

**Fundação Pedro Leopoldo**  
**Mestrado Profissional em Administração**

**Maturidade organizacional sob a ótica dos efeitos *exploration-exploitation* e  
ambidestria: Proposta de um modelo conceitual aplicado a organizações  
desenvolvedoras de software**

**Henrique Augusto Reis Rodrigues**

**Pedro Leopoldo**  
**2023**

**Henrique Augusto Reis Rodrigues**

**Maturidade organizacional sob a ótica dos efeitos *exploration-exploitation* e ambidestria: Proposta de um modelo conceitual aplicado a organizações desenvolvedoras de software**

Dissertação apresentada ao Curso de Mestrado Profissional em Administração da Fundação Pedro Leopoldo, como requisito parcial para a obtenção do título de Mestre em Administração.

Área de concentração: Gestão e Estratégia em Organizações.

Linha de pesquisa: Gestão Estratégica e Mercados.

Orientadora: Professora Dra. Ester Eliane Jeunon

**Pedro Leopoldo**  
**Fundação Pedro Leopoldo**  
**2023**

658.406 RODRIGUES, Henrique Augusto Reis  
R685m Maturidade organizacional sob a ótica dos efeitos  
exploration-exploitation e ambidestria: proposta de  
um modelo conceitual aplicado a organizações desen-  
volvedoras de software / Henrique Augusto Reis Ro-  
drigues.  
- Pedro Leopoldo: FPL, 2023.

149 p.

Dissertação Mestrado Profissional em Administração.  
Fundação Cultural Dr. Pedro Leopoldo – FPL, Pedro  
Leopoldo, 2023.

Orientadora: Prof<sup>ª</sup>. Dra. Ester Eliane Jeunon

1. Ambidestria Organizacional.
  2. Dualidades Estratégicas.
  3. Engenharia de Software.
  4. Metodologia Ágil.
  5. Modelos de Maturidade.
- I. JEUNON, Ester Eliane, orient. II. Título.

**Dados Internacionais de Catalogação na Publicação**  
**Ficha Catalográfica elaborada por Maria Luiza Diniz Ferreira – CRB6-1590**

## FOLHA DE APROVAÇÃO

Título da Dissertação: “MATURIDADE ORGANIZACIONAL SOB A ÓTICA DOS EFEITOS EXPLORATION-EXPLOITATION E AMBIDESTRIA: Proposta de um modelo conceitual aplicado a organizações desenvolvedoras de software”.

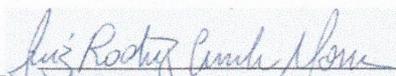
Nome do Aluno: **Henrique Augusto Reis Rodrigues**

Dissertação de mestrado, modalidade Profissionalizante, defendida junto ao Programa de Pós-Graduação em Administração da Faculdade Pedro Leopoldo, aprovada pela banca examinadora constituída pelos professores:



---

Profa. Dra. Ester Eliane Jeunon - Orientadora - FPL Educacional



---

Prof. Dr. Luiz Rodrigo Cunha Moura - FPL Educacional



Documento assinado digitalmente

CARLOS ALBERTO GONCALVES

Data: 09/08/2023 17:08:37-0300

Verifique em <https://validar.it.gov.br>

---

Prof. Dr. Carlos Alberto Gonçalves - UFMG

Pedro Leopoldo (MG), 13 de julho de 2023.

Ao meu pai (em lembrança), com amor.

## **Agradecimentos**

A Deus por esta conquista.

A Prof. Dra. Ester Eliane Jeunon, pela orientação eficiente e segura, pelas valiosas sugestões e estímulos dados.

Aos professores do mestrado em estrutura da Faculdade de Pedro Leopoldo, pelo interesse e prol da pesquisa e do crescimento acadêmico da instituição.

Aos amigos e funcionários do mestrado, pela convivência e amizade, das quais jamais me esquecerei.

A minha mãe Sônia, aos meus irmãos Raphael e Guilherme, pelo amor incondicional.

A minha esposa Márcia pelo carinho e disponibilidade.

Eduardo e Pedro, amo vocês no sentido mais amplo da palavra.

## RESUMO

**Objetivo:** o objetivo da pesquisa foi a proposição de um modelo conceitual de maturidade organizacional para aplicação em organizações desenvolvedoras de software ambidestras. **Relevância/originalidade:** Organizações desenvolvedoras de software enfrentam hoje em dia o desafio de atuar concomitantemente em projetos que exigem tanto a prática de Metodologias Tradicionais quanto a de Metodologias Ágeis, metodologias de desenvolvimento de software com características opostas e conflitantes. Antes mesmo de tornar-se uma organização orientada à construção de softwares, estas empresas atendem demandas comuns à dimensão de dualidades estratégicas frequentes, como a exploração das competências atuais (*exploitation*) e na exploração de novas competências (*exploration*). A relevância deste trabalho para a academia se demonstra a partir das discussões sob a ótica da ambidestria em torno da maturidade das organizações de administrarem as atividades de forma analítica e intuitiva, e atribuindo especialmente as organizações do setor de software, que operam em situações de grande dinamismo, incertezas tecnológicas e influenciada simultaneamente pela adoção de metodologias ágeis. **Metodologia/abordagem:** os procedimentos metodológicos adotados caracterizam esta pesquisa como de natureza descritiva de abordagem qualitativa. Utilizou-se a entrevista como técnica de coleta de dados. Após o desenvolvimento do modelo de maturidade, o mesmo foi avaliado por um grupo de cinco profissionais da área de gestão de projetos e de desenvolvimento de softwares, para verificarem a validade e confiabilidade do modelo bem como avaliar se o modelo é consistente com as práticas reais do mercado. **Principais resultados:** Como resultado, verificou-se que o modelo proposto excedeu o esperado em sua avaliação. Neste sentido, o modelo conceitual apresentado foi considerado relevante para as empresas de software que desejam avaliar a sua maturidade organizacional levando em consideração a influência das dualidades estratégicas e adotando metodologias ágeis simultaneamente. **Contribuições teórico-metodológicas:** buscou-se com a pesquisa contribuir para a academia, fomentando novas discussões sobre a habilidade das organizações do setor de software em equilibrar os processos estratégicos e de inovação, enquanto adequando as metodologias ágeis na geração de produtos de software competitivos e com qualidade. **Contribuições sociais/para a gestão:** o presente estudo apresenta relevância ao discutir questões de conflitos estratégicos nas organizações do setor de software, impactadas por competências necessárias como: a inovação incremental, concorrência e desempenho, e influenciadas diretamente pela ambidestria, ao assumir o uso de métodos ágeis como primeira opção metodológica. **Aderência a linha de pesquisa:** Esta pesquisa atende aos objetivos de um mestrado profissional, cujo objetivo é aproximar a academia do ambiente de negócios com estudos aplicados. Está vinculado à linha de pesquisa de inovação e organizações, no Núcleo de Estudos de Gestão em Organizações do Mestrado Profissional em Administração da Faculdade Pedro Leopoldo.

**Palavras-chaves:** Ambidestria organizacional. Dualidades estratégicas. Engenharia de Software. Metodologia ágil. Modelos de maturidade.

## Abstract

**Objective:** The objective of this research was to propose a conceptual maturity model for organizational application in ambidextrous software development organizations.

**Relevance/originality:** Software development organizations currently face the challenge of simultaneously working on projects that require both Traditional and Agile Methodologies, which have opposite and conflicting characteristics. Even before becoming software-oriented organizations, these companies meet demands common to the frequent strategic dualities dimension, such as the exploitation of current competencies (exploitation) and the exploration of new competencies (exploration). The relevance of this work for the academy is demonstrated by the discussions from the ambidexterity perspective around the maturity of organizations to manage activities in an analytical and intuitive way, especially for organizations in the software sector, which operate in situations of great dynamism, technological uncertainties, and are simultaneously influenced by the adoption of agile methodologies.

**Methodology/approach:** The methodological procedures adopted characterize this research as descriptive in nature and of a qualitative approach. Surveys were used as the data collection technique. After the development of the maturity model, it was evaluated by a group of five professionals from the project management and software development areas, in order to verify the validity and reliability of the model, as well as to assess whether the model is consistent with real market practices.

**Main results:** As a result, it was found that the proposed model exceeded expectations in its evaluation. In this sense, the conceptual model presented was considered relevant for software companies that wish to evaluate their organizational maturity taking into consideration the influence of strategic dualities and adopting agile methodologies simultaneously.

**Theoretical-methodological contributions:** The research sought to contribute to academia by fostering new discussions on the ability of software sector organizations to balance strategic and innovation processes while adapting agile methodologies in the generation of competitive and high-quality software products.

**Social/management contributions:** This study is relevant in discussing issues of strategic conflicts in software sector organizations, impacted by necessary competencies such as incremental innovation, competition, and performance, and directly influenced by ambidexterity, assuming the use of agile methods as the first methodological option.

**Adherence to research line:** This research meets the objectives of a professional master's degree, whose goal is to bring the academy closer to the business environment through applied studies. It is linked to the research line of innovation and organizations in the Center for Studies in Management in Organizations of the Professional Master's Program in Administration at Faculdade Pedro Leopoldo.

**Keywords:** Organization ambidexterity. Agile methodology. Maturity models. Software Engineering. Strategic dualities.

## Lista de Figuras

Figura 1 - Disposição dos níveis da Engenharia de Software. ....	54
Figura 2 - Modelo cascata de desenvolvimento. ....	56
Figura 3 - Manifesto ágil da Agile Alliance .....	62
Figura 4 - O Ciclo do Scrum .....	66
Figura 5 - Decomposição do problema de adaptação parcial do método ágil .....	75
Figura 6 - Visão geral dos modelos de maturidade .....	81
Figura 7 - Fases de maturidade propostos .....	90
Figura 8 - Framework de construção do modelo de maturidade proposto .....	125
Figura 9 - Modelo de maturidade proposto .....	126
Figura 10 - Consenso entre os profissionais acerca do framework do modelo de maturidade proposto .....	128
Figura 11 - Avaliação dos níveis de maturidade do modelo proposto. ....	129
Figura 12 - Avaliação dos componentes dos domínios do modelo de maturidade proposto .....	130
Figura 13 - Avaliação do mecanismo de autoavaliação para modelo de maturidade proposto. ....	131

## Lista de Tabelas

Tabela 1 - Distribuição dos especialistas de acordo com as variáveis: sexo, faixa etária, qualificação e experiência .....	127
----------------------------------------------------------------------------------------------------------------------------	-----

## Lista de Quadros

<b>Quadro 1:</b> Demandas simultaneamente conflitantes .....	40
<b>Quadro 2 :</b> Estrutura conceitual de <i>exploitation</i> e <i>exploration</i> .....	42
<b>Quadro 3:</b> Estudos em ambidestria organizacional no setor de software .....	45
<b>Quadro 4:</b> Perfis ocupacionais em uma equipe clássica/tradicional .....	58
<b>Quadro 5:</b> Estrutura comparativa entre as metodologias tradicionais e ágeis .....	72
<b>Quadro 6:</b> Subproblemas da adaptação parcial do método ágil.....	75
<b>Quadro 7:</b> Parâmetros que afetam a agilidade .....	77
<b>Quadro 8:</b> Modelos de maturidade e suas estruturas.....	79
<b>Quadro 9:</b> Síntese do referencial teórico .....	84
<b>Quadro 10:</b> Requisitos avaliados para o modelo de maturidade. ....	87
<b>Quadro 11:</b> Síntese da metodologia .....	88
<b>Quadro 12:</b> Características do escopo do modelo proposto .....	91
<b>Quadro 13:</b> Decisões para construção do modelo.....	91
<b>Quadro 14:</b> Composição estrutural das dimensões.....	94
<b>Quadro 15:</b> Atividades características de <i>exploration</i> - Práticas de Conhecimento .	94
<b>Quadro 16:</b> Atividades características de <i>exploration</i> - Práticas Inovadoras.....	96
<b>Quadro 17:</b> Atividades características desdobradas na dimensão <i>exploitation</i> - Concorrência.....	98
<b>Quadro 18:</b> Atividades características desdobradas na dimensão <i>exploitation</i> - Eficiência.....	99
<b>Quadro 19:</b> Atividades características desdobradas na dimensão <i>exploitation</i> - Qualidade.....	101
<b>Quadro 20:</b> Atividades características desdobradas na dimensão - Orientação Estratégica .....	102
<b>Quadro 21:</b> Formação da dimensão Agilidade .....	105
<b>Quadro 22:</b> Formação da dimensão Agilidade - Dinamismo .....	106
<b>Quadro 23:</b> Formação da dimensão Agilidade - Equipe .....	108
<b>Quadro 24:</b> Formação da dimensão Agilidade - Comunicação .....	109
<b>Quadro 25:</b> Formação da dimensão Agilidade - Teste .....	110
<b>Quadro 26:</b> Formação da dimensão Agilidade - Perfil e Conhecimento dos desenvolvedores .....	111

<b>Quadro 27:</b> Formação da dimensão Agilidade - Cultura .....	113
<b>Quadro 28:</b> Questões de autoavaliação para modelo teórico de pesquisa qualitativa. .....	114
<b>Quadro 29:</b> Níveis de maturidade propostos nesta pesquisa .....	119
Quadro 30 - Níveis de maturidade e respectivas características propostos no modelo de maturidade nesta pesquisa .....	120
<b>Quadro 31:</b> Processos de conhecimento incorporados ao modelo de maturidade proposto .....	123
<b>Quadro 32:</b> Respostas dos avaliadores para a questão aberta proposta.....	131

## Lista de Abreviaturas e Siglas

ASD	<i>Adaptive Software Development</i>
CHAOS	<i>Comprehensive Human Appraisal for Originating Software</i>
CMM	<i>Capability Maturity Model</i>
DSDM	<i>Dynamic Systems Development Method</i>
IBM FSD	<i>International Business Machines Corporation Federal Systems Division</i>
IGTI	Núcleo de Estudos em Inteligência, Gestão e Tecnologia para Inovação
IID	Desenvolvimento incremental iterativo
ISO/IEC	<i>International Organization for Standardization</i>
MPS.BR	Melhoria do Processo de Software Brasileiro
NASA	Administração Nacional da Aeronáutica e Espaço
OGC	<i>Office of Government Commerce</i>
OPM3	<i>Organization Project Management Maturity Model</i>
PMI	<i>Project Management Institute</i>
PMMM	<i>Project Management Maturity Model</i>
Prince	<i>Project In Controlled Environment</i>
TDD	<i>Test-Driven Development</i>
TI	Tecnologia da Informação
VUCA	<i>Volatility, Uncertainty, Complexity e Ambiguity</i>

## Sumário

1	Introdução .....	29
1.2	Problematização .....	33
1.3	Objetivos.....	35
1.3.1	Objetivo geral .....	35
1.3.2	Objetivos específicos .....	35
1.4	Justificativas .....	36
1.5	Organização do estudo.....	37
2	Referencial Teórico .....	38
2.1	Ambidestria organizacional.....	39
2.2	Ambidestria organizacional no setor de software .....	44
2.2.1	Ambidestria organizacional: vantagens.....	48
2.3	Engenharia de software .....	52
2.3.1	Metodologias de processos de software .....	55
2.4	Metodologias tradicionais .....	55
2.5	Metodologias ágeis.....	59
2.5.1	Scrum.....	64
2.6	Confronto entre as metodologias.....	71
2.7	Modelos de maturidades.....	78
2.7.1	Design e construção dos modelos de maturidade .....	82
2.8	Contribuição do referencial teórico para a pesquisa .....	83
3	Metodologia.....	84
3.1	Caracterização da pesquisa .....	85
3.2	Unidade de observação .....	85
3.3	Procedimento para coleta de dados .....	86
3.4	Síntese da metodologia .....	88

4 Apresentação e Discussão dos Resultados .....	89
4.1 Apresentação dos resultados .....	90
4.1.1 Composição arquitetural do modelo .....	90
4.1.2 Composição estrutural das dimensões .....	93
4.1.3 Componentes do modelo teórico de autoavaliação .....	113
4.1.4 <i>Níveis de maturidade</i> .....	118
4.1.5 <i>Framework do modelo teórico de maturidade</i> .....	124
4.2 Validação do modelo por entrevista.....	126
4.2.1 <i>Caracterização dos pesquisados</i> .....	127
4.2.2 Discussão dos resultados da avaliação do modelo proposto .....	128
5 Considerações Finais .....	132
5.2 Contribuições gerenciais.....	134
5.3 Propostas para estudos futuros .....	135
Referências .....	136
Apêndices.....	143

## 1 Introdução

Nota-se que tudo muda o tempo todo, temos vários cenários de caos de complexas previsões e variáveis de muita incerteza, principalmente nas recentes organizações constantemente desafiadas pelos conflitos geracionais e nas tradicionais organizações que precisam se reinventar para adequar-se à nova conjuntura econômica e social.

Procedente do vocabulário militar americano, o uso comum do termo VUCA começou no final dos anos 1990 e foi, posteriormente, utilizado nas ideias de liderança estratégica aplicadas em uma ampla gama de organizações. Em 1998, o *United States War College* apresentou este conceito no relatório “*Training and education army officers for the 21st Century: Implications for the United States*” (Whiteman,1998).

O termo VUCA é uma sigla utilizada para expressar a volatilidade (*volatility*), a incerteza (*uncertainty*), a complexidade (*complexity*) e a ambiguidade (*ambiguity*) no cenário atual e tem sido adotado por organizações e líderes empresariais para reproduzir conceitualmente o ecossistema de negócios caótico, turbulento e em constante transformação. Expõe a complexidade de nossa sociedade contemporânea, devido à interdependência e à globalização, circunstâncias que anteriormente tinham pouco impacto, mas que agora refletem em toda sociedade. Estas características do novo ambiente de negócios, reunidas no termo VUCA, refletem o atual cenário das organizações desenvolvedoras de software frente aos avanços tecnológicos, das formas de comunicação e, principalmente, na elaboração do planejamento estratégico, tornando-o arriscado e improdutivo quando projetado para 5 ou 10 anos adiante.

As organizações constituídas anteriormente a este novo modelo de negócios e que mantém em seu portfólio produtos de software e, portanto, derivam a um modelo organizacional orientado ao desenvolvimento de software também enfrentam atualmente uma estimulação de atender simultaneamente à ambidestria organizacional (Cordeiro, 2014).

A gênese dessa estimulação procede do campo da Gestão da Inovação e tem sido fomentada ao longo dos anos em diferentes áreas de gestão organizacional, tais como a Gestão Estratégica e o Design Organizacional, como advertem Burgelman (1991), Aprendizado Organizacional, March (1991), a Adaptação Organizacional, sempre com o objetivo de retratar a solução encontrada pelas organizações acerca do paradoxo da dualidade entre eficiência e flexibilidade, ponderado por Thompson (1967).

Em contribuição, March (1991) alinha-se a Henri (2006) e Tushman & O'Reilly (1996) salientando que a ambidestria expressa-se quando os gestores são demandados a cumprir os objetivos estratégicos, quando há necessidade de equilibrar as dualidades conflitantes entre si em termos de recursos e objetivos para garantir o sucesso e a sobrevivência no longo prazo. No entanto, faz-se necessário compreender os efeitos destas dualidades e explicar como elas afetam diretamente a performance e o desempenho das organizações. Diante disso, particularmente, o trabalho de Simon (2010) observou uma lacuna nas pesquisas em administração quanto à ambidestria organizacional. Designadamente, esta lacuna está direcionada à falta de compreensão sobre a capacidade das organizações de se engajarem simultaneamente na exploração das competências, dos recursos e das capacidades atuais (*exploitation*) e na exploração de novas competências (*exploration*) como uma estratégia consonante pela busca de sobrevivência e de melhoria de desempenho.

Simon (2010), assegura que ainda há um potencial abandono conceitual das soluções de gestão estratégica e gerencial por focar excessivamente no desenvolvimento da eficiência e da eficácia dos recursos organizacionais, abstendo em consideração os distintos modos de alocar recursos e tarefas com o objetivo simultâneo em *exploitation* e *exploration*. Uma forma de deliberar essa lacuna é na atribuição dos recursos de *exploitation* e *exploration* de modo conjunto e equilibrado, objetivando alcançar a ambidestria organizacional.

Um ponto universal entre todas as disciplinas de estudo que, de alguma forma se moldaram em torno do conceito, é que a Ambidestria Organizacional é considerada pré-requisito ou condição necessária para a sobrevivência e sucesso das organizações, como abordado por Raisch e Birkinshaw (2008) e isso permanece

autêntico quando analisamos a Ambidestria Organizacional nas organizações desenvolvedoras de software, particularmente.

Informações levantadas durante o planejamento do trabalho de pesquisa, que serão apresentados adiante, levantaram a suspeita de que a provável correlação entre as dualidades dimensionais possa ser diferente quando analisadas de modo comparativo em empresas desenvolvedoras de software e demais organizações, dadas as diferentes condições a que empresas de ambos os perfis estão sujeitas.

Em consideração a isto, endereçamos essas lacunas e sugerimos que a ambidestria organizacional influencia o nível de desempenho, performance e na capacidade dessas organizações em praticar simultaneamente ou alternadamente técnicas, metodologias ou processos nativos de culturas e práticas organizacionais tão incongruentes quanto às demandadas pelas metodologias ágeis e metodologias tradicionais.

Esta pode ser uma importante contribuição para o entendimento das condições em que a ambidestria organizacional se instala nas empresas. Pela evolução das práticas de engenharia dos softwares, particularmente nas organizações desenvolvedoras de software, o cenário mais comum acerca da ambidestria organizacional é na incorporação de uma metodologia ágil em uma organização onde metodologias tradicionais já estão aderentes. Nerur (2005), assim como Vinekar (2006), sugerem que a dinâmica de introdução a metodologias ágeis está sujeita a desafios que podem se tornar bloqueios ou impedimentos para adoção da organização ambidestra ideal.

Vinekar (2006) foi o signatário pela apropriação do conceito e reputação de Ambidestria Organizacional no campo do desenvolvimento de software, reconhecendo na adoção simultânea de metodologias tradicionais e ágeis a mesma qualificação para o conflito entre a busca da flexibilidade e da eficiência, de forma congruente.

Assim, obtém-se a sintonia das dualidades em organizações tradicionais frente a apropriação de metodologias ágeis em organizações desenvolvedoras de software.

Diante deste cenário, a área de desenvolvimento de software se tornou um nicho vantajoso para as organizações que atuam na área de Tecnologia da Informação. Buscando uma maior aderência no mercado de desenvolvimento de software, particularmente na qualidade dos softwares produzidos, diversas corporações começaram a fazer grandes investimentos para desenvolver sistemas diferenciados com maior excelência. Para isto, foram adotados modelos de qualidade de software, alguns com reconhecimento internacional, que pudessem atestar os sistemas desenvolvidos pela organização, conferindo a eles maior confiabilidade junto ao cliente.

A vista disso, foram concebidos modelos de qualidade de software que têm como objetivo garantir a qualidade do produto através da definição e normatização de processos organizacionais a serem aplicados durante o desenvolvimento do software. Os mais conhecidos no cenário brasileiro são: ISO/IEC 15504, VUCA (SEI 2006) e MPS.BR (SOFTEX 2009).

Desta forma, hoje, um dos maiores desafios das empresas produtoras de software é justamente dominar as práticas de qualidade preconizadas por esses modelos de processos de software frente a alta demanda de incorporação de metodologias ágeis, dadas as condições ambíguas naturais destas organizações.

A orientação deste estudo encontra oportunidade e motivação para propor um modelo conceitual de maturidade organizacional nas empresas desenvolvedoras de software, desde o desejado equilíbrio entre as dualidades: *exploitation* e *exploration*, balanceando o foco que a empresa dá nas atividades de criação, inovação, adaptabilidade ao ambiente, com o alinhamento e a eficiência dos processos e simultaneamente na adoção de metodologias ágeis aos seus processos de desenvolvimento de softwares.

## 1.2 Problematização

As empresas desenvolvedoras de software enfrentam hoje o desafio de atuar em projetos que exigem tanto a prática de metodologias tradicionais quanto a de metodologias ágeis, metodologias de desenvolvimento de software com características opostas e conflitantes. Além disso, estas empresas também enfrentam a demanda de equilibrar as dualidades estratégicas de *exploitation* e *exploration*, buscando explorar tanto as competências atuais quanto a de novas competências, segundo O'Reilly & Tushman (2013). Nesse contexto, os autores sugerem que a busca pela ambidestria na metodologia ágil e tradicional surge como uma estratégia para melhorar as entregas dos produtos e aumentar o desempenho e a inovação. Entretanto, a literatura ainda carece de um modelo de maturidade organizacional que possa ser aplicado nessas organizações, que considere a influência das dualidades estratégicas e o uso simultâneo de metodologias ágeis e tradicionais.

Os autores, O'Reilly & Tushman (2013), reforçam que a falta de um modelo de maturidade específico para as organizações desenvolvedoras de software que buscam a ambidestria na metodologia ágil e tradicional é outro ponto crítico. As metodologias de maturidade atuais, são baseadas em práticas estruturadas e processos bem definidos, mas não levam em consideração a ambidestria e as dualidades estratégicas enfrentadas por essas organizações. Sem um modelo de maturidade específico, essas empresas enfrentam dificuldades para avaliar e melhorar suas práticas de desenvolvimento de software, o que pode prejudicar sua capacidade de inovação e competição em um mercado cada vez mais exigente e dinâmico. Diante desse cenário, há uma necessidade premente de desenvolver um modelo de maturidade que considere métricas de avaliação em torno da ambidestria e as dualidades estratégicas para avaliar e melhorar a maturidade organizacional das empresas desenvolvedoras de software.

Os tipos de métricas a serem considerados para medir a maturidade de uma organização desenvolvedora de software, enquanto estão influenciados por metodologias ou processos nativos de culturas e práticas organizacionais exigidos pelas metodologias ágeis passa pela discussão de algumas hipóteses, desde a busca

pelo equilíbrio em ambas as capacidades de *exploitation-exploration* simultaneamente, conforme O'Reilly & Tushman (2013) afirmam, atribuindo especialmente as organizações de software que operam em situações de grande dinamismo e incertezas tecnológicas e de mercado, até no ciclo de vida dos processos de produção de software que detêm.

Sob a ótica da ambidestria, as empresas com capacidades para administrarem as atividades de forma analítica e intuitiva são denominadas de Organizações Ambidestras. A Ambidestria organizacional não envolve simplesmente se uma empresa pode alcançar a eficiência e inovação ou competir em vários mercados. Este conceito engloba o desenvolvimento das capacidades necessárias para competir em novos mercados e tecnologias que permitam à empresa sobreviver em detrimento das condições do ambiente em que ela se encontra, afirma O'Reilly & Tushman (2008).

Nesse sentido, não é mais aceitável a escolha por uma ou outra forma de pensamento para liderar, gerenciar e inovar as empresas e a forma como oferecem ou vendem seus softwares e/ou produtos. Ao mesmo tempo em que é preciso analisar os gargalos internos e melhorar continuamente para garantir a competitividade, é necessário também estar atento às novas oportunidades para se diferenciar no mercado e prover um software e/ou produto cada vez mais alinhado às expectativas do cliente, como sugerido por Brown e Wyatt (2010).

