-	-

Fonte: A autoria própria (2021).

Conforme apresenta a Tabela 6, de acordo com os especialistas para a dimensão de pessoal (D3), o fator de impacto capacidade do programador (FI12) tem maior importância para o desenvolvimento de *software web*, seguido dos fatores experiência de aplicações (FI13) capacidade do analista (FI11), linguagem e experiência com ferramentas (FI15), experiência de plataforma (FI14), continuidade de pessoal (FI16) em sua respectiva ordem.

Os resultados das avaliações de consistência assim como peso dos fatores de impacto da Dimensão de Projeto (D4) são apresentados na Tabela 7.

**Tabela 7 - Pesos dos Fatores de Impacto da Dimensão 4 (método AHP) conforme opinião dos especialistas**

Fatores D4	Pesos dos Especialistas			Peso Agregado	Ranking
	E1	E2	E3		
FI17	0,135	0,229	0,455	0,266	2°
FI18	0,242	0,069	0,078	0,120	4°
FI19	0,326	0,295	0,370	0,363	1°
FI20	0,298	0,408	0,097	0,251	3°
RCH < 0,08	0,062	0,014	0,072	-	-

Fonte: A autoria própria (2021).

Ao observar a Tabela 7, pode-se constatar que de acordo com os especialistas o fator de impacto cronograma de desenvolvimento requerido (FI19), maior importância para o desenvolvimento de software web, seguido respectivamente dos fatores uso de ferramentas de software (FI17), processamento de dados distribuído (FI20), desenvolvimento multisite (FI18).

Os julgamentos dos especialistas foram consistentes conforme o Índice Harmônico Randômico (HRI) explanado na Tabela 1 na seção 3.2.1 foi realizada a agregação e na então a determinação de importância dos fatores de impacto. O resultado é apresentado na Tabela 8.

**Tabela 8 - Peso global dos fatores de impacto obtidos pelo método AHP especialistas**

		<b>Alternativa</b>	<b>Peso</b>
<b>D1 - Dimensão Técnica</b>	0,168	F1 -Restrição de tempo de execução	0,0705
		F2 - Restrição de armazenamento principal	0,0121
		F3 -Volatilidade da plataforma	0,0573
		F4 -Taxa de transação	0,0282
<b>D2 - Dimensão Produto</b>	0,432	F5 - Confiabilidade de <i>software</i> exigida	0,1265
		F6 -Dimensão da base de dados	0,0158
		F7- Complexidade do produto	0,1153
		F8 - Reutilização necessária	0,0407
		F9 - Documentação compatível com as necessidades do ciclo de vida	0,0526
<b>D3 - Dimensão Pessoal</b>	0,243	F10 - Eficiência do usuário final	0,0812
		F11 - Capacidade do analista	0,0432
		F12 - Capacidade do programador	0,0662
		F13 - Experiência de aplicações	0,0492
		F14 - Experiência de plataforma	0,0260
		F15 - Linguagem e experiência com ferramentas	0,0363
<b>D4 - Dimensão Projeto</b>	0,157	F16 - Continuidade de pessoal	0,0220
		F17 - Uso de ferramentas de <i>software</i>	0,0417
		F18 - Desenvolvimento Multisite	0,0189
		F19 - Cronograma de Desenvolvimento Requerido	0,0569
		F20 - Processamento de dados distribuído	0,0394

**Fonte: Autoria própria (2021).**

Conforme pode-se observar na Tabela 8, os fatores da Dimensão de Produto (D2) apresentaram maior peso, seguidos pelos fatores da Dimensão de Pessoal (D3), posteriormente os fatores da Dimensão Técnica (D1). O fator de impacto confiabilidade de software exigida foi o que ficou com maior peso (0,1265), seguido pelos fatores complexidade do produto(0,1153) e eficiência do usuário final (0,0812), todos da dimensão de produto (D2). Na sequência tem-se os fatores restrição de tempo de execução (0,0705) e capacidade do programador (0,0662) estes das dimensões técnica e de produto respectivamente.

Nesse sentido é possível afirmar que de acordo com os especialistas, os fatores de impacto da dimensão de produto são os fatores que possuem maior importância para o desenvolvimento de *software web* de maneira geral.

## **4.2 Seleção e avaliação dos requisitos para desenvolvimento**

Com o modelo elaborado, a próxima fase foi sua aplicação em uma empresa de desenvolvimento de *software*. Os dez requisitos do projeto de desenvolvimento selecionado para aplicação, são aqui denominados apenas de Requisito (RQ), para garantir o sigilo de informações a pedido da empresa, por se tratar de um projeto comercial.

Conforme explicado na seção 3.3.1, foram três integrantes da equipe, que realizaram de forma individual as avaliações de complexidade dos fatores de impacto para os requisitos definidos. As tabelas com a avaliação usando as variáveis linguísticas, conforme o instrumento apresentado na seção 3.2.3 estão disponíveis no APÊNDICE B.

As avaliações individualmente conforme foram sendo realizadas, foram também convertidas em sua respectiva representação *Fuzzy* Trapezoidal. As avaliações já nas suas correspondentes representação em *Fuzzy* Trapezoidal foram agregadas em um único resultado de acordo com a equação (10) detalhada na seção 3.3.3 e são apresentadas nas Tabela 9

**Tabela 9 - Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C1-C5) em números *Fuzzy* Trapezoidais**

Alternativa	C1				C2				C3				C4				C5			
RQ1	0,000	0,133	0,167	0,300	0,200	0,467	0,533	0,800	0,100	0,200	0,200	0,300	0,000	0,233	0,267	0,600	0,000	0,200	0,300	0,500
RQ2	0,200	0,367	0,433	0,600	0,000	0,133	0,167	0,300	0,000	0,133	0,167	0,300	0,100	0,233	0,267	0,500	0,100	0,400	0,400	0,600
RQ3	0,100	0,333	0,367	0,600	0,000	0,000	0,100	0,200	0,000	0,000	0,100	0,200	0,000	0,067	0,133	0,300	0,000	0,133	0,167	0,300
RQ4	0,000	0,000	0,100	0,200	0,000	0,000	0,100	0,200	0,000	0,000	0,100	0,200	0,000	0,067	0,133	0,300	0,000	0,133	0,167	0,300
RQ5	0,400	0,533	0,567	0,800	0,100	0,433	0,467	0,800	0,100	0,200	0,200	0,300	0,000	0,400	0,500	0,800	0,100	0,333	0,367	0,800
RQ6	0,000	0,000	0,100	0,200	0,000	0,000	0,100	0,200	0,000	0,000	0,100	0,200	0,000	0,067	0,133	0,300	0,000	0,233	0,267	0,600
RQ7	0,100	0,367	0,433	0,800	0,000	0,167	0,233	0,500	0,000	0,167	0,233	0,500	0,000	0,467	0,533	1,000	0,100	0,533	0,567	1,000
RQ8	0,100	0,267	0,333	0,500	0,000	0,100	0,200	0,500	0,000	0,167	0,233	0,500	0,000	0,467	0,533	1,000	0,400	0,633	0,667	1,000
RQ9	0,000	0,200	0,300	0,500	0,000	0,167	0,233	0,500	0,100	0,233	0,267	0,500	0,000	0,367	0,433	0,800	0,400	0,600	0,600	0,900
RQ10	0,000	0,267	0,333	0,600	0,000	0,133	0,167	0,300	0,000	0,167	0,233	0,500	0,000	0,300	0,400	0,800	0,200	0,433	0,467	0,600

Fonte: Dados da pesquisa. (2021).

**Tabela 10 - Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C6-C10) em números *Fuzzy* Trapezoidais**

Alternativa	C6				C7				C8				C9				C10			
RQ1	0,000	0,167	0,233	0,500	0,400	0,600	0,600	0,900	0,000	0,000	0,100	0,200	0,400	0,633	0,667	0,900	0,000	0,067	0,133	0,300
RQ2	0,000	0,200	0,300	0,500	0,200	0,400	0,500	0,800	0,000	0,067	0,133	0,300	0,200	0,467	0,533	0,800	0,000	0,067	0,133	0,300
RQ3	0,000	0,000	0,100	0,200	0,000	0,133	0,167	0,300	0,000	0,100	0,200	0,500	0,000	0,000	0,100	0,200	0,000	0,167	0,233	0,600
RQ4	0,000	0,000	0,100	0,200	0,000	0,133	0,167	0,300	0,000	0,100	0,200	0,500	0,000	0,000	0,100	0,200	0,000	0,167	0,233	0,600
RQ5	0,000	0,167	0,233	0,500	0,200	0,533	0,567	0,900	0,000	0,000	0,100	0,200	0,500	0,667	0,733	0,900	0,000	0,133	0,167	0,300
RQ6	0,000	0,000	0,100	0,200	0,000	0,133	0,167	0,300	0,000	0,000	0,100	0,200	0,000	0,000	0,100	0,200	0,000	0,133	0,167	0,300
RQ7	0,000	0,267	0,333	0,600	0,200	0,467	0,533	0,900	0,000	0,233	0,267	0,600	0,100	0,400	0,400	0,900	0,000	0,133	0,167	0,300
RQ8	0,000	0,167	0,233	0,500	0,200	0,467	0,533	0,900	0,000	0,233	0,267	0,600	0,100	0,400	0,400	0,900	0,000	0,467	0,533	1,000
RQ9	0,100	0,467	0,533	1,000	0,200	0,467	0,533	0,900	0,000	0,233	0,267	0,600	0,100	0,400	0,400	0,900	0,100	0,533	0,567	1,000
RQ10	0,000	0,200	0,300	0,500	0,200	0,467	0,533	0,900	0,000	0,133	0,167	0,300	0,200	0,533	0,567	0,900	0,000	0,133	0,167	0,300

Fonte: Dados da pesquisa (2021).

**Tabela 11 - Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C11-C15) em números *Fuzzy* Trapezoidais**

Alternativa	C11				C12				C13				C14				C15			
RQ1	0,100	0,300	0,300	0,600	0,200	0,467	0,533	0,900	0,100	0,267	0,333	0,500	0,100	0,233	0,267	0,500	0,500	0,600	0,700	0,800
RQ2	0,000	0,267	0,333	0,800	0,000	0,267	0,333	0,800	0,100	0,300	0,300	0,600	0,200	0,467	0,533	0,800	0,200	0,433	0,467	0,600
RQ3	0,000	0,067	0,133	0,300	0,000	0,000	0,100	0,200	0,000	0,167	0,233	0,600	0,000	0,067	0,133	0,300	0,000	0,133	0,167	0,300
RQ4	0,000	0,067	0,133	0,300	0,000	0,000	0,100	0,200	0,000	0,167	0,233	0,600	0,000	0,067	0,133	0,300	0,000	0,133	0,167	0,300
RQ5	0,000	0,000	0,100	0,200	0,000	0,300	0,400	0,800	0,500	0,600	0,700	0,800	0,100	0,400	0,400	0,600	0,400	0,533	0,567	0,800
RQ6	0,000	0,300	0,400	1,000	0,000	0,000	0,100	0,200	0,000	0,000	0,100	0,200	0,000	0,100	0,200	0,500	0,000	0,133	0,167	0,300
RQ7	0,000	0,067	0,133	0,300	0,100	0,367	0,433	0,800	0,100	0,467	0,533	0,800	0,200	0,367	0,433	0,600	0,400	0,500	0,500	0,600
RQ8	0,000	0,067	0,133	0,300	0,100	0,367	0,433	0,800	0,100	0,367	0,433	0,800	0,200	0,367	0,433	0,600	0,400	0,567	0,633	0,800
RQ9	0,000	0,067	0,133	0,300	0,100	0,367	0,433	0,800	0,100	0,400	0,400	0,600	0,200	0,367	0,433	0,600	0,400	0,600	0,600	0,900
RQ10	0,000	0,233	0,267	0,600	0,000	0,233	0,267	0,600	0,100	0,433	0,467	0,800	0,200	0,433	0,467	0,600	0,200	0,367	0,433	0,600

Fonte: Dados da pesquisa. (2021).

**Tabela 12 – Avaliação Agregada de Complexidade dos Requisitos quanto aos Fatores de Impacto (C16-C20) em números *Fuzzy* Trapezoidais**

Alternativa	C16				C17				C18				C19				C20			
RQ1	0,000	0,267	0,333	0,600	0,100	0,233	0,267	0,500	0,000	0,100	0,200	0,500	0,100	0,367	0,433	0,800	0,000	0,167	0,233	0,600
RQ2	0,000	0,333	0,367	0,600	0,100	0,233	0,267	0,500	0,000	0,000	0,100	0,200	0,100	0,433	0,467	0,800	0,000	0,167	0,233	0,600
RQ3	0,000	0,167	0,233	0,600	0,000	0,067	0,133	0,300	0,000	0,067	0,133	0,300	0,000	0,200	0,300	0,800	0,000	0,067	0,133	0,300
RQ4	0,000	0,167	0,233	0,600	0,000	0,067	0,133	0,300	0,000	0,067	0,133	0,300	0,000	0,200	0,300	0,800	0,000	0,067	0,133	0,300
RQ5	0,000	0,367	0,433	0,800	0,400	0,533	0,567	0,800	0,000	0,267	0,333	0,900	0,400	0,567	0,633	0,800	0,000	0,233	0,267	0,600
RQ6	0,000	0,167	0,233	0,600	0,000	0,067	0,133	0,300	0,000	0,067	0,133	0,300	0,000	0,200	0,300	0,800	0,000	0,067	0,133	0,300
RQ7	0,000	0,367	0,433	0,800	0,000	0,400	0,500	0,800	0,000	0,200	0,300	0,800	0,100	0,533	0,567	0,900	0,000	0,067	0,133	0,300
RQ8	0,000	0,333	0,367	0,600	0,000	0,400	0,500	0,800	0,000	0,200	0,300	0,800	0,100	0,533	0,567	0,900	0,000	0,067	0,133	0,300
RQ9	0,000	0,333	0,367	0,600	0,000	0,367	0,433	0,800	0,000	0,267	0,333	0,900	0,100	0,533	0,567	0,900	0,000	0,067	0,133	0,300
RQ10	0,000	0,333	0,367	0,600	0,000	0,100	0,200	0,500	0,000	0,067	0,133	0,300	0,100	0,533	0,567	0,900	0,000	0,067	0,133	0,300

Fonte: Dados da pesquisa. (2021)

### 4.3 Ordenação dos requisitos para desenvolvimento

Após a agregação das avaliações de complexidade, em uma única representação *Fuzzy* Trapezoidal, conforme seção 3.3.2, os requisitos são ordenados aplicando os pesos obtidos na avaliação de importância pelos especialistas conforme explicado na seção 3.3.3 e dessa forma obtém-se a pontuação para cada fator de impacto e conseqüentemente para cada requisito.

A tabela completa com a posição da ordenação final dos requisitos de *software* do projeto com os fatores de impacto e seus respectivos valores está disponível no APÊNDICE C. Já a ordenação final dos requisitos foi sintetizada e apresentada a seguir na Tabela 13.

**Tabela 13 - Ordenação dos Requisitos para Desenvolvimento**

<i>Ranking</i>	$CC_i$	Alternativas
1º	0,631	RQ5
2º	0,622	RQ8
3º	0,609	RQ9
4º	0,607	RQ7
5º	0,581	RQ1
6º	0,499	RQ2
7º	0,478	RQ10
8º	0,104	RQ6
9º	0,101	RQ3
10º	0,051	RQ4

Fonte: Dados da pesquisa (2021).