Ao abordamos os efeitos das dualidades na adoção de metodologias ágeis particularmente nas empresas desenvolvedoras de software, Simons (2010) observou uma lacuna nas pesquisas em administração quanto ao conceito de ambidestria organizacional, enquanto estratégia empresarial em qualquer nicho de negócio, direcionada a falta de compreensão das organizações se engajarem concomitantemente na exploração das competências atuais (*exploitation*) e na exploração de novas competências (*exploration*).

Além da discussão sobre estas dualidades exploradas, é suficiente a necessidade de interpolar a ambidestria metodológica, ou seja, os efeitos da escolha de novas abordagens e competências ágeis, em organizações que operam em elementos tradicionais, Simons (2010).

Sendo assim, este estudo visa responder a seguinte questão:

Quais as dimensões a serem consideradas para assim, elaborar uma escala de maturidade para organizações desenvolvedoras de software enquanto simultaneamente são influenciadas pelas dualidades estratégicas: *exploration* e *exploitation* e na apropriação de metodologias ágeis, dadas as condições ambidestras naturais destas organizações?

### **1.3 Objetivos**

#### **1.3.1 Objetivo geral**

Elaborar um modelo de maturidade organizacional conceitual para aplicação em organizações desenvolvedoras de software ambidestras.

#### **1.3.2 Objetivos específicos**

- a) Identificar os componentes empresariais com impacto direto no desempenho das organizações relacionados às dualidades *exploitation-exploration*, e respectivas simultaneidades;
- b) Identificar as dimensões relacionadas a ambidestria metodológica, ao utilizar de técnicas e práticas da metodologia ágil *Scrum*, em paralelo ao uso de metodologias tradicionais no desenvolvimento de softwares;
- c) Definir categorias de maturidade específicas, em escalas representativas, divididos em níveis e priorizadas de forma lógica;
- d) Propor modelo conceitual para nortear as organizações de software enquanto adotando metodologias ágeis. O modelo descreve os elementos conceituais e suas relações através dos princípios da ambidestria organizacional;

e) Identificar a percepção de especialistas em gestão de projetos e gestores de equipes de desenvolvimento de software para avaliar a aderência e confiabilidade do modelo bem como a consistência com as práticas reais do mercado.

É importante destacar que NÃO faz parte do escopo deste trabalho:

a) Propor um modelo de maturidade de sucesso. O sucesso da aplicabilidade do modelo de maturidade proposto não está relacionado apenas com a geração de informações para elaborar o modelo futuro para nortear as empresas na jornada ambidestra, mas também com o sucesso de sua implementação, o que não faz parte do escopo deste trabalho.

b) Criar novas técnicas analíticas e intuitivas, ou seja, novas demandas (dualidades) para auxiliar a análise de processos de negócio. Este trabalho foca na utilização integrada dos pensamentos e técnicas analíticas e intuitivas existentes na literatura.

#### **1.4 Justificativas**

O interesse deste estudo para a academia se convence a partir das lacunas existentes enquanto organizações que, necessitam equilibrar demandas (dualidades) conflitantes entre si, especialmente as empresas desenvolvedoras de software que lidam com as incertezas mercadológicas e técnicas, e desejam adaptar-se ao ambiente, com o alinhamento e a eficiência dos processos e vantagens na adoção de metodologias ágeis.

Contribuir com o trabalho docente já é motivação suficiente para a concepção de uma escala de maturidade que possa nortear as organizações de software a atingir o melhor desempenho. É explícito a dificuldade que as organizações desenvolvedoras de software têm ao se adequar às novas metodologias ágeis e gerar produtos de software competitivos e com qualidade.

Ao analisar os trabalhos acadêmicos sobre o tema, fica evidente a preocupação de entender melhor quais as principais métricas que as organizações devem orientar

para norteá-las a um melhor desempenho estratégico. A principal crítica por March (1991), atualmente é que, por utilizar metodologias tradicionais e não conhecer profundamente as dualidades conflitantes, os gestores não conseguem conquistar os objetivos estratégicos e tampouco garantir o sucesso e desempenho a longo prazo, também reportado por O'Reilly & Tushman (2008).

É possível identificar na literatura diversos estudos orientados aos efeitos das dimensões *exploitation-exploration*, sob a justificativa de desempenho organizacional. Encontra-se também trabalhos variados contemplando as escolhas do modelo organizacional frente aos processos de Engenharia de Software contemporâneos.

No entanto, não foram identificadas pesquisas que propõem um modelo conceitual de maturidade que aborda desde o conhecimento e a utilização de estratégias e técnicas intuitivas até entendimento e análise de processos de negócio que se baseiam nos princípios da ambidestria organizacional, particularmente na adoção de metodologias ágeis.

Os procedimentos metodológicos utilizados nesta pesquisa a caracterizam como descritiva de abordagem qualitativa. A pesquisa aqui realizada tem interesses quantitativos, para analisar a melhor aderência das métricas e práticas na organização pesquisada, mas também componentes predominantemente qualitativos, para se identificar as motivações e principais influências na adoção de metodologias ágeis. Para essas questões, o foco não é somente saber as motivações mais frequentes, mas principalmente identificar os fatores que conduziram as organizações para a decisão de adoção e como esses fatores moldaram essas mesmas organizações, com respaldo nas dualidades conflitantes.

## **1.5 Organização do estudo**

A estrutura desta dissertação está subdividida em cinco capítulos e seções. Neste primeiro capítulo foi apresentado a introdução do trabalho, na qual constam a contextualização, problematização, os objetivos e justificativas.

O segundo capítulo apresenta o referencial teórico, no qual se registram as pesquisas na área e a discussão sobre as origens das dualidades estratégicas e particularmente das competências acerca da *exploração-exploração* e suas vantagens. Em seguida, aborda-se a Engenharia de Software com fundamentos conceituais para posteriormente colaborar com o entendimento e aplicabilidade dos processos de software e o uso de metodologias.

Na sequência, trata-se das metodologias tradicionais e ágeis, com ênfase no entendimento do *framework Scrum*, utilizado como referência nesta pesquisa. Por fim, confronta-se as metodologias e abordamos os elementos estruturais dos modelos de maturidade atuais, finalizando com a contribuição do referencial teórico para a pesquisa.

O terceiro capítulo descreve a metodologia empregada na pesquisa, iniciando pela caracterização da pesquisa e em seguida a unidade de observação, procedimento para a coleta de dados e finalizando com uma síntese da metodologia.

No quarto capítulo relatam-se e discutem-se os resultados alcançados, identificando e discutindo as relações existentes entre as variáveis obtidas a partir da literatura estudada, a composição arquitetural e estrutura das dimensões, os componentes do modelo teórico de autoavaliação com a percepção dos avaliadores e profissionais a respeito da aplicabilidade do modelo de maturidade proposto neste trabalho.

Por fim, o quinto capítulo traz as considerações finais desta pesquisa, incluindo as reflexões sobre os resultados encontrados, mostrando as contribuições gerenciais aplicáveis e sugestões para estudos futuros.

## **2 Referencial Teórico**

Neste capítulo serão discutidas questões sobre ambidestria organizacional tradicionalmente aplicada ao estudo da estratégia empresarial, e com o foco em empresas desenvolvedoras de software.

Estabeleceu-se uma dinâmica de leitura com exposição das informações e autores de forma matricial, onde são apontadas as pesquisas de maior relevância para a obtenção de uma visão geral sobre as dualidades conflitantes *exploitation-exploration*. Adiante, explora-se os conceitos de engenharia de software e sua importância para o desenvolvimento de produtos de software com qualidade.

Em seguida, aborda-se as metodologias tradicionais e as metodologias ágeis e analisados os aspectos envolvidos na adoção de metodologias ágeis no desenvolvimento de software e, por fim, descreve-se os conceitos e definições dos aspectos mais importantes dos modelos de maturidade para processos de software, que contribuirão para a recomendação de um modelo de maturidade em organizações desenvolvedoras de software que passam por uma influência ambidestra.

## **2.1 Ambidestria organizacional**

A ambidestria manifesta-se em organizações de diferentes ramos da atividade econômica. No âmbito deste presente estudo, optou-se pela compreensão de uma abordagem teórica não somente vinculada às empresas desenvolvedoras de software mas como a ambidestria é fundamentada em torno da noção de *exploitation* e *exploration*, desenvolvidas a partir do trabalho de March (1991). A criação de organizações tende a ser realizada por um agente ou um grupo de agentes, com a missão de executar, ao longo do tempo, as diferentes etapas desse processo, segundo Gartner (2020).

Para Gartner & Brush (2016), as etapas abrangem situações como a concepção de uma ideia, a descoberta ou criação de oportunidades, a aquisição de recursos necessários para a realização do negócio, a exploração de mecanismos inovadores, até o resultado desse processo, que é de fato, a efetiva construção da organização e o desenvolvimento futuro do negócio em termos intraempreendedores. Trata-se de demandas simultaneamente conflitantes, quando colocadas em uma mesma perspectiva de análise, viabilizam a melhoria contínua e o refinamento de processos, métodos, técnicas, ações, produtos ou serviços, novos meios de produção ou de execução de uma atividade, implementação de novas práticas, e com eficiência organizacional, como orienta Borges, Lima & Brito (2017).

No Quadro 1, visualizam-se os principais autores e as demandas abordadas.

**Quadro 1:**  
Demandas simultaneamente conflitantes

<b>Autores</b>	<b>Dualidade A</b>	<b>Dualidade B</b>
March (1991); He e Wong (2004); Hill e Birkinshaw (2014); Simons (2010)	<i>Exploitation</i>	<i>Exploration</i>
Tushman e O'Reilly (1996)	Mudança Evolutiva	Mudança Revolucionária
Henri (2006)	Cultura voltada controle	Cultura voltada flexibilidade
Mundy (2010)	Uso de controle	Uso da Estratégia
Gibson e Birkinshaw (2004)	Adaptabilidade	Alinhamento
Porter (1986)	Posicionamento estratégico baixo custo	Diferenciação
Adler (1999); Jorgenses e Messner (2009)	Eficiência	Flexibilidade
Vinekar, Slinkman e Nerur (2006)	Sistemas Tradicionais	Sistemas ágeis

Fonte: adaptado de Severgnini, E., Galdamez, E., Vieira, V. (2019)

March (1991), por sua vez, ao sintetizar o quadro teórico sobre a referida abordagem destaca que a referida abordagem é baseada nas dualidades *exploration* e *exploitation*.

O conceito de *exploration* relaciona-se à criação e/ou descoberta de novas oportunidades de negócios, por meio do uso da criatividade, aplicação de inovações, e desenvolvimentos experimentais, de acordo com Gupta, Smith & Shalley (2006). De forma similar, autores, Hill e Birkinshaw (2014), definem *exploration* como atividades organizacionais que se associam a elementos como o foco em pesquisa e desenvolvimento, a capacidade de assumir riscos, o exercício de autonomia, a relação com o ambiente externo em estado de alerta empreendedor, e a promoção de um ambiente interno inovativo e adaptativo, com abertura para novas descobertas e novas experiências.

Os estudos sobre ambidestria são originários do trabalho de Duncan (1976). Duncan foi o primeiro a utilizar o termo “ambidestro” no contexto organizacional. Ele sugeriu que fosse necessário às organizações alterarem suas estruturas ao longo do tempo para alinhar a inovação ou a eficiência à estratégia de negócio. Essa concepção pode ser observada por meio de análises sobre dualidades contraditórias em nível organizacional, como eficiência e eficácia, inovação radical e incremental, e *exploration* e *exploitation*, de acordo com Birkinshaw & Gupta (2013), Chen (2017), Gibson & Birkinshaw (2004), Raisch & Birkinshaw, (2008).

Neste cenário, as organizações tomam decisões sobre aspectos como a aplicação de recursos entre as estratégias de flexibilidade e eficiência, a adaptação e a renovação estratégica, a exploração de novas competências e a de competências atuais Koriak *et al.* (2018); Popadiuk & Bido (2016). Logo, pode-se dizer que a ambidestria retrata a capacidade da organização em melhorar e aperfeiçoar as rotinas e métodos existentes e, ao mesmo tempo, adaptar-se a mudanças que ocorrem a partir do desenvolvimento de novas oportunidades. Com isso, a referida concepção se constitui como base para o desempenho, competitividade, e sobrevivência organizacional no mercado, enfatizado por Birkinshaw & Gupta (2013).

As organizações naturalmente ao longo do seu ciclo de vida, por exemplo, devem participar de deliberações que abrangem a aplicação e manutenibilidade de recursos entre as estratégias acerca da flexibilidade e da eficiência, como preconizado por Adler (1999); da adaptabilidade e do alinhamento, indicado por Gibson e Birkinshaw (2004); da exploração de novas competências (*exploration*) e da exploração das competências contemporâneas (*exploitation*), proposto por He e Wong (2004), Hill e Birkinshaw (2014), March (1991) e Simons (2010), da diferenciação e do baixo custo por Porter (1996), das inovações incrementais e das revolucionárias sugeridas por Tushman e O’Reilly (1996) e do uso do controle e da transigência apontadas por Mundy (2010).

Para que o gerenciamento dessas demandas conflitantes ocorra, é recomendado a harmonia na aplicação de recursos, como forma de equilíbrio sugerida por Gupta, Smith e Shalley (2006), de modo que as sequências conflitantes - *exploration* e *exploitation* - sejam alcançadas simultaneamente, potencializando o desempenho e

garantindo a continuidade. Portanto, a organização deve buscar a eficiência, a efetividade e a conformidade (*exploitation*) e, ao mesmo tempo, implementar a criatividade e a exploração de seus recursos (*exploration*), como mencionam os autores.

Como abordado anteriormente, entre os estudos sobre as demandas conflitantes, um dos mais representativos na literatura de ambidestria organizacional é o de March (1991, p. 71) o qual sugere que *exploitation* e *exploration* são "atividades de aprendizagem essencialmente diferentes, entre as quais, concorrem com a atenção e com os recursos escassos da empresa".

Logo, as organizações ambidestras são aquelas que demonstram semelhante destreza tanto em atividades de aproveitamento das capacidades atuais (*exploitation*), quanto em atividades de prospecção de novas competências e exploração de novas oportunidades (*exploration*), como ressaltado por Simsek (2009).

O Quadro 2, apresenta o entendimento conceitual sobre *exploitation* e *exploration*, na ótica dos autores March (1991), Benner e Tushman (2003, p. 245), He e Wong (2004) e Zimmermann, Raisch e Birkinshaw (2015, p. 1120).

**Quadro 2 :**  
Estrutura conceitual de *exploitation* e *exploration*

<b>Autores</b>	<b><i>Exploitation</i></b>	<b><i>Exploration</i></b>
March (1991, p. 71)	Denominado como "velhas certezas". Possui elementos como o refinamento, escolha, produção, eficiência, seleção, implementação e execução.	Denominado como "Novas possibilidades" Traz elementos compreendidos em termos de pesquisa, variação, tomada de riscos, experimentação, flexibilidade, descoberta e inovação.
Benner e Tushman (2003, p. 245)	Abrangem "mudanças incrementais, técnicas de gestão por processos e de design", explora recursos existentes e atende às pressões de desempenho de curto prazo.	Abrange mudanças radicais, criação e inovação, além de estar voltada para adaptações de desempenho de longo prazo.

He e Wong (2004)	Associado com "estruturas mecanicistas, sistemas fortemente conectados, caminhos contínuos, rotinas, controle, burocracia, mercados e tecnologias estáveis" (p. 481). Objetivo primário na melhoria de mercados e produtos existentes.	Associada com as "estruturas orgânicas, sistema de baixa conexão, novos caminhos, improvisação, autonomia, caos, mercados e tecnologias emergentes" (p. 483). Foca em novos mercados e produtos.
Zimmermann, Raisch e Birkinshaw (2015, p. 1120)	Abrange a utilização e o compartilhamento do conhecimento existente entre parceiros.	Abrange a criação de novos conhecimentos entre os parceiros envolvidos.

Fonte: adaptado de Severgnini, E., Galdamez, E., Vieira, V. (2019)

Conforme estabelecido no Quadro 2, uma das principais diferenças entre as duas estratégias é que a *exploration* é mais voltada para o longo prazo, enquanto a *exploitation* é mais voltada para o curto prazo. A *exploration* envolve mais riscos e incertezas, pois as novas oportunidades podem não ser tão bem-sucedidas quanto o esperado. Já a *exploitation* é mais segura e consistente, pois se baseia em recursos e processos existentes que já foram testados e aprimorados ao longo do tempo.

Ainda que existem diferenças entre as demandas simultaneamente conflitantes, a estrutura conceitual definida por March (1991) sugere que transcorra um equilíbrio entre *exploitation* e *exploration* de modo que uma organização fortaleça o desempenho e amplie a competitividade, enquanto alcança o sucesso e a sobrevivência de longo prazo. O esforço excessivo na exploração das capacidades atuais, pode conduzir a organização para um estado de equilíbrio aquém do ideal, enquanto o estímulo no foco apenas na conquista de novas capacidades pode envolver altos custos de experimentação, abundantes ideias a serem desenvolvidas e poucas competências distintivas, apontado por Heavey e Simsek (2017). Assim, March (1991), afirma que obter um equilíbrio entre atividades distintas que competem entre si por recursos insuficientes é um fator primordial para a sobrevivência e prosperidade da organização.

Ainda, March (1991), já na década de 90, defendeu que toda organização é desafiada a equilibrar essas duas vertentes, constatando que a prática da *exploitation* garante a viabilidade dos negócios a curto prazo e a *exploration* seu sucesso a longo prazo.

Apesar de cada uma apresentar objetivos divergentes, trabalhos acadêmicos recentes confirmaram que ambas podem ser realizadas através de uma organização ambidestra. Diante disso, verifica-se que um desafio fundamental no contexto empresarial é acomodar ambas essas estratégias, pois isoladamente nenhuma promove um sucesso organizacional duradouro. Por exemplo, ao focar majoritariamente em *exploitation*, uma companhia conseguirá gerar capital por um certo período de tempo limitado, mas falhará mais adiante em se adaptar aos avanços da tecnologia e às mudanças de mercado. De modo similar, entidades que focam apenas em inovação tendem a ser prejudicadas por não gerenciar eficientemente suas operações, de acordo com O'Reilly & Tushman (2013).

Ainda assim, Gupta *et al.* (2006) oferecem uma visão alternativa na qual o desequilíbrio para uma demanda em detrimento à outra pode fazer a organização mais específica, eficiente e melhorar o desempenho. Portanto, há diferentes modos de exercitar as capacidades de *exploration* e *exploitation*, os quais este trabalho endereça.

## **2.2 Ambidestria organizacional no setor de software**

Organizações que estão inseridas no setor de software enfrentam constantemente escolhas estratégicas contraditórias, tais como o desenvolvimento e/ou manutenção de soluções completas para diferentes clientes ou soluções customizadas, como argumenta Cordeiro (2014), a necessidade de escolher entre melhorar a eficiência no desenvolvimento de processos ou ajustar as necessidades de novos clientes, decidir sobre adaptação do portfólio de produtos, melhoramento do seu software em relação aos clientes existentes e competidores diretos ou explorar novas tecnologias e oportunidades de mercado e, por fim, considerar ambas as inovações como incrementais ou profundos, propõem Napier, Mathiassen, & Robey (2011).

Dado a circunstância de extrema complexidade, pesquisadores têm buscado compreender como empresas de software lidam com a ambidestria organizacional, conforme trabalhos apresentados no Quadro 3.

**Quadro 3:**

## Estudos em ambidestria organizacional no setor de software

<b>Autores</b>	<b>Tipo de Pesquisa</b>	<b>Amostra</b>	<b>Objetivos</b>	<b>Dualidades de Ambidestria</b>	<b>Conclusão</b>
Boehm e Turner (2004)	Qualitativo	2 organizações	Examinar aspectos de projetos relacionados aos métodos ágeis e disciplinados e prover uma abordagem para equilibrá-los	Agilidade e Disciplina	Os autores sugerem que as características de projetos, tais como: conjunto de habilidades desenvolvidas, disponibilidade do cliente e antecipação de requisitos são utilizados para guiar a seleção de um desenvolvimento de processos que melhor se ajusta à organização.
Vinekar <i>et al.</i> (2006)	Discussão Técnica	Não se aplica	Compreender as razões que levam as organizações sentem a necessidade de conviver com os desafios para sustentar culturas opostas.	Metodologias tradicionais e Metodologias Ágeis	Os autores recomendam que as empresas do setor de software criem uma subunidade tradicional, focando em <i>exploitation</i> e uma subunidade ágil, focando em <i>exploration</i> , sugerido por Tushman & O'Reilly, (1996).
Venkatraman <i>et al.</i> (2006)	Quantitativo	1005 organizações	Avaliar o impacto da ambidestria no desempenho de empresas do setor de software ao longo de um período de 12 anos.	<i>Exploitation</i> e <i>Exploration</i>	As evidências sugerem que o equilíbrio sequencial entre <i>exploitation</i> e <i>exploration</i> , ao longo do tempo, possui efeitos significativos no desempenho das empresas.

Napier, Mathiassen e Robey (2011)	Qualitativo	1 organização	Desenvolver, aplicar e analisar um framework que integra a teoria existente da ambidestria contextual (Gibson & Birkinshaw, 2004) com os processos de coordenação, projeto e inovação para o aperfeiçoamento das organizações de software.	Alinhamento e Adaptabilidade	Os autores propõem que gestores desenvolvam um alto desempenho contextual e aumentem a habilidade da organização de software para alinhar-se e adaptar-se diante das demandas atuais e futuras de mercado.
Cordeiro (2014)	Qualitativo e Quantitativo	466 organizações(Survey) e 6 organizações (Estudo de caso)	Buscar evidências que possam comprovar se a escolha da solução organizacional para a criação da ambidestria em empresas desenvolvedoras de software é influenciada pelas barreiras encontradas no processo de introdução de uma metodologia ágil como segunda opção metodológica.	Metodologias Ágeis e Metodologias Tradicionais	O autor encontrou indícios de que algumas das barreiras enfrentadas pelas organizações na introdução de metodologias ágeis, entre elas a diversidade de clientes e de projetos, podem influenciar a opção das empresas na escolha do tipo de ambidestria a ser adotado. O estudo apontou também, que 48,8% das empresas que adotaram a prática de metodologias ágeis praticam a ambidestria.

Martini (2015)	Qualitativo	7 organizações	Investigar quais são os fatores que impedem grandes empresas de software de se tornarem ambidestras. Prover soluções para mitigar tais desafios	Curto Prazo e Longo Prazo	O estudo apontou três principais desafios que estão dificultando o alcance da ambidestria: (a) conflito entre metodologias ágeis e tradicionais; (b) complexidade de equilibrar metas de curto e longo prazo entre grande número de stakeholders com diferentes visões e especialidades e (c) a tendência arriscada de desenvolver sistemas que não se sustentam com uma entrega de longo-prazo com novas características. O autor propõe uma solução que inclui papéis, times e práticas, as quais são necessárias para integrar os stakeholders.
Tonelli <i>et al.</i> (2016)	Qualitativo	1 organização	Apontar e analisar, sob a perspectiva da teoria da prática, o processo de formação de ambidesteridades dentro do processo de inovação, tendo como fundamento a constituição mútua entre ações localizadas e aspectos da cultura.	Ação e Estrutura	Os resultados apontaram que o processo de construção de ambidesteridade baseia-se em 3 estágios: geração de conflitos iniciais; reação aos conflitos e estabilização via dinâmicas entre lógicas e ações.

Fonte: adaptado de Severgnini, E., Galdamez, E., Vieira, V. (2019)

Por exemplo, os estudos apresentados por Boehm e Turner (2004) depararam com limitações na capacidade das organizações de software alcançarem a ambidestria devido: (a) ao esforço unicamente no nível de projeto e (b) ao desconhecimento dos efeitos das metodologias ágeis e tradicionais.

Vinekar *et al.* (2006) sugerem que as organizações promovam estruturas distintas a fim de provocar a ambidestria natural, ponderando a falta de recursos, o conhecimento e a informação como fortes influenciadores da ambidestria. Esses influenciadores também são elementos sugeridos por Katila e Ahuja (2002). Os estudos de Martini (2015) e Tonelli, Zambalde, Brito & Bermejo (2016), por se tratar de estudos de caso, limitam a generalização das barreiras que dificultam a ambidestria nas indústrias de software. Por fim, Venkatraman, Lee e Iyer (2006), únicos autores que investigaram a dualidade *exploitation* e *exploration*, apresentaram evidências de uma relação sequencial no desempenho, na qual a dimensão *exploration* impacta a dimensão de *exploitation* que, por sua vez, influencia o desempenho.

Portanto, dado as lacunas levantadas na revisão de literatura, provoca-se uma análise mais ampla da relação existente entre a ambidestria organizacional e o desempenho no setor de software, seguindo as sugestões de Gibson e Birkinshaw (2004) e utilizando a congruência.

### **2.2.1 Ambidestria organizacional: vantagens**

Através de uma extensa revisão de literatura, O'Reilly e Tushman (2013) observaram que sob certas condições de incerteza mercadológica e tecnológica, especialmente no contexto de organizações de grande porte, as quais geralmente possuem maior disponibilidade de recursos, a ambidestria organizacional afeta positivamente o seu desempenho financeiro, o seu potencial inovador e a sua longevidade. Vinekar *et al.* (2016) descobriram, também, que empresas de software as quais adotam uma estratégia ambidestra possuem uma tendência maior em obter sucesso na introdução de novos produtos e a aumentar o lucro e a satisfação dos seus clientes.

Tais evidências confirmam que a ambidestria organizacional é uma capacidade fundamental para a sustentabilidade tanto a curto quanto a longo prazo das

organizações. Faz-se, portanto, imprescindível que as instituições criem mecanismos adequados para avaliar qual abordagem de ambidestria se faz apropriada para a realidade delas e em que níveis seus esforços devem ser direcionados ao alcance da *exploitation* e da *exploration*.

### **2.2.2 Orientação Estratégica**

A literatura de orientação estratégica emerge dentro do campo de estudo de estratégia empresarial. Ainda que essa última área seja relativamente recente, a teorização e sistematização do conhecimento sobre estratégia tem seu início a mais de 2.500 anos na China com a obra de Sun Tzu denominada “A Arte da Guerra”, na qual, dentre outros ensinamentos, o autor pondera sobre a necessidade do conhecimento pessoal e do adversário para sair vitorioso nas diversas empreitadas (TZU, 2003).

De forma análoga, transportando para o campo dos negócios, também é necessário, no momento de formulação das estratégias empresariais, o conhecimento próprio da organização (conhecimento de si mesmo) e do ambiente externo (conhecimento do adversário). Pode-se ver, portanto, dentre outros fatores, que os campos de estudo em estratégia militar e estratégia empresarial guardam muitas similaridades.

Tais similaridades se devem, sobretudo, ao fato de que a partir da segunda metade do século XX, o conceito de estratégia militar foi transportado para dentro do campo de estudos de administração empresarial. O ano de 1965 poderia ser lembrado como um marco com a publicação de “*Corporate Strategy*” de Igor Ansoff, na qual o autor apresenta quatro tipos de estratégia, que ficaram conhecidas como a “matriz produto/mercado” ou “matriz Ansoff” (ALDAY, 2000).

Ainda segundo Alday (2000) foi a partir da década de 1970 que o estudo de estratégia empresarial eclodiu dentro da literatura de administração aumentando a demanda por estudos e publicações acerca do tema.

Hofer e Schendel (1978) caracterizam a estratégia como a união entre as capacidades, no sentido de aptidão, e recursos internos da própria organização com as oportunidades e riscos presentes no ambiente externo à firma, de tal forma que

uma estratégia eficiente seja capaz de utilizar os recursos internos da firma na captação de oportunidades presentes no ambiente econômico em que a empresa opera.

Para Porter (1980) a estratégia é vista como a junção por um lado das metas e objetivos almejados pela organização e por outro das ações e esforços despendidos para o alcance dos mesmos.

A orientação estratégica de uma organização é um componente fundamental de sua filosofia de negócios, transcende as competências individuais e reflete os valores essenciais que guiam suas decisões. Segundo Day (1994), essa filosofia não se baseia apenas nas habilidades pessoais dos administradores, mas sim na coletividade da organização. Ela unifica os recursos e competências da empresa, influenciando diretamente as práticas gerenciais dos dirigentes, como destaca Rajagopalan (1996).

A estabilidade da orientação estratégica ao longo do tempo é crucial, pois, conforme aponta Venkatraman (1985), ela deve ser mais duradoura do que simples iniciativas individuais para manter a posição de mercado. Esse elemento revela-se enraizado nas práticas gerenciais da empresa, proporcionando consistência e direção. Além disso, a variabilidade da orientação estratégica entre empresas de um mesmo setor, como observado por Venkatraman (1985), destaca que diferentes organizações podem adotar abordagens distintas, mesmo operando no mesmo segmento de mercado. Nesse contexto, Machado da Silva e Nogueira (2001) enfatizam que a filosofia de negócios, ao ser moldada por valores resultantes de um processo de decisão racional, reflete as preferências da empresa por determinados cursos de ação, destacando a importância desses valores na construção da identidade organizacional e nas escolhas estratégicas.

Dentro da literatura de orientação estratégica, há três grandes categorias gerais que podem ser encontradas a partir de uma análise de seu conteúdo. Essas categorias são apontadas por Morgan e Strong (2003) e apresentadas a seguir:

- Narrativa: os estudos narrativos de orientação estratégica descrevem a natureza holística da filosofia de negócios, a qual é única para o caso, para a

situação e para a organização. A ênfase dessa abordagem quanto a sua metodologia é qualitativa e quase sempre assume a forma de estudo de caso.

- **Classificatória:** esses estudos classificam as filosofias de negócios em tipos já existentes, por isso mesmo as distintas classificações são conhecidas como tipologias (conceitual) ou taxonomias (empírica) de orientação estratégica.
- **Comparativa:** dentro dessa categoria são identificadas combinações de atributos e dimensões estratégicas. Ela pretende descrever a orientação estratégica adotada por uma firma, portanto, por meio de múltiplas habilidades (ou atributos) e dimensões comuns a um conjunto de firmas.

Portanto, os autores Morgan e Strong (2003), consideram a proposta de uma abordagem flexível e dinâmica na análise das orientações estratégicas, não se buscando uma classificação estrita e pré-determinada. Em vez disso, adota-se uma perspectiva que permite a construção de tipologias com base no comportamento estratégico observado nas empresas. Essa flexibilidade reconhece a complexidade e a natureza mutável do ambiente de negócios, destacando a importância de uma abordagem adaptativa para compreender as diferentes manifestações da orientação estratégica.

A capacidade de adaptar e reconfigurar estratégias, incorporando elementos de *exploitation* e *exploration* conforme necessário, é desafiador para o sucesso organizacional em um cenário empresarial em constante evolução.

Nesse contexto, ainda os autores, destacam que a análise das orientações estratégicas não se limita a categorias estáticas, mas busca capturar a dinâmica e a diversidade que caracterizam as decisões estratégicas nas organizações contemporâneas.

## 2.3 Engenharia de software

Desde a década de 60, profissionais da área de Tecnologia da Informação discutem a necessidade e importância da organização dos processos de desenvolvimento como forma de se atingir a melhor qualidade dos produtos de software, de acordo com Fleury & Vargas (1983). O entendimento de que o projeto, desenvolvimento e manutenção de softwares também devem ser visto como um processo de produção, tal qual processos produtivos aplicados a bens materiais, levou estudiosos da área a propor a adoção de práticas de engenharia para sua organização.

Segundo o autor, Sommerville (2011, p.5), a disciplina de engenharia de software é relevante por dois motivos, o primeiro em relação a sociedade, cada vez mais, depende de sistemas de software avançados. O segundo motivo, a longo prazo, normalmente, é mais barato usar métodos e técnicas propostos pela engenharia de software, ao invés de somente escrever os programas como se fossem algum projeto pessoal.

O autor ainda contribui complementando, que a engenharia de software se torna essencial para o total funcionamento da vida em sociedade, seja ela nacional ou internacional, pois está presente nos mais diversos segmentos.

A disciplina de engenharia de software, manifestou-se na década de 60, *inicialmente* proposto no ano de 1968, em uma convenção organizada para tratar a “Crise do Software”, onde diversos cenários foram discutidos para a introdução de um novo hardware de computador baseado em circuitos integrados e sistemas eletrônicos. O não atendimento dos prazos, falta de documentação dos requisitos funcionais e técnicos, e a baixa qualidade dos produtos de software foram alguns dos problemas encontrados durante a “Crise do software”.

Assim, Souza (2014), reforça que, como solução para estes problemas, surgiu a engenharia de software com suas técnicas, ferramentas e métodos. A partir disso, a

sua ascensão foi excepcional, e a engenharia de software passou a figurar um papel fundamental como boas práticas de desenvolvimento de softwares.

Pressman (2016), define engenharia de software como uma aplicação de abordagem sistemática e disciplinada. O autor ainda complementa que a aplicação da engenharia de software é quantificável para o desenvolvimento, operação e conservação de produtos de software.

Segundo Williams (2010), a engenharia de software segue o conceito de disciplina na produção de software. Afirma ainda, que é procedente em metodologias, e que por sua vez, seguem métodos que utilizam ferramentas e *frameworks* automáticos, para abarcar as principais atividades do processo de produção. De acordo com suas apurações, Pressman (2016), reitera que: “software, em todas as suas formas e em todos seus campos de aplicação, deve passar pelos processos de engenharia”.

Pressman (2016), define software como:

Software de computador é o produto que os profissionais de software constroem e, depois, mantêm ao longo do tempo. Abrangem programas que executem em computadores de qualquer tamanho e arquitetura. É um conjunto composto por instruções de computador, estruturas de dados e documentos.

O que mais interessa neste estudo é entender como a disciplina de engenharia de software está conectada com obter resultados de qualidade, por isso a gênese que a sustenta é o foco na qualidade, sendo que os principais elementos de um produto de software, são manutenibilidade, confiança, proteção, eficiência e aceitabilidade.

Adiante, acerca da definição da engenharia de software, em conformidade com Pressman (2006), a engenharia de software “é uma tecnologia em três camadas: processos, métodos e ferramentas”.

Cada atuação apresentada por estas três camadas, observado na Figura 1, são requisitos primários para a construção de um software, logo que, promovem o arcabouço necessário para a produção dos artefatos e são responsáveis em apresentar um resultado com um produto final.



Figura 1 - Disposição dos níveis da Engenharia de Software.  
Fonte: Pressman (2006, p.17)

A engenharia de software, compreende todos esses fatores especificados para o desenvolvimento. Na Figura 1, ilustra-se bem os elementos e a importância da disposição dos níveis; o foco na qualidade é a primeira camada, portanto, a engenharia de software é fundamentada na qualidade e se justifica o fato de estar no princípio. Adiante os processos, que é a abordagem que favorece a seleção de tarefas e ritos necessários para tornar possível o produto de software.

Na sequência, pelos métodos e as ferramentas. Os métodos promovem técnicas para desenvolver o software, envolve diversas tarefas como comunicação, refinamento de requisitos funcionais e técnicos entre outros, e por fim as ferramentas, que propiciam o suporte automatizado para a camada de processo e métodos.

Os ritos e condutas sistemáticas usadas na engenharia de software é, em algumas vezes, chamada processo de software”, Sommerville (2011, p.5). Este argumento será tratado no capítulo seguinte. Para assegurar a produção de um software, são necessárias diversas etapas, precedidas por métodos muito bem estabelecidos e compostas de uma série de tarefas em cada uma delas, conforme apresentado ao longo do próximo tópico.

### **2.3.1 Metodologias de processos de software**

A adoção de uma metodologia para o desenvolvimento de um software, é o ponto chave para o sucesso do projeto ou produto. Cockburn (2002) define o termo metodologia como “uma série de métodos e técnicas”, onde o método se refere a “procedimentos sistemáticos”. Segundo Sommerville (2011), um modelo de processo de software, é uma representação abstrata, simplória, de um processo de software. Os modelos de processo, incluem as tarefas que fazem parte do processo. “Cada modelo reproduz uma perspectiva particular de um processo e, portanto, fornece informações parciais sobre ele” Sommerville (2011, p.19).

Em resumo, o objetivo desses modelos, é demonstrar um esboço conceitual para sustentar o desenvolvimento e fornecer um suporte maior ao desenvolvedor da aplicação. A classificação destes modelos encontrada na literatura especializada SANTOS (2007, p.34), com Pressman (2016), Sommerville (2011), Cockburn (2002), Fowler (2003), é a seguinte:

- Metodologias Tradicionais, ou “pesadas”;
- Metodologias Ágeis, ou “leves”.

A seguir, os modelos de processo que serão apresentados e descritos, foram recolhidos de Sommerville (2011) e Pressman (2016), e de alguns outros autores que serão mencionados ao longo do estudo.

### **2.4 Metodologias tradicionais**

As metodologias clássicas desenvolvidas para suportar os processos de engenharia de software, bem como as definições dos papéis e estruturas organizacionais desenhadas desde as primeiras fábricas de software na década de 70, segundo Cusumano (1991), e que ainda persistem em grande parte das equipes de desenvolvimento de software nos mais diversos tipos de ambientes empresariais, guardam enormes similaridades com as estruturas organizacionais Tayloristas, com a divisão das tarefas e alocação a profissionais especializados SENAI (2004).

As abordagens de desenvolvimento de software tradicionais são orientadas em processos e conduzidas pela convicção de que as fontes de variações, seja nas

particularidades dos requisitos técnicos e funcionais dos usuários/clientes, são identificáveis e podem ser subtraídas pela contínua mensuração e refino dos processos. Sendo assim, o seu objetivo está na implementação de processos repetitivos e altamente otimizados, afirmam Cockburn e Highsmith (2001).

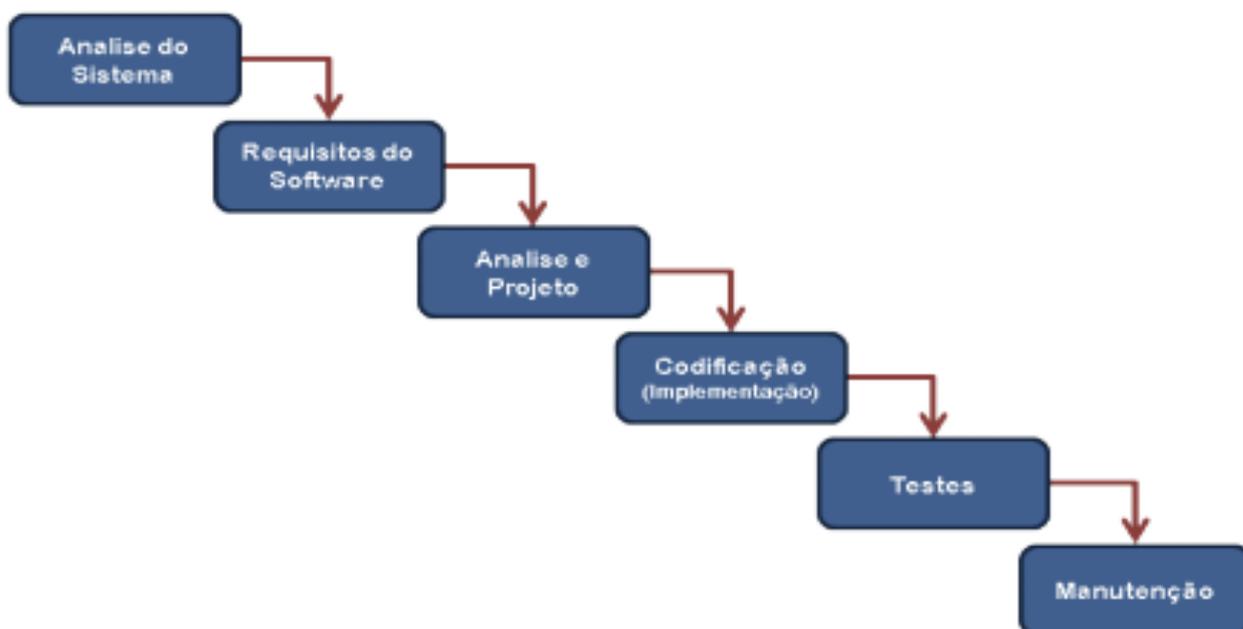


Figura 2 - Modelo cascata de desenvolvimento.  
Fonte: Pressman (2010)

Ao analisarmos o padrão clássico do ciclo de desenvolvimento de software, na perspectiva tradicional, o modelo cascata, ilustrado na Figura 2, nota-se como um processo sequencial, linear, do mesmo arquétipo das linhas de montagem industriais, estabelecendo etapas bem definidas e evolutivas desde o levantamento de requisitos, até os testes e liberação do produto de software aplicável.

Segundo Pressman (2010), apesar de ter sido ajustado ao longo do tempo para adequação à necessidade de otimização do processo de desenvolvimento de software, possibilitando incorporar etapas de prototipagem e homologação junto aos usuários e clientes, e para promover entregas incrementais, evolutivas e modularizadas através de ciclos iterativos, as etapas básicas previstas no Modelo Cascata de desenvolvimento de software estão ainda presentes em praticamente todos os processos metodológicos atuais. Com excessiva ênfase na documentação, as metodologias tradicionais despendem, portanto, grande esforço no descobrimento, análise e especificação dos requisitos do sistema nas etapas iniciais do

desenvolvimento, assumindo que seja possível mantê-los estáveis e imutáveis ao longo do ciclo de vida.

Os registros e as documentações geradas nesta etapa inicial compreenderão uma formalização entre o cliente e o desenvolvedor, auxiliando também como um mecanismo de comunicação formal de especificações funcionais e técnicas entre a equipe de desenvolvimento ao longo de todo o ciclo de vida do processo de desenvolvimento do software. Observa-se neste cenário uma exposição das características das metodologias tradicionais a rigidez e burocracia formal, disciplinada, do processo de desenvolvimento repetitivo para suportar as demandas pré-definidas, com a exploração de competências e práticas já estabelecidas. Estas características possibilitam as condições ideais para a condução de projetos com maior previsibilidade, estabilidade e garantia de qualidade, de acordo com Boehm (2003).

O modelo cascata de desenvolvimento pressupõe a definição de tarefas específicas, que exigem profissionais com papéis muito bem estabelecidos e competências específicas.

Apesar de cada novo sistema computacional ou funcionalidade a ser desenvolvido a partir de uma nova especificação ou requisito funcional e técnico, e, portanto, se caracterizar como um novo produto, cujo conteúdo é construído de forma inédita, diferente dos sistemas produzidos anteriormente, os profissionais de software possuem competências claramente definidas (capacidade de abstração, raciocínio lógico, domínio de técnicas, etc.), usam ferramentas padronizadas (linguagens, modelos, plataformas de desenvolvimento, etc.) e têm sua produtividade mensurável (homem/hora por funcionalidade, linhas de código por homem/hora, pontos por função, etc.), como reconhecido por Pressman (2010).

Os grupos de projeto formados para o desenvolvimento de sistemas, orientados a funcionalidade ou pelo projeto integral, dirigidos por metodologias clássicas também possuem características organizacionais convencionais. Estruturas organizacionais departamentalizadas conciliam perfis diferentes de profissionais, separando em equipes de analistas, equipes de programadores, equipes de testes e qualidade e assim por diante, como células organizadas em função de suas habilidades.

O Quadro 4 apresenta as principais atribuições presentes em uma equipe clássica de desenvolvimento de software e sua ligação com as atividades de desenvolvimento do modelo.

**Quadro 4:**

Perfis ocupacionais em uma equipe clássica/tradicional

<b>Atividade</b>	<b>Perfil Ocupacional</b>	<b>Descrição</b>
Análise de Sistema	Analista de Sistemas	Compreende a necessidade do cliente/usuário e cria as especificações iniciais e o detalhamento do sistema
Análise de Requisitos	Analista de Sistemas	
Análise e Projeto	Projetista	Partindo das especificações iniciais do sistema, elabora o projeto detalhado, traduzindo as especificações em modelos e recomendações de construção.
Implementação	Programador	Codifica o software e gera conteúdo para sua documentação. É responsável por testes preliminares,
Testes	Testador	Integra diferentes componentes e módulos e testar o sistema buscando identificar falhas
Manutenção	Programador	Proceder às manutenções corretivas e evolutivas do software, com base nos resultados dos testes aplicados pelo Testador e/ou de novas especificações geradas.

Fonte: baseado em Fleury e Fleury (2007)

O texto acima descreve as funções e atividades presentes em uma equipe clássica de desenvolvimento de software, incluindo a identificação de necessidades, especificação técnica, codificação, testes e manutenção. Cada membro da equipe tem atribuições específicas, mas todas estão interligadas e devem trabalhar em conjunto para garantir o sucesso do projeto. Assumindo a formação original destes grupos de projeto, com a alocação de profissionais específicos de cada perfil para a participação dedicada em um determinado esforço de desenvolvimento, esses profissionais mantêm seus papéis originais, suas hierarquias matricialmente organizadas junto aos

departamentos da organização, com um baixo nível de interação e autonomia, garantido muito mais pela formalização e troca de documentos de especificação e modelagem de projeto, do que pela interação direta, colaborativa e autônoma.

## 2.5 Metodologias ágeis

Apreciamos a jornada das organizações voltadas à produção, para viabilizar linhas mais flexíveis e dinâmicas com a manifestação das abordagens sociotécnicas, que formalizaram a ideia de grupos de trabalho semiautônomos, as organizações com orientação à fábricas de software também buscaram o entendimento da adequação ao impacto da formação de grupos capazes de tornar o desenvolvimento de software mais dinâmico, flexível e ágil.

Ao oposto das metodologias tradicionais, as metodologias ágeis toleram a imprevisibilidade, e fomentam a potencialidade das pessoas e sua criatividade para a condução de programas e projetos ao invés de se apoiar em processos, sendo assim adequadas ao desenvolvimento de softwares em ambientes de maior instabilidade, onde os requisitos do software não se apresentam claros desde o início e com potenciais redefinições ao longo de seu desenvolvimento. Para este estudo, é relevante explorar as origens das abordagens iterativas e incrementais que contribuíram para a existência dos métodos ágeis atuais, destacando-se o panorama nos quais foram aplicados, Cordeiro (2014).

Larman & Basili (2003) e Whiteley *et al.* (2021) mostram que as primeiras iniciativas de desenvolvimento incremental iterativo de produtos surgiram na década de 1930 com Walter Shewhart, da Bell Labs, propondo melhoria da qualidade a partir de ciclos curtos de PDSA (Planejar ⇒ Fazer ⇒ Estudar ⇒ Agir). Recebeu incrementos na década seguinte, por parte de Edwards Deming.

Na década de 1960, a Administração Nacional da Aeronáutica e Espaço (NASA) usou IID no software do Projeto *Mercury*, sendo em seguida levado para a *International Business Machines Corporation Federal Systems Division* (IBM FSD) por membros que trabalharam no projeto. Larman & Basili (2003) e Whiteley *et al.* (2021) atestam que muitos líderes importantes da engenharia de softwares daquela década apoiaram as práticas de desenvolvimento incremental e iterativo. Ocorreram então a inclusão

da prototipagem rápida pelos projetos aeroespaciais e a introdução do design colaborativo com foco no engajamento e na dinâmica de poder, como estratégias para viabilizar o desenvolvimento incremental e iterativo. Sobre a década de 1970, Larman & Basili (2003) indicam ser Winston Royce o criador do modelo em cascata, tendo proposto a utilização de um processo de desenvolvimento iterativo e evolutivo com *feedback*. Na mesma década, Harlan Mills propôs o uso, no IBM FSD, de uma metodologia conhecida como *Top Down Programming in Large Systems*, programação de cima para baixo em grandes sistemas.

A década de 80 foi bastante latente, tendo marcado tendências no uso de métodos precursores das metodologias ágeis e contribuindo para a consolidação de práticas. Whiteley *et al.* (2021) entendem que a combinação dos princípios e técnicas de fabricação japonesa com as técnicas de gestão da qualidade americanas foi crucial para a melhoria da qualidade da manufatura, com destaque para os círculos de qualidade de Ishikawa e suas aplicações em subcontratação e marketing. Outra abordagem a se destacar foi proposta por Hirotaka Takeuchi e Ikujiro Nonaka, em 1986, na publicação do artigo pela revista Harvard Business Review, com o título *The new product development game* (Takeuchi & Nonaka, 1986) - o novo jogo de desenvolvimento de novos produtos, que mais tarde, em 1993, inspirou a criação do *framework* ágil *Scrum*.

Ainda, na década de 1990 ocorreu um forte movimento de consolidação do uso de métodos incrementais iterativos, na crescente indústria de software. Muitas publicações científicas e didáticas que tratavam do tema colaboraram para a difusão e fortalecimento das práticas que envolviam iterações. Um exemplo é a adoção do *timebox* semanal. Havia um forte movimento de rejeição de práticas de especificação herdadas do modelo cascata, preferindo-se uma análise evolucionária, tendo o Departamento de Defesa Americano como forte impulsionador ao substituir o modelo em cascata pelo IID nas suas contratações. Segundo Larman & Basili (2003), surgiram nessa época os primeiros trabalhos de Jeff Sutherland e Ken Schwaber, que se tornaram o método *Scrum*, empregando as iterações e o limite de tempo, chamados de *timebox* ou *sprint*. O método foi inspirado nas iterações japonesas usadas na manufatura da Honda, Canon e Fujitsu, na década de 1980. Posteriormente, teve-se uma versão mais refinada do *Scrum* contida no 24 artigo escrito por Takeuchi &

Nonaka (1986) e, por fim, na pesquisa de Beedle, Devos, Sharon, Schwaber & Sutherland (1999).

Contudo, foi entre 11 e 13 de fevereiro de 2001 que 17 pessoas ligadas ao uso de variadas metodologias de desenvolvimento de software, mas que até então nenhuma era chamada de ágil, se reuniram e nesse encontro foi criado o Manifesto para Desenvolvimento Ágil de Software (Beck *et al.*, 2001). Havia representantes da *Extreme Programming*, *Scrum*, *DSDM*, *Adaptive Software Development*, *Crystal (ASD)*, *Feature-Driven Development*, *Pragmatic Programming*, entre outros simpáticos à necessidade de uma alternativa aos processos de desenvolvimento de software tradicionais em cascata, orientados à documentação. O manifesto para o desenvolvimento de software ágil criado é composto de valores e princípios.

Em cada um dos quatro valores expressos é correto afirmar que há mais estima pelos itens à esquerda, mesmo havendo valor nos itens à direita (Beck *et al.*, 2001 p.1):

- a) Indivíduos e interações mais que processos e ferramentas;
- b) software em funcionamento mais que documentação abrangente;
- c) colaboração com o cliente mais que negociação de contratos;
- d) resposta a mudanças mais que seguir um plano;

Os 12 princípios do manifesto ágil são (Beck *et al.*, 2001, pp.1-12):

- Nossa maior prioridade é satisfazer o cliente por meio da entrega contínua e adiantada de software com valor agregado;
- mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;
- entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
- pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
- construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho;

- o método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é a partir de conversa face a face;
- software funcionando é a medida primária de progresso;
- os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
- contínua atenção à excelência técnica e bom design aumenta a agilidade;
- simplicidade - a arte de maximizar a quantidade de trabalho não realizado - é essencial;
- as melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis;
- em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Pelo manifesto ágil abstrai-se que os métodos ágeis são modelos organizacionais baseados nas pessoas, na colaboração e na construção de comunidades organizacionais. Ghezzi & Cavallo (2020) asseguram que métodos ágeis são estruturas conceituais que apresentam métodos e técnicas capazes de responder às mudanças, em ambientes incertos e turbulentos, os quais permitem a identificação, enfrentamento e adaptação.

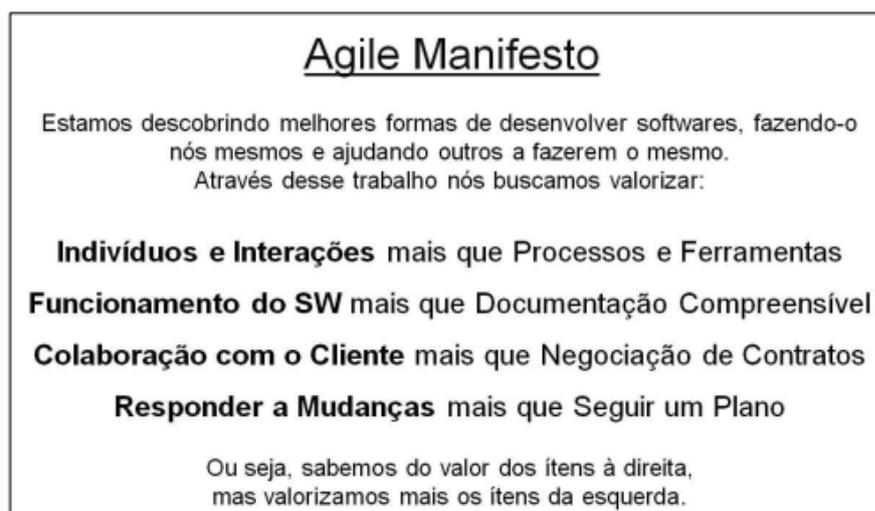


Figura 3 - Manifesto ágil da Agile Alliance  
Fonte: Baseado em [www.agilealliance.org](http://www.agilealliance.org) (Dezembro/2022)

A Figura 3 apresenta os princípios do manifesto ágil, documento redigido pela *Agile Alliance* para tornar público os objetivos comuns das metodologias ágeis.

Dados estes princípios e características das metodologias ágeis sua aplicação é orientada para a busca da satisfação do usuário final, com flexibilidade para a adoção de mudanças nos requisitos ao longo do desenvolvimento, menor número de incidentes em função do ciclo constante de testes e validações junto aos clientes, tempos de desenvolvimento mais curtos com entregas frequentes e incrementais. Adequado em projetos onde os requisitos ainda não são totalmente conhecidos desde o início.

Processos ágeis são caracterizados pela realização de ciclos iterativos de curta duração definidos pelas funcionalidades dos produtos, além de contar com períodos de reflexão, adaptabilidade e introspecção, tomada de decisões em modo colaborativo, rápida reação a feedbacks e mudanças de requisitos ou escopo.