O *Fuzzy* TOPSIS, baseia-se na escolha das melhores alternativas em relação a sua aproximação com a solução ideal positiva e maior distanciamento da solução ideal negativa (Chen, 2000) portanto conforme a Tabela 13, o requisito que obteve a maior aproximação a solução ideal positiva ( $CC_i$ ) foi o RQ5 (0,631) seguido do requisito RQ8 (0,622), ou seja, são os requisitos de *software* que foram considerados os mais complexos e portanto, requerem maior atenção no gerenciamento do projeto de desenvolvimento, enquanto que o requisito que obteve a menor aproximação a solução ideal positiva, e portanto ficando em último no *ranking* de complexidade, e assim demandando menor atenção, de acordo com as avaliações foi o RQ10 (0,051).

#### 4.3.1 Apresentação dos requisitos ordenados com os principais fatores de impacto

Com base nos dados disponíveis no APÊNDICE B os requisitos foram ordenados e todos os seus fatores de impacto também. Foi aplicado uma linha de corte, em que foram selecionados os 4 fatores de impacto com maior valor de complexidade obtido de acordo com as avaliações realizadas, e então ordenados e listados na Tabela 14. A tabela completa está disponível no APÊNDICE D.

**Tabela 14 - Requisitos de *Software* ordenados e com os principais fatores de impactos também ordenados**

Ordem	Requisito	Ordenação Fator	Fatores de Impacto	$CC_i$	Dimensão
1	RQ5 0,631	1	Experiência de aplicações (FI13)	0,550	De pessoal
		2	Cronograma de Desenvolvimento Requerido (FI19)	0,484	De projeto
		3	Restrição de tempo de execução (FI1)	0,468	Técnica
		4	Uso de ferramentas de software (FI17)	0,467	De projeto
2	RQ8 0,622	1	Confiabilidade de <i>software</i> exigida (FI5)	0,517	De produto
		2	Linguagem e experiência com ferramentas (FI15)	0,484	De projeto
		3	Cronograma de Desenvolvimento Requerido (FI19)	0,349	De projeto
		4	Complexidade do produto (FI7)	0,347	De produto
3	RQ9 0,609	1	Confiabilidade de software exigida (FI5)	0,501	De produto
		2	Linguagem e experiência com ferramentas (FI15)	0,500	De projeto
		3	Cronograma de Desenvolvimento Requerido (FI19)	0,349	De projeto
		4	Complexidade do produto (FI7)	0,347	De produto
4	RQ7 0,607	1	Linguagem e experiência com ferramentas (FI15)	0,451	De projeto
		2	Confiabilidade de <i>software</i> exigida (FI5)	0,351	De produto
		3	Cronograma de Desenvolvimento Requerido (FI19)	0,349	De projeto
		4	Complexidade do produto (FI7)	0,347	De produto
5	RQ1	1	Linguagem e experiência com ferramentas (FI15)	0,550	De projeto

		2	Documentação compatível com as necessidades do ciclo de vida (F9)	0,516	De produto
	0,581	3	Complexidade do produto (FI7)	0,501	De produto
		4	Capacidade do programador (FI11)	0,346	De pessoal
	RQ2	1	Experiência de plataforma (FI14)	0,346	De pessoal
		2	Linguagem e experiência com ferramentas (FI15)	0,327	De projeto
6		3	Complexidade do produto (FI7)	0,310	De produto
	0,499	4	Cronograma de Desenvolvimento Requerido (FI19)	0,295	De projeto
	RQ10	1	Cronograma de Desenvolvimento Requerido (FI19)	0,349	De projeto
		2	Complexidade do produto (FI7)	0,347	De produto
7		3	Confiabilidade de software exigida (FI5)	0,328	De produto
	0,478	4	Experiência de plataforma (FI14)	0,327	De pessoal
	RQ6	1	Capacidade do analista (FI11)	0,198	De pessoal
		2	Confiabilidade de software exigida (FI5)	0,157	De produto
8		3	Cronograma de Desenvolvimento Requerido (FI19)	0,135	De projeto
	0,104	4	Continuidade de pessoal (FI16)	0,114	De projeto
	RQ3	1	Restrição de tempo de execução (FI1)	0,237	Técnica
		2	Cronograma de Desenvolvimento Requerido (FI19)	0,135	De projeto
9		3	Experiência de aplicações (FI13)	0,114	De pessoal
	0,101	4	Continuidade de pessoal (FI16)	0,114	De projeto
	RQ4	1	Cronograma de Desenvolvimento Requerido (FI19)	0,135	De projeto
		2	Experiência de aplicações (FI13)	0,114	De pessoal
10		3	Continuidade de pessoal (FI16)	0,114	De projeto
	0,051	4	Linguagem e experiência com ferramentas (FI15)	0,092	De projeto

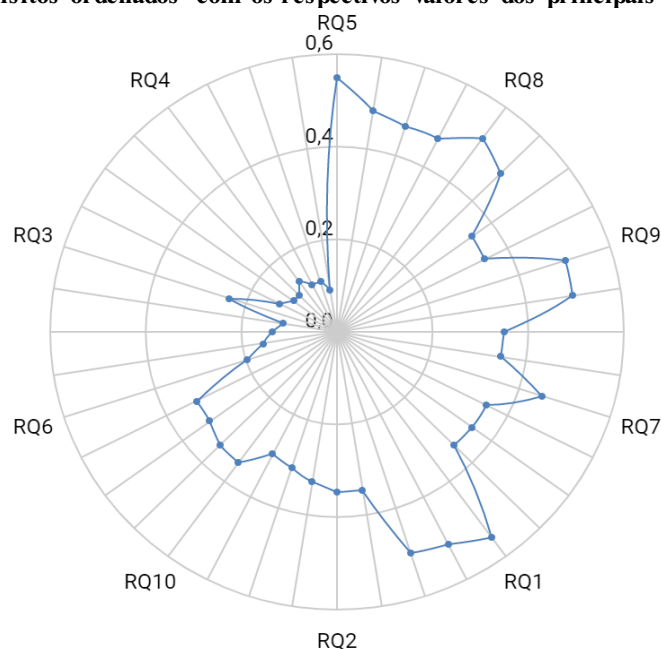
**Fonte: Dados da pesquisa (2021)**



Ao observar os dados apresentados na Tabela 14 pode-se observar que apenas um fator de impacto da dimensão técnica apareceu, o fator Restrição de Tempo de Execução e apenas para os requisitos RQ5 que é o primeiro da ordenação final e novamente no requisito RQ3 que ocupa a nona posição.

Ao considerar os requisitos ordenados e seus fatores de impacto de maior complexidade avaliada, conforme mostra a Tabela 14 acima, foi elaborado o gráfico da Figura 14 para apresentar de forma visual tais dados.

**Figura 14 - Requisitos ordenados com os respectivos valores dos principais fatores ordenados**

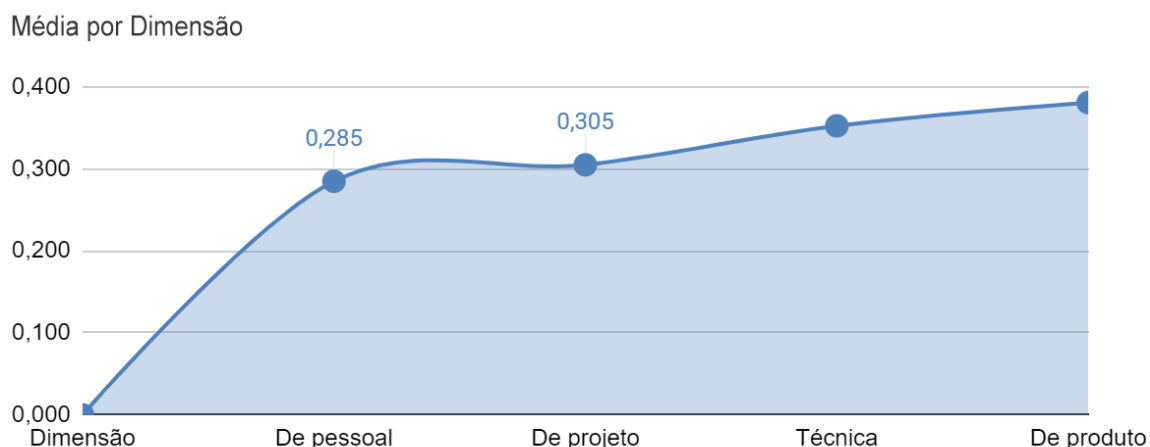


**Fonte: Dados da pesquisa (2021).**

Ao observar a disposição dos requisitos ordenados com apenas a pontuação dos quatro fatores de impacto avaliados com maior complexidade, o RQ5 se destaca dos demais mantendo os fatores de impacto com altos valores. Enquanto que o RQ4 apresentou o valor final bem abaixo dos demais.

É possível analisar o reflexo da ordenação dos requisitos de *software* e também dos principais fatores de impacto conforme a avaliação de complexidade quanto às dimensões que tais fatores estão agrupados, conforme a Figura 15 que apresenta uma média dos valores de  $CC_i$  índice de proximidade do *Fuzzy* TOPSIS obtidos por dimensão.

**Figura 15 - Média dos valores de CCI por dimensão**

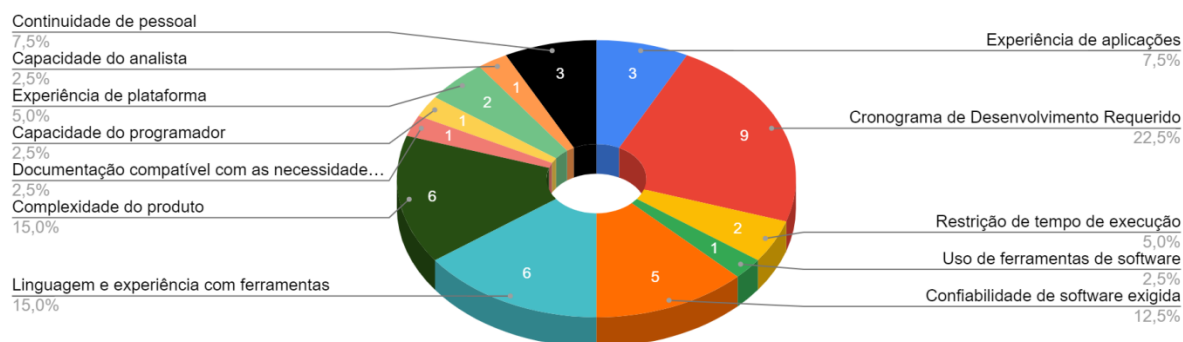


**Fonte: Dados da pesquisa (2021).**

No âmbito dos fatores de impacto ranqueados entre os quatro de maior pontuação, ao analisar o gráfico da Figura 15 o qual apresenta a média das pontuações obtidas, é possível perceber que a dimensão de produto (0,383), foi a que obteve a maior pontuação quanto a complexidade dos requisitos avaliados, enquanto que a dimensão de pessoal (0,285) foi a que obteve a menor. Nesse sentido, pode-se afirmar às questões como documentação necessária, facilidades para gerar eficiência para os usuários, e complexidades relacionadas às necessidades da área de negócio que o *software* em desenvolvimento atende, são os fatores de impacto que mais geram complexidade para o desenvolvimento dos requisitos avaliados. Além disso, questões atreladas a experiência da equipe com relação a linguagens de programação, ou a plataformas usadas para o desenvolvimento, e a rotatividade de profissionais não foi algo que aumentou de forma significativa a complexidade para concluir os requisitos deste projeto em questão.

No gráfico da Figura 16 é apresentada a média dos valores de CCI índice de proximidade do *Fuzzy-TOPSIS* obtidos, considerando apenas os fatores de impacto que ficaram nas quatro primeiras posições de complexidade para os requisitos avaliados.

**Figura 16 - Média dos valores de CCI dos fatores de impacto mais complexos**

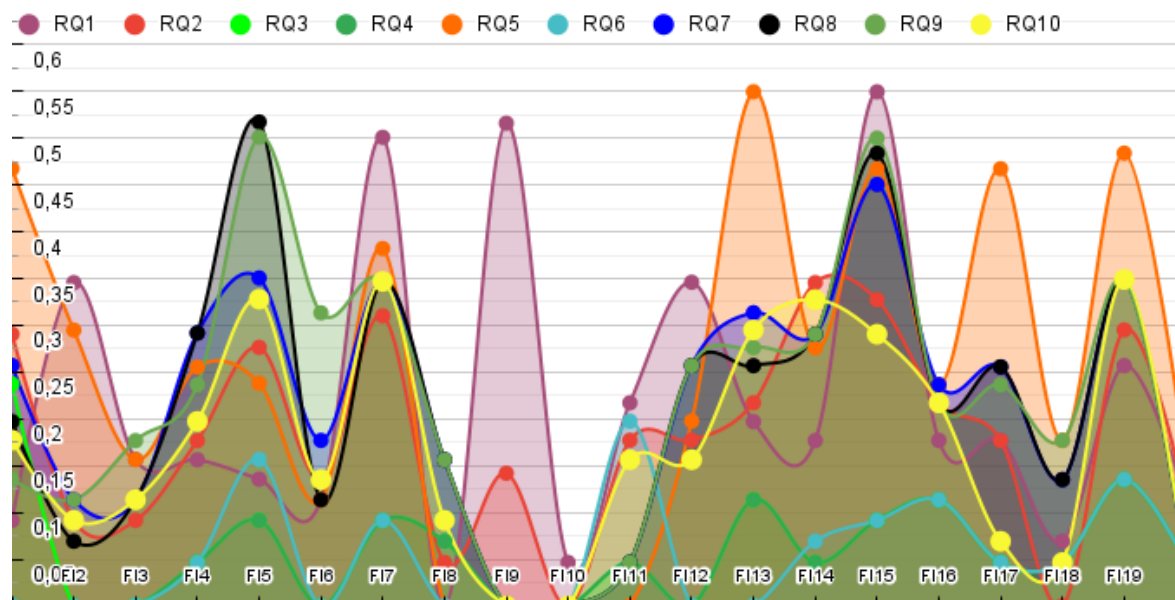


Fonte: Dados da pesquisa (2021).

Assim, em análise aos resultados apresentados na Figura 16 é possível identificar que Cronograma de Desenvolvimento Requerido (22%), Linguagem e experiência com ferramentas (15%), Complexidade do produto (15%) e Confiabilidade de *software* exigida (12,5%), foram os fatores de impacto que se fizeram presente maior número de vezes entre os quatro principais fatores de impacto quanto a complexidade para os requisitos de *software* avaliados.

Ao considerar todos os fatores de impacto em cada requisito de *software* avaliado, pode-se observar na Figura 17, quais os fatores foram avaliados com maior complexidade.

**Figura 17 - Requisitos versus fatores de impacto e suas pontuações**



Fonte: Dados da pesquisa (2021).

A Figura 17, demonstra que ao considerar todos os fatores de impacto para o requisitos avaliados, os fatores Restrição de tempo de execução (FI1), Confiabilidade de *software* exigida (FI5), Complexidade do produto (FI7), Documentação compatível com as necessidades do

ciclo de vida (FI9), Experiência de aplicações (FI13), Linguagem e experiência com ferramentas (FI15), Uso de ferramentas de *software* (FI17), Cronograma de Desenvolvimento Requerido (FI19) foram considerados mais complexos, pois obtiveram as maiores valores de CCI (índice de proximidade do *Fuzzy TOPSIS*) em determinados requisitos, contudo os fatores Confiabilidade de software exigida (FI5), Complexidade do produto (FI7), Linguagem e experiência com ferramentas (FI15) e Cronograma de Desenvolvimento Requerido (FI19) foram considerados com complexidade elevada em vários requisitos.

#### 4.3.2 Avaliação da percepção da aplicação e aceitação do modelo elaborado

A avaliação da percepção referente a aplicação do modelo, coletada por meio do *Google Forms* é apresentada em sua íntegra no APÊNDICE F.

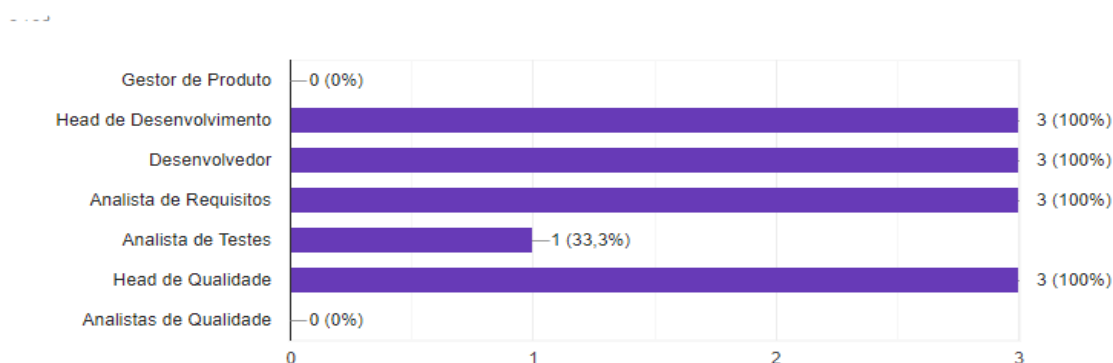
Quanto à percepção de relevância questionada na primeira pergunta, os respondentes dentre as opções irrelevante, pouco relevante, moderadamente relevante, muito relevante e extremamente relevante, consideraram que os fatores de impacto são Muito Relevantes e Extremamente Relevantes, não houve outras marcações.

Em relação a questão sobre a experiência e a percepção sobre o impacto da aplicação do modelo de apoio para o gerenciamento do projeto, foi comentado que seria importante aplicar em mais ciclos, também foi sugerido usar os resultados obtidos com a aplicação do modelo como indicadores de desempenho.

As funções que na visão dos respondentes devem participar da avaliação de complexidade dos requisitos são apresentados na Figura 18.

**Figura 18 - Funções que devem participar da avaliação na percepção dos respondentes.**

Na sua opinião, qual desses papéis deve participar da avaliação de complexidade do modelo aplicado?



Fonte: Dados da pesquisa (2021)

Ao avaliar a Figura 18, pode-se constatar que foi unânime que as funções de *Head* de desenvolvimento, desenvolvedor, analista de requisitos e *Head* de qualidade devem realizar a avaliação de complexidade dos requisitos de *software* do projeto em desenvolvimento. Além disso, o analista de teste também é tido como necessário para a avaliação por 33,3%. Não houve votação para que o analista de qualidade nem o gestor de produto avaliem.

Em relação à questão para comentarem a percepção quanto à ferramenta utilizada na aplicação do modelo proposto, em relação a sua facilidade de uso ou outros pontos que julgaram importantes, foi apontada a quantidade de fatores de impacto, um ponto de atenção, podendo ser um processo cansativo conforme a quantidade de requisitos a serem avaliados, embora a forma de responder a avaliação, foi considerada fácil: *“Foi tranquilo, super fácil de responder tendo a descrição dos fatores pra consultar quando havia dúvidas, um pouco cansativo se for fazer a avaliação de muitos requisitos”*. Ou ainda: *“A forma de resposta era fácil, não houve uma dificuldade, mas ficou cansativo quanto a quantidade de fatores a serem avaliados”*. Nessa mesma ótica, outro respondente afirmou: *“A necessidade de rever os conceitos de tantos fatores tornou um pouco cansativo, embora a forma de responder a avaliação tenha sido muito simples e prática e acho que isso pode dificultar a adesão”*.

Outro ponto importante referente a percepção, foi que no dia-a-dia do processo de desenvolvimento os fatores de impacto são considerados empiricamente, e que ao ter que formalizar essa avaliação, gera maior aprofundamento e detalhamento do fator o que pode ser considerado algo bom para o processo: *“embora se saiba que são fatores que consideramos normalmente de forma subjetiva e até mentalmente, mas ao ter que definir e trazer para fora do 'pensamento', faz considerar com mais profundidade do que quando é só no pensamento”*.

Os respondentes deixaram como sugestão utilizar alguma forma de selecionar os fatores de impacto como forma de reduzir os fatores aplicados para a avaliação, conforme segue: *“Deixar a possibilidade de selecionar quais destes fatores são mais relevantes para cada projeto, por exemplo dentre os 20, escolher 10 deles e aí avaliar a complexidade somente dos 10 escolhidos”* e *“Algum checklist ou algo do tipo para selecionar/reduzir os fatores de impacto a serem avaliados conforme o projeto”*. Além disso, também foi sugerido apresentar algum exemplo de uso do fator em sua descrição: *“apresentar uma descrição com exemplos, pode simplificar bastante”*.

Em relação ao TAM o resultado completo está no Apêndice G, foram atribuídos os adjetivos a seguir: assertivo; diferente; promissor interessante; cansativo; valoroso, importante, diferente, eficaz. Já em relação a deixar o modelo mais simples de utilizar houveram sugestões para disponibilizar em um sistema : *“Uma interface mais amigável, um sistema específico*

*possivelmente com flexibilidade para seleção de fatores.”, reduzir a quantidade de fatores de impacto: “Diminuir a quantidade de fatores e refinar as descrições deles” além de fornecer exemplos dos fatores de impacto: “Implementado em um sistema integrado com a visualização dos detalhes do projeto e exemplos do uso dos fatores em relação aos requisitos”*

Com relação aos constructos de Facilidade de uso percebida e Utilidade percebida os participantes houve ponto de atenção para a variável “*A Avaliação de Complexidade do Requisitos é Rápida*” pois houveram respostas “Neutro”, assim como a variável “*A descrição dos fatores de impacto do modelo, são de fácil entendimento*”. As demais prevaleceram na resposta positiva escala de “*Concordo*” ou “*Concordo Totalmente*”

#### 4.3.3 Comparativo com trabalhos relacionados

A técnica AHP utiliza comparação entre pares de requisitos, recebendo avaliações de usuários humanos para embasar sua definição de prioridades (OGNJANO VIC et al., 2011), (ALAWNEH, 2018). Assim como a AHP, as técnicas variáveis para, posteriormente, definir seus grupos de prioridades. Portanto, a principal diferença entre essas técnicas diz respeito aos critérios de avaliação e a equipe envolvida.

Shao *et al.* (2017) apresentaram o método DRank baseado em preferências e dependências para a priorização de requisitos de software visando utilizar essas dependências para melhorar a qualidade da priorização de acordo com as preferências dos *stakeholders* de um projeto. AHMAD et. al. (2018) desenvolveram um sistema *fuzzy* em termos de variáveis de entrada/saída. Os parâmetros selecionados para entrada são Custo, Tempo de Design, Desempenho, Tempo de Resposta, Número de Partes Interessadas. Diferentemente do modelo aqui proposto, pois a equipe envolvida é apenas a equipe técnica. Além disso os critérios utilizados também se distinguem com relação aos fatores de impacto.

Já ROTTOLI e CASANOVA(2021) propuseram um método que considera as opiniões de muitos especialistas sobre critérios de decisão múltipla, usando rótulos linguísticos *fuzzy*, o que permite capturar a imprecisão do julgamento de cada especialista. As opiniões são agregadas usando um operador de Iowa linguístico guiado por maioria considerando diferentes pesos para cada especialista e, em seguida, os requisitos são priorizados considerando as opiniões agregadas e diferentes pesos para cada dimensão avaliada.

## 5 CONCLUSÃO

O processo de desenvolvimento de *software* é uma atividade complexa . Deste modo o gerenciamento de projetos de desenvolvimento de *software* precisa corresponder em termos de assertividade e eficácia. Para alcançar o sucesso de um projeto de implementação de *software*, seja ele para um novo produto ou a manutenção de produto existente, é necessário tomar decisões de forma rápida e assertiva, as ferramentas de apoio têm um papel importante nesse processo. Deste modo, este trabalho propôs o desenvolvimento de um modelo de apoio ao gerenciamento de projetos de desenvolvimento de *software*, que por meio de avaliação qualitativa dos fatores de impacto, quanto a sua complexidade em relação a requisitos de *software*, usando os métodos multicritério AHP e *Fuzzy* TOPSIS para o processamento das avaliações. A partir das informações de revisões sistemáticas de literatura e na adaptação de metodologias existentes o modelo proposto introduz a aplicação dos fatores de impacto de modelos de estimativas de *software*, na avaliação de complexidade de requisitos de *software*.

Para a elaboração do modelo, foram submetidos os fatores de impacto para avaliação de importância por especialistas, em relação a construção de *software web* de maneira geral. Para o desenvolvimento dessas avaliações, foi aplicado o método multicritério AHP, e assim foram obtidos os pesos para cada fator de impacto do modelo. Os pesos obtidos são a base para a aplicação em projetos de desenvolvimento de *software web*, a partir do modelo elaborado.

O modelo é aplicado por meio de avaliações qualitativas da complexidade dos fatores de impacto para o desenvolvimento de requisitos de *software*, com base na determinação da complexidade com o Peso Linguístico *Fuzzy* TOPSIS. O modelo foi validado na aplicação do desenvolvimento de projeto de manutenção de um *software web*. Sendo assim, considerou-se que os objetivos foram atingidos.

De acordo com a avaliação das dimensões e seus fatores de impacto realizada pelos especialistas quanto a sua importância para o desenvolvimento de *software web*, a dimensão que se destacou como a mais importante foi da dimensão de produto (D2) (0,432) seguida respectivamente pela dimensão de pessoal (D3)(0,243), dimensão técnica(D1)(0,168) e por último a dimensão de projeto (D4)(0,157). Quanto aos fatores de impacto, considerando todos de maneira geral, independente as quais dimensões pertencem, os cinco fatores de impacto de maior importância de acordo com os especialistas são respectivamente os fatores Confiabilidade de *software* exigida (0,1265), Complexidade do produto (0,1153) e Eficiência do usuário final (0,0812), todos da dimensão de produto (D2), Restrição de tempo de execução (0,0705) e Capacidade do programador (0,0662) estes das dimensões técnica e de produto

respectivamente. Nesse sentido é possível afirmar que de acordo com os especialistas, os fatores de impacto da dimensão de produto são os fatores que possuem maior importância para o desenvolvimento de *software web* de maneira geral.

A aplicação do modelo proposto ocorreu em um projeto com dez requisitos de *software* em que a avaliação de complexidade dos requisitos em relação aos fatores de impacto resultou no RQ5 (0,631) como o mais complexo, enquanto que o RQ4 (0,051) como o menos complexo, ocupando a décima posição na ordenação. Embora tenha havido uma diferença significativa entre o primeiro e o último requisito, os fatores Experiência de aplicações e Cronograma de Desenvolvimento Requerido se fizeram presente em ambos os casos, ficando entre os quatro fatores de maior complexidade avaliada. Além disso, no projeto aplicado, o fator de Cronograma de Desenvolvimento Requerido esteve presente entre os fatores com maior complexidade, para nove dos dez requisitos avaliados, bem como os fatores e Complexidade do produto e Linguagem e experiência com ferramentas que se mantiveram em seis dos dez, seguido do fator Confiabilidade de software exigida que esteve entre os fatores de maior complexidade para os dez requisitos.

Dessa forma, para o cenário do projeto aplicado, a dimensão de produto se manteve em destaque quanto a seu impacto na complexidade dos requisitos do projeto aplicado, alcançando a média de 0,380 (relativa aos valores de índice de proximidade do *Fuzzy TOPSIS*), seguida da dimensão técnica que chegou a média de 0,350 na sequência, está a dimensão de projeto que obteve 0,305 e por último a dimensão de pessoal que resultou na média de 0,285.

O modelo proposto nesta dissertação pode contribuir para os gestores, oferecendo a contribuição do conhecimento empírico da equipe de forma simples, por meio das avaliações de complexidade. Além disso, pode servir como indicador de desempenho da equipe..

A contribuição prática desta dissertação ocorreu pela aplicação modelo proposto em uma empresa de desenvolvimento de *software* com experiência no desenvolvimento de projetos. Espera-se que com ajustes, possa ser aplicada em qualquer empresa com projetos de desenvolvimento de *software* de qualquer porte e independente da metodologia usada no processo de desenvolvimento, uma vez que a base teórica do modelo proposto permite a sua generalização. O modelo foi aplicado em um único projeto de desenvolvimento de *software web*, bem como em uma única organização, portanto, a aplicação em outras empresas pode contribuir no desenvolvimento do modelo proposto. Além disso, por meio dos construtos do Modelo de Aceitação de Tecnologia (TAM) de Davis (1989), constatou-se que o modelo teve uma boa aceitação pelos envolvidos no projeto aplicado, por perceberem a Utilidade Percebida e Facilidade de Uso, o que contribui efetivamente pela Atitude e Intenção Comportamental para



o Uso Real, embora tenha recebido sugestões de ajustes quanto a possibilidade de escolha e quantidade de fatores

Por fim, como trabalho futuro, se propõem a realização de estudo de caso múltiplo para aplicar o modelo em diferentes projetos e organizações. Como este estudo ocorreu em um projeto de manutenção de *software*, pode-se sugerir a aplicação em projeto de desenvolvimento de novo *software*. Além disso, sugere-se construir uma ferramenta computacional específica, que possibilite a aplicação do modelo de maneira mais ágil e com melhor experiência ao usuário avaliador, conforme seção 3.2.4 que foi apresentado uma sugestão em protótipo visual.

Como outro estudo que poderia ser realizado, se sugere a inserção de uma etapa anterior à avaliação dos requisitos, para que ao responder um breve questionário, sejam selecionados apenas os fatores de impacto que melhor se adequem ao contexto do projeto do *software* que esteja em desenvolvimento. Com essa nova etapa, se pode melhorar a praticidade da aplicação do modelo.

## REFERÊNCIAS

AASEM, M. Ramzan M. e A. Jaffar, "Análise e otimização de técnicas de priorização de requisitos de software", 2010 **International Conference on Information and Emerging Technologies** , 2010, pp. 1-6, doi: 10.1109/ICIET.2010.5625687.

ABBAS, M. et al. MBRP: Model-Based Requirements Prioritization Using PageRank Algorithm. 2019, **IEEE Computer Society**. p.31-38.

ABDELAZIM, K.; MOAWAD, R.; ELFAKHARANY, E. A Framework for Requirements Prioritization Process in Agile Software Development. 2020, 1: **Institute of Physics Publishing**.

ACHIMUGU, Philip; SELAMAT, A.; IBRAHIM, R.; MAHRIN, M. N. R. A systematic literature review of software requirements prioritization research. **Information and software technology**, v. 56, n. 6, p. 568-585. 2014.

ADHIM, K. K.; HUDAIB, A.; AL-SHBOUL, B. Efficient requirement prioritization based on enhanced multi-verse optimizer. **Journal of Theoretical and Applied Information Technology**, v. 97, n. 19, p. 4966-4977, 2019. ISSN 19928645 (ISSN).

AFRAZ, S.; RASHIDI, H.; MIKAEILVAND, N. CSOP+RP: A novel constraints satisfaction model for requirements prioritization in large-scale software systems. *Turkish Journal of Electrical Engineering and Computer Sciences*, v. 29, n. 7, p. 3074-3091, 2021. ISSN 13000632 (ISSN).