O projeto é dividido em um conjunto de funcionalidades, são discutidos e decididos de forma democrática e colaborativa pela equipe de desenvolvimento e pelo representante do cliente com o objetivo de entregar a funcionalidade utilizável do produto final a cada fim de ciclo. Cada ciclo de desenvolvimento é realizado por um pequeno grupo composto por desenvolvedores e clientes que normalmente envolvem o planejamento, o desenvolvimento, integração e teste de código, de acordo com Nerur, Mahapatra & Mangalaraj (2005).

As atribuições dentro do time de desenvolvimento são definidas pelo próprio grupo, sendo encorajado o frequente intercâmbio de funções e papéis entre cada membro da equipe, que atua de forma semiautônoma e autoadministrada.

Esta configuração organizacional possibilita a interação entre desenvolvedores e também entre o cliente ou o usuário final, possibilitando a tomada de decisões colaborativa e a substituição de documentações formais de requisitos pela comunicação direta e clara dentro da equipe, criando assim condições para a rápida assimilação de mudanças dos requisitos e para a rápida atuação na correção de defeitos ou inadequações de código no momento de sua geração.

É fácil perceber que um dos aspectos críticos para o sucesso das metodologias ágeis é a participação ativa, colaborativa e presencial do cliente junto à equipe de

desenvolvimento. É essa participação ativa que irá viabilizar o comprometimento entre as partes quanto a prazos e escopo, a troca das especificações documentadas pela comunicação oral, e a rápida validação, teste e aceitação de cada componente desenvolvido.

É importante frisar que uma estrutura ágil, seja um *framework* ou uma metodologia ágil, é um conjunto de diretrizes e princípios para o desenvolvimento de software que enfatiza a iteração contínua, entrega rápida e colaboração próxima entre desenvolvedores e partes interessadas. Existem vários métodos ágeis diferentes, que seguem esses valores fundamentais. De acordo com levantamento do IGTI, o *Scrum* é o *framework* ágil mais famoso entre as empresas brasileiras, presente em 74% delas por sua eficiência na otimização de processos e sua versatilidade: ele se adapta facilmente a diversas áreas e não fica restrito apenas ao desenvolvimento de tecnologias. Nesta pesquisa dá-se mais ênfase ao *Scrum*.

### **2.5.1 Scrum**

O desenvolvimento de software é tratado por metodologias tradicionais como um conjunto de processos definidos, o que não tem historicamente levado a bons resultados. Processos definidos, ao receberem um conjunto bem definido de entradas, resultam sempre nas mesmas saídas e com riscos altos de imprevisibilidade de requisitos funcionais. De forma contrária, o *framework Scrum* trata o desenvolvimento de software como um conjunto de processos empíricos, pois acredita que a complexidade do desenvolvimento de software, por ser uma atividade criativa e de trabalho intelectual intenso, leva a resultados imprevisíveis. Por essa razão, *Scrum* é fundamentado na teoria de controle de processos empíricos, que possui três pilares: a visibilidade, que garante que aspectos do processo que afetam os resultados devem estar visíveis e ser bem compreendidos por aqueles que gerenciam os resultados; a inspeção, que significa que os diversos aspectos do processo devem ser inspecionados com uma frequência suficiente para que variações inaceitáveis no processo possam ser detectadas; e a adaptação, de forma que se quem inspeciona o processo determinar, a partir da inspeção, que um ou mais aspectos do processo estão fora dos limites aceitáveis e que, assim, o produto resultante será inaceitável,

ele deverá ajustar o processo ou o material sendo processado o mais rápido possível, de forma a minimizar desvios posteriores Schwaber & Beedle (2002).

Práticas do *Scrum* incluem o uso de equipes autogerenciadas e multidisciplinares, o bloqueio à adição de novos requisitos durante o curso da iteração de desenvolvimento, o uso de eventos de duração determinada (como suas reuniões e a própria iteração de desenvolvimento), uma demonstração realizada às partes interessadas no final de cada iteração, e o planejamento adaptativo orientado às necessidades do cliente, ao final de cada iteração, segundo Larman (2003).

#### 2.5.1.1 Visão geral do ciclo do scrum

No *Scrum*, o produto é desenvolvido em ciclos ou iterações com duração fixa entre duas e quatro semanas, chamados de sprints. No início do projeto, o cliente estabelece uma visão relativamente superficial para o sistema a ser desenvolvido, que se tornará mais clara no transcorrer do desenvolvimento. Essa visão deve ser determinante para todas as novas funcionalidades que serão desenvolvidas para o produto.

Antes do início de cada sprint, um representante do cliente devidamente escolhido, chamado de *Product Owner*, realiza um levantamento de requisitos com o cliente considerado os mais prioritários para o cliente naquele momento, de acordo com o valor de negócio. O *Product Owner* insere, então, esses requisitos no *product backlog*, que é uma lista priorizada de todos os requisitos do produto conhecidos até o momento, e a reprioriza, atualizando requisitos já existentes e eliminando aqueles que não são mais necessários para o cliente. No primeiro dia da *sprint*, a equipe de desenvolvimento se reúne com o *Product Owner* na reunião de *sprint planning* (planejamento da *sprint*), onde é decidido qual conjunto de itens será retirado do *product backlog* para ser desenvolvido durante a *sprint*. Esses itens são então detalhados em tarefas necessárias para implementá-los pela equipe de desenvolvimento, e esse conjunto de itens com suas tarefas correspondentes passam a ser chamados de *sprint backlog*. Após a reunião (geralmente no dia seguinte), a equipe inicia o trabalho de desenvolvimento do software propriamente dito, conforme planejado, até o final do penúltimo dia da *sprint*. Em cada dia de trabalho da *sprint*, a equipe realiza a reunião chamada *daily scrum* (*scrum* diário), em que se reúne durante

quinze minutos para promover visibilidade do andamento da iteração para seus membros, comunicação entre os envolvidos e tomada rápida de decisões. No último dia da *sprint*, a equipe realiza duas reuniões: a reunião de *sprint review* (revisão da *sprint*) e a reunião de *sprint retrospective* (retrospectiva da *sprint*). Na reunião de *sprint review*, a equipe apresenta o que foi feito ao *Product Owner* e a todas as partes interessadas. Na reunião de *sprint retrospective*, a equipe realiza a inspeção do que funcionou bem e do que pode melhorar para o próximo ciclo de desenvolvimento, e planeja a adaptação, ou seja, como irá realizar essas melhorias. Ao final da *sprint*, a equipe terá um incremento no produto que pode ou não ser entregue ao cliente naquele momento, mas que por regra deve constituir funcionalidades completas que adicionem valor de negócio para o cliente. O *Product Owner* determina, de acordo com a necessidade do cliente, quando é o momento mais adequado para realizar a entrega para o cliente do produto com os incrementos até então desenvolvidos. Essa entrega é conhecida como *release*, e pode ou não ser planejada previamente, segundo Schwaber & Beedle (2002).

Uma representação do ciclo do *Scrum* pode ser vista na Figura 4.

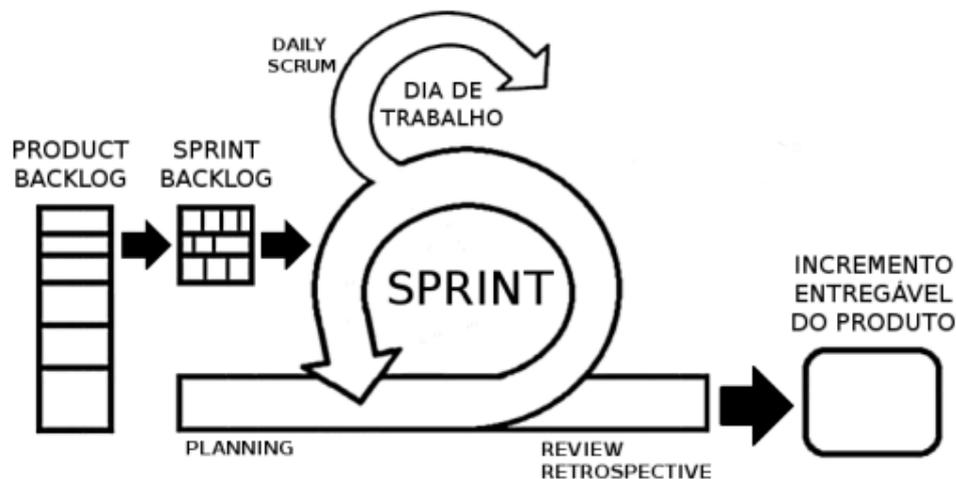


Figura 4 - O Ciclo do Scrum  
Fonte: Adaptado de Schwaber & Beedle (2002)

### 2.5.1.2 Os papéis do scrum

O *Scrum* define três papéis: o *Scrum Master*, o *Product Owner* e a equipe de desenvolvimento, ou os desenvolvedores de software. Uma das principais atribuições

do *Scrum Master* é a de garantir a adesão da equipe de desenvolvimento e do *Product Owner* aos valores do *Scrum*, a suas práticas e as suas regras. Ele deve trabalhar junto à equipe e à organização para ajudá-los na adoção do *Scrum*, ensinando a equipe a usar o autogerenciamento e a multidisciplinaridade para ser mais produtivo e ter mais qualidade.

O *Scrum Master* é também responsável por remover os eventuais bloqueios ou impedimentos (ou tratar e acompanhar sua remoção), apontados pela equipe, que estejam atrapalhando seu trabalho, como interferências externas, tentativas de mudanças de escopo durante a iteração de desenvolvimento, problemas de infraestrutura etc. O *Scrum Master*, no entanto, não gerencia a equipe, pois ela é autogerenciada, como exposto por Schwaber & Beedle (2002).

O *Product Owner*, é o responsável por assegurar que o trabalho da equipe de desenvolvimento proporcione retorno ao investimento feito pelo cliente. Para realizar esse trabalho, ele gerencia a lista de requisitos do produto, chamada de *product backlog*, através do contato frequente com os clientes do projeto. Nessa atividade de gerência estão incluídos os trabalhos de levantamento desses requisitos, a sua inserção e priorização na lista, a atualização dos requisitos à medida que mudam, a remoção de requisitos não mais necessários e que não mais serão desenvolvidos, o detalhamento progressivo dos requisitos à medida que ganham prioridade e a garantia de sua visibilidade a todos os envolvidos no projeto. O *Product Owner* é também responsável por estabelecer a visão do produto junto ao cliente, além de mantê-la e comunicá-la às partes interessadas. A gerência das entregas (*releases*) do produto também é uma atribuição do *Product Owner*. Uma observação importante é que esse papel é exercido por uma pessoa, não por um comitê. No entanto, essa pessoa pode ser aconselhada e influenciada por outras pessoas envolvidas no projeto, mas suas decisões devem ser respeitadas, de forma que ninguém além dela pode mudar as prioridades estabelecidas por ela. Outra observação importante é que o *Product Owner*, dependendo de questões muito específicas de cada organização e projeto, pode ser proveniente tanto da organização fornecedora quanto do cliente do projeto, de acordo com Schwaber & Beedle (2002).

A equipe de desenvolvimento ideal deve ter entre cinco e nove membros. Essa equipe é responsável por gerenciar o trabalho interno da iteração de desenvolvimento (*sprint*), de forma a transformar a lista de requisitos (*product backlog*) em incrementos de

funcionalidades que podem ser entregues ao cliente ao final de cada iteração. A equipe de desenvolvimento deve ser multidisciplinar, ou seja, todo conhecimento necessário para gerar os incrementos deve estar dentro da equipe e, assim, seus membros idealmente devem ocupar diversos papéis técnicos e realizar diferentes tipos de atividades. Deve ser também auto gerenciada, de forma que ninguém deve dizer à equipe como ela deve transformar o *product backlog* em incrementos entregáveis para o cliente. Uma vez identificados obstáculos ou impedimentos ao trabalho da equipe, eles devem ser imediatamente informados ao *Scrum Master* que deve tomar as ações necessárias para removê-los, quando possível, sugerem Schwaber & Beedle (2002); Schwaber (2004).

#### 2.5.1.3 Os artefatos do scrum

O *Scrum* define o uso de cinco artefatos: o *product backlog*, o *release burndown*, o *sprint backlog*, o *sprint burndown* e a definição de pronto ou *Definition of Done*. O *product backlog* é uma lista incompleta e dinâmica dos requisitos do produto, ordenados pela prioridade que eles têm em serem colocados em desenvolvimento. Assim, os itens do alto da lista, que serão implementados primeiro, devem ser de menor granularidade e devem possuir mais detalhes, como uma melhor descrição e podem possuir algum tipo de estimativa de esforço de desenvolvimento. Essa prioridade é dada, para um determinado momento, pelo valor que irá gerar para o cliente ou pelo risco que a sua não implementação representa. O *product backlog* nunca está completo, pois deve evoluir à medida que o produto e seu ambiente evoluem. Essa lista pode conter funcionalidades, correções de problemas, tecnologias e melhorias que constituem a mudança que deverá ser implementada no produto. Conforme descrito anteriormente, o *Product Owner* é o responsável por atualizar, priorizar e dar visibilidade ao *product backlog*. O *release burndown* é um gráfico utilizado para acompanhar o progresso do desenvolvimento em direção a uma entrega (*release*). Seu uso não é obrigatório, mas caso seja utilizado, deve ser criado quando se faz o planejamento da entrega, e deve ser atualizado ao final de cada *sprint*. Esse gráfico mostra no eixo vertical a soma dos esforços estimados restantes dos itens (que, a princípio, constarão na entrega conforme o planejado) e mostra no eixo horizontal o tempo, representado por cada *sprint* planejada para aquela entrega, segundo Schwaber & Beedle (2002); Schwaber (2004).

O *sprint backlog* é composto por uma lista dos itens que serão desenvolvidos durante a *sprint*, suas tarefas correspondentes, o andamento do desenvolvimento dessas tarefas e suas estimativas de tempo de desenvolvimento, (se a equipe optar por utilizar estimativas). Seus itens são selecionados do *product backlog* na reunião de planejamento da *sprint*, onde cada um dos itens selecionados é quebrado em tarefas. O *sprint backlog* é modificado ao longo da *sprint*, de forma que suas estimativas, quando utilizadas, são atualizadas e tarefas podem surgir para os itens já existentes. A equipe de desenvolvimento é responsável pelo *sprint backlog* e deve garantir que ele tenha alta visibilidade e que se mantenha atualizado. Uma representação comum do *sprint backlog* é um quadro afixado na parede da sala de trabalho dividido em colunas, contendo cartões ou papéis adesivos que indicam, dependendo de sua coluna, cada item que faz parte da *sprint* e as cada uma das tarefas correspondente a cada um dos itens (alinhadas horizontalmente a eles), divididas geralmente em três colunas que indicam o progresso do desenvolvimento da tarefa: “a fazer”, “em desenvolvimento” (ou “em progresso”) e “pronto”. O *sprint burndown* é um gráfico utilizado para acompanhar o progresso do desenvolvimento em direção ao final de uma *sprint*. Esse gráfico é criado na reunião de planejamento da *sprint* e deve ser atualizado diariamente. Ele mostra a soma dos tempos estimados restantes (ou outra unidade de medida de trabalho restante em uso) das tarefas que fazem parte do *sprint backlog* pelo tempo, representado pelos dias de desenvolvimento da *sprint*, segundo Schwaber & Beedle (2002); Schwaber (2004).

A definição de pronto ou *Definition of Done* significa somente que, quando alguém diz que alguma parte do trabalho está pronta, a equipe, o *Product Owner* e todas as partes interessadas devem entender o que isso significa. Por exemplo, um item estar pronto pode significar que esteja desenvolvido para a equipe, mas para o *Product Owner* pode significar que esteja desenvolvido, testado e documentado. Para resolver essa questão, uma definição de pronto em comum deve ser estabelecida, e essa definição deve ser a mais rigorosa possível, afirmam Schwaber & Beedle (2002); Schwaber (2004).

#### 2.5.1.4 Os eventos do scrum

Os autores Schwaber & Beedle (2002); Schwaber (2004). também descrevem como os eventos do *Scrum* ocorrem. Os eventos do *Scrum*, que possuem sua duração

determinada, são a reunião de *release planning* (planejamento da entrega), a *sprint*, a reunião de *sprint planning* (planejamento da *sprint*), a reunião de *sprint review* (revisão da *sprint*), a reunião de *sprint retrospective* (retrospectiva da *sprint*) e o *daily scrum* (*scrum* diário).

A reunião de *release planning* serve para estabelecer a meta da entrega, os itens do *product backlog* que deverão ser desenvolvidos e entregues e a data da entrega. Esse conjunto de itens, no entanto, poderá ser modificado a cada *sprint*. O progresso em direção a essa entrega planejada deve ser inspecionado a cada *sprint*, com o auxílio do *release burndown*. A reunião de *release planning* não é obrigatória, mas quando é feita, geralmente, planeja-se a entrega para somente quando incrementos no produto suficientes tiverem sido feitos para gerar valor para o cliente.

A *sprint* é a iteração ou ciclo de desenvolvimento, que contém a reunião de planejamento, o trabalho de desenvolvimento propriamente dito, a reunião de revisão e a reunião de retrospectiva. Cada *sprint* tem uma meta bem definida, que determina o que é imprescindível que esteja pronto ao seu final. As *sprints* têm duração fixa de duas semanas a um mês e ocorrem uma atrás da outra, sempre com a mesma duração. A duração da *sprint* é determinada de forma que seu horizonte seja curto o suficiente para que mudanças necessárias não alterem a meta que foi estabelecida para aquela *sprint*. Essa duração varia conforme o tipo de projeto e pode mudar dentro do mesmo projeto, de forma não frequente, conforme as condições do projeto e do ambiente mudam. Cada *sprint* deve ter como saída um incremento do produto pronto, que pode ser entregue para o cliente se assim tiver sido planejado. Para garantir a integridade da *sprint*, o *Scrum Master* tem o papel de assegurar que não seja feita nenhuma mudança que possa afetar a meta estabelecida para a *sprint*. Mas, caso a meta da *sprint* perca o sentido em termos de negócios, a *sprint* pode ser cancelada por decisão do *Product Owner*. Cancelamentos de *sprint*, no entanto, devem ser raros.

A reunião de *sprint planning* (planejamento da *sprint*) ocorre no primeiro dia e é dividida em duas partes. Na primeira parte, o *Product Owner* e a equipe de desenvolvimento se reúnem para a escolha dos itens mais prioritários do *product backlog* a serem implementados. O *Product Owner* define também a meta da *sprint*. Na segunda parte, a equipe de desenvolvimento quebra os itens escolhidos em tarefas e, opcionalmente, as estima em horas. Na reunião de *sprint review* (revisão da *sprint*), a equipe mostra, informalmente, tudo o que foi desenvolvido durante a iteração

às partes interessadas. A equipe demonstra o que fez e o *Product Owner* faz suas observações, que deverão servir de entrada para as *sprints* seguintes. O *Product Owner*, a partir do que foi e do que não foi feito na *sprint*, estabelece se a equipe conseguiu ou não atingir a meta da *sprint*. A reunião de *sprint retrospective* (retrospectiva da *sprint*) serve para que a equipe inspecione a *sprint* com relação a seu processo de desenvolvimento e planeje sua adaptação, dentro das regras do *Scrum*, de forma a torná-lo mais eficaz para a próxima *sprint*. Nessa reunião, os membros da equipe identificam e priorizam o que consideram que foi bom e o que consideram que pode melhorar, traçando planos de ação para realizar essas melhorias.

A *daily scrum* (*scrum* diário) é uma reunião de, no máximo, quinze minutos, realizada todo dia preferencialmente no mesmo local e à mesma hora. Essa reunião serve para proporcionar visibilidade do trabalho entre os membros da equipe, promover sua comunicação, levantar quais obstáculos atrapalham o desenvolvimento desde a última *daily scrum* e servir de oportunidade para decisões rápidas com relação ao progresso da *sprint*. Nessa reunião, cada membro da equipe detalha o que ele completou desde a última *daily scrum*, o que ele pretende fazer até a próxima *daily scrum* e quais obstáculos estiveram em seu caminho, determinam Schwaber & Beedle (2002); Schwaber (2004).

## **2.6 Confronto entre as metodologias**

Nerur *et al.* (2005) relacionaram as diferenças fundamentais entre as metodologias tradicionais e as metodologias ágeis, com o objetivo de mapear potenciais desafios a serem enfrentados pelas empresas que destinam-se a migrar da primeira para a segunda.

No Quadro 5, condensa-se as principais diferenças entre os processos metodológicos tradicionais e as metodologias ágeis, fomentando a percepção de quão opostas são as proposições e os princípios de cada uma em seus aspectos fundamentais.

**Quadro 5:**

## Estrutura comparativa entre as metodologias tradicionais e ágeis

	<b>Tradicional</b>	<b>Ágil</b>
Pressupostos básicos	Os requisitos do Sistema são totalmente especificados, previsíveis, e podem ser construídos através de um planejamento meticuloso e de longo prazo e visibilidade	Sistemas adaptativos de alta qualidade podem ser desenvolvidos por equipes pequenas usando princípios de melhoria e testes contínuos do design, baseado no rápido feedback e entendimento das necessidades de mudança.
Controle	Abrangem "mudanças incrementais, técnicas de gestão por processos e de design", explora recursos existentes e atende às pressões de desempenho de curto prazo.	Abrange mudanças radicais, criação e inovação, além de estar voltada para adaptações de desempenho de longo prazo.
Estilo Gerencial	Associado com "estruturas mecanicistas, sistemas fortemente conectados, caminhos contínuos, rotinas, controle, burocracia, mercados e tecnologias estáveis" (p. 481). Objetivo primário na melhoria de mercados e produtos existentes.	Associada com as "estruturas orgânicas, sistema de baixa conexão, novos caminhos, improvisação, autonomia, caos, mercados e tecnologias emergentes" (p. 483). Foca em novos mercados e produtos.
Gestão de Conhecimento	Abrange a utilização e o compartilhamento do conhecimento existente entre parceiros.	Abrange a criação de novos conhecimentos entre os parceiros envolvidos.
Atribuição de Papéis	Foco no Processo	Foco nas Pessoas
Comunicação	Comando e Controle	Liderança e Colaboração
Papel do Cliente	Explícito	Tácito
Ciclo do Projeto	Individual - favorece a especialização	Times auto-organizáveis - encorajando intercâmbio de papéis
Modelo de Desenvolvimento	Formal	Informal

Estrutura Organizacional Ideal	Importante	Crítico
Tecnologias de Desenvolvimento	Guiado por tarefas e atividades	Guiado por funcionalidades do produto

Fonte: Baseado em Nerur *et al* (2005)

O quadro acima apresenta uma comparação entre pressupostos básicos, controle, estilo gerencial, gestão de conhecimento, atribuição de papéis, comunicação, papel do cliente, ciclo do projeto, modelo de desenvolvimento e tecnologias de desenvolvimento de duas abordagens distintas de desenvolvimento de software: uma baseada em processos metodológicos tradicionais e outra em metodologias ágeis. As diferenças entre as abordagens são significativas em vários aspectos, incluindo o foco nas pessoas e na colaboração, a utilização de times auto-organizáveis, a valorização da criação de novos conhecimentos, a flexibilidade em relação a mudanças e a adoção de tecnologias guiadas pelas funcionalidades do produto.

As proposições e princípios presentes nas metodologias tradicionais são resultados de décadas de busca por processos repetíveis e otimizados como fonte de eficiência. Essa busca levou as organizações a uma estabilidade nos processos de desenvolvimento, nos elementos organizacionais e nas práticas gerenciais que hoje estão penetradas na cultura dos desenvolvedores tradicionais.

As diferenças conceituais, técnicas e culturais entre ambas as linhas metodológicas de desenvolvimento acabam por repartir a comunidade de desenvolvedores de software e colocando em campos opostos os “tradicionalistas” e os “agilistas”, cada qual proclamando a superioridade de sua respectiva metodologia, segundo Nerur *et al.* (2005)

Ainda o autor complementa que, aqueles que adotaram as práticas ágeis há mais tempo acreditam que seu uso afeta positivamente sua taxa de sucesso na entrega de projetos. Em contrapartida, os seguidores de metodologias mais tradicionais classificam os métodos ágeis como caóticos e carentes de um rigor procedural formal, só presente em seus próprios métodos.

Dadas tais diferenças e potenciais conflitos pode-se afirmar que as organizações desenvolvedoras de software que praticam metodologias tradicionais e pretendem adotar metodologias ágeis, ampliando assim seu portfólio de metodologias e criando

a capacidade ambidestra, precisam empreender cuidados especiais e analisar de forma clara seus objetivos, devendo se preparar para a necessidade de reconfigurar seu quadro de profissionais, seu estilo gerencial e sua adaptabilidade organizacional, entre as dualidades *exploitation* e *exploration*, já examinadas anteriormente, a fim de obter sucesso na adoção da nova prática.

Lycett *et al.* (2003) colocam que a abordagem ágil é fortemente a favor da comunicação humana e colaboração ao invés de atividades repetidas e definidas como mecanismo para desenvolver softwares de qualidade. Concluem que, na prática, a dificuldade que muitas organizações enfrentam com a abordagem ágil reside na mudança cultural necessária para convencer a gerência dos benefícios e a viabilidade no contexto de pressão do mercado global e práticas regulatórias.

Diante de todas essas características do desenvolvimento ágil, Nielsen & Mcmunn (2005) enfatizam a importância de implantar esses conceitos de projeto em projeto, diluindo o impacto e os riscos das mudanças ao longo do tempo. Dessa forma, os usuários também são introduzidos gradualmente e podem perceber, ao longo do tempo, a importância da participação ativa no projeto.

Nessa linha de introduzir os conceitos ágeis gradualmente, Mikulenas, Butleris & Nemuraite (2011) argumentam que passados somente 8 anos desde a publicação do Manifesto Ágil, o conceito de desenvolvimento ágil ganhou forte destaque no campo do desenvolvimento de sistemas de informação. Mas muitas organizações receiam mudar suas metodologias de trabalho de forma drástica e ter que assumir riscos pela escolha, ainda mais se considerar o último relatório CHAOS publicado em 2011 pela Standish Group, no qual é apresentado que somente 32% dos projetos de software podem ser classificados como sendo de sucesso, por terem atingido a meta de custo e tempo. As organizações geralmente não preferem reconstruir seus métodos e processos de imediato, em sua maioria optam por fazer por projeto ou mesmo estender o método existente para agregar algumas partes do método ágil. O problema é que os métodos ágeis são apresentados, atualmente, como uma solução monolítica sem um roadmap formal de como customizar e configurá-lo para uma adaptação parcial, fora que há uma variedade de métodos ágeis para escolher.

Os autores propõem uma visão do problema, generalizado em seis subproblemas, de adaptar parcialmente os métodos ágeis, conforme ilustrado na Figura 5.



Figura 5 - Decomposição do problema de adaptação parcial do método ágil  
 Fonte: Baseado em Nerur *et al* (2005)

O estudo de Nerur *et al.* (2005) identificou que a adaptação parcial de métodos ágeis pode levar a problemas de comunicação, coordenação e integração. Isso ocorre porque os processos ágeis são altamente interdependentes e a adaptação parcial pode afetar a interação entre as equipes, a colaboração e a compreensão mútua dos processos. Além disso, a falta de uma estratégia clara de adaptação pode dificultar a integração de diferentes partes do processo e levar a inconsistências e conflitos. Portanto, é importante considerar cuidadosamente os impactos potenciais antes de fazer adaptações parciais em métodos ágeis.