AHMAD, K. S. et al. Fuzzy-MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements. 2018, **Institute of Electrical and Electronics Engineers Inc.** p.433-437.

AHUJA, H.; SUJATA; PUROHIT, G. N. Understanding requirement prioritization techniques. In: SWAROOP, A.; SINGH, M., et al, 2017, **Institute of Electrical and Electronics Engineers Inc.** p.257-262.

AL-ADWAN, A.; ALADWAN, A. Using Interdependencias for the Prioritization and Reprioritization of Requirements in Incremental Development. **International Journal of Advanced Computer Science and Applications**, v. 11, n. 11, p. 224-230, 2020. ISSN 2158107X (ISSN).

ALAWNEH, L. Requirements prioritization using hierarchical dependencies. LATIFI, S.: Springer Verlag. 558: 459-464 p. 2018.

ALBRECHT, A. J., **Measuring application development productivity**, Joint Application Development Symposium, 1979, p. 83-92.

ALBUGA, S.; ODEH, Y. Towards Prioritizing Software Business Requirements in Startups. 2018, **Institute of Electrical and Electronics Engineers Inc.** p.257-265.

ALI, A. et al. Role of Requirement Prioritization Technique to Improve the Quality of Highly-Configurable Systems. **IEEE Access**, v. 8, p. 27549-27573, 2020. ISSN 21693536 (ISSN).

ALI, S. et al. Aspect-based requirements mining technique to improve prioritisation process: Multi-stakeholder perspective. **IET Software**, v. 14, n. 5, p. 482-492, 2020. ISSN 17518806 (ISSN).

AL-TA'ANI, R. H.; RAZALI, R. Process model for systematic requirements prioritisation process in an agile software development environment based on 5S approach: Empirical study. **Journal of Theoretical and Applied Information Technology**, v. 95, n. 8, p. 1715-1736, 2017. ISSN 19928645 (ISSN).

ALZQAQBAH, A.; MASADEH, R.; HUDAIB, A. Whale optimization algorithm for requirements prioritization. 2018, **Institute of Electrical and Electronics Engineers Inc.** p.84-89.

AMAZAL, F. A., IDRI, A. E ABRAN, A. (2014). **Improving fuzzy analogy based software development effort estimation**. Em Software Engineering Conference (APSEC), 2014 21st Asia-Pacific, volume 1, páginas 247–254. IEEE.

AMELIA, T.; MOHAMED, R. B. A Proposed Requirements Prioritization Model Based on Cost-Value Approach with Collaboration Perspective. 2021, **Institute of Electrical and Electronics Engineers Inc.** p.10-15.

ARRUDA, D. et al. Engenharia de Requisitos : Um Survey Realizado no Porto Digital, Recife/Brasil. 2014.

ASSOCIAÇÃO BRASILEIRA DE EMPRESAS DE SOFTWARE, ABES. **Estudo do Mercado Brasileiro de Software: Panoramas e Tendências 2019**. 2005-. Anual.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12207: Tecnologia de Informação - Processos de ciclo de vida de software**. Rio de Janeiro, 2009.

AZIZ, N. A. A.; HASSAN, S.; ADMODISASTRO, N. Prioritizing based on crowd preferences to requirements elicited from crowd's sentiments. **International Journal of Innovative Technology and Exploring Engineering**, v. 8, n. 8 S, p. 341-346, 2019. ISSN 22783075 (ISSN).

BARBOSA, P. A. M. et al. Selection and prioritization of software requirements using the Verbal Decision Analysis paradigm. 2017, **Knowledge Systems Institute Graduate School**. p.213-217.

BARBOSA, P. A. M. et al. Selection and Prioritization of Software Requirements Applying Verbal Decision Analysis. **Complexity**, v. 2019, 2019. ISSN 10762787 (ISSN).

BARBOSA, P. A. M.; PINHEIRO, P. R.; SILVEIRA, F. R. V. Towards the verbal decision analysis paradigm for implementable prioritization of software requirements. **Algorithms**, v. 11, n. 11, 2018. ISSN 19994893 (ISSN).

BECK, K. **Extreme Programming explained: Embrace Change**. Edição US. Boston: Addison-Wesley, 2000.

BIRGUN CERKEZOGLU, B. T., S. A Systematic Approach for Improving the Software Management Process. **International Journal of Innovation and Technology Management**, v. 16, n. 4, 2019.

BOEHM, Barry W. **Software Cost Estimation Meets Software Diversity**, IEEE International Conference on Systems (IEEE/ACM), 2017.

BOEHM, Barry W. **Software engineering economics**. New Jersey: Prentice-Hall, 1981.

BOEHM, Barry W., FAIRLEY, Richard E. Software estimation perspectives. **IEEE Software**, v. 17, n. 6, November 2000, p. 22-26, 2000.

BOEHM, Barry, C. et al. **Software Cost Estimation with COCOMO II**, Prentice Hall, 2000.

BORHAN, N.H. et al. Requirements prioritization techniques focusing on agile software development: A systematic literature review. **International Journal of Scientific and Technology Research**, v. 8, n. 11, p. 2118-2125, 2019. ISSN 22778616 (ISSN).

BRAGA, Antônio. **Análise de pontos de função**. Rio de Janeiro: Infobook, 1996.

BUSETTA, P. et al. Tool-Supported Collaborative Requirements Prioritisation. In: DEMARTINI, C.;CONTE, T., et al, 2017, **IEEE Computer Society**. p.180-189.

C. Jones, **Estimating Software Costs**, McGraw Hill, 1998.

CABRERA, M. A. C. et al. Prioritization of non-functional requirements in a mobile application for panic button system using neutrosophic decision maps. **Neutrosophic Sets and Systems**, v. 34, p. 153-158, 2020. ISSN 23316055 (ISSN).

CANTANHEDE, Marco André Dias. **Lean thinking em desenvolvimento de software**: estudo e aplicação de ferramenta para avaliação do lean em software. 2014. 142 f. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Tecnologia, Limeira, SP.

CARIGNANO, C. G. **Tomada de decisões em cenários complexos**: introdução aos métodos discretos de apoio à decisão. São Paulo: Thompson Learning. 2005.

CAUCHICK, M. P. A.; RUI, S. O Método do Estudo de Caso na Engenharia de Produção. **Elsevier**. v.2, pp. 131-148, 2012.

CAVALCANTI, C. et al. Mechanisms to support requirements prioritization: A systematic mapping review. In: SILVA, C.;HADAD, G., et al, 2018, **PUC-Rio**, Pontificia Universidade Catolica do Rio de Janeiro.

CHALLA, Jagat Sesh et al. Integrated software quality evaluation: a fuzzy multi-criteria approach. **Journal of Information Processing Systems**, v. 7, n. 3, p. 473-518, 2011.

CHEN, C. T Extensions of the TOPSIS for group decision-making under fuzzy environment. **Fuzzy Sets and Systems**, v. 114, n. 1, p. 1-9, 2000 [http://dx.doi.org/10.1016/S0165-0114\(97\)00377-1](http://dx.doi.org/10.1016/S0165-0114(97)00377-1) .

CHEN, C.T., Extensions of the TOPSIS for group decision-making under fuzzy environment, **Fuzzy Sets Syst**. 114 2000 1–9.

CMUSEI, Carnegie Mellon University, **Software Engineering Institute**. CMMI® for Development, Version 1.2. Pensilvania: Carnegie Mellon University, 2006. 20p.

DANEVA, Maya et al. Agile Requirements Prioritization in Large-scale outsourced system Projects: An empirical study. **Journal of Software and Systems**, v. 86, ed. 5, p. 1333-1353, maio 2013.

DAVIS, A.M. The Art of Requirements Triage. IEEE Computer Society. 2003.

DEVADAS, R.; SRINIVASAN, G. N. Review of different fuzzy logic approaches for prioritizing software requirements. **International Journal of Scientific and Technology Research**, v. 8, n. 9, p. 296-298, 2019. ISSN 22778616 (ISSN).

DHINGRA, S. et al. Selection of prioritization technique for software requirement using Fuzzy Logic and Decision Tree. 2017, **Institute of Electrical and Electronics Engineers Inc.**

DOS SANTOS, J. R. F.; ALBUQUERQUE, A. B.; PINHEIRO, P. R. Requirements prioritization in market-driven software: A survey based on large numbers of stakeholders and requirements. In: PAULK, M.;BRITO, M. A., et al, 2017, **Institute of Electrical and Electronics Engineers Inc.** p.67-72.

DUBOIS, D., Prade, H.: **Fuzzy Sets and Systems: Theory and Applications**. Academic Press, New York 1980

FALAHAH. Defining and prioritizing software requirement using gIBIS and AHP method. **Universal Journal of Electrical and Electronic Engineering**, v. 6, n. 2, p. 32-44, 2019. ISSN 23323280 (ISSN).

FUGGETTA, Alfonso. **Software process: a roadmap**. In: **The Future of Software Engineering**, ICSE'2000, Limerick, Ireland, 2000.

GALARZA, F. P. C. et al. Prioritization of Software Requirements Using Neutrosophic TOPSIS. **Neutrosophic Sets and Systems**, v. 44, p. 199-206, 2021. ISSN 23316055 (ISSN).

GARG, U.; SINGHAL, A. Software requirement prioritization based on non-functional requirements. 2017, **Institute of Electrical and Electronics Engineers Inc.** p.793-797.

GIL, A. C. **Como elaborar projetos de pesquisa**. 6. ed. São Paulo: Atlas, 2017.

GILLAIN, J.; JURETA, I.; FAULKNER, S. Planning optimal agile releases via requirements optimization. 2017, **Institute of Electrical and Electronics Engineers Inc.** p.10-16.

GLIGORIC, Z., BELJIC, C., SIMEUNOVIC, V. Shaft Location Selection At Deep Multiple Orebody Deposit By Using Fuzzy TOPSIS Method And Network Optimization. **Expert Systems with Applications**, v.37, n.2, pp.c1408–1418, 2010.

GOMES, L. F. A. M., e Gomes, C. F. S. **Princípios e métodos para a tomada de decisão: Enfoque multicritério** (6a ed.). São Paulo: Atlas. 2019.

GONÇALVES, C. D. F.; DIAS, J. A. M.; MACHADO, V. A. C. Multi-criteria decision methodology for selecting maintenance key performance indicators. **International Journal of Management Science and Engineering Management**, v. 10, n. 3, p. 215-223, 2015.

GOVIL, N.; SHARMA, A. Information Extraction on Requirement Prioritization Approaches in Agile Software Development Processes. 2021, **Institute of Electrical and Electronics Engineers Inc.** p.1097-1100.

GULZAR, K. et al. Fuzzy approach to prioritize usability requirements conflicts: An experimental evaluation. **IEEE Access**, v. 5, p. 13570-13577, 2017. ISSN 21693536 (ISSN).

GUPTA, V. et al. Requirements engineering in software startups: A systematic mapping study. **Applied Sciences** (Switzerland), v. 10, n. 17, 2020. ISSN 20763417 (ISSN).

H.J. Zimmermann, **Fuzzy Set Theory and its Applications**, second ed., Kluwer Academic Publishers, Boston, Dordrecht, London, 1991.

HAIDER, W. et al. Improving requirement prioritization and traceability using artificial intelligence technique for global software development. 2019, **Institute of Electrical and Electronics Engineers Inc.**

HÅKANSSON, A. Portal of Research Methods and Methodologies for Research Computer Engineering, and Applied Computing, **WORLDCOMP'13**; Las Vegas, Nevada, USA, p. 22-25, 2013

HAMDY, A.. Genetic fuzzy system for enhancing software estimation models, **International Journal of Modeling and Optimization**, Vol. 4, No. 3, June 2014, páginas 227–232. 2014

HAZAN, C. Análise de pontos de função - uma aplicação nas estimativas de tamanho de projetos de software, **Engenharia de Software Magazine**, No. 2, 2009, p. 25-31.

HICDURMAZ, M.: A fuzzy multi criteria decision making approach to software life cycle model selection. In: **38th Euromicro Conference on Software Engineering and Advanced Applications**, pp. 384–391 2012

HOTA, H. S.; SINGHAI, Sanjay Kumar; SHUKLA, Ragini. Application of Fuzzy Analytic Hierarchy Method in Software Engineering Scenario. **International Journal of Computer Applications**, v. 57, n. 21, 2012.

HUDAIB, A.; ALHAJ, F. Self-organizing maps for agile requirements prioritization. 2019, **Institute of Electrical and Electronics Engineers Inc.**

HUDAIB, A.; MASADEH, R.; ALZAQBAH, A. WGW: A hybrid approach based on whale and grey wolf optimization algorithms for requirements prioritization. **Advances in Systems Science and Applications**, v. 18, n. 2, p. 63-83, 2018. ISSN 10786236 (ISSN).

HUJAINAH, F. et al. Interrelated Elements in Formulating an Efficient Requirements Prioritization Technique: Review. 2020, **Institute of Electrical and Electronics Engineers Inc.** p.97-101.

HUJAINAH, F. et al. Software Requirements Prioritisation: A Systematic Literature Review on Significance, Stakeholders, Techniques and Challenges. **IEEE Access**, v. 6, p. 71497-71523, 2018. ISSN 21693536 (ISSN).

HUJAINAH, F. et al. SRPTackle: A semi-automated requirements prioritisation technique for scalable requirements of software system projects. **Information and Software Technology**, v. 131, 2021. ISSN 09505849 (ISSN).

IBRIWESH, I. et al. Prioritizing solution-oriented software requirements using the multiple perspective prioritization technique algorithm: An empirical investigation. **Concurrent Engineering Research and Applications**, v. 27, n. 1, p. 68-79, 2019. ISSN 1063293X (ISSN).

IJAZ, K. B.; INAYAT, I.; ALLAH BUKHSH, F. Non-functional Requirements Prioritization: A Systematic Literature Review. In: STARON, M.;CAPILLA, R., et al, 2019, **Institute of Electrical and Electronics Engineers Inc.** p.379-386.

ISHIZAKA A, LABIB A. Review of the main developments in the analytic hierarchy process. *Expert Syst Appl* 2011;v.38:14336–14345.

ISO/IEC 20926:2009, Software and systems engineering - Software measurement. **IFPUG functional size measurement method 2009**, ISO/IEC, 2009.

JONES, C. *Applied Software Measurement: Global Analysis of Productivity and Quality*, 3ed. McGraw Hill. 2008.

JØRGENSEN, M. “Evidence-based guidelines for assessment of software development cost uncertainty”, *Software Engineering*, **IEEE Transactions** on, vol. 31–11, 2005, pp. 942–954.

JØRGENSEN, M.; Shepperd, M. “A systematic review of software development cost estimation studies”, *Software Engineering*, **IEEE Transactions** on, vol. 33–1, 2007, pp. 33–53.

KAMAL, T. et al. Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development. **IEEE Access**, v. 8, p. 44714-44726, 2020. ISSN 21693536 (ISSN).

KARLSSON, J.: Software requirements prioritizing. **In: Proceedings of the Second International Conference on Requirements Engineering**, pp. 110–116 1996.

KAUFMANN, A., M.M. Gupta, **Introduction to Fuzzy Arithmetic: Theory and Applications**, Van Nostrand Reinhold, New York, 1991.

KAZEMIFARD, M., Z. A. N. M. A. M. F. Fuzzy emotional cocomo ii software cost estimation (fecsce) using multi-agent systems. **Applied Software Computing Journal**, 11(12):2260–2270. 2011

KIFETEW, F. et al. DMGame: A Gamified Collaborative Requirements Prioritisation Tool. 2017, **Institute of Electrical and Electronics Engineers Inc.** p.468-469.