No Quadro 6, Nerur *et al.* (2005) expõem uma breve explicação sobre os subproblemas derivados da adaptação parcial do método ágil.

**Quadro 6:**  
 Subproblemas da adaptação parcial do método ágil

Subproblema	Descrição
Como avaliar a compatibilidade do método ágil?	Métodos ágeis são frequentemente utilizados como alternativa ao modelo tradicional. Há restrições tanto no nível do

	projeto quanto organizacional que devem ser ajustadas para sua adoção por TI.
Como preparar os métodos ágeis para adaptação parcial?	A demanda atual é estender o método existente na empresa por adicionar algumas práticas ágeis, há necessidade de identificar quais práticas serão utilizadas no contexto atual.
Como iniciar o processo? Qual é o ponto de partida?	A atualização de um método é uma atividade que gera alto custo para TI tanto pelo risco da implantação quanto de treinamento da equipe que associado a pressão de mercado direciona a buscar o momento mais propício para início da atualização.
Onde é a fronteira entre a possibilidade e o desejo?	Dependendo do método existente na empresa pode haver espaço para agregar muitas práticas ágeis, podendo elevar e muito o esforço de adaptação, sendo assim, deve-se balancear os elementos que são possíveis ou desejáveis de serem utilizados.
Como é tomada a decisão de quando configurar ou selecionar?	Envolvem questões de como selecionar e construir fragmentos de um método ágil concreto.
Como é integrá-lo a um método customizado existente?	Depois de estender o método existente com práticas ágeis deve-se tomar a decisão de implementá-lo.

Fonte: Mikulenas, Butleris & Nemuraite (2011)

Conforme demonstrado no quadro acima, Mikulenas, Butleris & Nemuraite (2011) sugerem que a adaptação parcial de métodos ágeis pode gerar subproblemas que precisam ser considerados. Um deles é a avaliação da compatibilidade do método ágil com a cultura e a estrutura da organização, para garantir que a adaptação não comprometa o funcionamento geral. Outro subproblema é a preparação adequada do método ágil para a adaptação parcial, que envolve definir claramente os processos a serem adaptados e quaisquer limitações a serem impostas. Além disso, é importante determinar o ponto de partida da adaptação e onde está a fronteira entre a possibilidade e o desejo de adaptação parcial. A decisão de quando configurar ou selecionar o método ágil também é crucial e deve levar em conta as necessidades específicas do projeto. Finalmente, a integração do método ágil adaptado a um

método customizado existente deve ser cuidadosamente planejada e executada para garantir a harmonia entre os processos ágeis e os existentes.

Mikulenas, Butleris & Nemuraite (2011) concluem que não há um consenso universal de que os métodos ágeis são facilmente adaptáveis, há um conjunto de características de ambiente tanto da organização quanto do projeto que facilitam sua utilização, o qual os autores denominam como requisitos de agilidade descritos como constituído de dois parâmetros: tipo de medida (quantitativa ou qualitativa) e tipo de conteúdo (técnico, social, negócio, psicológico, etc.).

Já o estudo de Mafakheriab *et al.* (2008) propõem indicadores de agilidade para identificar as empresas que adotam o desenvolvimento ágil que seriam compostos pelos parâmetros apresentados no Quadro 7.

**Quadro 7:**  
Parâmetros que afetam a agilidade

N.	Parâmetro	Objetivo
1	Dinamismo	Habilidade de mudar os requisitos Entregas frequentes de softwares funcionais
2	Tamanho da Equipe	Ter um time pequeno
3	Comunicação	Ter proximidade do cliente Redução da documentação
4	Teste	Ser capaz de testar os resultados mais frequentemente
5	Perfil e conhecimento dos desenvolvedores	Não ter nenhum pessoal no nível 1B (aprendiz) Ter pelo menos 30% no nível 2 e 3 (pleno e sênior)
6	Cultura	Ter alto grau de liberdade para qualquer um envolvido na equipe

Fonte: Modelo de Mafakheriab *et al.* (2008)

O modelo proposto por Mafakheriab *et al.* (2006), através do quadro acima, apresenta uma visão abrangente dos parâmetros que afetam a agilidade em um ambiente de desenvolvimento de software. Entre esses parâmetros, podemos destacar o

dinamismo do ambiente, que se refere à capacidade de resposta da equipe às mudanças de requisitos. Além disso, o tamanho da equipe pode influenciar a agilidade, uma vez que equipes menores tendem a ser mais ágeis que equipes grandes. A comunicação entre os membros da equipe também é um fator importante, uma vez que uma comunicação eficiente e eficaz pode ajudar a garantir a colaboração e o alinhamento dos esforços. A capacidade de teste é outro parâmetro importante, pois a agilidade requer ciclos de feedback rápidos para permitir mudanças e melhorias contínuas. O perfil e conhecimento dos desenvolvedores também podem afetar a agilidade, pois profissionais com habilidades e experiências diversas podem contribuir para o desenvolvimento ágil. Por fim, a cultura da organização também é um parâmetro importante, pois a adoção da agilidade requer uma cultura organizacional que valorize a colaboração, a experimentação e a melhoria contínua.

Com base na experiência das organizações estudadas, Lindvall *et al.* (2004) acreditam que as práticas ágeis cobrem as necessidades das grandes organizações e especialmente das pequenas. O desafio não é aplicar práticas ágeis ao projeto, mas sim como difundi-las pela organização, evitando duplo trabalho da equipe causado pelo conflito das práticas ágeis e tradicionais.

## **2.7 Modelos de maturidades**

O desenvolvimento de modelos de maturidade é visto como uma questão de pesquisa científica por alguns pesquisadores do campo de sistemas de informação Becker, Knackstedt, Poppelbub (2009); Mettler (2009); Rohner (2009)

A aplicação dos modelos de maturidade teve início nos anos 90, em particular o *Capability Maturity Model* (CMM) proposto pelo *Software Engineering Institute* segundo, Fincher & Ginger (1997). O uso de modelos de maturidade foi estabelecido no campo de operações, sendo relativamente novo para gerenciamento de projeto, emergindo nos últimos vinte anos, como menciona Andersen (2003).

A origem destes modelos se deu no domínio do gerenciamento da qualidade na gestão de operações, com foco na identificação, documentação, controle e otimização dos processos de produção, sendo expandido para a gestão de projetos de negócio e software e mais recente, para a gestão de projetos genérica, afirma Pinto (2016). Nos

projetos de software, a maturidade foi expandida além da visão unicamente em processos para sistemas completos, segundo Cleland & Ireland (2007). E mais recentemente, modelos de maturidade integrados que avaliam processos, sistemas e fatores contextuais organizacionais surgiram, adicionam Zwikael, Levin & Rad (2008), com pontos fortes particulares na avaliação de práticas e medidas de desempenho nas organizações, fornecendo um meio de avaliação além do processo e perspectiva de sistema.

Com a aplicação do gerenciamento de projetos na multiplicidade de indústrias, a necessidade de avaliar sua maturidade parece ser crucial de acordo com a exigência de aumentar sua eficácia no desenvolvimento de novos produtos e aspectos associados de gerenciamento de projetos em manufatura, sugerem Allenbach & Huffman (2000). Pela constante necessidade de avaliação, o *Project Management Institute* (PMI) desenvolveu o *Organizational Project Management Maturity Model* (OPM3) - PMI (2013) e no Reino Unido, o *Office of Government Commerce*, o progenitor da metodologia de gerenciamento de projeto PRINCE2, desenvolveu o *Portfolio, Programme and Project Management Maturity Model (P3M3)* - OGC (2010).

O Quadro 8 resume alguns dos modelos de maturidade com suas estruturas e avaliações.

**Quadro 8:**  
Modelos de maturidade e suas estruturas

Modelos de Maturidade	Estrutura	Avaliação
Capability Maturity Model for Software (CMM®) ou CMMSW®	Modelo em estágios, com cinco níveis (inicial, repetível, definido, gerenciável e otimizado). Cada nível de maturidade é composto por áreas chave de processos (KPA's) que, quando desempenhados corretamente, identificam os requisitos necessários para alcançar cada nível de maturidade (PAULK; WEBER, 1996).	O CMM não aborda todos os problemas que são importantes para projetos bem sucedidos. Por exemplo, o CMM não aborda a especialização em domínios específicos de aplicativos, defende tecnologias de software específicas ou sugere como selecionar, contratar, motivar e reter pessoas competentes. (PAULK; WEBER, 1996).
Project Management Maturity Model (PMMM)	Um modelo baseado em sistema com cinco níveis de maturidade (linguagem comum, processos comuns, metodologia singular, benchmarking e melhoria contínua) (KERZNER, 2009)	Sua natureza geral e descritiva oferece pouca ajuda sobre como uma organização pode realmente subir na curva de maturidade; Exceto para uso em informações de benchmarking para melhorar a metodologia de GP (ALAMI; BOUKSOUR; BEIDOURI, 2015).

<p>Organization Project Management Maturity Model (OPM3)</p>	<p>Uma das principais características do OPM3® é a mensuração da maturidade, através da existência de um conjunto padronizado de Melhores Práticas (PMI, 2013). Uma das principais diferenças do OPM3® para outros modelos de maturidade é o fato de ser um modelo multidimensional, sendo possível determinar a maturidade de uma organização sob diferentes perspectivas (LIMA; ANSELMO, 2004).</p>	<p>O questionário de avaliação é considerado repetitivo e burocrático (SOLER, 2005) e a grande abrangência de seu escopo e estrutura extremamente complexa desencoraja seu uso (ZAGUIR; MARTINS, 2007). Apesar das críticas, é um dos mais completos modelos de maturidade em Gerenciamento de Projetos.</p>
<p>Portfolio, Programme and Project Management Maturity Model (P3M3).</p>	<p>O P3M3 possui perspectivas de gestão de Portfólio, Programa e Projeto, descritos em cinco níveis de maturidade, como o CMM (consciência do processo, processo repetitivo, processo definido, processo gerenciado e processo otimizado). O P3M3 foca em sete perspectivas (governança organizacional, controle de gestão, gestão de benefícios, gestão de riscos, gestão de partes interessadas, gestão financeira e gestão de recursos) (OGC, 2017).</p>	<p>O P3M3 oferece uma abordagem por estágios (identificando, analisando, gerenciando e revisando), onde a jornada de uma organização é suportada pela maturidade progressiva em portfólio, programa e projeto (OGC, 2017).</p>

Fonte: Modelo de OGC (2010)

De acordo com o quadro acima, os modelos de maturidade são ferramentas que ajudam a avaliar a capacidade e o nível de maturidade de uma organização em relação a um determinado conjunto de práticas e processos. Essas ferramentas fornecem um quadro de referência para identificar áreas de melhoria e estabelecer metas realistas para o desenvolvimento de processos. Os modelos de maturidade são comumente usados na indústria de software para avaliar a eficácia de processos de desenvolvimento e gerenciamento de projetos. A avaliação dos processos de uma organização em relação a esses modelos pode ajudar a melhorar a eficiência, qualidade e produtividade dos projetos.

Diferentes tipos de modelos de maturidade existem hoje, sendo muitos deles inspirados no modelo CMM sugerem Backlund, Chroneer & Sundqvist (2014).

A Figura 6 mostra a evolução dos modelos durante os anos e seus três paradigmas de maturidade: perspectiva de controle de processos, perspectiva sistêmica e perspectiva sistêmica e organizacional.

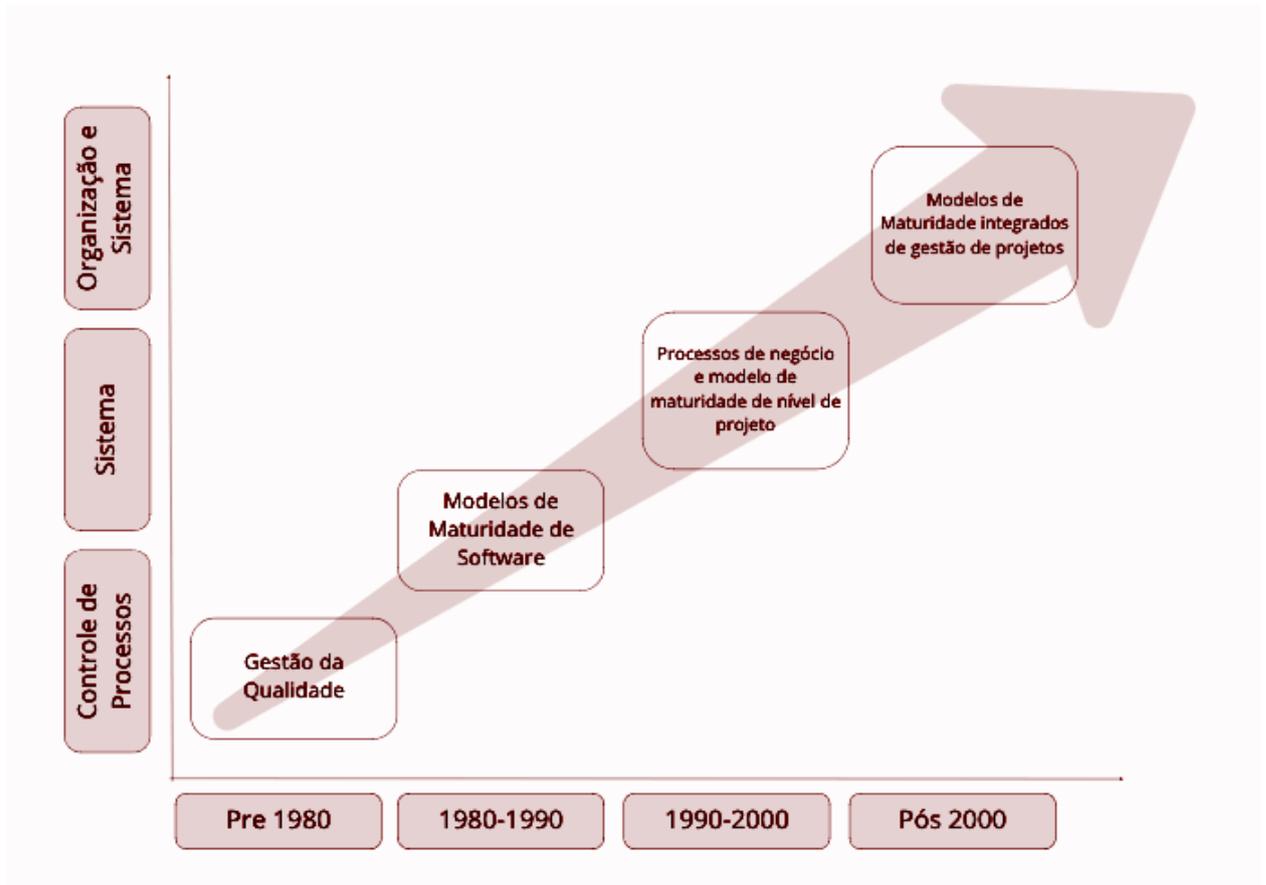


Figura 6 - Visão geral dos modelos de maturidade  
 Fonte: Elaborado pelo autor, adaptado de Pinto (2016)

De acordo com Pinto (2016), a evolução dos modelos de maturidade começou com a gestão da qualidade na década de 1950, que buscava a melhoria contínua dos processos para garantir a qualidade do produto final. Na década de 1980, surgiu o conceito de modelos de maturidade de software, que propunham um conjunto de práticas para melhorar a qualidade e a produtividade do desenvolvimento de software. Em seguida, os modelos de maturidade foram aplicados aos processos de negócio, com a criação de modelos específicos para a melhoria da gestão empresarial. Na década de 1990, surgiram os modelos de maturidade de nível de projeto, que buscavam aprimorar a gestão de projetos, identificando boas práticas e fornecendo um caminho para a melhoria contínua. Finalmente, na década de 2000, foram desenvolvidos modelos integrados de gestão de projetos, que visam à melhoria dos processos de gestão de projetos de forma holística, abordando aspectos relacionados à gestão de portfólio, riscos, qualidade, entre outros. Essa evolução demonstra uma preocupação crescente em melhorar a qualidade, a eficiência e a eficácia dos

processos, tanto em nível de projeto quanto organizacional, visando à melhoria contínua e ao sucesso nos negócios.

### **2.7.1 Design e construção dos modelos de maturidade**

De Bruin *et al.* (2005) e Becker, Knackstedt e Pöppelbuß (2009) sugerem procedimentos para o processo de concepção de modelos de maturidade. Enquanto De Bruin *et al.* (2005) propõem seis fases destinadas a orientar a criação de um modelo, Becker, Knackstedt e Pöppelbuß (2009) derivaram os requisitos e modelo do guia de design científico de Heyner *et al.* (2004), onde são distinguidas oito fases que fornecem um manual para o desenvolvimento teoricamente fundado de avaliação de modelos de maturidade. Contudo, Becker, Knackstedt e Pöppelbuß (2009) propõem um processo similar, enfatizando o uso de modelos de maturidade já existentes e um desenvolvimento interativo:

1. Escopo. A fase escopo define o foco e identifica as partes interessadas e o público-alvo. Ele determina o equilíbrio entre a realidade complexa e a simplicidade do modelo.
2. Design. A fase de concepção aborda os requisitos e descreve os principais conceitos de maturidade, estrutura de níveis, dimensões e subdimensões (meta-modelo).
3. População. Na fase de preenchimento, as características correspondentes são determinadas e a avaliação da maturidade é definida, o qual inclui a especificação dos instrumentos de avaliação.
4. Teste. O modelo construído é testado em seu completo conteúdo, avaliando também o rigor do escopo do modelo pretendido e a precisão do instrumento de avaliação para validade e confiabilidade.
5. Implantar. O modelo é implantado para os primeiros interessados e para uma comunidade independente.
6. Manter. Uma vez implementado, o modelo precisa ser mantido em uso e por um período de tempo suficiente, para garantir sua evolução.

Para projetar e preencher os modelos de maturidade, diferentes métodos de investigação exploratória e a combinação destes métodos são propostos. Métodos comumente mencionados são de análise de literatura, Delphi e estudos de caso e grupos focais. Os métodos quantitativos são menos utilizados para a construção de modelos de maturidade, uma vez que estes modelos requerem uma base teórica sólida, sugerem Becker, Knackstedt e Pöppelbuß (2009).

Os testes também feitos são principalmente qualitativos. Por fim, a escolha do método de investigação relevante é influenciada pelo escopo, partes interessadas, e direcionamento de público, de acordo com Becker, Knackstedt e Pöppelbuß (2009).

## **2.8 Contribuição do referencial teórico para a pesquisa**

O referencial teórico contribuiu para a elaboração da base da pesquisa, dando aderência às implicações, para elucidar as métricas e variáveis a serem correlacionadas na pesquisa, bem como contribuiu para estabelecer fronteiras para a coleta e análise dos dados. Os autores referenciados permitiram abstrair com clareza os constructos e os elementos históricos do surgimento da agilidade como se conhece hoje, além de direcionar ao entendimento de questões ligadas à gestão ágil aplicada à organização desenvolvedora de softwares. Compreende-se, desse modo, as abordagens da ambidestria organizacional enquanto simultaneamente impactadas pelas dualidades estratégicas, a evolução das organizações de software na adoção de metodologias ágeis e a discussão de uma escala de maturidade em relação aos processos adotados (Quadro 9).

**Quadro 9:**  
Síntese do referencial teórico

<b>Conceitos</b>	<b>Principais Autores</b>
<b>Ambidestria Organizacional</b>	March (1991); He e Wong (2004); Hill e Birkinshaw (2014); Simons (2010);Tushman e O'Reilly (1996); Henri (2006); Mundy (2010); Gibson e Birkinshaw (2004) ; Porter (1986); Adler (1999); Jorgenses e Messner (2009) ; Vinekar, Slinkman e Nerur (2006); Severgnini, E., Galdamez, E., Vieira, V. (2019); Gupta, Smith & Shalley, (2006);Raisch & Birkinshaw (2008);Severgnini, Takahashi & Abib (2019); Benner e Tushman (2003);Zimmermann, Raisch e Birkinshaw (2015);Boehm e Turner (2004);
<b>Ambidestria Organizacional no setor de Software</b>	Napier, Mathiassen, & Robey(2011);Vinekar, Slinkman e Nerur (2006);Cordeiro (2014);Martini (2015); Severgnini, E., Galdamez, E., Vieira, V. (2019);Gibson e Birkinshaw (2004);O'Reilly e Tushman (2013)
<b>Engenharia de Software; Metodologias no Processo de Software</b>	Souza (2014);Pressman (2016);Cockburn (2002);Fowler (2003);BOEHM (2003);Fleury e Fleury (2007);Larman & Basili (2003) e Whiteley <i>et al.</i> (2021) ;Beck <i>et al.</i> (2001);Ghezzi & Cavallo (2020); Schwaber & Beedle (2002); Schwaber (2004);Larman (2003);Nerur et al (2005);Nielsen e McMunn (2005); Mikulenas, Butleris e Nemuraite (2011); Mafakheriab <i>et al.</i> (2008)
<b>Modelos de Maturidade</b>	De Bruin <i>et al.</i> (2005); Ahern, Clouse e Turner (2008);Garcia e Pacheco (2010);Sommerville(2011);Huang e Han (2006);SEI (2006)

Fonte: elaborada pelo autor.

### 3 Metodologia

Neste capítulo são apresentados os procedimentos metodológicos adotados para o desenvolvimento desta pesquisa. Segundo Kourganoff (1990), “Pesquisa é o conjunto de investigações, operações e trabalhos intelectuais ou práticos que tenham como objetivo a descoberta de novos conhecimentos, a invenção de novas técnicas e a exploração ou a criação de novas realidades”.

Mediante isto, esse capítulo concentra-se na descrição dos procedimentos metodológicos utilizados nesta pesquisa, proporcionando alcançar os objetivos propostos, bem como o modelo em si.

### **3.1 Caracterização da pesquisa**

De acordo com Gerhard e Silveira (2009), as pesquisas podem ser classificadas quanto à sua abordagem, à natureza, aos objetivos e aos procedimentos. Esta pesquisa, caracteriza-se por ser um estudo descritivo com abordagem qualitativa e aplicada a pesquisa de campo. Segundo Gil (2017, p. 26), “as pesquisas descritivas têm como objetivo a descrição das características de determinada população ou fenômeno. Podem ser elaboradas também com a finalidade de identificar possíveis relações entre variáveis”. Quanto à abordagem, foi realizada uma pesquisa qualitativa.

Minayo (2009) mostra que a pesquisa qualitativa responde a questões particulares em nível não quantificável, buscando informações no universo dos significados, motivos, aspirações, crenças, valores e atitudes, aprofundando o conhecimento das relações, dos processos e dos fenômenos.

O método adotado foi a pesquisa de campo, que de acordo com Gil (2012), trata-se de entender a diferença entre um indivíduo e outro, a partir da análise da interação entre as pessoas de um grupo ou comunidade, extraindo dados diretamente por meio da realidade dos indivíduos. Ainda, definiu-se a utilização desta abordagem por estar centrada na identificação das características de situações, eventos e organizações, segundo Llewellyn & Northcott (2007), com clareza lógica das decisões, aplicação dos métodos e registro cuidadoso do processo de coleta, organização e interpretação.

### **3.2 Unidade de observação**

Para a avaliação do modelo proposto nesta pesquisa, foram consultados especialistas na área de gestão de projetos e gestores de times de desenvolvimento, de empresas brasileiras do setor de software.

Embora a pesquisa não tenha fundamentação quantitativa, utilizou-se amostra por julgamento, onde, segundo Freitas (2018), o pesquisador faz a seleção da população em estudos que sejam mais acessíveis, disponíveis ou colaborativos para participar

do processo. É adotado neste trabalho as recomendações de Lynn (1986), que define um número mínimo de cinco e máximo de dez participantes.

A amostra populacional proporciona traduzir o problema da pesquisa, como sugere Malhorta (2005), e é formada por dois grupos: (a) especialistas em gestão de projetos, (b) gestores de equipes de desenvolvimento de software, certificados em um ou mais métodos ou boas práticas de gerenciamento de projetos. Isto possibilita, por exemplo, avaliar a diferença ou semelhanças das respostas entre os diferentes tipos de especialistas, aqueles familiarizados ou não com as metodologias propostas neste estudo.

Assim, foram selecionados cinco profissionais atuantes em empresas brasileiras desenvolvedoras de software, dentre eles especialistas em gestão de projetos de formação não específica, certificados em alguma metodologia ou boas práticas de gerenciamento de projetos.

Além disso, este trabalho é amparado por um extenso levantamento bibliográfico onde foi possível levantar informações, conceitos, teorias já publicadas em livros, artigos, teses e dissertações. É uma etapa fundamental, pois permite aprofundar o conhecimento sobre o tema em questão e embasar a argumentação com referências teóricas consistentes. Neste trabalho de dissertação, a pesquisa bibliográfica foi utilizada como base para a construção do modelo de maturidade proposto e discussão dos resultados obtidos, contribuindo para a construção de uma reflexão crítica e embasada sobre o objeto de estudo.

### **3.3 Procedimento para coleta de dados**

Esta pesquisa utilizou-se do processo de avaliação (*survey*), que consiste, basicamente, em um conjunto de questões padronizadas apresentadas a uma amostra populacional (Babbie, 2014). É caracterizado como um *survey* de exploração, pois funciona como um mecanismo exploratório, procurando não deixar que elementos críticos deixem de ser investigados, apontando novas oportunidades, e interseccional ou transversal, visto que a coleta de dados ocorreu em um único intervalo de tempo (Creswell, 2010; Babbie, 2014).

O survey foi auto-administrado, oferecido pela internet através de e-mail, por melhor atingir os especialistas, pelo seu baixo custo, pela sua velocidade de condução e ausência da interferência do entrevistador (Babbie, 2014). Então, para a coleta de dados, é apresentado no Apêndice B o questionário para a avaliação com os especialistas. Tal questionário é apresentado sempre da mesma forma e seguindo a mesma ordem, mantendo o padrão para todos. Os especialistas foram contactados previamente para definição de data e horário adequado para a aplicação do questionário.

Foi explicado cara cada um dos especialistas:

- Objetivo da avaliação;
- Os níveis de maturidade;
- Os objetos de medição (componentes do domínio);
- A estrutura do modelo de maturidade proposto;
- O *framework* proposto.

O estudo foi realizado em etapas não consecutivas e interdependentes, com a elaboração do instrumento para avaliação do modelo proposto, a análise do modelo proposto pelos profissionais, levando-se em consideração a aparência geral do instrumento, facilidade de entendimento, a viabilidade de aplicação, itens contemplados, a pertinência dos tópicos e sua avaliação. Os entrevistados responderam às perguntas, conforme o Quadro 10.

**Quadro 10:**

Requisitos avaliados para o modelo de maturidade.

Requisitos a serem analisados no modelo de maturidade		
1	Aparência	Itens estruturados e seguem uma sequência lógica
2	Entendimento	Explicitado de forma clara, simples e inequívoca
3	Viabilidade	Permite ação de avaliação clara e precisa
4	Itens contemplados	Cada item de identificação e avaliação é útil, não se confunde com os demais, permite resposta objetiva e atende ao objetivo proposto.

Fonte: Baseado em Figueira (2020)

Para medição das percepções demonstrados no quadro acima, os respondentes avaliaram a aparência geral, facilidade de entendimento, viabilidade para prática e os itens contemplados no modelo, atribuindo uma nota apropriada (1- insuficiente; 2- abaixo do esperado; 3- atende o esperado; 4- excede o esperado; 5- excelente) a cada item analisado.

A coleta de dados desta pesquisa de dissertação foi realizada no período de 12 de janeiro de 2023 a 19 de janeiro de 2023 através de um questionário online, respondido por 5 (cinco) pessoas, dentro do critério de inclusão, utilizando-se a ferramenta do *Google Forms*.

Lakatos e Marconi (2003) consideram que a habilidade de usar preceitos ou normas na obtenção dos propósitos científicos corresponde à parte prática de coleta de dados.

No desenvolvimento do questionário, o pesquisador necessita cuidadosamente considerar inúmeros aspectos, como os conceitos que quer medir, as variáveis e o tipo de relação entre as variáveis, segundo De Bruin *et al.* (2005). Sua qualidade pode ser afetada por inúmeros fatores como as palavras utilizadas, expressões, layouts das questões e sua sequência. Assim sendo, De Bruin *et al.* (2005), sugerem que a adaptação da ferramenta de autoavaliação é debruçada na revisão da literatura relevante.

### **3.4 Síntese da metodologia**

O Quadro 11 apresenta uma síntese metodológica adotada para a investigação do tema de estudo.

**Quadro 11:**  
Síntese da metodologia

<b>Objetivos específicos</b>	<b>Autores</b>	<b>Técnica de coleta de dados</b>	<b>Técnica de análise dos dados</b>
Identificar os componentes empresariais com impacto direto no desempenho das organizações relacionados às dualidades <i>exploitation-exploration</i> , e respectivas simultaneidades.	March (1991); He e Wong (2004); Hill e Birkinshaw (2014); Simons (2010); Tushman e O'Reilly (1996)	Levantamento bibliográfico	Análise de Conteúdo
Identificar as dimensões relacionadas a ambidestria metodológica, ao utilizar de técnicas e práticas da metodologia ágil <i>Scrum</i> , em paralelo ao uso de metodologias tradicionais no desenvolvimento de softwares;	Schwaber & Beedle (2002); Schwaber Nerur <i>et al.</i> (2005); Popadiuk (2012); Mafakheriab <i>et al.</i> (2008)	Levantamento bibliográfico	Análise de Conteúdo
Definir categorias de maturidade específicas, em escalas representativas, divididos em níveis e priorizadas de forma lógica.	Bruin <i>et al.</i> (2005); Axelos (2016);	Levantamento bibliográfico	Análise de Conteúdo
Propor modelo conceitual para nortear as organizações de software enquanto adotando metodologias ágeis. O modelo descreve os elementos conceituais e suas relações através dos princípios da Ambidestria Organizacional.	Nerur <i>et al.</i> (2005); De Bruin <i>et al.</i> (2005); Axelos (2016); Schwaber & Beedle (2002); Schwaber (2004); De Bruin <i>et al.</i> (2005); March (1991); Popadiuk (2012); Mafakheriab <i>et al.</i> (2008); Figueira (2020)	Levantamento bibliográfico	Análise de Conteúdo
Identificar a percepção de especialistas em gestão de projetos e gestores de equipes de desenvolvimento de software para avaliar a aderência e confiabilidade do modelo bem como a consistência com as práticas reais do mercado	De Bruin <i>et al.</i> (2005); March (1991); Popadiuk (2012); Figueira (2020)	Questionário Apêndice B	Análise de Conteúdo

Fonte: Elaborado pelo autor.

#### 4 Apresentação e Discussão dos Resultados

Este capítulo apresenta e discute os resultados da análise dos dados obtidos pelas entrevistas.

## 4.1 Apresentação dos resultados

Os resultados são apresentados primeiramente pelos componentes do modelo a partir da pesquisa bibliográfica e depois para o *framework* desenvolvido, como estrutura de avaliação e melhoria de práticas e processos.

### 4.1.1 Composição arquitetural do modelo

Foi utilizado como referência a representação por estágios das áreas chaves de processo de De Bruin *et al.* (2005), que é composto por seis fases: (i) Escopo, (ii) Design, (iii) Preenchimento, (iv) Teste e (v) Implementação (vi). As três últimas fases não fazem parte do escopo desta pesquisa, pois após o preenchimento do modelo, o mesmo é avaliado por especialistas da área conforme dito no item 3.3.

A Figura 7 demonstra as fases utilizadas.

#### *Fases de Desenvolvimento de modelos de maturidade*



#### *Fases utilizadas*



Figura 7 - Fases de maturidade propostos  
Fonte: Adaptado de De Bruin *et al.* (2005)

A primeira fase no desenvolvimento do modelo de maturidade desta pesquisa é determinar o escopo, que irá influenciar as demais fases subsequentes, devendo então definir o foco do modelo e determinar as partes interessadas. O foco refere-se a qual domínio o modelo deverá atender, seja ele específico ou genérico. Finalmente, determina-se os envolvidos que irão utilizar o modelo: academia, profissional, governo ou combinado.

O Quadro 12 reúne as características do escopo.

**Quadro 12:**

Características do escopo do modelo proposto

<b>Critério</b>	<b>Característica</b>			
<b>Foco do Modelo</b>	Domínio Específico		Genérico	
<b>Partes Interessadas</b>	Academia	Profissionais	Governo	Híbrido

Fonte: Adaptado de De Bruin *et al.* (2005)

A partir destas características, segundo o quadro acima, o modelo de maturidade proposto deve ser independente, amplamente usado por profissionais nas organizações e presente em diversas pesquisas acadêmicas. Sendo assim, é caracterizado com uma perspectiva combinada de múltiplas partes interessadas.

A segunda fase, design, é determinar a arquitetura do modelo incorporando as necessidades do público-alvo e como estas necessidades serão atendidas. Estas necessidades são refletidas no porquê da aplicação do modelo, como o modelo pode ser aplicado, variando pela estrutura organizacional, quem necessita estar envolvido na aplicação do modelo e o que pode ser alcançado com a aplicação. Para atingir essas necessidades, o modelo necessita ser apropriadamente equilibrado entre uma realidade frequentemente complexa e a simplicidade do modelo.

Um modelo muito simples pode não ter todas as informações necessárias para o público, enquanto complexo, pode gerar confusão e limitar seu uso, recomenda De Bruin *et al.* (2005).

O Quadro 13 exprime as principais decisões a serem tomadas.

**Quadro 13:**

Decisões para construção do modelo

<b>Critério</b>	<b>Característica</b>		
<b>Audiência</b>	Interno		Externo
	Executivos / Gestão		Audidores / Parceiros
<b>Método de Aplicação</b>	Autoavaliação	Assistido por terceiros	Praticante Certificado
<b>Driver de Aplicação</b>	Requerimento Interno	Requerimento Externo	Ambos

Respondente	Gestão	Staff	Parceiros
Aplicação	1 entidade/região	múltiplas entidades	múltiplas entidades / múltiplas regiões

Fonte: Adaptado de De Bruin *et al.* (2005)

Seguindo as diretrizes estabelecidas por De Bruin *et al.* (2005), evidenciado no quadro acima, o público-alvo são os clientes internos e externos, executivos e gestores, auditores e parceiros, ou seja, todos aqueles que possuem papéis envolvidos na gestão de portfólio, programa e projeto. O método de aplicação será utilizado o de autoavaliação, conforme mencionado no item 4.1.3. O *driver* de aplicação é advindo tanto de requerimento interno quanto externo, onde pode haver a necessidade de se analisar sua própria organização. O respondente é a gestão, fazendo parte um indivíduo com conhecimento da organização, um grupo de pessoas ou um facilitador interno ou externo, como recomenda Axelos (2016). O modelo tem múltipla aplicação pela sua abrangência. Há a possibilidade de a aplicação ser limitada para somente uma divisão da organização ou até mesmo para uma perspectiva de gestão, como o setor de desenvolvimento de software ou pesquisa e desenvolvimento de novos produtos, por exemplo, segundo Axelos (2016).

Para o preenchimento, De Bruin *et al.* (2005) argumenta que se deve identificar o que é necessário medir na avaliação de maturidade e como isso irá ser mensurado pelos especialistas. A identificação dos componentes é crítica para domínios complexos visto que isto permite um entendimento profundo da maturidade.

O uso de dimensões e domínios-chaves, como recomenda (Jeunon, 2014), viabiliza a operacionalização dos construtos em medidas quantitativas, ou seja, permitem transformar características abstratas ou qualitativas em variáveis mensuráveis, garantindo a validade e a confiabilidade do instrumento de coleta de dados pertinente ao modelo de maturidade em avaliação. Portanto, a definição e operacionalização dos domínios chaves, pelos quais a escala de dimensão será composta serão as seguintes: *Exploration*, *Exploitation*, Orientação Estratégica e Agilidade.

#### **4.1.2 Composição estrutural das dimensões**

O uso de dimensões para evidenciar os resultados de maturidade organizacional em empresas ambidestras é de extrema importância para uma avaliação completa e precisa. Ao utilizar essas dimensões em conjunto com a análise das práticas de conhecimento, é possível identificar os pontos fortes e fracos da empresa em relação à sua ambidestria e, assim, definir estratégias para promover uma gestão mais equilibrada e eficiente das atividades.

##### *4.1.2.1 Componentes dos domínios: exploration, exploitation e orientação estratégica*

A revisão da literatura é considerada suficiente para fornecer a teoria e identificação do necessário juntamente com modelos já testados, que reduzem a preocupação de saber se os componentes são corretos e funcionais, segundo De Bruin *et al.* (2005). Logo, além dos dados teóricos levantados, é utilizado como base o modelo de March (1991) e Popadiuk (2012) para os componentes do domínio *exploration* e *exploitation* no novo modelo proposto.

Há, entretanto, algumas diferenças entre gestão de processos relacionados a práticas de maturidade organizacional e as dualidades estratégicas *exploitation-exploration*. Enquanto os processos de maturidade cobrem uma ampla gama de questões e é mais estruturado quando comparado aos conflitos de equilíbrio das dualidades, suas atividades são também menos padronizadas e os resultados menos quantificáveis. Consequentemente, a maturidade vista em modelos conceituais deve ser julgada de múltiplas perspectivas, incluindo tecnologias, processos, colaboradores, para assim, efetuar uma avaliação holística, apontam, Kulkarni & Freeze (2004).

March (1991) e Popadiuk (2012) oferecem uma visão para melhorar a compreensão das variáveis que influenciam e se relacionam com as dimensões *exploration* e *exploitation*. O modelo proposto por Popadiuk (2012) é dividido em duas determinantes: *exploration* e *exploitation*.

Nesse sentido, tem-se estruturado as seguintes variáveis e dimensões, conforme o Quadro 14.

**Quadro 14:**  
Composição estrutural das dimensões

Dimensão	Formação da Dimensão	Principais autores pesquisados no estudo
Exploration	Práticas de conhecimento	March (1991) e Popadiuk (2012).
	Práticas Inovadoras	March (1991) e Popadiuk (2012).
Exploitation	Concorrência	March (1991) e Popadiuk (2012).
	Eficiência	March (1991) e Popadiuk (2012).
	Qualidade	March (1991) e Popadiuk (2012).
Orientação Estratégica		March (1991) e Popadiuk (2012).

Fonte: baseado em March (1991) e Popadiuk (2012)

Como explorado acima, a dimensão *exploration* está associada às Práticas do conhecimento e às Práticas inovadoras das organizações. Já o *exploitation* refere-se à associação da Concorrência, Orientação estratégica, Eficiência, Qualidade e Parcerias. No entanto, esta pesquisa adaptou e utilizou este modelo de Popadiuk (2012) para avaliar a orientação estratégica da empresa frente a geração de conhecimento, bem como sua utilização na indústria e no setor de tecnologia.

#### 4.1.2.1.1 Componentes da dimensão: *exploration*

Neste caminho, as dimensões do modelo de maturidade proposto neste trabalho contêm a processos orientados às práticas de conhecimento que devem ser alcançadas para melhorar os processos e, uma vez que estes processos são alcançados, a organização atinge um nível específico de maturidade.

O Quadro 15 exprime estes processos orientados a dualidade *exploration* com foco em práticas de conhecimento.

**Quadro 15:**  
Atividades características de *exploration* - Práticas de Conhecimento

Dimensões do modelo de maturidade proposto	<i>Exploration</i> - Práticas de Conhecimento
--------------------------------------------	-----------------------------------------------

<i>Exploration</i> - Práticas de conhecimento	Volume de novas ideias
	Utilização de novas fontes de conhecimento
	Conhecimento existente em base de dados
	Compartilhamento de conhecimento interno
	Processo individuais de aprendizagem
	Processo de aprendizagem coletiva
	Formação de capacidades de equipe
	Intensidade do desenvolvimento pessoal
	Valorização do conhecimento individual

Fonte: baseado em March (1991) e Popadiuk (2012)

Os tópicos do campo da estratégia e práticas de conhecimento, abordadas no quadro acima, descrevem aspectos da visão corporativa e definição de objetivos em relação à organização, analisando o comportamento da alta gestão em relação às tomadas de decisão acerca de novas ideias, na base de conhecimento e nos processos de aprendizagem. Estruturas de conhecimento e formas de aprendizagem descrevem aspectos da estrutura da base do conhecimento organizacional, cobrindo o critério de classificação para conhecimento e documentos. A área de processos, funções e organização descreve assuntos relacionados à estrutura organizacional e tarefas dos papéis individuais e coletivos. Foca nos aspectos procedurais no contexto de uma organização baseada em processos, objetivando descobrir como as práticas de conhecimento podem ser avaliadas nos processos de negócio March (1991).

Ainda, dentro da dimensão *exploration*, propostas por March (1991) e Popadiuk (2012), podem ser utilizadas como referência para a análise das práticas inovadoras em organizações. Segundo March (1991), as empresas podem adotar estratégias que buscam aprimorar suas capacidades e conhecimentos existentes que envolvem a busca por novas oportunidades e a experimentação de novas ideias. Já Popadiuk (2012) propõe a dualidade entre inovação incremental, que se baseia em melhorias

graduais de produtos ou processos existentes, e inovação radical, que envolve mudanças significativas e disruptivas. Nesse contexto, as práticas inovadoras podem ser entendidas como ações e estratégias adotadas pelas empresas para gerar valor e melhorar sua posição competitiva, seja por meio da exploração ou da exploração, e da inovação incremental ou radical. Para alcançar esses objetivos, as organizações precisam ser capazes de equilibrar essas dualidades estratégicas e implementar práticas inovadoras que se adequem às suas necessidades e objetivos.

Dessa forma, a adoção de práticas inovadoras pode ser vista como um processo dinâmico e complexo que envolve a gestão de dualidades estratégicas e a busca por soluções que permitam a organização se adaptar às mudanças do mercado e do ambiente competitivo. Portanto, para construir práticas inovadoras bem-sucedidas, é necessário que as empresas estejam atentas às dualidades estratégicas propostas por March (1991) e Popadiuk (2012). No Quadro 16, identificou-se as práticas inovadoras que permitem equilibrar esses elementos que agreguem valor à organização.

**Quadro 16:**  
Atividades características de *exploration* - Práticas Inovadoras

Dimensão	<i>Exploration</i> - Práticas inovadoras
<b><i>Exploration</i> - Práticas inovadoras</b>	Concentração de produtos ou processos novos
	Desenvolvimento de protótipos
	Taxa de inovação de produto
	Inovação de técnica de Marketing
	Abertura de novos canais de distribuição
	Concentração em inovações radicais nos produtos
	Concentração em inovações tecnológicas radicais
	Busca incessante por novos mercados
	Desenvolvimento de novos produtos e serviços

	Participação agressiva em novas alianças baseadas em tecnologia
--	-----------------------------------------------------------------

Fonte: baseado em March (1991) e Popadiuk (2012)

A dimensão práticas inovadoras, segundo o quadro acima, refere-se à busca incessante por novos mercados, produtos e serviços, através da aplicação de práticas inovadoras. Nessa dimensão, é comum a concentração em inovações radicais nos produtos, tecnologias ou técnicas de marketing. Empresas que buscam essa dimensão podem investir na concentração de produtos ou processos novos, desenvolvimento de protótipos, abertura de novos canais de distribuição e participação agressiva em novas alianças baseadas em tecnologia. Além disso, é importante medir a taxa de inovação do produto e estar constantemente focado em desenvolver novos produtos e serviços.

Segundo Popadiuk (2012), às práticas de concentração de produtos ou processos novos, desenvolvimento de protótipos e inovação de técnica de marketing têm como objetivo melhorar a qualidade e a eficiência dos produtos e processos existentes, bem como aumentar a atratividade da oferta para a organização.

Já a taxa de inovação de produto e a concentração em inovações radicais nos produtos buscam desenvolver novos produtos ou melhorar os existentes, oferecendo algo novo e diferenciado ao mercado.

Popadiuk (2012), ainda, reforça que a concentração em inovações tecnológicas radicais, por sua vez, tem como objetivo desenvolver e implementar novas tecnologias que possam oferecer vantagens competitivas à organização. A abertura de novos canais de distribuição, a busca incessante por novos mercados e a participação agressiva em novas alianças baseadas em tecnologia têm como objetivo expandir a atuação da empresa, buscando novas oportunidades de negócio e parcerias estratégicas.

Por fim, o desenvolvimento de novos produtos e serviços visa atender às necessidades e demandas do mercado, oferecendo soluções inovadoras e relevantes para os clientes.

#### 4.1.2.1.2 Componentes da dimensão: *exploitation*

A dualidade estratégica da exploração (*exploitation*), conforme proposta por March (1991) e Popadiuk (2016), é baseada na busca por aprimorar as capacidades e os conhecimentos existentes na organização. Em outras palavras, trata-se de maximizar a eficiência e eficácia operacional da organização, aprimorando continuamente processos, produtos e serviços. Essa abordagem pode ser vista como uma estratégia defensiva, com foco em manter e consolidar a posição da empresa no mercado, fortalecendo suas capacidades internas.

O Quadro 17 exprime estes processos orientados a dualidade *exploitation* com foco no componente concorrência.

#### **Quadro 17:**

Atividades características desdobradas na dimensão *exploitation* - Concorrência

<b>Dimensão</b>	<b><i>Exploitation</i> - Concorrência</b>
<b><i>Exploitation</i> - Concorrência</b>	Aparecimento de novos concorrentes
	Existência de produtos ou processos substitutos
	Concorrência no mercado local
	Concorrência no setor empresarial
	Existência de guerra de promoção no setor
	Concorrência abrange oferta facilmente

Fonte: Elaborado pelo autor, baseado em March (1991) e Popadiuk (2012)

Como visto acima, a dimensão *exploitation*, conforme proposta por March (1991) e Popadiuk (2012), pode ser aplicada em uma organização desenvolvedora de software por meio da mensuração das práticas que a caracterizam. Para mensurar o aparecimento de novos concorrentes, por exemplo, a organização pode monitorar o mercado em busca de empresas que atuem em áreas semelhantes e acompanhar seus produtos e serviços.

A existência de produtos ou processos substitutos pode ser mensurada por meio da análise dos produtos oferecidos pela concorrência e da avaliação de suas vantagens competitivas.

Da mesma forma, a concorrência no mercado local e no setor empresarial pode ser avaliada por meio da análise da participação de mercado da organização e de seus concorrentes, bem como das estratégias de marketing e vendas adotadas por eles. A existência natural da competição direta no setor, por exemplo, pode ser mensurada por meio da análise das campanhas publicitárias e promoções realizadas pelos concorrentes, enquanto a abrangência da concorrência pode ser avaliada pela facilidade de oferta de produtos e serviços similares ao da organização desenvolvedora de software.

Ao mensurar essas práticas, a organização pode obter informações valiosas sobre a concorrência e utilizar essas informações para aprimorar suas estratégias de negócios e manter sua competitividade no mercado. Dentro do arcabouço de *exploitation* proposto por March (1991) e Popadiuk (2012), a eficiência é um dos principais componentes que caracterizam essa dimensão.

A eficiência pode ser avaliada por meio da análise do uso de recursos, processos organizacionais, e a identificação de gargalos que possam estar afetando a produtividade da empresa.

No Quadro 18, relacionamos as práticas orientadas à eficiência, seguindo modelo de March (1991) e Popadiuk (2012).

**Quadro 18:**

Atividades características desdobradas na dimensão *exploitation* - Eficiência

<b><i>Exploitation</i> - Eficiência</b>	<b><i>Exploitation</i> - Eficiência</b>
	Criação de rotinas detalhadas
	Importância da eficiência

	Foco na realização das atividades
	Preocupações com ganhos de escala
	Mecanismo de controle organizacional
	Foco nos custos
	Foco inclinado para a produção

Fonte: Baseado em March (1991) e Popadiuk (2012)

Como evidenciado acima, para mensurar as práticas de eficiência em empresas desenvolvedoras de software, March (1991) e Popadiuk (2012), orientam que, para cada uma dessas práticas, é possível estabelecer um conjunto de indicadores que possam ser monitorados e avaliados ao longo do tempo, permitindo que a organização identifique oportunidades de melhoria e aprimoramento. Por exemplo, a criação de rotinas detalhadas pode ser mensurada por meio da avaliação da documentação de processos e procedimentos da empresa, enquanto a importância da eficiência pode ser medida pelo grau de ênfase dado à produtividade e à otimização dos recursos.

O foco na realização das atividades pode ser avaliado por meio de indicadores de desempenho individuais e coletivos, enquanto as preocupações com ganhos de escala podem ser mensuradas pelo grau de investimento em tecnologia e automação.

O mecanismo de controle organizacional pode ser avaliado pela eficácia dos processos de gestão e supervisão, enquanto o foco nos custos pode ser mensurado pelo controle e redução de despesas operacionais. Por fim, a inclinação para a produção pode ser avaliada por meio do cumprimento de metas e prazos estabelecidos, além do nível de satisfação do cliente com os produtos e serviços entregues.

Ainda, dentro dos limites da dimensão *exploitation*, o componente qualidade é uma prática importante para as empresas desenvolvedoras de software. March (1991) e Popadiuk (2012), propõe que essa prática visa maximizar a eficiência das atividades, garantindo um alto nível de qualidade do produto final entregue ao cliente. Para

mensurar essa prática em um modelo de maturidade em empresas desenvolvedoras de software, é necessário realizar uma avaliação constante dos processos e sistemas de qualidade, visando identificar pontos de melhoria e corrigir possíveis falhas. Além disso, é importante definir indicadores de qualidade, que possam ser mensurados e avaliados regularmente, a fim de garantir que o produto entregue ao cliente esteja dentro dos padrões de qualidade estabelecidos.

A busca constante pela qualidade pode ser um importante diferencial competitivo para as empresas desenvolvedoras de software, portanto, relacionamos as práticas de qualidade, no Quadro 19, seguindo o modelo de March (1991) e Popadiuk (2012).

**Quadro 19:**

Atividades características desdobradas na dimensão *exploitation* - Qualidade

<b><i>Exploitation</i> - Qualidade</b>	<b><i>Exploitation</i> - Qualidade</b>
	Equipe de testes dedicada
	Controle de qualidade estabelecido
	Controle de qualidade compartilhado
	Preocupação com a qualidade dos processos
	Preocupação com a qualidade dos produtos
	Foco na qualidade

Fonte: Elaborado pelo autor, baseado em March (1991) e Popadiuk (2012)

No quadro anterior, a dimensão *Exploitation* do modelo de maturidade trata da eficiência e da qualidade na execução das atividades da organização. Para alcançar esse nível de maturidade, a empresa deve estabelecer uma equipe de testes dedicada e um controle de qualidade compartilhado. É importante também que haja uma preocupação constante com a qualidade dos processos e dos produtos, com o objetivo de garantir que os produtos finais atendam aos padrões estabelecidos pela empresa. Além disso, é fundamental que haja um foco na qualidade, buscando

sempre aprimorar os processos e produtos, a fim de satisfazer as necessidades dos clientes.

#### 4.1.2.1.3 Componentes da dimensão: orientação estratégica

A teoria das dualidades estratégicas de March (1991) e Popadiuk (2012) nos fornece uma base para entender a dinâmica de inovação em organizações. Essa teoria se baseia na distinção entre duas estratégias opostas e complementares: *exploration* e *exploitation*.

No entanto, a compreensão das dualidades estratégicas não se limita a essas duas estratégias. Os autores propõem uma terceira dimensão, a orientação estratégica, que é importante para entendermos a posição da organização em relação às dualidades estratégicas. A orientação estratégica é composta por duas estruturas: a visão estratégica focada no presente e as estratégias focadas no curto prazo.

Neste sentido, este estudo buscará explorar a importância dessas estruturas para entender a relação entre as dualidades estratégicas e as práticas de inovação em organizações, como observado no Quadro 20.

#### **Quadro 20:**

Atividades características desdobradas na dimensão - Orientação Estratégica

<b>Dimensão</b>	<b>Orientação Estratégica</b>
<b>Orientação Estratégica</b>	Visão Estratégica focada no presente
	Estratégias focadas no curto prazo

Fonte: Baseado em March (1991) e Popadiuk (2012)

Como descrito no quadro acima, a orientação estratégica é um elemento-chave na definição e implementação de estratégias empresariais. É essencial que a visão estratégica da organização esteja focada tanto no presente quanto no futuro, considerando as mudanças e desafios que possam surgir ao longo do caminho. No

entanto, existem situações em que a visão estratégica pode ser limitada, com estratégias voltadas principalmente para o curto prazo. Nesse caso, a empresa pode enfrentar dificuldades em se adaptar às mudanças de mercado e competição, perdendo sua capacidade de se manter competitiva a longo prazo. Por isso, é importante que as empresas busquem sempre uma orientação estratégica que equilibre as necessidades presentes e futuras, com estratégias capazes de gerar valor a longo prazo.

Com relação às métricas de avaliação, é possível utilizar alguns indicadores para avaliar a visão estratégica focada no presente e a estratégia focada no curto prazo em uma organização.

**Quadro 21:**  
Métricas de avaliação e indicadores de visão estratégica

Visão Estratégica	Métricas
Focada no presente	Desempenho Financeiro atual da empresa
	Taxa de crescimento dos lucros
	Participação no mercado
	Satisfação do cliente
	Eficiência Operacional
Focada no curto prazo	Capacidade de resposta as mudanças do mercado
	Agilidade em lançar novos produtos e serviços
	Capacidade de adaptação as novas tecnologias
	Habilidade de aproveitar oportunidades a curto prazo

Fonte: baseado em March (1991) e Popadiuk (2012)

Como descrito no quadro acima, para March (1991), a visão estratégica focada no presente, pode-se avaliar o desempenho financeiro atual da empresa, a taxa de crescimento dos lucros, a participação de mercado, a satisfação do cliente e a eficiência operacional. Já para a estratégia focada no curto prazo, as métricas podem incluir a capacidade de resposta da empresa às mudanças do mercado, a agilidade em lançar novos produtos ou serviços, a capacidade de adaptação a novas tecnologias e a habilidade de aproveitar oportunidades de curto prazo. Popadiuk

(2012), no entanto, ressalta que a escolha das métricas a serem utilizadas deve ser alinhada aos objetivos estratégicos da organização e estar em consonância com o ambiente externo em que a empresa atua.