KIFETEW, F. M. et al. Towards multi-decision-maker requirements prioritisation via multi-objective optimisation. In: FRANCH, X.;WIERINGA, R., et al, 2017, **CEUR-WS.** p.137-144.

KOLPONDINOS, M. Z. H.; GLINZ, M. Behind Points and Levels-The Influence of Gamification Algorithms on Requirements Prioritization. 2017, **Institute of Electrical and Electronics Engineers Inc.** p.332-341.

LEE, O. L.; KAMALRUDIN, M.; SIDEK, S. Requirements prioritization using risk assessment for analyzing requirements. **International Journal of Recent Technology and Engineering**, v. 8, n. 2 Special Issue 11, p. 785-791, 2019. ISSN 22773878 (ISSN).

LIMA JUNIOR, Francisco Rodrigues; CARPINETTI, Luiz Cesar Ribeiro. A comparison between TOPSIS and *Fuzzy* TOPSIS methods to support multicriteria decision making for supplier selection. *Gestão & Produção*, v. 22, n. 1, p. 17-34, 2015.

MISAGHIAN, N.; MOTAMENI, H.; RABBANI, M. Prioritizing interdependent software requirements using tensor and fuzzy graphs. **Turkish Journal of Electrical Engineering and Computer Sciences**, v. 27, n. 4, p. 2697-2717, 2019. ISSN 13000632 (ISSN).

MKPOJIOGU, E. O. C.; HASHIM, N. L. Quality-based prioritization: An approach for prioritizing software requirements. **Journal of Telecommunication, Electronic and Computer Engineering**, v. 9, n. 2-2, p. 17-21, 2017. ISSN 21801843 (ISSN).

MOHARRERI, K., SAPRE, A. V., RAMANATHAN, J. e RAMNATH, R. Cost-effective supervised learning models for software effort estimation in agile environments. Em 2016 **IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)**, volume 2, páginas 135–140. IEEE.

MORABITO, R. et al. **Modelagem e Simulação. In: Metodologia de pesquisa em engenharia de produção e gestão de operações.** 2. Ed. ed. [s.l.] Elsevier Editora Ltda., 2018.

MUÑANTE, D.; KIFETEW, F. M.; ALBRECHT, O. Modelling prioritisation decision-making in software evolution. In: AMELLER, D.; DIESTE, O., et al, 2017, **CEUR-WS**.

NGUYEN, V.; Steece, B.; Boehm, B. A constrained regression technique for cocomo calibration. **In: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement**, 2008, pp. 213–222.

NIKITINA, N.; KAJKO-MATTSSON, M.; STRALE, M. **From scrum to scrumban: A case study of a process transition.** 2012 International Conference on Software and System Process (ICSSP), p. 140-149, jun. 2012.

NISAR M, WANG Y-J, ELAHI M. Software development effort estimation using fuzzy logic—a survey. **In: Fifth international conference on fuzzy systems and knowledge discovery**, vol 1. IEEE, Shandong, China, pp 421–427, 2008

NIVOIT, J. B. Henri. **Issues in Strategic Management of Large-Scale Software Product Line Development. Composite Information Systems Laboratory (CISL).** Massachusetts Institute of Technology Cambridge. 2013.

NUSEIBEH, B. & EASTERBROOK, S. Requirements engineering: a roadmap. In: **Proceedings of the Conference on the Future of Software Engineering.** ACM, 2000



OCHODEK, Mirosław; KOPCZYŃSKA, Sylwia. Perceived importance of agile requirements engineering practices—a survey. **Journal of Systems and Software**, v. 143, p. 29-43, 2018.

PAETSCH, F. A. Eberlein and F. Maurer, "Requirements engineering and agile software development," WET ICE 2003. Proceedings. Twelfth **IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises**, 2003., 2003, pp. 308-313, doi: 10.1109/ENABL.2003.1231428.

PERNSTAL, J.; FELDT, R.; GORSCHKEK, T. The lean gap: A review of lean approaches to large-scale software systems development. **Journal of Systems and Software**, v. 86, n. 11, p. 2797–2821, 2013.

PMI. **Um guia do conhecimento em gerenciamento de projetos**. Guia PMBOK. 5. ed. EUA: Project Management Institute, 2013.

POPPENDIECK, M.; POPPENDIECK, T. **Lean Software Development: An Agile Toolkit**. Addison-Wesley Professional, 2003. p. 203 Prentice Hall, 2011.

PRESSMAN, Roger S. **Software Engineering: a practitioner's approach**. 8<sup>a</sup> ed. Nova Iorque:McGraw-Hill, 2014.

PRESSMAN, Roger S. **Software Engineering**, McGraw Hill Higher Education. 2005.

PRESSMAN, Roger. **Engenharia de software**. 7 ed. Porto Alegre: AMGH, 2011.

Projects and Degree Projects. The 2013 World Congress in Computer Science, **True Planning User Manual**, Price Systems. 2002-2016.

PUTNAM, L. H. "A general empirical solution to the macro software sizing and estimating problem.", **IEEE Trans. Software Eng.**, vol. 4-4, 1978, pp. 345–361.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informações**. Rio de Janeiro: Brasport, 1999.

ROTTOLI, G. D.; CASANOVA, C. Multi-criteria group requirement prioritization in software engineering using fuzzy linguistic labels. In: FLOREZ, H.; GALPIN, I., et al, 2021, **CEUR-WS**. p.16-28.

ROY, B. Multicriteria methodology for decision aiding. Netherlands: Kluwer Academic Publishers, 1996

SAATY, T. L., **Fundamentals of Decision Making and Priority Theory With the Analytic Hierarchy Process**. Pittsburgh: RWS publications, 2006. 478 p.

SACHDEVA, S. et al. Prioritizing User Requirements for Agile Software Development. 2018, **Institute of Electrical and Electronics Engineers Inc**. p.495-498.

SADIA, H.; ABBAS, S. Q.; FAISAL, M. Volatile requirement prioritization: A fuzzy based approach. **International Journal of Engineering and Advanced Technology**, v. 8, n. 5, p. 2467-2472, 2019. ISSN 22498958 (ISSN).

SADIQ, M.; HASSAN, T.; NAZNEEN, S. AHP-GORE-PSR: Applying analytic hierarchy process in goal oriented requirements elicitation method for the prioritization of software requirements. 2017, **Institute of Electrical and Electronics Engineers Inc.**

SALINAS, T. C. J, SEDENÕ, J., ESCALONA, M., MEJÍAS, M. Estimating, planning and managing Agile Web development projects under a value-based perspective. **Information and Software Technology**. 2014.

SEHRA, Sumeet Kaur; BRAR, Yadwinder Singh; KAUR, Navdeep. Applying fuzzy-AHP for software effort estimation in data scarcity. **International Journal of Engineering Trends and Technology(IJETT)**, v. 45, p. 4-9, 2017.

SEHRA, Sumeet Kaur; BRAR, Yadwinder Singh; KAUR, Navdeep. Multi Criteria Decision Making Approach for Selecting Effort Estimation Model. **International Journal of Computer Applications**, v. 39-1, p. 10-17, 2012.

SEO, Y.-S.; Yoon, K.-A.; Bae, D.-H. "An empirical analysis of software effort estimation with outlier elimination". **In: Proceedings of the 4th international workshop on Predictor models in software engineering**, 2008, pp. 25–32

SHAW, I. S.; SIMÕES, M. G. **Controle e modelagem Fuzzy**. São Paulo: E. Blücher. 200 p., 1999.

SHEPPERD, M. J.; Schofield, C. "Estimating software project effort using analogies.", **IEEE Trans. Software Eng.**, vol. 23–11, 1997, pp. 736–743.

SHER, F.; JAWAWI, D. N. A.; MOHAMMAD, R. Requirements prioritization aspects quantification for value-based software developments. **Journal of Theoretical and Applied Information Technology**, v. 97, n. 14, p. 3969-3979, 2019. ISSN 19928645 (ISSN).

SHIM, W. An agile method of representing, organizing, and (re) prioritizing requirements in a large enterprise. In: DAMIAN, D.; PERINI, A., et al, 2019, **IEEE Computer Society**. p.464-469.

SILVA, E.L.; MENEZES, E.M. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis: [S. n.], 2000. Society, 2003, p. 244-254.

SILVA, V. B. S., Schramm, F., Damasceno, A. C.: A multicriteria approach for selection of agile methodologies in software development projects. In: **IEEE International Conference on Systems, Man, and Cybernetics (SMC)**, pp. 2056–2060 (2016)

SILVAX, A. et al. Ordering the product backlog in agile software development projects: A systematic literature review. 2017, **Knowledge Systems Institute Graduate School**. p.74-80.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall. 2013.

SOMMERVILLE, I. **Software Engineering**. 10th. ed. Scotland: Pearson Addison-Wesley, 2016. 816 p.

SONG, Y. W.; WINDHEIM, M.; BENDER, B. Challenges in the definition and prioritisation of requirements: A case study. In: STORGA, M.; PAVKOVIC, N., et al, 2018, Faculty of Mechanical Engineering and Naval Architecture. p.1337-1344.

SRDJEVIC, B. Combining different prioritization methods in the analytic hierarchy process synthesis, **Computers & Operations Research**, v.32, n7, p.1897-1919, 2005.

SUFIAN, M. et al. A systematic literature review: Software requirements prioritization techniques. 2019, **Institute of Electrical and Electronics Engineers Inc.** p.35-40.

SVENSSON, R. B., GORSCHKE, T., REGNELL, B., TORKAR, R., SHAHROKNI, A., FELDT, R. Prioritization of Quality Requirements : State of Practice in Eleven Companies, pg. 69–78, 2011.

TÁLELE, P.; PHALNIKAR, R. Classification and prioritisation of software requirements using machine learning - A systematic review. 2021, **Institute of Electrical and Electronics Engineers Inc.** p.912-918.

TAVARES, H. C. A. B, CARVALHO, A. E. S. J. F. B. CASTRO. Medição de Pontos por Função a Partir da Especificação de Requisitos, Workshop em Engenharia de Requisitos (WER 2002), p. 278-298, 2002.

TCHEMRA, A. H. **Tabela de decisão adaptativa na tomada de decisões multicritério.** Tese (doutorado). Escola politécnica da Universidade de São Paulo. Departamento de engenharia de computação e sistemas digitais. São Paulo. 172 p. 2009.

THE STANDISH GROUP INTERNATIONAL INC. **Chaos. Technical Report -1994.** Disponível em: <[www.standishgroup.com](http://www.standishgroup.com)> Acesso em 03 mai. 2019.

THE STANDISH GROUP INTERNATIONAL. **Chaos. Technical Report - 2009.** Disponível em <[www.standishgroup.com](http://www.standishgroup.com)> Acesso em 03 mai. 2019.

THE STANDISH GROUP INTERNATIONAL. **Chaos. Technical Report - 2012.** Disponível em <[www.standishgroup.com](http://www.standishgroup.com)> Acesso em 03 mai. 2019.

TI MAIOR. **Programa estratégico de software e serviços de tecnologia da informação 2012 – 2015.** ASSESPRO - Associações das Empresas Brasileiras de Tecnologia da Informação Disponível em: <<http://assespro.org.br/biblioteca/documentos/2012-08-20-ti-maior-programa-estrategico-de-software-e-servicos-de-tecnologia-da-informacao/>> Acesso em: 16 set. 2019.

TSUNODA, M.; Monden, A.; Keung, J. W.; ichi Matsumoto, K. “Incorporating expert judgment into regression models of software effort estimation.” In: **APSEC**, 2012, pp. 374–379.

TUFAIL, H. et al. Towards the selection of Optimum Requirements Prioritization Technique: A Comparative Analysis. 2019, **Institute of Electrical and Electronics Engineers Inc.** p.227-231.

UPADHYAY, N.; SHARMA, A. Requirement Prioritization Based on Cost using Artificial Bee Colony Algorithm. 2020, **Institute of Electrical and Electronics Engineers Inc.** p.426-430.

VALLON, R.; MÜLLER-WERNHART, M.; SCHRAMM, W.; GRECHENIG, T. Combination of agile and lean in software development: Concept and realization. **Informatik-Spektrum**, v. 37, n. 1, p. 28–35, 2014. Springer Verlag.

VAZQUEZ, C. E., SIMÕES, G. S., ALBERT, R. M., 2011, Análise de Pontos de Função: **Medição, Estimativas e Gerenciamento de Projetos de Software**, 11 ed. São Paulo, SP, Brasil, Editora Érica

VAZQUEZ. C. E. SIMÕES, G. S. ALBERT, R. M., Análise de pontos de função. **Medição, estimativas e gerenciamento de projetos de software**, 8ª ed. São Paulo: Érica, 2008.

VICTOR, M., UPADHYAY, N.: Selection of Software Testing Technique: A Multi Criteria Decision Making Approach. **Springer Berlin Heidelberg**, Berlin, Heidelberg, pp. 453–462 2011

WANG, Xinyu et al. The role of requirements engineering practices in agile development: an empirical study. In: Requirements engineering. **Springer**, Berlin, Heidelberg, 2014. p. 195-209.

WHEATCRAFT, L.S. Triple your chances of project success risk and requirements. **INCOSE International Symposium**, Denver, CO, June 23. 2011.

WIN, T. Z.; MOHAMED, R.; SALLIM, J. Requirement Prioritization Based on Non-Functional Requirement Classification Using Hierarchy AHP. 2020, 1: **Institute of Physics Publishing**.

WOHLIN, Claes. Guidelines for snowballing in systematic literature studies and a replication in software engineering. **ACM International Conference Proceeding Series**. 2014

YASEEN, M.; IBRAHIM, N.; MUSTAPHA, A. Requirements prioritization and using iteration model for successful implementation of requirements. **International Journal of Advanced Computer Science and Applications**, v. 10, n. 1, p. 121-127, 2019. ISSN 2158107X (ISSN).

YASEEN, M.; MUSTAPHA, A.; IBRAHIM, N. Prioritization of Software Functional Requirements: Spanning Tree based Approach. **International Journal of Advanced Computer Science and Applications**, v. 10, n. 7, p. 489-497, 2019. ISSN 21565570

YASEEN, M.; MUSTAPHA, A.; IBRAHIM, N. Prioritization of software functional requirements from developers perspective. **International Journal of Advanced Computer Science and Applications**, v. 11, n. 9, p. 210-224, 2020. ISSN 2158107X (ISSN).

ZACARIAS, R. O. et al. Prioritization of software requirements derived from a business process model: Approach using the kano model. **RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao**, v. 2020, n. E38, p. 431-447, 2020. ISSN 16469895 (ISSN).

ZADEH, L. A Outline of a new approach to the analysis of complex systems and decision processes. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 3, n. 1, p. 28-44, 1973

ZADEH, L.A.; **Fuzzy sets**. Inform Control, 8:338–353. 196.

## APÊNDICE A

### Lista dos Artigos do Portfólio

**Quadro 12 - Lista dos artigos do portfólio bibliográfico**

<b>Autor (es)</b>	<b>Título</b>	<b>Journal</b>	<b>Ano</b>
Abbas, M. Et Al.	MBRP: Model-Based Requirements Prioritization Using PageRank Algorithm.	IEEE Computer Society	2019
Abdelazim, K.; Moawad, R.; Elfakharany, E.	A Framework for Requirements Prioritization Process in Agile Software Development.	Institute of Physics Publishing	2020
Adhim, K. K.; Hudaib, A.; Al- Shboul, B.	Efficient requirement prioritization based on enhanced multi-verse optimizer.	Journal of Theoretical and Applied Information Technology	2019
Ahmad, K. S. Et Al.	Fuzzy-MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements.	Institute of Electrical and Electronics Engineers	2018
Ahuja, H.; Sujata; Purohit, G. N.	Understanding requirement prioritization techniques	Institute of Electrical and Electronics Engineers Inc	2017
Al-Adwan, A.; Aladwan, A.	Using Interdependencias for the Prioritization and Reprioritization of Requirements in Incremental Development	International Journal of Advanced Computer Science and Applications	2020
Albuga, S.; Odeh, Y.	Towards Prioritizing Software Business Requirements in Startups.	Institute of Electrical and Electronics Engineers	2018
Ali, S. Et Al.	Aspect-based requirements mining technique to improve prioritisation process: Multi-stakeholder perspective.	IET Software	2020
Al-Ta'ani, R. H.; Razali, R.	Process model for systematic requirements prioritisation process in an agile software development environment based on 5S approach: Empirical study.	Journal of Theoretical and Applied Information Technology	2017
Alzaqebah, A.; Masadeh, R.; Hudaib, A.	Whale optimization algorithm for requirements prioritization	Institute of Electrical and Electronics Engineers	2018
Amelia, T.; Mohamed, R. B.	A Proposed Requirements Prioritization Model Based on Cost-Value Approach with Collaboration Perspective	Institute of Electrical and Electronics Engineers	2021
Aziz, N. A. A.; Hassan, S.; Admodisastro, N.	Prioritizing based on crowd preferences to requirements elicited from crowd's sentiments	International Journal of Innovative Technology and Exploring Engineering	2019
Barbosa, P. A. M. Et Al.	Selection and prioritization of software requirements using the Verbal Decision Analysis paradigm	Knowledge Systems Institute Graduate School	2017
Barbosa, P. A. M.; Pinheiro, P. R.; Silveira, F. R. V.	Towards the verbal decision analysis paradigm for implementable prioritization of software requirements	Algorithms	2018
Barbosa, P. A. M. Et Al.	Selection and Prioritization of Software Requirements Applying Verbal Decision Analysis	Complexity	2019
Borhan, N. H. Et Al.	Requirements prioritization techniques focusing on agile software development: A systematic literature review	International Journal of Scientific and Technology Research	2019
Busetta, P. Et Al.	Tool-Supported Collaborative Requirements Prioritisation.	IEEE Computer Society	2017

Cabrera, M. A. C. Et Al.	Prioritization of non-functional requirements in a mobile application for panic button system using neutrosophic decision maps.	Neutrosophic Sets and Systems	2020
Cavalcanti, C. Et Al.	Mechanisms to support requirements prioritization: A systematic mapping review. In:	PUC-Rio	2018
Devadas, R.; Srinivasan, G. N.	Review of different fuzzy logic approaches for prioritizing software requirements	International Journal of Scientific and Technology Research	2019
Dhingra, S. Et Al.	Selection of prioritization technique for software requirement using Fuzzy Logic and Decision Tree	Institute of Electrical and Electronics Engineers Inc	2017
Dos Santos, J. R. F.; Albuquerque, A. B.; Pinheiro, P. R.	Requirements prioritization in market-driven software: A survey based on large numbers of stakeholders and requirements.	Institute of Electrical and Electronics Engineers Inc	2017
Falahah	Defining and prioritizing software requirement using gIBIS and AHP method.	Universal Journal of Electrical and Electronic Engineering,	2019
Galarza, F. P. C. Et Al.	Prioritization of Software Requirements Using Neutrosophic TOPSIS.	Neutrosophic Sets and Systems	2021
Garg, U.; Singhal, A.	Software requirement prioritization based on non-functional requirements.	Institute of Electrical and Electronics Engineers Inc	2017
Gillain, J.; Jureta, I.; Faulkner, S.	Planning optimal agile releases via requirements optimization.	Institute of Electrical and Electronics Engineers Inc	2017
Govil, N.; Sharma, A.	Information Extraction on Requirement Prioritization Approaches in Agile Software Development Processes	Institute of Electrical and Electronics Engineers Inc	2021
Gulzar, K. Et Al.	Fuzzy approach to prioritize usability requirements conflicts: An experimental evaluation.	IEEE Access	2017
Gupta, V. Et Al.	Requirements engineering in software startups: A systematic mapping study.	Applied Sciences	2020
Haider, W. Et Al.	Improving requirement prioritization and traceability using artificial intelligence technique for global software development	Institute of Electrical and Electronics Engineers Inc	2019
Hudaib, A.; Alhaj, F.	Self-organizing maps for agile requirements prioritization	Institute of Electrical and Electronics Engineers Inc.	2019
Hudaib, A.; Masadeh, R.; Alzaqebah, A.	WGW: A hybrid approach based on whale and grey wolf optimization algorithms for requirements prioritization.	Advances in Systems Science and Applications	2018
Hujainah, F. Et Al.	Interrelated Elements in Formulating an Efficient Requirements Prioritization Technique: Review	Institute of Electrical and Electronics Engineers Inc	2020
Hujainah, F. Et Al.	SRPTackle: A semi-automated requirements prioritisation technique for scalable requirements of software system projects.	Information and Software Technology	2021
Ibriwesh, I. Et Al.	Prioritizing solution-oriented software requirements using the multiple perspective prioritization technique algorithm: An empirical investigation.	Concurrent Engineering Research and Applications	2019
Ijaz, K. B.; Inayat, I.; Allah Bukhsh, F.	Non-functional Requirements Prioritization: A Systematic Literature Review. In:	Institute of Electrical and Electronics Engineers Inc	2019
Kamal, T. Et Al.	Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development.	IEEE Access	2020
Kifetew, F. Et Al.	DMGame: A Gamified Collaborative Requirements Prioritisation Tool.	Institute of Electrical and Electronics Engineers Inc	2017
Kifetew, F. M. Et Al.	Towards multi-decision-maker requirements prioritisation via multi-objective optimisation. In:	CEUR-WS	2017

Kolpondinos, M. Z. H.; Glinz, M.	Behind Points and Levels-The Influence of Gamification Algorithms on Requirements Prioritization	Institute of Electrical and Electronics Engineers Inc.	2017
Misaghian, N.; Motameni, H.; Rabbani, M.	Prioritizing interdependent software requirements using tensor and fuzzy graphs.	Turkish Journal of Electrical Engineering and Computer Sciences	2019
Mkpojiogu, E. O. C.; Hashim, N. L.	Quality-based prioritization: An approach for prioritizing software requirements.	Journal of Telecommunication, Electronic and Computer Engineering	2017
Muñante, D.; Kifetew, F. M.; Albrecht, O.	Modelling prioritisation decision-making in software evolution. In:	CEUR-WS	2017
Rottoli, G. D.; Casanova, C.	Multi-criteria group requirement prioritization in software engineering using fuzzy linguistic labels. In:	CEUR-WS	2021
Sachdeva, S. Et Al.	Prioritizing User Requirements for Agile Software Development	Institute of Electrical and Electronics Engineers Inc	2018
Sadia, H.; Abbas, S. Q.; Faisal, M.	Volatile requirement prioritization: A fuzzy based approach.	International Journal of Engineering and Advanced Technology	2019
Sadiq, M.; Hassan, T.; Nazneen, S.	AHP-GORE-PSR: Applying analytic hierarchy process in goal oriented requirements elicitation method for the prioritization of software requirements	Institute of Electrical and Electronics Engineers Inc	2017
Sher, F.; Jawawi, D. N. A.; Mohammad, R.	Requirements prioritization aspects quantification for value-based software developments.	Journal of Theoretical and Applied Information Technology	2019
Shim, W.	An agile method of representing, organizing, and (re) prioritizing requirements in a large enterprise. In:	IEEE Computer Society	2019
Silvax, A. Et Al.	Ordering the product backlog in agile software development projects: A systematic literature review.	Knowledge Systems Institute Graduate School	2017
Song, Y. W.; Windheim, M.; Bender, B.	Challenges in the definition and prioritisation of requirements: A case study. In:	Faculty of Mechanical Engineering and Naval Architecture	2018
Tálele, P.; Phalnikar, R.	Classification and prioritisation of software requirements using machine learning - A systematic review.	Institute of Electrical and Electronics Engineers Inc	2021
Tufail, H. Et Al.	Towards the selection of Optimum Requirements Prioritization Technique: A Comparative Analysis.	Institute of Electrical and Electronics Engineers Inc	2019
Upadhyay, N.; Sharma, A.	Requirement Prioritization Based on Cost using Artificial Bee Colony Algorithm	Institute of Electrical and Electronics Engineers Inc	2020
Vegendla, A. Et Al.	A systematic mapping study on requirements engineering in software ecosystems.	Journal of Information Technology Research	2018
Win, T. Z.; Mohamed, R.; Sallim, J.	Requirement Prioritization Based on Non-Functional Requirement Classification Using Hierarchy AHP	Institute of Physics Publishing	2020
Yaseen, M.; Ibrahim, N.; Mustapha, A.	Requirements prioritization and using iteration model for successful implementation of requirements.	International Journal of Advanced Computer Science and Applications	2019
Yaseen, M.; Mustapha, A.; Ibrahim, N.	Prioritization of Software Functional Requirements: Spanning Tree based Approach.	International Journal of Advanced Computer Science and Applications	2019

Yaseen, M.; Mustapha, A.; Ibrahim, N.	Prioritization of software functional requirements from developers perspective.	International Journal of Advanced Computer Science and Applications	2020
Zacarias, R. O. Et Al.	Prioritization of software requirements derived from a business process model: Approach using the kano model.	RISTI - Revista Iberica de Sistemas e Tecnologias de Informação	2020

**Fonte: Dados da pesquisa (2019).**



## APÊNDICE B

### Quadros de Avaliação de Complexidade dos Requisitos de Software

**Quadro 13 - Avaliação de Complexidade dos Requisitos do Integrante I**

Alternativas	Critérios																			
	F11	F12	F13	F14	F15	F16	F1_07	F18	F19	F110	F111	F112	F113	F114	F115	F116	F117	F118	F119	F20
RQ-1	VL	ML	L	M	ML	ML	M	VL	MH	VL	L	ML	L	L	MH	M	ML	VL	MH	VL
RQ-2	ML	L	L	ML	M	ML	ML	L	M	VL	L	L	L	M	M	M	L	VL	MH	VL
RQ-3	L	VL	VL	L	L	VL	L	ML	VL	VL	L	VL	M	VL	L	M	L	VL	MH	VL
RQ-4	VL	VL	VL	L	L	VL	L	ML	VL	VL	L	VL	M	VL	L	M	L	VL	MH	VL
RQ-5	M	M	L	MH	L	ML	M	VL	MH	VL	VL	ML	MH	L	M	M	M	VL	MH	VL
RQ-6	VL	VL	VL	L	L	VL	L	VL	VL	VL	VL	VL	VL	VL	L	M	L	VL	MH	VL
RQ-7	ML	ML	L	M	M	ML	ML	L	L	VL	L	ML	MH	ML	M	M	MH	VL	MH	VL
RQ-8	ML	ML	L	M	M	ML	ML	L	L	VL	L	ML	ML	ML	MH	M	MH	VL	MH	VL
RQ-9	ML	ML	L	M	M	ML	ML	L	L	VH	L	ML	M	ML	M	M	MH	VL	MH	VL
RQ-10	M	L	L	ML	M	ML	ML	L	M	VL	L	L	MH	ML	ML	M	ML	VL	MH	VL

**Quadro 14 -Avaliação de Complexidade dos Requisitos do Integrante II**

Alternativas	Critérios																			
	F11	F12	F13	F14	F15	F16	F17	F18	F19	F110	F111	F112	F113	F114	F115	F116	F117	F118	F119	F20
RQ-1	L	M	L	L	ML	VL	H	VL	H	L	M	H	ML	ML	MH	ML	L	ML	ML	VL
RQ-2	M	VL	L	L	L	VL	MH	VL	MH	L	MH	MH	M	ML	ML	M	ML	VL	M	VL
RQ-3	ML	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL
RQ-4	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL
RQ-5	M	L	L	MH	MH	VL	H	VL	H	L	VL	MH	MH	M	MH	MH	MH	H	MH	L
RQ-6	VL	VL	VL	VL	VL	VL	VL	VL	VL	L	VL	VL	VL	VL	VL	VL	VL	VL	VL	VL
RQ-7	MH	VL	ML	VH	VH	VL	H	M	H	L	VL	MH	MH	M	M	MH	MH	MH	H	VL
RQ-8	ML	VL	ML	VH	VH	VL	H	M	H	VH	VL	MH	MH	M	MH	M	MH	MH	H	VL
RQ-9	ML	VL	ML	MH	H	VH	H	M	H	M	VL	MH	M	M	H	M	M	H	H	VL
RQ-10	ML	VL	ML	MH	ML	VL	H	L	H	L	M	M	M	M	ML	M	VL	L	H	VL

**Quadro 15 -Avaliação de Complexidade dos Requisitos do Integrante III**

Alternativas	Critérios																			
	F11	F12	F13	F14	F15	F16	F17	F18	F19	F110	F111	F112	F113	F114	F115	F116	F117	F118	F119	F20
RQ-1	L	MH	L	VL	VL	L	M	VL	M	VL	L	ML	ML	L	MH	VL	L	VL	L	M
RQ-2	ML	L	VL	L	M	ML	ML	VL	ML	VL	VL	VL	L	MH	M	VL	L	VL	L	M
RQ-3	M	VL	VL	VL	L	VL	L	VL	VL	M	VL	VL	VL	L	L	VL	VL	L	VL	L
RQ-4	VL	VL	VL	VL	L	VL	L	VL	VL	M	VL	VL	VL	L	L	VL	VL	L	VL	L
RQ-5	MH	MH	L	VL	L	L	ML	VL	MH	L	VL	VL	MH	M	M	VL	M	VL	M	M
RQ-6	VL	VL	VL	VL	M	VL	L	VL	VL	L	VH	VL	VL	ML	L	VL	VL	L	VL	L
RQ-7	L	L	VL	VL	L	M	ML	VL	L	L	VL	L	L	ML	M	VL	VL	VL	L	L
RQ-8	L	VL	VL	VL	M	L	ML	VL	L	M	VL	L	L	ML	M	VL	VL	VL	L	L
RQ-9	VL	L	L	VL	M	L	ML	VL	L	L	VL	L	L	ML	M	VL	VL	VL	L	L
RQ-10	VL	L	VL	VL	M	ML	ML	VL	ML	L	VL	VL	L	M	M	VL	VL	VL	L	L

## APÊNDICE C

### Tabela de Requisitos de Software e Fatores de Impacto com o Valor e a Respectiva Posição na Ordenação

Tabela 15 - Resultado da Avaliação de Complexidade dos Requisitos de Software com a Respectiva Posição