#### 4.1.2.2 Componentes dos domínios agilidade

A combinação das dimensões *exploration* e *exploitation* com a dimensão agilidade é fundamental para o sucesso de organizações desenvolvedoras de software que adotam metodologias ágeis, apontam Mafakheriub *et al.* (2008).

A dimensão *exploration*, caracterizada pela busca incessante por novos mercados e desenvolvimento de novos produtos e serviços, pode ser aprimorada com a adoção de rotinas e procedimentos metodológicos ágeis, uma vez que essas metodologias permitem um maior grau de flexibilidade para a identificação de novas oportunidades e experimentação de ideias.

Por outro lado, a dimensão *exploitation*, que se concentra na otimização dos processos existentes e no aumento da eficiência, também pode ser aprimorada com a adoção da agilidade organizacional, através da sua abordagem iterativa e incremental, que possibilita ajustes constantes no processo e melhoria contínua da eficiência. A dimensão agilidade, por sua vez, está intrinsecamente relacionada ao *Scrum*, como principal metodologia ágil na atualidade, pois essa metodologia valoriza a entrega contínua de valor ao cliente e a adaptabilidade aos requisitos em constante mudança, Mafakheriub *et al.* (2008).

Dessa forma, a adoção do *Scrum* possibilita que organizações desenvolvedoras de software combinem as dimensões *exploration* e *exploitation* com a dimensão agilidade, permitindo que elas se adaptem rapidamente às mudanças do mercado e mantenham uma posição de vantagem competitiva.

Nesse sentido, tem-se estruturado as seguintes variáveis e dimensões, conforme o Quadro 21.

**Quadro 21:**

## Formação da dimensão Agilidade

Dimensão	Formação da Dimensão	Principais autores pesquisados no estudo
Agilidade	Dinamismo	Mafakheriab <i>et al.</i> (2008)
	Equipe	Mafakheriab <i>et al.</i> (2008)
	Comunicação	Mafakheriab <i>et al.</i> (2008)
	Teste	Mafakheriab <i>et al.</i> (2008)
	Perfil e conhecimento dos desenvolvedores	Mafakheriab <i>et al.</i> (2008)
	Cultura	Mafakheriab <i>et al.</i> (2008)

Fonte: Baseado em Mafakheriab *et al.* (2008)

A dimensão agilidade é composta por seis elementos fundamentais, conforme descrito por Mafakheriab *et al.* (2008), no quadro acima. O primeiro elemento é o dinamismo, que envolve a capacidade da equipe de se adaptar rapidamente a mudanças e demandas do projeto. O segundo elemento é a equipe, que deve ser composta por membros engajados, colaborativos e capacitados para trabalhar em equipe. A comunicação é outro elemento crucial, permitindo que as informações sejam compartilhadas de forma clara e eficiente. Os testes são fundamentais para garantir a qualidade do produto final, e por isso constituem o quarto elemento. O perfil e o conhecimento dos desenvolvedores também são essenciais para a dimensão agilidade, sendo que os membros da equipe devem possuir as habilidades necessárias para a realização das tarefas propostas. Por fim, a cultura organizacional deve ser propícia à adoção de metodologias ágeis, com um ambiente favorável à experimentação, inovação e colaboração. Todos esses elementos, quando considerados em conjunto, contribuem para a construção de uma dimensão agilidade forte e bem-sucedida.

#### 4.1.2.2.1 Componentes da dimensão: dinamismo

Os autores, Schwaber & Beedle (2002), ressaltam que o dinamismo é um componente essencial da metodologia *Scrum*. Ele se refere à capacidade de adaptação rápida e contínua do processo de desenvolvimento de software às mudanças nas necessidades e demandas do mercado. Essa flexibilidade é fundamental para que as empresas possam responder de forma eficaz às mudanças em um ambiente de negócios em constante evolução. O dinamismo é alcançado por meio de uma abordagem iterativa e incremental que permite a entrega contínua de software funcional, permitindo feedbacks constantes do cliente e ajustes na estratégia de desenvolvimento.

Com o objetivo de aprimorar a capacidade de adaptação, essa dimensão é composta por diversas características, tais como a habilidade de mudar requisitos, entregas frequentes, softwares funcionais, ritos definidos de Inspeção, Adaptação e Transparência, iteração contínua e geração de artefatos, apontam Schwaber & Beedle (2002). No Quadro 22, elencamos, baseado no *Scrum*, quais as práticas necessárias a serem avaliadas em uma organização em relação a sua capacidade de implementar processos de forma consistente e eficiente.

**Quadro 22:**  
Formação da dimensão Agilidade - Dinamismo

Dimensão Agilidade	Dinamismo
Dinamismo	Habilidade de mudar requisitos
	Entregas frequentes
	Softwares funcionais
	Ritos definidos - Inspeção, Adaptação e Transparência
	Iteração contínua
	Geração de artefatos

Fonte: Elaborado pelo autor, baseado em Mafakheriab *et al.* (2008), Schwaber & Beedle (2002); Schwaber (2004)

Como descrito no quadro acima, entre as suas principais características, o modelo de Mafakheriab *et al.* (2006), preconiza que a habilidade de mudar requisitos a qualquer momento, sem prejudicar o projeto como um todo, só é possível devido à entrega frequente de funcionalidades, o que permite aos desenvolvedores testarem o software e darem *feedbacks* para que ajustes possam ser feitos rapidamente.

A entrega de softwares funcionais é outra característica do *Scrum*, uma vez que cada iteração resulta em um produto finalizado, mesmo que não contenha todas as funcionalidades desejadas.

Os ritos definidos pelo *Scrum* - Inspeção, Adaptação e Transparência - garantem que o processo seja transparente e eficiente, permitindo a inspeção constante do produto e o ajuste das estratégias para melhorar a qualidade do software. A iteração contínua é uma outra característica do *Scrum*, já que as *sprints* - períodos de tempo definidos para entrega de um conjunto de funcionalidades - ocorrem de forma sucessiva até que o software esteja pronto para ser lançado.

Por fim, a geração de artefatos - como o *Product Backlog* e o *Sprint Backlog* - permitem que o projeto seja gerenciado de forma mais eficiente, com a documentação necessária para auxiliar na tomada de decisões e na avaliação do progresso do projeto.

#### 4.1.2.2.2 Componentes da dimensão: equipe

Segundo o *framework Scrum*, a equipe de desenvolvimento deve ser auto-organizável e multidisciplinar, ou seja, deve possuir todas as habilidades necessárias para entregar um incremento de produto funcional. Isso significa que a equipe não precisa depender de pessoas externas para realizar suas tarefas e que deve ser composta por profissionais de diferentes áreas, como desenvolvedores, designers, testadores, entre outros. Dessa forma, a equipe é capaz de tomar decisões e solucionar problemas de forma ágil e eficiente, sem depender de hierarquias ou burocracias

excessivas. No Quadro 23, estabelecemos as práticas recomendadas pelo *framework Scrum*, na formação da dimensão equipe.

**Quadro 23:**  
Formação da dimensão Agilidade - Equipe

Dimensão Agilidade	Equipe
Equipe	Definição de atribuições e papéis
	Atuação semiautônoma e autoadministrada
	Participação ativa e colaborativa

Fonte: Elaborado pelo autor, baseado em Mafakheriab *et al.* (2008); Schwaber & Beedle (2002); Schwaber (2004)

De acordo com os autores, Schwaber & Beedle (2002), enquanto organizações adotando práticas ágeis, pressupõe a mudança no modelo de gestão de equipes de desenvolvimento de software. Nesse sentido, conforme explorado no quadro acima, três aspectos se destacam como fundamentais: definição clara de atribuições e papéis, atuação semiautônoma e autoadministrada dos membros da equipe e participação ativa e colaborativa em todas as etapas do processo de desenvolvimento. A definição de atribuições e papéis busca estabelecer a clareza e a transparência nas responsabilidades e atividades de cada membro da equipe, permitindo uma melhor gestão dos recursos e uma maior eficiência no processo.

A atuação semiautônoma e autoadministrada dos membros da equipe pressupõe uma maior autonomia na tomada de decisões e na gestão dos processos, o que pode resultar em maior eficiência, produtividade e engajamento da equipe.

Por fim, a participação ativa e colaborativa em todas as etapas do processo de desenvolvimento contribui para a criação de um ambiente colaborativo, em que as ideias e sugestões dos membros da equipe são valorizadas e incorporadas ao processo, resultando em um produto final de maior qualidade.

#### 4.1.2.2.3 Componentes da dimensão: comunicação

A comunicação efetiva entre os membros da equipe é essencial para o sucesso do projeto, especialmente em um ambiente ágil em que a colaboração e a transparência são enfatizadas. Nesse sentido, o *Scrum* oferece um conjunto de práticas e ferramentas que ajudam a melhorar a comunicação dentro da equipe e com os demais envolvidos externos e interessados.

No Quadro 24, estabelecemos as práticas recomendadas pelo *framework Scrum*, na formação da dimensão comunicação, com elementos essenciais a serem avaliados em uma organização, segundo os autores Schwaber & Beedle (2002).

#### **Quadro 24:**

Formação da dimensão Agilidade - Comunicação

<b>Dimensão Agilidade</b>	<b>Comunicação</b>
<b>Comunicação</b>	Decisões rápidas
	Papel do Cliente
	Redução da documentação

Fonte: Elaborado pelo autor, baseado em Mafakheriab *et al.* (2008); Schwaber & Beedle (2002); Schwaber (2004)

Como visto no quadro acima, dentro do *framework Scrum*, é possível adotar diversas práticas para alcançar um alto nível de maturidade em termos de agilidade e comunicação. Dentre essas práticas, destacam-se a realização de reuniões diárias para tomada de decisões rápidas e eficientes, na participação ativa e colaborativa do cliente em todo o processo de desenvolvimento, e a redução da documentação, evitando excessos que possam prejudicar a agilidade e a comunicação.

A definição clara de atribuições e papéis, aliada a uma atuação semiautônoma e autoadministrada da equipe, também é essencial para que haja uma comunicação eficiente e uma tomada de decisão rápida e precisa, de acordo com Schwaber (2004).

#### 4.1.2.2.4 Componentes da dimensão: teste

A realização de testes frequentes em produtos desenvolvidos é essencial para garantir a qualidade e confiabilidade do software. Nesse contexto, o modelo de maturidade proposto por Mafakheriab (2008) destaca a importância dos testes frequentes em diversas fases do ciclo de vida do produto, desde a etapa de requisitos até a entrega do software.

A implementação de testes frequentes possibilita a detecção precoce de erros e falhas, permitindo que sejam corrigidos de forma mais ágil e eficiente, reduzindo assim os custos de desenvolvimento e garantindo um produto final de alta qualidade.

Além disso, a realização de testes frequentes também contribui para aumentar a confiança do cliente no produto desenvolvido e reduzir o risco de retrabalho e refatoração em etapas posteriores do ciclo de vida do software.

No Quadro 25, estabelecemos as práticas recomendadas pelo *framework Scrum*, na formação da dimensão teste, sendo a frequência de testes o único construto avaliado para compor a escala de maturidade.

#### **Quadro 25:**

Formação da dimensão Agilidade - Teste

<b>Dimensão Agilidade</b>	<b>Teste</b>
<b>Teste</b>	Frequência de testes

Fonte: Elaborado pelo autor, baseado em Mafakheriab *et al.* (2008); Schwaber & Beedle (2002); Schwaber (2004)

Como observado no quadro acima, dentro da perspectiva do *Scrum*, a dinâmica dos testes de software deve ser ágil e contínua, visando garantir a qualidade do produto desenvolvido. Essa prática é conhecida como *Test-Driven Development* (TDD) e consiste em desenvolver os testes antes mesmo de começar a escrever o código, garantindo assim que o produto final atenda aos requisitos definidos e funcione corretamente.

Os testes devem ser automatizados e integrados ao processo de desenvolvimento, permitindo uma rápida identificação e correção de possíveis falhas. Além disso, o *Scrum* prevê a realização de testes durante as iterações, a fim de validar as funcionalidades implementadas e garantir que o produto esteja de acordo com as expectativas do cliente.

#### 4.1.2.2.5 Componentes da dimensão: perfil e conhecimento dos desenvolvedores

A senioridade dos desenvolvedores de software é um fator crucial para a efetividade do *framework Scrum*. Considerando que o *Scrum* se baseia em equipes auto-organizáveis e auto-gerenciáveis, é importante que os membros da equipe tenham um certo nível de experiência e conhecimento para tomarem decisões técnicas e estratégicas de forma independente. A senioridade dos desenvolvedores pode influenciar no sucesso do projeto, uma vez que profissionais mais experientes têm mais facilidade em identificar riscos e oportunidades, propor soluções eficazes e lidar com situações adversas.

Por isso, é fundamental que a equipe seja composta por membros de diferentes níveis de senioridade, de forma a promover um ambiente de aprendizado e troca de conhecimentos. No Quadro 26, estabelecemos as práticas recomendadas pelo *framework Scrum*, na formação da dimensão perfil e conhecimento dos desenvolvedores.

#### **Quadro 26:**

Formação da dimensão Agilidade - Perfil e Conhecimento dos desenvolvedores

<b>Dimensão Agilidade</b>	<b>Perfil e Conhecimento dos desenvolvedores</b>
<b>Perfil e Conhecimento dos desenvolvedores</b>	Senioridade da equipe

Fonte: Elaborado pelo autor, baseado em Mafakheriab *et al.* (2008); Schwaber & Beedle (2002); Schwaber (2004)

Tal como indicado no quadro acima, desenvolvedores mais experientes trazem maior senioridade e conhecimento técnico, que são fundamentais para uma equipe de desenvolvimento de software ágil. Eles podem ajudar a equipe a tomar decisões

melhores e mais informadas, além de fornecer orientação e liderança para os membros mais novos da equipe. Com sua experiência, eles podem antecipar possíveis problemas e resolver problemas de forma mais rápida e eficaz. Além disso, eles podem trazer novas ideias e soluções inovadoras para desafios técnicos.

A presença de desenvolvedores seniores também pode contribuir para a construção de uma cultura de aprendizado contínuo, onde os membros da equipe podem se desenvolver e melhorar suas habilidades técnicas ao trabalhar em conjunto com os membros mais experientes.

Em última análise, ter desenvolvedores seniores na equipe pode elevar significativamente o nível de maturidade da organização e contribuir para o sucesso do projeto.

#### *4.1.2.2.6 Componentes da dimensão: cultura*

A adoção de metodologias ágeis, como o *Scrum*, em organizações desenvolvedoras de softwares, tem se tornado cada vez mais comum para as que buscam maior eficiência e flexibilidade em seus processos. No entanto, a implementação dessas metodologias não se trata apenas de uma mudança de processos, mas também de uma mudança de cultura.

A dimensão cultura é fundamental para que a adoção de práticas ágeis seja bem-sucedida, pois envolve uma mudança de *mindset* e a criação de um ambiente colaborativo, onde os membros da equipe se sintam encorajados a experimentar, inovar e compartilhar conhecimento.

Nesse sentido, compreender a importância da dimensão cultura em um ambiente orientado a metodologias ágeis torna-se fundamental para o sucesso da implementação dessas metodologias. Associamos a autonomia como elemento predominante na cultura, relacionado no Quadro 27.

### **Quadro 27:**

#### Formação da dimensão Agilidade - Cultura

<b>Dimensão Agilidade</b>	<b>Cultura</b>
<b>Cultura</b>	Alto grau de liberdade para qualquer um da equipe

Fonte: Elaborado pelo autor, baseado em Mafakheriab *et al.* (2008); Schwaber & Beedle (2002); Schwaber (2004)

Conforme apontado no quadro acima, a cultura deve ser moldada de forma a incentivar a colaboração, a transparência, a comunicação aberta, a melhoria contínua e a valorização do *feedback*. Além disso, é preciso que a liderança da organização esteja engajada e comprometida com a mudança cultural necessária para que as práticas ágeis sejam efetivamente implementadas.

A cultura ágil deve permear toda a organização, desde a equipe de desenvolvimento até a alta gerência, para que haja uma visão compartilhada do valor entregue aos clientes e dos objetivos da organização.

Portanto, Schwaber (2004), ressalta que a cultura é um elemento crucial para o sucesso da adoção das metodologias ágeis, permitindo que a empresa obtenha benefícios como maior flexibilidade, capacidade de resposta às mudanças e aumento da satisfação do cliente.

#### **4.1.3 Componentes do modelo teórico de autoavaliação**

Como demonstrado na definição e operacionalização dos domínios chaves na coleta de dados bibliográfica, a escala de dimensão foi composta pelas competências: *Exploration*, *Exploitation*, Orientação Estratégica e Agilidade.

Assim, conta-se com três modelos orientados a competências nas dualidades estratégicas *exploitation-exploration*, onde cada um destes modelos são avaliados em seis perspectivas (práticas de conhecimento, práticas inovadoras, concorrência, eficiência, qualidade e orientação estratégica) e com, um modelo orientado as competências ambidestras, ao assumir o uso do *Scrum* como metodologia ágil

concorrentemente ao uso de metodologias tradicionais de desenvolvimento de software. Deriva-se em seis perspectivas (Dinamismo, Equipe, Comunicação, Teste, Perfil e conhecimento e Cultura). Todas as perspectivas descritas contam com perguntas da ferramenta de autoavaliação focadas em práticas recomendadas para se atingir um nível de maturidade, e processo de conhecimento que devem ser implementados.

Por exemplo, no modelo *Exploration* na perspectiva Práticas Inovadoras, procura-se analisar práticas de conhecimento relativas à organização. São levados em consideração os processos propostos para cada nível de maturidade, as áreas de processos inseridas no componente do domínio citado, e neste exemplo, foca-se concentração de produtos ou processos novos, desenvolvimento de protótipos, busca por novos mercados, inovação técnica em marketing, taxa de inovação de produtos e participação agressiva em novas alianças.

Os quadros a seguir possuem as questões propostas para o modelo *exploration*, *exploitation*, orientação estratégica e agilidade, em todas as doze perspectivas de derivadas dos modelos.

O Quadro 28 apresenta as questões do modelo de autoavaliação de maturidade para os modelos estudados neste trabalho.

**Quadro 28:**

Questões de autoavaliação para modelo teórico de pesquisa qualitativa.

<i>Exploration</i>	
Práticas de Conhecimento	Participamos de um ambiente propício à geração de novas ideias, com processos e incentivos para que os colaboradores possam contribuir ativamente com sugestões de melhorias, inovações e soluções criativas. <b>Há cultura de valorização e um volume de ideias significativas na organização.</b>
	Temos um nível de concordância em relação a utilização de novas fontes de conhecimento para a tomada de decisão, embora não seja uma prática comum. <b>A organização tem intenção de fomentar o uso de novas fontes de conhecimento.</b>
	Utilizamos o uso de conhecimento existente para melhorar os processos, mas nem sempre temos as melhores informações atualizadas. <b>Existe uma abordagem de armazenar e usar o conhecimento, mas não há fomento na manutenção de novos processos.</b>
	Incentivamos o compartilhamento do conhecimento interno. <b>As atividades de conhecimento interno são enfim registradas e armazenadas e de certa</b>

	<p><b>forma disponíveis a todos.</b></p> <p>Estimulamos o aprendizado individual, embora não seja possível controlar o nível de conhecimento. <b>Não utiliza-se métricas para o aumento da produtividade.</b></p> <p>Estimulamos o aprendizado coletivo, embora não seja possível controlar o nível de conhecimento. <b>Não utiliza-se métricas para o aumento da produtividade.</b></p> <p>Temos programas e incentivos para o desenvolvimento de habilidades e conhecimentos específicos. <b>Há intenção de gerenciar o conhecimento e melhoria contínua de produtividade,</b></p>
Práticas Inovadoras	<p>Atendemos melhor às necessidades dos clientes, devido a grande concentração de produtos enquanto a concentração de processos nos permite operar de maneira mais eficiente e eficaz. <b>Há uma consciência organizacional competitiva sobre o equilíbrio entre produtos e atendimento</b></p> <p>Utilizamos ferramentas e técnicas de inovação aberta para apoiar o desenvolvimento de protótipos. <b>Há uma abordagem comum incorporada na organização para promover a inovação de novos produtos</b></p> <p>Preocupa-se com a promoção da inovação de produtos, garantindo uma cultura de criatividade e aprendizado contínuo em todos os níveis. <b>Práticas estabelecidas para compartilhamento e uso do conhecimento é parte integrante da organização e continuamente melhorada.</b></p> <p>Temos inovado em relação às novas tendências e técnicas de marketing. <b>Há consciência da necessidade de inovação e abertura de penetração no mercado</b></p> <p>Temos aberto novos canais de distribuição em relação aos canais existentes. <b>Há intenção em desenvolver e gerenciar o equilíbrio estratégico.</b></p> <p>Há um incentivo na inovação radical sem comprometer a base de clientes atual. <b>Reconhecimento da necessidade de inovar otimizando o desempenho na manutenibilidade dos clientes.</b></p> <p>Há um incentivo na inovação tecnológica radical sem comprometer os produtos e serviços existentes. <b>Reconhecimento da necessidade de inovação tecnológica integrada com o portfólio de produtos atual.</b></p> <p>Preocupamos em buscar novos mercados de atuação. <b>Necessidade de expansão e diversificação de fontes de receita e dependência de mercados únicos.</b></p> <p>Preocupamos em formar novas alianças estratégicas baseadas em tecnologias. <b>Necessidade de maior eficiência operacional e competitividade</b></p>
<i>Exploitation</i>	
Concorrência	<p>Monitoramos o surgimento de novos concorrentes em seu mercado-alvo. <b>Necessidade de identificação de possíveis ameaças e oportunidades de negócio.</b></p> <p>Monitoramos continuamente o mercado em busca de possíveis produtos ou serviços mais eficientes e que potencialmente podem impactar nos produtos e</p>

	processos atuais. <b>Adaptação às novas tendências, tecnologias e aumentar a eficiência dos produtos existentes.</b>
	Monitoramos continuamente a concorrência no mercado local. <b>Necessidade de ajustar estratégias e táticas do negócio</b>
	Avaliamos com frequência o resultado das ações promocionais em relação aos concorrentes. <b>Necessidade de ajustar estratégias de marketing e identificação de vantagens competitivas.</b>
Eficiência	Buscamos garantir a conformidade com normas e regulamentações das rotinas e processos atuais. <b>Manutenção da confiabilidade e integridade dos negócios e processos.</b>
	Avaliamos através de indicadores de desempenho a eficiência da empresa. <b>Identificação de possíveis oportunidades de melhoria.</b>
	Preocupamos em equilibrar a busca pela redução de custos com a necessidade de manter ou melhorar a qualidade dos produtos/serviços oferecidos. <b>Necessidade manter a rentabilidade da empresa e aumentar sua capacidade de investimento em novos projetos</b>
	Preocupamos em buscar constantemente melhorias nos processos produtivos. <b>Avaliação contínua dos processos existentes para identificar oportunidades de otimização e melhoria</b>
Qualidade	Existe uma equipe de testes dedicada para garantir a qualidade do produto. <b>Garantir a qualidade do produto e a satisfação do cliente, além de melhorar a reputação da empresa</b>
	Há uma preocupação em relação a conformidade nos processos e padrões de qualidade. <b>Garantir a qualidade do produto ou serviço oferecido, mas também a proteger a empresa de possíveis consequências legais</b>
Orientação Estratégica	
Orientação Estratégica	A visão estratégica é focada no presente. <b>Adaptação às mudanças do mercado e mantenha uma vantagem competitiva</b>
	Estratégias focadas no curto prazo. <b>Capacidade de lidar com questões imediatas, como a necessidade de aumentar as vendas, melhorar a eficiência operacional ou reduzir custos</b>
Agilidade	
Dinamismo	Adotamos metodologias ágeis nos processos de desenvolvimento de software, você diria que o time de desenvolvimento tem habilidades em alterar requisitos de forma frequente. <b>Alta capacidade de responder às mudanças, em vez de seguir um plano rígido e inflexível.</b>
	Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza entregas frequentes. <b>Entrega iterativa e incremental de funcionalidades do software, de forma que o cliente possa ter acesso a versões do produto que já tenham algum valor.</b>
	Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza entregas funcionais de software. <b>Objetivo é</b>

	<p><b>entregar uma versão do software com funcionalidades básicas, mas que possa ser utilizada pelo cliente para obter algum benefício ou resultado.</b></p> <p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza os eventos (<i>Release Planning, Sprint, sprint planning, sprint review, retrospective e daily scrum</i>). <b>Necessidade de garantir que o processo de desenvolvimento seja iterativo e incremental, com entregas frequentes de software funcional.</b></p> <p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza iterações frequentes. <b>Durante cada sprint, o time de desenvolvimento trabalha em um conjunto de atividades previamente definidas, que incluem a análise de requisitos, o desenvolvimento de funcionalidades, testes e integração.</b></p> <p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento utiliza-se dos artefatos (<i>product backlog, release burndown, sprint backlog, sprint burndown e o DoD</i>) em cada sprint. <b>Necessidade de alinhamento em relação aos objetivos do projeto, que todas as funcionalidades estejam claramente definidas e priorizadas, e que o progresso do projeto possa ser medido e monitorado</b></p>
Equipe	<p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento possui as atribuições e papéis (<i>Scrum Master, Product Owner, Desenvolvedores de Software</i>) estabelecidos? <b>Necessidade de garantir que cada membro do time de desenvolvimento tenha responsabilidades claras e que as decisões sejam tomadas de forma colaborativa e transparente</b></p> <p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento possui atuação semiautônoma e auto administrada. <b>Liberdade de tomar decisões e gerenciar o próprio trabalho, em vez de depender de direções e ordens hierárquicas</b></p> <p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento tem participação ativa e colaborativa. <b>Necessidade de compartilhamento de conhecimentos e habilidades entre os membros da equipe, além da participação em reuniões diárias, revisões e retrospectivas.</b></p>
Comunicação	<p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, embora temos alta frequência de mudanças de prioridades em projetos em andamento. <b>A frequência de mudanças de prioridades pode variar dependendo do projeto e das necessidades do cliente, mas não deve ser uma prática comum.</b></p> <p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, a equipe de desenvolvimento tem autonomia para realizar alterações no produto com base no feedback do cliente. <b>Liberdade para cada membro da equipe contribuir com ideias e soluções, é possível obter resultados melhores e mais inovadores</b></p> <p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, com qual frequência a equipe de desenvolvimento revisa e valida os artefatos produzidos sem documentação detalhada. <b>Necessidade de validação dos artefatos é contínuo e realizado de forma iterativa e incremental durante as sprints.</b></p>
Testes	<p>Adotamos metodologias ágeis nos processos de desenvolvimento de software, e com frequência a equipe de desenvolvimento realiza os testes dos potenciais</p>

	entregáveis. <b>Necessidade de garantir que o software esteja sempre funcionando e atendendo aos requisitos definidos</b>
Perfil e Conhecimento	Adotamos metodologias ágeis nos processos de desenvolvimento de software, e com frequência a equipe atua utilizando alta senioridade dos desenvolvedores. <b>Importante que a equipe tenha uma combinação adequada de habilidades e conhecimentos, com membros experientes</b>
Cultura	Entendemos a importância do nível de concordância em relação ao alto grau de liberdade a todos da equipe. <b>Importante que haja um nível de concordância em relação aos objetivos e metas a serem alcançados, para que a equipe trabalhe de forma unida e em direção ao mesmo propósito.</b>

Fonte: Elaborado pelo autor.