Alternativas	FI1	FI2	FI3	FI4	FI5	FI6	FI7	FI8	FI9	FI10	FI11	FI12	FI13	FI14	FI15	FI16	FI17	FI18	FI19	FI20
<b>RQ1</b>	0,092	0,346	0,157	0,156	0,136	0,114	0,501	0,001	0,516	0,046	0,217	0,346	0,197	0,177	0,550	0,177	0,177	0,069	0,257	0,114
Ranking	17	5	12	13	14	16	3	20	2	19	7	4	8	11	1	9	10	18	6	15
<b>RQ2</b>	0,291	0,092	0,092	0,177	0,276	0,135	0,310	0,046	0,142	0,000	0,177	0,177	0,217	0,346	0,327	0,217	0,177	0,000	0,295	0,114
Ranking	5	16	17	12	6	14	3	18	13	20	10	9	7	1	2	8	11	19	4	15
<b>RQ3</b>	0,237	0,000	0,001	0,046	0,091	0,000	0,092	0,069	0,000	0,000	0,046	0,001	0,114	0,046	0,092	0,114	0,046	0,046	0,135	0,046
Ranking	1	18	16	12	7	17	6	8	19	20	9	15	3	13	5	4	10	14	2	11
<b>RQ4</b>	0,001	0,000	0,001	0,046	0,091	0,000	0,092	0,069	0,000	0,000	0,046	0,001	0,114	0,046	0,092	0,114	0,046	0,046	0,135	0,046
Ranking	14	18	16	11	6	17	5	7	19	19	8	15	2	12	4	3	9	13	1	10
<b>RQ5</b>	0,468	0,295	0,157	0,255	0,238	0,114	0,382	0,001	0,000	0,000	0,001	0,198	0,550	0,276	0,467	0,237	0,467	0,177	0,484	0,156
Ranking	3	7	14	9	10	16	6	18	19	19	17	12	1	8	5	11	4	13	2	15
<b>RQ6</b>	0,001	0,000	0,001	0,046	0,157	0,000	0,092	0,001	0,000	0,000	0,198	0,001	0,001	0,069	0,092	0,114	0,046	0,046	0,135	0,046
Ranking	12	18	14	10	2	17	6	16	19	19	1	13	15	7	5	4	8	11	3	9
<b>RQ7</b>	0,257	0,114	0,114	0,292	0,351	0,177	0,347	0,156	0,000	0,000	0,046	0,257	0,313	0,290	0,451	0,237	0,256	0,135	0,349	0,046
Ranking	8	16	15	6	2	12	4	13	19	19	17	9	5	7	1	11	10	14	3	18
<b>RQ8</b>	0,197	0,069	0,114	0,292	0,517	0,114	0,347	0,156	0,000	0,000	0,046	0,257	0,257	0,290	0,484	0,217	0,256	0,135	0,349	0,046
Ranking	11	16	14	5	1	15	4	12	19	19	17	7	8	6	2	10	9	13	3	18
<b>RQ9</b>	0,135	0,114	0,177	0,237	0,501	0,313	0,347	0,156	0,000	0,000	0,046	0,257	0,276	0,290	0,500	0,217	0,237	0,177	0,349	0,046
Ranking	15	16	13	10	1	5	4	14	19	19	17	8	7	6	2	11	9	12	3	18
<b>RQ10</b>	0,177	0,092	0,114	0,197	0,328	0,135	0,347	0,092	0,000	0,000	0,156	0,156	0,295	0,327	0,290	0,217	0,069	0,046	0,349	0,046
Ranking	9	14	13	8	3	12	2	15	19	19	11	10	5	4	6	7	16	18	1	17

Fonte: Dados da pesquisa(2021)

## APÊNDICE D

### Ranking de Requisitos de Software e Fatores de Impacto

Quadro 16 - Ranking de Requisitos de Software e Fatores de Impacto

(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
1	0,6314	RQ5	1	0,54965	FI13	Experiência de aplicações
			2	0,48400	FI19	Cronograma de Desenvolvimento Requerido
			3	0,46753	FI1	Restrição de tempo de execução
			4	0,46735	FI17	Uso de ferramentas de software
			5	0,46733	FI15	Linguagem e experiência com ferramentas
			6	0,38206	FI7	Complexidade do produto
			7	0,29487	FI2	Restrição de armazenamento principal
			8	0,27600	FI14	Experiência de plataforma
			9	0,25547	FI4	Taxa de transação
			10	0,23806	FI5	Confiabilidade de software exigida
			11	0,23654	FI16	Continuidade de pessoal
			12	0,19767	FI12	Capacidade do programador
			13	0,17702	FI18	Desenvolvimento Multisite
			14	0,15658	FI3	Volatilidade da plataforma
			15	0,15629	FI20	Processamento de dados distribuído
			16	0,11351	FI6	Dimensão da base de dados
			17	0,00061	FI11	Capacidade do analista
			18	0,00058	FI8	Reutilização necessária
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
19	0,00000	FI10	Eficiência do usuário final			

(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
2	0,6219	RQ8	1	0,51738	FI5	Confiabilidade de software exigida
			2	0,48387	FI15	Linguagem e experiência com ferramentas
			3	0,34896	FI19	Cronograma de Desenvolvimento Requerido
			4	0,34691	FI7	Complexidade do produto
			5	0,29177	FI4	Taxa de transação
			6	0,29047	FI14	Experiência de plataforma
			7	0,25704	FI12	Capacidade do programador
			8	0,25689	FI13	Experiência de aplicações
			9	0,25556	FI17	Uso de ferramentas de software
			10	0,21715	FI16	Continuidade de pessoal
			11	0,19709	FI1	Restrição de tempo de execução
			12	0,15629	FI8	Reutilização necessária
			13	0,13511	FI18	Desenvolvimento Multisite
			14	0,11357	FI3	Volatilidade da plataforma
			15	0,11351	FI6	Dimensão da base de dados
			16	0,06919	FI2	Restrição de armazenamento principal
			17	0,04648	FI11	Capacidade do analista
			18	0,04648	FI20	Processamento de dados distribuído
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
19	0,00000	FI10	Eficiência do usuário final			

(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
3	0,6086	RQ9	1	0,50111	FI5	Confiabilidade de software exigida
			2	0,50009	FI15	Linguagem e experiência com ferramentas
			3	0,34896	FI19	Cronograma de Desenvolvimento Requerido
			4	0,34691	FI7	Complexidade do produto
			5	0,31332	FI6	Dimensão da base de dados
			6	0,29047	FI14	Experiência de plataforma
			7	0,27605	FI13	Experiência de aplicações
			8	0,25704	FI12	Capacidade do programador
			9	0,23664	FI17	Uso de ferramentas de software
			10	0,23657	FI4	Taxa de transação
			11	0,21715	FI16	Continuidade de pessoal
			12	0,17702	FI18	Desenvolvimento Multisite
			13	0,17681	FI3	Volatilidade da plataforma
			14	0,15629	FI8	Reutilização necessária
			15	0,13522	FI1	Restrição de tempo de execução
			16	0,11351	FI2	Restrição de armazenamento principal
			17	0,04648	FI11	Capacidade do analista
			18	0,04648	FI20	Processamento de dados distribuído
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
19	0,00000	FI10	Eficiência do usuário final			

(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
4	0,6066	RQ7	1	0,45053	FI15	Linguagem e experiência com ferramentas
			2	0,35060	FI5	Confiabilidade de software exigida
			3	0,34896	FI19	Cronograma de Desenvolvimento Requerido
			4	0,34691	FI7	Complexidade do produto
			5	0,31349	FI13	Experiência de aplicações
			6	0,29177	FI4	Taxa de transação
			7	0,29047	FI14	Experiência de plataforma
			8	0,25709	FI1	Restrição de tempo de execução
			9	0,25704	FI12	Capacidade do programador
			10	0,25556	FI17	Uso de ferramentas de software
			11	0,23654	FI16	Continuidade de pessoal
			12	0,17699	FI6	Dimensão da base de dados
			13	0,15629	FI8	Reutilização necessária
			14	0,13511	FI18	Desenvolvimento Multisite
			15	0,11357	FI3	Volatilidade da plataforma
			16	0,11351	FI2	Restrição de armazenamento principal
			17	0,04648	FI11	Capacidade do analista
			18	0,04648	FI20	Processamento de dados distribuído
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
19	0,00000	FI10	Eficiência do usuário final			

(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
5	0,5805	RQ1	1	0,54958	FI15	Linguagem e experiência com ferramentas
			2	0,51603	FI9	Documentação compatível com as necessidades do ciclo de vida
			3	0,50092	FI7	Complexidade do produto
			4	0,34602	FI12	Capacidade do programador
			5	0,34559	FI2	Restrição de armazenamento principal
			6	0,25695	FI19	Cronograma de Desenvolvimento Requerido
			7	0,21717	FI11	Capacidade do analista
			8	0,19704	FI13	Experiência de aplicações
			9	0,17700	FI16	Continuidade de pessoal
			10	0,17679	FI17	Uso de ferramentas de software
			11	0,17677	FI14	Experiência de plataforma
			12	0,15658	FI3	Volatilidade da plataforma
			13	0,15626	FI4	Taxa de transação
			14	0,13553	FI5	Confiabilidade de software exigida
			15	0,11355	FI20	Processamento de dados distribuído
			16	0,11351	FI6	Dimensão da base de dados
			17	0,09152	FI1	Restrição de tempo de execução
			18	0,06920	FI18	Desenvolvimento Multisite
			19	0,04650	FI10	Eficiência do usuário final
			20	0,00058	FI8	Reutilização necessária

(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
6	0,4990	RQ2	1	0,34563	FI14	Experiência de plataforma
			2	0,32747	FI15	Linguagem e experiência com ferramentas
			3	0,31013	FI7	Complexidade do produto
			4	0,29509	FI19	Cronograma de Desenvolvimento Requerido
			5	0,29062	FI1	Restrição de tempo de execução
			6	0,27644	FI5	Confiabilidade de software exigida
			7	0,21718	FI13	Experiência de aplicações
			8	0,21715	FI16	Continuidade de pessoal
			9	0,17730	FI12	Capacidade do programador
			10	0,17712	FI11	Capacidade do analista
			11	0,17679	FI17	Uso de ferramentas de software
			12	0,17678	FI4	Taxa de transação
			13	0,14190	FI9	Documentação compatível com as necessidades do ciclo de vida
			14	0,13509	FI6	Dimensão da base de dados
			15	0,11355	FI20	Processamento de dados distribuído
			16	0,09154	FI2	Restrição de armazenamento principal
			17	0,09153	FI3	Volatilidade da plataforma
			18	0,04648	FI8	Reutilização necessária
			19	0,00027	FI18	Desenvolvimento Multisite
			20	0,00000	FI10	Eficiência do usuário final



(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
7	0,4779	RQ10	1	0,34896	FI19	Cronograma de Desenvolvimento Requerido
			2	0,34691	FI7	Complexidade do produto
			3	0,32795	FI5	Confiabilidade de software exigida
			4	0,32745	FI14	Experiência de plataforma
			5	0,29503	FI13	Experiência de aplicações
			6	0,29049	FI15	Linguagem e experiência com ferramentas
			7	0,21715	FI16	Continuidade de pessoal
			8	0,19735	FI4	Taxa de transação
			9	0,17717	FI1	Restrição de tempo de execução
			10	0,15636	FI12	Capacidade do programador
			11	0,15629	FI11	Capacidade do analista
			12	0,13509	FI6	Dimensão da base de dados
			13	0,11357	FI3	Volatilidade da plataforma
			14	0,09154	FI2	Restrição de armazenamento principal
			15	0,09153	FI8	Reutilização necessária
			16	0,06924	FI17	Uso de ferramentas de software
			17	0,04648	FI20	Processamento de dados distribuído
			18	0,04647	FI18	Desenvolvimento Multisite
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
19	0,00000	FI10	Eficiência do usuário final			

(continua)

<b>Ranking do Requisito</b>	<b>Valor Requisito</b>	<b>Requisito</b>	<b>Ranking Fator de Impacto</b>	<b>Valor do Fator de Impacto</b>	<b>Sigla do Fator de Impacto</b>	<b>Nome do Fator de Impacto</b>
8	0,1040	RQ6	1	0,19755	FI11	Capacidade do analista
			2	0,15668	FI5	Confiabilidade de software exigida
			3	0,13535	FI19	Cronograma de Desenvolvimento Requerido
			4	0,11352	FI16	Continuidade de pessoal
			5	0,09153	FI15	Linguagem e experiência com ferramentas
			6	0,09151	FI7	Complexidade do produto
			7	0,06921	FI14	Experiência de plataforma
			8	0,04648	FI17	Uso de ferramentas de software
			9	0,04648	FI20	Processamento de dados distribuído
			10	0,04647	FI4	Taxa de transação
			11	0,04647	FI18	Desenvolvimento Multisite
			12	0,00100	FI1	Restrição de tempo de execução
			13	0,00093	FI12	Capacidade do programador
			14	0,00081	FI3	Volatilidade da plataforma
			15	0,00070	FI13	Experiência de aplicações
			16	0,00058	FI8	Reutilização necessária
			17	0,00022	FI6	Dimensão da base de dados
			18	0,00017	FI2	Restrição de armazenamento principal
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
19	0,00000	FI10	Eficiência do usuário final			

(continua)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
9	0,1011	RQ3	1	0,23722	FI1	Restrição de tempo de execução
			2	0,13535	FI19	Cronograma de Desenvolvimento Requerido
			3	0,11358	FI13	Experiência de aplicações
			4	0,11352	FI16	Continuidade de pessoal
			5	0,09153	FI15	Linguagem e experiência com ferramentas
			6	0,09151	FI7	Complexidade do produto
			7	0,09150	FI5	Confiabilidade de software exigida
			8	0,06924	FI8	Reutilização necessária
			9	0,04648	FI11	Capacidade do analista
			10	0,04648	FI17	Uso de ferramentas de software
			11	0,04648	FI20	Processamento de dados distribuído
			12	0,04647	FI4	Taxa de transação
			13	0,04647	FI14	Experiência de plataforma
			14	0,04647	FI18	Desenvolvimento Multisite
			15	0,00093	FI12	Capacidade do programador
			16	0,00081	FI3	Volatilidade da plataforma
			17	0,00022	FI6	Dimensão da base de dados
			18	0,00017	FI2	Restrição de armazenamento principal
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
			20	0,00000	FI10	Eficiência do usuário final

(conclusão)

Ranking do Requisito	Valor Requisito	Requisito	Ranking Fator de Impacto	Valor do Fator de Impacto	Sigla do Fator de Impacto	Nome do Fator de Impacto
10	0,0512	RQ4	1	0,13535	FI19	Cronograma de Desenvolvimento Requerido
			2	0,11358	FI13	Experiência de aplicações
			3	0,11352	FI16	Continuidade de pessoal
			4	0,09153	FI15	Linguagem e experiência com ferramentas
			5	0,09151	FI7	Complexidade do produto
			6	0,09150	FI5	Confiabilidade de software exigida
			7	0,06924	FI8	Reutilização necessária
			8	0,04648	FI11	Capacidade do analista
			9	0,04648	FI17	Uso de ferramentas de software
			10	0,04648	FI20	Processamento de dados distribuído
			11	0,04647	FI4	Taxa de transação
			12	0,04647	FI14	Experiência de plataforma
			13	0,04647	FI18	Desenvolvimento Multisite
			14	0,00100	FI1	Restrição de tempo de execução
			15	0,00093	FI12	Capacidade do programador
			16	0,00081	FI3	Volatilidade da plataforma
			17	0,00022	FI6	Dimensão da base de dados
			18	0,00017	FI2	Restrição de armazenamento principal
			19	0,00000	FI9	Documentação compatível com as necessidades do ciclo de vida
19	0,00000	FI10	Eficiência do usuário final			

## APÊNDICE E

### Instrumentos de Avaliação de Percepção e Aceitação do Modelo

**Quadro 17 - Instrumento para coleta da percepção da aplicação do modelo elaborado.**

<b>Percepção da aplicação do Modelo de Apoio ao Gerenciamento de Projeto de Desenvolvimento de <i>Software</i></b>	
<p>Termo de Consentimento Lido e Esclarecido</p> <p>Prezado(a) Respondente, você está sendo convidado a responder este questionário intitulado “Percepção da aplicação do Modelo De Apoio Ao Gerenciamento De Projeto De Desenvolvimento De Software”, como parte da pesquisa realizada por Aline Bampi, aluna do programa de mestrado em Engenharia de Produção e Sistemas (PPGEPS) da Universidade Tecnológica Federal do Paraná, Campus Pato Branco.</p> <p>O objetivo deste questionário é avaliar a percepção da aplicação do modelo, não havendo respostas “certas” ou “erradas” às perguntas formuladas. Você foi um dos(as) envolvidos(as) na avaliação de complexidade dos requisitos quanto aos fatores de impacto na etapa anterior, e por esse motivo está recebendo este formulário.</p> <p>Como participante do estudo, você deve estar ciente de que:</p> <p>I) Tem plena liberdade para deixar de responder ao questionário a qualquer momento, sem que ocorra qualquer prejuízo pessoal.</p> <p>II) Os dados obtidos serão mantidos em sigilo pelo pesquisador, assegurando ao participante a privacidade quanto aos dados confidenciais envolvidos na pesquisa.</p> <p>III) Os resultados poderão ser divulgados em publicações científicas mantendo o sigilo dos respondentes;</p> <p>IV) Não receberei remuneração nessa pesquisa, sendo minha participação voluntária.</p> <p>V) Ao responder este questionário estou declarando que li e concordo com o termo.</p> <p>Desde já, agradeço a sua atenção e me coloco a disposição para o caso de dúvidas através do e-mail: <a href="mailto:alinebampi@gmail.com">alinebampi@gmail.com</a></p> <p>Tempo médio de resposta: 7 minutos.</p>	
Em relação ao modelo aplicado e a sua composição, qual a sua percepção quanto a relevância de cada item abaixo mencionado, para o processo de tomada de decisão no planejamento e acompanhamento do projeto?	
Fatores de Impacto utilizados:	Avaliação de Complexidade do Requisitos:
<input type="checkbox"/> irrelevante <input type="checkbox"/> pouco relevante <input type="checkbox"/> moderadamente relevante <input type="checkbox"/> muito relevante <input type="checkbox"/> extremamente relevante	<input type="checkbox"/> irrelevante <input type="checkbox"/> pouco relevante <input type="checkbox"/> moderadamente relevante <input type="checkbox"/> muito relevante <input type="checkbox"/> extremamente relevante
Se desejar, comente como foi a sua experiência e sua percepção sobre o impacto da aplicação do modelo de apoio para o gerenciamento do projeto?	
Na sua opinião, qual desses papéis deve participar da avaliação de complexidade do modelo aplicado? <input type="checkbox"/> Gestor de Produto	

<input type="checkbox"/> Head de Desenvolvimento <input type="checkbox"/> Desenvolvedor <input type="checkbox"/> Analista de Requisitos <input type="checkbox"/> Analista de Testes <input type="checkbox"/> Head de Qualidade <input type="checkbox"/> Analistas de Qualidade <input type="checkbox"/> Outro: _____
<p>Se desejar, comente qual foi sua percepção sobre a ferramentas utilizada para aplicação do modelo, quanto a sua facilidade de uso e outros pontos que julgar importantes.</p>
<p>Se desejar, deixe sugestões para melhorar a aplicação ou o aproveitamento do modelo.</p>

Fonte: Aatoria Própria (2021).

**Quadro 18 - Instrumento para avaliação da aceitação do modelo elaborado - TAM.**

<p><b>Percepção da aplicação do Modelo Multicritério de Priorização de Complexidade de Requisitos de <i>Software</i></b></p>					
<p>Termo de Consentimento Lido e Esclarecido</p> <p>Prezado(a) Respondente, você está sendo convidado a responder este questionário intitulado “Percepção da aplicação do Modelo Multicritério de Priorização de Requisitos de Software”, como parte da pesquisa realizada por Aline Bampi, aluna do programa de mestrado em Engenharia de Produção e Sistemas (PPGEPS) da Universidade Tecnológica Federal do Paraná, Campus Pato Branco</p> <p>O objetivo deste questionário é avaliar a percepção da aplicação do modelo, não havendo respostas “certas” ou “erradas” às perguntas formuladas. Você foi um dos(as) envolvidos(as) na avaliação de complexidade dos requisitos quanto aos fatores de impacto na etapa anterior, e por esse motivo está recebendo este formulário.</p> <p>Como participante do estudo, você deve estar ciente de que:</p> <p>I) Tem plena liberdade para deixar de responder ao questionário a qualquer momento, se m que ocorra qualquer prejuízo pessoal.</p> <p>II) Os dados obtidos serão mantidos em sigilo pelo pesquisador, assegurando ao participante a privacidade quanto aos dados confidenciais envolvidos na pesquisa.</p> <p>III) Os resultados poderão ser divulgados em publicações científicas mantendo o sigilo dos respondentes;</p> <p>IV) Não receberei remuneração nessa pesquisa, sendo minha participação voluntária.</p> <p>V) Ao responder este questionário estou declarando que li e concordo com o termo.</p> <p>Desde já, agradeço a sua atenção e me coloco a disposição para o caso de dúvidas através do e-mail: <a href="mailto:alinebampi@gmail.com">alinebampi@gmail.com</a></p> <p>Tempo médio de resposta: 10 minutos.</p>					
<p>Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir</p>					
	Discordo Totalmente	Não concordo	Neutro	Concordo	Concordo Totalmente
Considero relevante a utilização dos Fatores de Impacto	( )	( )	( )	( )	( )

Considero relevante a Avaliação de Complexidade do Requisitos Aplicada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Considero o modelo relevante quanto a proporcionar benefícios para o processo de tomada de decisão no gerenciamento do projeto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Recomendo a utilização do modelo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar a Avaliação de Complexidade do Requisitos é Simples	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A Avaliação de Complexidade do Requisitos é Rápida	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Senti dificuldade em realizar a aplicação do Modelo de Priorização de Requisitos com a Avaliação de Complexidade	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
O resultado da aplicação do modelo auxiliou no processo de tomada de decisão no gerenciamento do projeto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A aplicação do modelo permitiu melhorar a assertividade no processo de tomada de decisão	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A aplicação do modelo facilitou a justificativa da ordem de entrega de versões para o cliente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A utilização do modelo favorece o alinhamento de prazos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A utilização do modelo favorece o acompanhamento de impedimentos técnicos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A utilização do modelo favorece a melhoria de performance das entregas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Recomento a utilização deste modelo de priorização de requisitos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aprender a aplicar o modelo de priorização de requisitos é fácil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A quantidade de fatores de impacto é adequada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Os fatores de impacto apresentados para a avaliação de complexidade na aplicação do modelo, são adequados	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
A descrição dos fatores de impacto do modelo, são de fácil entendimento	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Considero suficientes os cargos envolvidos na avaliação de complexidade durante a aplicação do modelo	( )	( )	( )	( )	( )
Se desejar cite 3 adjetivos que expressam sua opinião sobre o modelo de priorização de requisitos aplicado:					
Se desejar, complete a afirmação a seguir, conforme sua percepção: Para ser mais simples de utilizar, o modelo de priorização de requisitos deve ...					
Se desejar, complete a afirmação a seguir, conforme sua percepção: Para ser mais assertivo, o modelo de priorização de requisitos deve ...					

**Fonte: Aatoria Própria (2021).**



## APÊNDICE F

### Respostas - Avaliações de Percepção da Aplicação do Modelo

Quadro 19 - Avaliações de Percepção - Respostas

<b>Em relação ao modelo aplicado e a sua composição, qual a sua percepção quanto a relevância de cada item para o processo de tomada de decisão no planejamento e acompanhamento do projeto? [Fatores de Impacto utilizados]</b>		
Muito relevante	Muito relevante	Muito relevante
<b>Em relação ao modelo aplicado e a sua composição, qual a sua percepção quanto a relevância de cada item para o processo de tomada de decisão no planejamento e acompanhamento do projeto? [Avaliação de Complexidade do Requisitos]</b>		
Extremamente relevante	Extremamente relevante	Muito relevante
<b>Se desejar, comente como foi a sua experiência e sua percepção sobre o impacto da aplicação do modelo de apoio para o gerenciamento do projeto?</b>		
<p>Não consigo mensurar de forma efetiva o impacto, devido a não ter acompanhado as decisões após a aplicação do modelo (não é meu papel), mas acredito que por ser aplicado antes da execução das tarefas, ajudou a alinhar prazos, impedimentos técnicos e melhorias em performance de entrega do projeto dentro do esperado.</p>	<p>Permitiu avaliar de um novo ponto de vista, questões que até então parecem estar implícitas em nossas considerações, mesmo que até então de maneira informal. Permitiu organizar a ordem do desenvolvimento dos requisitos que não possuíam dependências entre si e que tinham maior complexidade e assim diminuindo as chances de ocorrer imprevistos e com esse requisito, e quando houve problema tivemos tempo para resolver de forma prioritária sem causar aqueles desesperos na equipe.</p>	<p>Foi bem interessante poder traduzir como geralmente já avalio na prática, se é algo complexo ou não, muitas vezes não é fácil explicar de forma clara por que é, mas no dia a dia a gente passa a entender, e essa aplicação meio que clarificou isso. Muito embora, tenha sido cansativo avaliar tantos fatores. No impacto da gestão não busquei entender o quanto foi válido para mudar algum planejamento. Mas foi priorizado em certo momento o que era mais complexo para remover problemas o quanto antes e isso evitou de deixar tarefas parcialmente realizadas para depois</p>
<b>Na sua opinião, qual desses papéis deve participar da avaliação de complexidade do modelo aplicado?</b>		
<p>Head de Desenvolvimento, Desenvolvedor, Analista de Requisitos, Analista de Testes, Head de Qualidade</p>	<p>Head de Desenvolvimento, Desenvolvedor, Analista de Requisitos, Head de Qualidade</p>	<p>Head de Desenvolvimento, Desenvolvedor, Analista de Requisitos, Head de Qualidade</p>

**Se desejar, comente qual foi sua percepção sobre a ferramentas utilizada para aplicação do modelo, quanto a sua facilidade de uso e outros pontos que julgar importantes.**

<p>A forma de resposta era fácil, não houve uma dificuldade, mas ficou cansativo quanto e quantidade de fatores a serem avaliados. Ou questão é que embora se saiba que são fatores que consideramos normalmente de forma subjetiva e até mentalmente, mas ao ter que definir e trazer para fora do 'pensamento', faz considerar com mais profundidade do que quando é só no pensamento. O que é muito bom, embora inicialmente parece que pode acarretar mais tempo do que estamos acostumados (se forem todos os fatores usados nessa aplicação). Ainda assim, achei que é uma forma relativamente prática de tornar o processo como um todo, mais eficaz.</p>	<p>Foi importante ter acesso a descrição dos fatores de impacto, alguns termos são de conhecimento prático, mas no momento da avaliação gera dúvidas se está alinhado com o tinha em meu entendimento. A necessidade de rever os conceitos de tantos fatores tornou um pouco cansativo, embora a forma de responder a avaliação tenha sido muito simples e prática e acho que isso pode dificultar a adesão.</p>	<p>Foi tranquilo, super fácil de responder tendo a descrição dos fatores pra consultar quando havia dúvidas, um pouco cansativo se for fazer a avaliação de muitos requisitos.</p>
--	--	--

**Se desejar, deixe sugestões para melhorar a aplicação ou o aproveitamento do modelo**

<p>Configurar no Jira no requisito essas avaliações. Deixar a possibilidade de selecionar quais destes fatores são mais relevantes para cada projeto, por exemplo dentre os 20, escolher 10 deles e ai avaliar a complexidade somente dos 10 escolhidos</p>	<p>Se reformulado a quantidade de quesitos a avaliar e apresentar uma descrição com exemplos, pode simplificar bastante.</p>	<p>Algum checklist ou algo do tipo para selecionar/reduzir os fatores de impacto a serem avaliados conforme o projeto.</p>
---	--	--

Fonte: Dados da Pesquisa(2021)

APÊNDICE G

Respostas da Aplicação do Modelo de Aceitação de Tecnologia

Figura 19 - Avaliação de Aceitação de Tecnologia

Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Considero relevante a utilização dos Fatores de Impacto]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Considero relevante a Avaliação de Requisitos Aplicada]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Considero o modelo relevante quanto a proporcionar benefícios para o processo de tomada de decisão no gerenciamento do projeto]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Recomendo a utilização do modelo]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Realizar a Avaliação de Complexidade do Requisitos é Simples]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A Avaliação de Complexidade do Requisitos é Rápida]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Senti dificuldade em realizar a aplicação do Modelo de Priorização de Requisitos com a Avaliação de Complexidade]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [O resultado da aplicação do modelo auxiliou no processo de tomada de decisão no gerenciamento do projeto]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A aplicação do modelo permitiu melhorar a assertividade no processo de tomada de decisão]	
Concordo Totalmente	Concordo	Concordo	Concordo	Neutro	Não concordo	Não concordo	Concordo Totalmente	Concordo	
Concordo Totalmente	Concordo Totalmente	Concordo	Concordo Totalmente	Concordo	Neutro	Discordo Totalmente	Concordo	Concordo Totalmente	
Concordo	Concordo Totalmente	Concordo Totalmente	Concordo Totalmente	Não concordo	Neutro	Discordo Totalmente	Concordo Totalmente	Concordo Totalmente	
Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A aplicação do modelo facilitou a justificativa da ordem de entrega de versões para o cliente]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A utilização do modelo favorece o alinhamento de prazos]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A utilização do modelo favorece o acompanhamento de impedimentos técnicos]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A utilização do modelo favorece a melhoria de performance da entregas]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Recomento a utilização deste modelo de priorização de requisitos]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Aprender a aplicar o modelo de priorização de requisitos é fácil]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A quantidade de fatores de impacto é adequada]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Os fatores de impacto apresentados para a avaliação de complexidade na aplicação do modelo, são adequados]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [A descrição dos fatores de impacto do modelo, são de fácil entendimento]	Em relação ao modelo aplicado e a sua composição, marque a opção que melhor representa a sua percepção quanto aos itens a seguir [Considero suficiente os cargos envolvidos na avaliação de complexidade durante a aplicação do modelo]
Concordo Totalmente	Concordo	Concordo Totalmente	Concordo	Concordo	Neutro	Não concordo	Concordo Totalmente	Não concordo	Não concordo
Concordo Totalmente	Concordo	Concordo Totalmente	Neutro	Concordo Totalmente	Concordo Totalmente	Concordo	Concordo Totalmente	Concordo	Não concordo
Concordo	Concordo	Concordo Totalmente	Neutro	Concordo Totalmente	Concordo Totalmente	Neutro	Concordo	Neutro	Neutro

Se desejar cite 3 adjetivos que expressam sua opinião sobre o modelo de priorização de requisitos aplicado

assertivo; diferente; promissor

Interessante; Cansativo; Valoroso

Importante, diferente, eficaz

Se desejar, complete a afirmação a seguir, conforme sua percepção: Para ser mais simples de utilizar, o modelo de priorização de requisitos deve ...

Diminuir a quantidade de fatores e refinar as descrições deles para facilitar a interpretação

Implementado em um sistema integrado com a visualização dos detalhes do projeto e exemplos do uso dos fatores em relação aos requisitos.

Uma interface mais amigável, um sistema específico possivelmente com flexibilidade para seleção de fatores.

Se desejar, complete a afirmação a seguir, conforme sua percepção: Para ser mais assertivo, o modelo de priorização de requisitos deve ...

Um mecanismo que permita fazer a avaliação de forma mais rápida. Sugerir ações conforme os fatores de maior impacto.

Possibilitar selecionar/excluir um ou outro fator de impacto e trazer sugestões de ações.

Dar flexibilidade de escolher fatores de impacto por requisitos