A explicação das relações entre as variáveis é uma das principais contribuições do modelo teórico de pesquisa qualitativa para o avanço do conhecimento científico. Esse modelo permite ao pesquisador identificar e descrever as diferentes variáveis envolvidas no fenômeno estudado e como elas interagem entre si, levando a uma compreensão mais completa e profunda do fenômeno em questão. Além disso, o modelo teórico pode fornecer explicações plausíveis para os resultados obtidos na pesquisa, permitindo ao pesquisador ir além da simples descrição dos dados e propor hipóteses explicativas para o fenômeno estudado.

Por fim, o modelo teórico de pesquisa qualitativa pode ser utilizado como base para o desenvolvimento de novas pesquisas, permitindo a elaboração de novas questões de pesquisa e a definição de novas variáveis a serem investigadas. Dessa forma, o modelo teórico de pesquisa qualitativa é uma ferramenta valiosa para o desenvolvimento de pesquisas de qualidade e para o avanço do conhecimento científico em diversas áreas do conhecimento.

#### **4.1.4 Níveis de maturidade**

Os modelos de maturidade são utilizados para avaliar a capacidade de uma organização em atingir seus objetivos em determinadas áreas, tais como gerenciamento de projetos ou desenvolvimento de software. Esses modelos são compostos por níveis que representam etapas de evolução da organização, onde cada nível apresenta um conjunto de práticas e processos mais maduros e eficientes do que o anterior. A utilização de níveis permite uma avaliação mais precisa da maturidade da organização, além de fornecer um caminho claro para a melhoria contínua de suas práticas e processos. Dessa forma, os níveis são essenciais para a

aplicação e compreensão dos modelos de maturidade, tornando-os uma ferramenta valiosa para a gestão e aprimoramento das práticas organizacionais.

Baseando-se ainda na revisão da literatura, particularmente nos modelos de De Bruin *et al.* (2005), propõe-se quatro níveis destinados ao modelo de maturidade desenvolvido nesta pesquisa: Ad Hoc, Padronizado, Gerenciado e Otimizado, descritos pelo Quadro 29.

**Quadro 29:**  
Níveis de maturidade propostos nesta pesquisa

<b>Nível</b>	<b>Modelo de maturidade proposto</b>
1	Ad hoc
2	Padronizado
3	Gerenciado
4	Otimizado

Fonte: Elaborado pelo autor.

Como ilustrado no quadro acima, a proposta de quatro níveis para o modelo de maturidade desenvolvido nesta pesquisa visa estabelecer uma estrutura clara e organizada para avaliar a evolução do processo ou sistema em estudo. Ainda, o uso de escalas de maturidade é fundamental para avaliar e monitorar a evolução do processo ou sistema em estudo, permitindo uma melhoria contínua e o alcance da excelência na execução das atividades.

Combinando as características dos modelos, em relação aos objetivos do atendimento das práticas estabelecidas no referencial teórico, sob a ótica das dualidades estratégicas e ambidestria ao uso de metodologias ágeis, consolida-se as características de cada nível de maturidade, descritas no Quadro 30.

Quadro 30 - Níveis de maturidade e respectivas características propostos no modelo de maturidade nesta pesquisa

Nível	Proposta do modelo de maturidade
1 - Ad hoc	A organização não tem uma abordagem estruturada para explorar novas oportunidades ou otimizar processos existentes. As práticas de <i>exploitation</i> e <i>exploration</i> não são reconhecidas ou realizadas de maneira informal e não documentada. Cultura organizacional não suporta práticas ágeis
2 - Padronizado	A organização começou a adotar práticas de <i>exploitation</i> e <i>exploration</i> , mas elas são realizadas de maneira desorganizada e ainda não há uma estrutura formal para gerenciá-las. A organização pode estar experimentando metodologias ágeis, como o <i>Scrum</i> , em alguns projetos, mas não respeitando todos os eventos e ritos definidos pelo <i>framework</i> , nem com atribuições de papéis e responsabilidade definidos.
3 - Gerenciado	A organização possui uma estrutura formal para gerenciar práticas de <i>exploitation</i> e <i>exploration</i> . As metodologias ágeis, como o <i>Scrum</i> , são amplamente utilizadas na organização e existem processos documentados para apoiar a adoção dessas práticas. A organização também possui processos bem estabelecidos, mas não otimizados, para gerenciar projetos tradicionais.
4 - Otimizado	A organização é altamente eficiente em gerenciar práticas de <i>exploitation</i> e <i>exploration</i> . As metodologias ágeis são utilizadas em todos os projetos e a organização tem uma cultura forte de inovação e melhoria contínua. A organização também tem processos bem estabelecidos para gerenciar projetos tradicionais e é capaz de adotar novas práticas com rapidez e eficácia.

Fonte: Elaborado pelo autor.

Como demonstrado no quadro acima, a primeira fase no desenvolvimento do modelo de maturidade é o Ad hoc. A expressão “ad hoc” é frequentemente utilizada na área de tecnologia da informação para se referir a soluções improvisadas, que são criadas de forma temporária para atender uma necessidade específica, sem seguir um padrão ou processo formal.

No contexto da maturidade de organizações desenvolvedoras de software, o nível "ad hoc" geralmente é utilizado para se referir a organizações que não possuem um processo formal de desenvolvimento de software e que adotam soluções improvisadas e não documentadas para atender às demandas de seus clientes ou projetos. A dualidade *exploitation-exploration* não é reconhecida e não há um esforço para equilibrá-las.

Organizações que se encontram no nível "ad hoc" geralmente têm dificuldade em gerenciar projetos de forma eficiente e em manter a qualidade do software

desenvolvido. As soluções criadas são muitas vezes únicas e não podem ser facilmente replicadas em outros projetos. Além disso, a falta de um processo formal de desenvolvimento pode levar a atrasos e inconsistências no produto final.

Organizações no nível padronizado, possuem pouca ou nenhuma intenção ou consciência em gerenciar formalmente o conhecimento (identificação) visto que o mesmo não é explicitamente reconhecido como um fator crítico de sucesso em projetos.

Organizações que se enquadram nesse nível de maturidade geralmente possuem equipes com pouca experiência em metodologias ágeis e em gerenciamento de projetos tradicionais. As práticas de *exploitation* e *exploration* são utilizadas de forma intuitiva e sem uma estrutura formal definida para gerenciá-las. Os projetos que utilizam metodologias ágeis, no caso o *Scrum*, são realizados de maneira desorganizada e os eventos e ritos definidos pelo *framework* são desconsiderados ou realizados de forma inadequada. As atribuições de papéis e responsabilidades também não são claramente definidas, o que pode gerar conflitos e atrasos no desenvolvimento dos projetos. Nesse cenário, é comum haver falta de comunicação efetiva entre os membros da equipe e com o cliente, o que pode levar a retrabalho e insatisfação do cliente. A falta de documentação também pode ser um fator comum nesse tipo de organização. Em resumo, a organização apresenta uma baixa maturidade em relação à utilização de metodologias ágeis e precisa de aprimoramento em seus processos para alcançar um nível de maturidade adequado.

No nível 2, padronizado, as organizações que se encaixam nesse perfil de maturidade costumam ter uma abordagem mais informal e improvisada em relação ao gerenciamento de projetos e processos de desenvolvimento de software. A falta de estrutura formal para gerenciar as práticas de *exploitation* e *exploration* pode levar a uma falta de clareza sobre os objetivos e prioridades da organização, bem como à falta de alinhamento entre as equipes e departamentos.

Além disso, a falta de respeito aos eventos e ritos do *Scrum* e a falta de atribuições de papéis e responsabilidades claras, podem levar a um trabalho desorganizado e

sem uma direção clara. A comunicação também pode ser prejudicada, com informações importantes perdidas ou mal interpretadas.

Essas organizações podem ter dificuldade em lidar com mudanças e incertezas, o que pode resultar em projetos atrasados, com custos elevados e baixa qualidade. A falta de disciplina e consistência em processos e práticas pode levar a um ambiente caótico e estressante para as equipes, com um grande risco de retrabalho e conflitos.

Dessa forma, é importante que essas organizações invistam em uma estrutura formal para gerenciar suas práticas de *exploitation* e *exploration*, bem como no treinamento e capacitação de suas equipes para entender e seguir as metodologias ágeis corretamente. A definição clara de papéis e responsabilidades e a adoção de práticas de comunicação eficazes também são fundamentais para melhorar a produtividade e qualidade do trabalho.

No nível 3 - gerenciado, a organização apresenta um nível de maturidade gerenciado em relação às práticas de *exploitation* e *exploration*, com uma estrutura formal estabelecida para gerenciá-las e metodologias ágeis, como o *Scrum*, sendo amplamente utilizadas e documentadas na organização. No entanto, a maturidade em relação a projetos tradicionais ainda não é otimizada, sugerindo que a organização pode estar buscando o equilíbrio entre as práticas ágeis e tradicionais. Além disso, é possível que a organização esteja trabalhando em uma cultura de melhoria contínua para aprimorar seus processos e torná-los mais eficientes.

No nível 4 - otimizado, as organizações que se encaixam nesse perfil de maturidade são caracterizadas por uma forte cultura de inovação e melhoria contínua, com uma liderança engajada em fomentar essa mentalidade. Essas organizações possuem um ambiente colaborativo e aberto, que encoraja a experimentação e o aprendizado constante. Além disso, a utilização de metodologias ágeis é amplamente difundida e implementada em todos os projetos, com uma atenção especial para o respeito aos eventos e ritos definidos pelo *framework* Scrum. Essas organizações priorizam a entrega de valor contínuo ao cliente, utilizando *sprints* regulares para fornecer incrementos funcionais do produto. As equipes são autogerenciáveis e auto-organizáveis, e têm um forte compromisso com a qualidade do produto. A gestão de

projetos tradicionais também é eficiente, mas sem perder de vista a necessidade de se adaptar às mudanças e evoluir para novas práticas. A organização é capaz de adotar novas tecnologias e abordagens com rapidez e eficácia, mantendo-se sempre atualizada e competitiva no mercado.

Sendo assim, uma organização sem maturidade não sabe, não tem consciência ou habilidade em explorar o equilíbrio entre dualidades estratégicas *exploration-exploitation* e não há uma cultura organizacional preparada para adoção de ambidestria metodológica.

Para o modelo proposto, além das características da descritas, foram incorporados os processos do conhecimento, derivados do modelo de De Bruin *et al.* (2005), apropriados para cada nível de maturidade apresentado no Quadro 31 com suas respectivas escalas.

**Quadro 31:**

Processos de conhecimento incorporados ao modelo de maturidade proposto

<b>Maturidade</b>	<b>Processos de conhecimento</b>				
<b>Nível</b>	<b>Identificar</b>	<b>Criar</b>	<b>Armazenar</b>	<b>Compartilhar</b>	<b>Usar</b>
Inconsciente	0	0	0	0	0
1 - Ad hoc	1	1/2	1/2	0	0
2 - Padronizado	1	1	1/2	1/2	0
3 - Gerenciado	1	1	1	1/2	1/2
4 - Otimizado	1	1	1	1	1

Fonte: Elaborado pelo autor.

A adoção de processos de conhecimento, como abordado no quadro acima, é uma estratégia importante para aprimorar o modelo de maturidade desenvolvido nesta pesquisa. A incorporação de processos de conhecimento significa a utilização de informações e experiências anteriores para aprimorar o modelo, tornando-o mais eficiente e eficaz. Além disso, a incorporação de processos de conhecimento permite ao modelo estar atualizado com as últimas tendências e melhores práticas em sua área de atuação, tornando-o mais competitivo e inovador. Dessa forma, a adoção de processos de conhecimento é uma estratégia essencial para o desenvolvimento e

aprimoramento do modelo de maturidade, contribuindo para o alcance da excelência e a melhoria contínua do processo ou sistema em estudo.

#### **4.1.5 Framework do modelo teórico de maturidade**

A utilização de um *framework* de um modelo teórico de maturidade traz diversos benefícios para as organizações. Um *framework* oferece uma estrutura para a avaliação e melhoria de práticas e processos, proporcionando um maior controle sobre as atividades e permitindo a identificação de áreas que precisam de melhorias. Segundo Silva *et al.* (2018), a utilização de frameworks consagrados é indicada e verificada em estudos que mediram as eficácias enquanto boas práticas organizacionais, mesmo assim, foi identificado no estudo que grande parte das empresas pequenas, médias e grandes, não implementam esses *frameworks*, preferindo utilizar metodologias próprias e desenvolvidas a partir de metodologias de tentativa e erro.

Sendo assim, um *framework* das dimensões avaliadas expandido foi criado para melhor descrever a forma da construção do modelo incorporado aos processos de conhecimento e níveis de maturidade, como demonstrado na Figura 8.

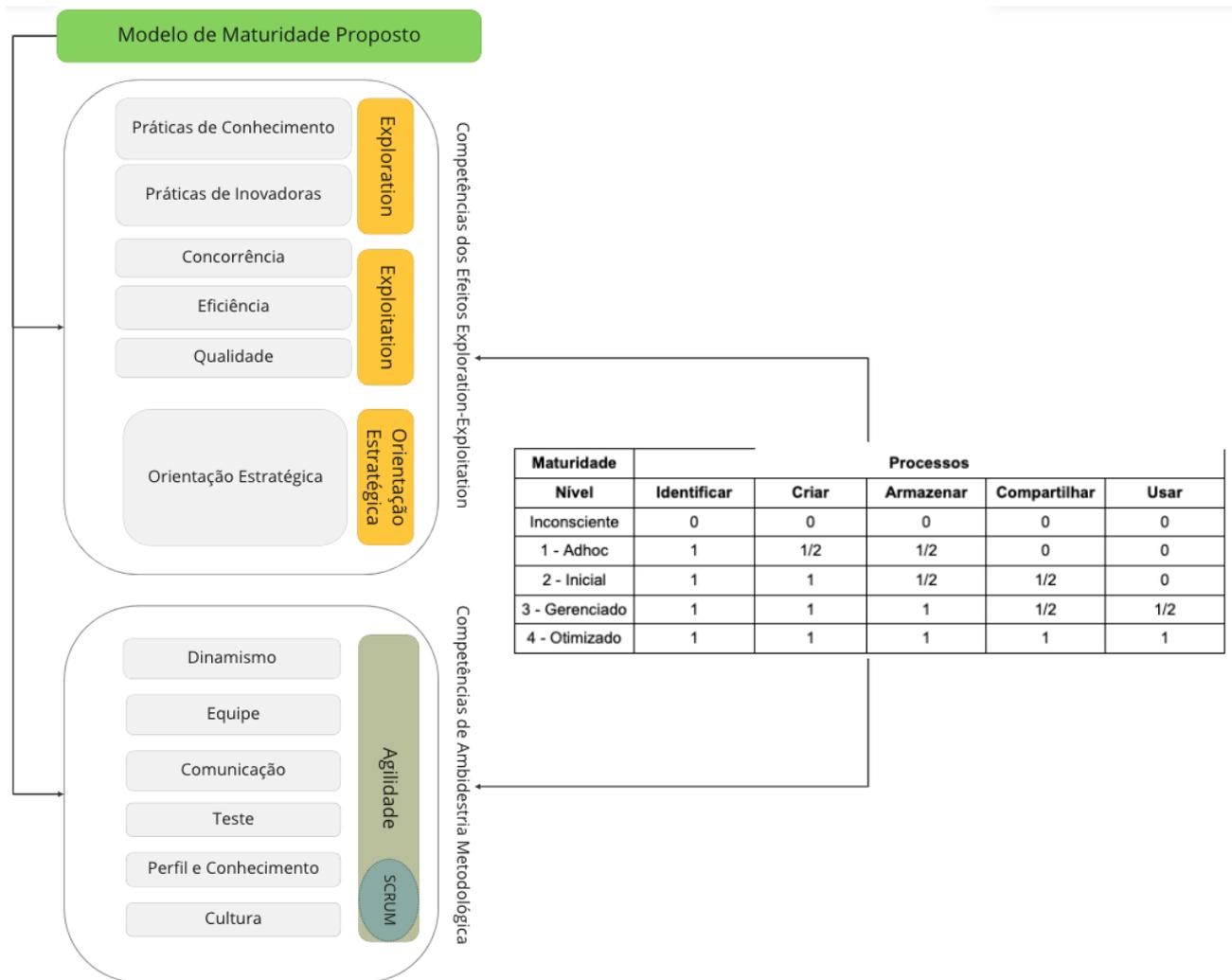


Figura 8 - Framework de construção do modelo de maturidade proposto  
 Fonte: Elaborado pelo autor

Este *framework* ajuda a padronizar as práticas e processos em toda a organização, o que pode aumentar a eficiência e a produtividade. A utilização do *framework* também pode levar a uma melhoria na qualidade dos produtos e serviços, pois promove a adoção de boas práticas e a eliminação de processos desnecessários ou ineficientes. Assim, o *framework* de um modelo teórico de maturidade pode ajudar as organizações a se tornarem mais competitivas no mercado, ao oferecer uma estrutura para a melhoria contínua e aprimoramento das práticas e processos.

Com a adaptação do instrumento de autoavaliação contendo os componentes de autoavaliação, o *framework* do modelo proposto com os respectivos processos de conhecimento incorporados, baseado por De Bruin *et al.* (2005) em todas as perspectivas de dimensão e componentes. A Figura 9 demonstra os processos de

conhecimento atuando de forma perpendicular às perspectivas, gerando pontos de intersecção com as mesmas.

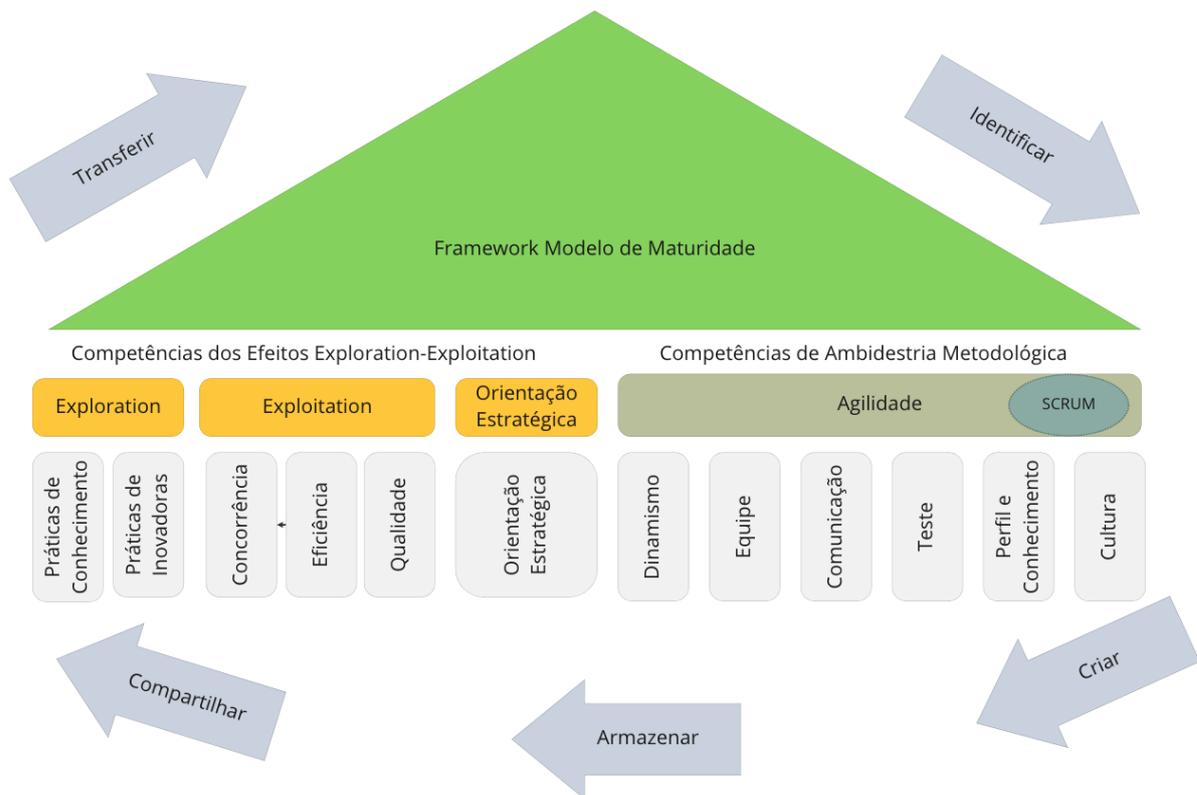


Figura 9 - Modelo de maturidade proposto  
Fonte: Elaborado pelo autor

Com as características do modelo proposto definidas, com a definição dos componentes do domínio (o que se quer avaliar) e seu agrupamento, com a definição dos níveis de maturidade, a inserção das atividades do conhecimento, a adaptação da ferramenta de autoavaliação e a proposição de *framework*, o pesquisador passa a adotar este modelo conceitual para medir a maturidade das organizações desenvolvedoras de software sob a ótica dos efeitos *exploration-exploitation* e ambidestria metodológica.

## 4.2 Validação do modelo por entrevista

Apresentaremos adiante os resultados da validação do modelo de maturidade teórico e as conclusões obtidas a partir da análise dos dados coletados.

#### 4.2.1 Caracterização dos pesquisados

Os entrevistados são compostos de profissionais que trabalham em empresas desenvolvedoras de software, no mercado brasileiro. Foram convidados 5 profissionais que compuseram o corpo de avaliadores do modelo conforme Tabela 1.

**Tabela 1**

Distribuição dos especialistas de acordo com as variáveis: sexo, faixa etária, qualificação e experiência.

Variáveis		Presença	%
Sexo	Feminino	1	20%
	Masculino	4	80%
	<b>Total</b>	<b>5</b>	<b>100%</b>
Faixa Etária	< 35 anos	0	0%
	36 a 41 anos	1	20%
	42 a 47 anos	3	60%
	48 a 53 anos	1	20%
	54 a 59 anos	0	0%
	> 59 anos	0	0%
	<b>Total</b>	<b>5</b>	<b>100%</b>
	Qualificação	Graduação	1
Especialização		2	40%
Certificação na área		1	20%
Mestrado		1	20%
Doutorado		0	0%
<b>Total</b>		<b>5</b>	<b>100%</b>
Experiência	< 5 anos	0	0%
	6 a 10 anos	2	40%
	11 a 20 anos	3	60%
	> 21 anos	0	0%
	<b>Total</b>	<b>5</b>	<b>100%</b>

Tabela 1 - Distribuição dos especialistas de acordo com as variáveis: sexo, faixa etária, qualificação e experiência

Fonte: Elaborado pelo autor.

Os profissionais selecionados correspondem a 80% do sexo masculino e 20% do sexo feminino. A faixa etária predominante é entre 42 a 47 anos, com 60% dos entrevistados. Em relação a qualificação, os profissionais foram desdobrados em: 20% graduados, 40% com alguma especialização e 20% com certificação. A experiência dominante entre os profissionais é entre 11 a 20 anos.

#### 4.2.2 Discussão dos resultados da avaliação do modelo proposto

Para os profissionais selecionados foi enviado o questionário de avaliação do modelo de maturidade proposto neste estudo apresentado no Apêndice A - Termo de Confidencialidade e no Apêndice B - Instrumento de Coleta de Dados juntamente com as instruções para o preenchimento.

O julgamento dos profissionais sobre o *framework* é apresentado na Figura 10. Essa ponderação traz uma análise das notas individuais, permitindo confrontar com a média de cada tópico. É possível analisar se há alguma provável relação das notas concedidas entre o nível de experiência dos profissionais.

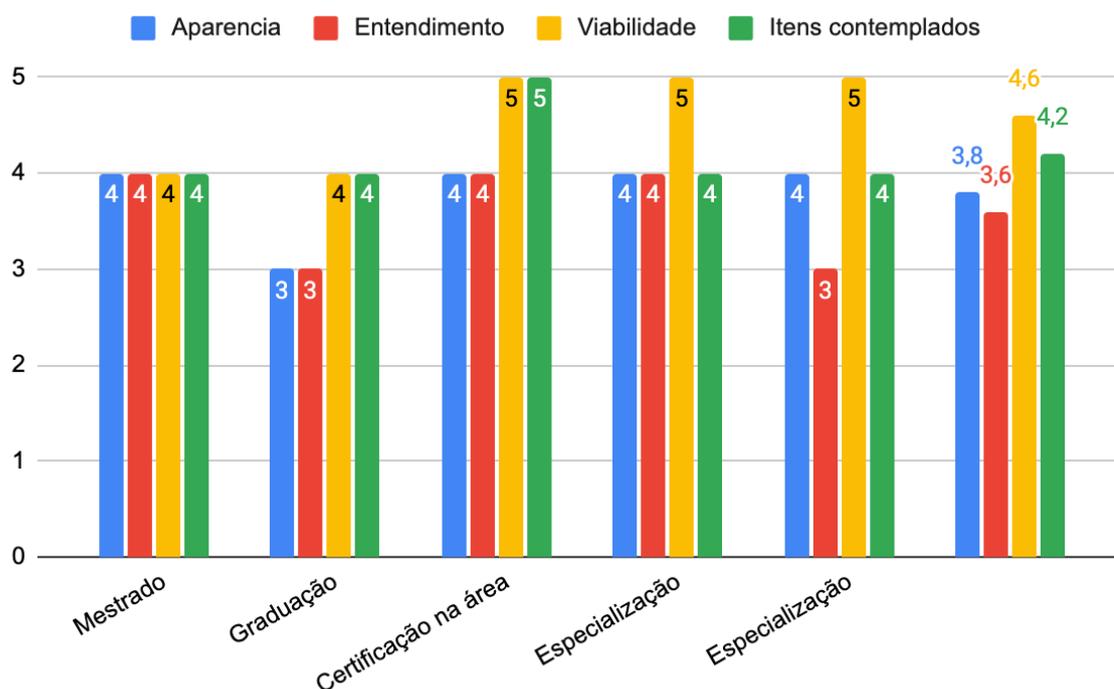


Figura 10 - Consenso entre os profissionais acerca do framework do modelo de maturidade proposto  
Fonte: Elaborado pelo autor

Dentre os cinco especialistas, um profissional com graduação classificou o *framework* como, atende o esperado, pela perspectiva da aparência. Para ambos os especialistas com especialização, classificaram como excede o esperado. Com relação ao entendimento do *framework*, um profissional com especialização e um com graduação, classificaram como atende o esperado, enquanto os demais com nível de qualificação profissional maior, classificaram como excede o esperado. A viabilidade do *framework*, para os profissionais certificados e com especialização na área, classificou como excelente. Quanto aos itens contemplados, somente o profissional com certificação avaliou como excelente. Todos os demais, neste quesito, avaliaram que excede o esperado.

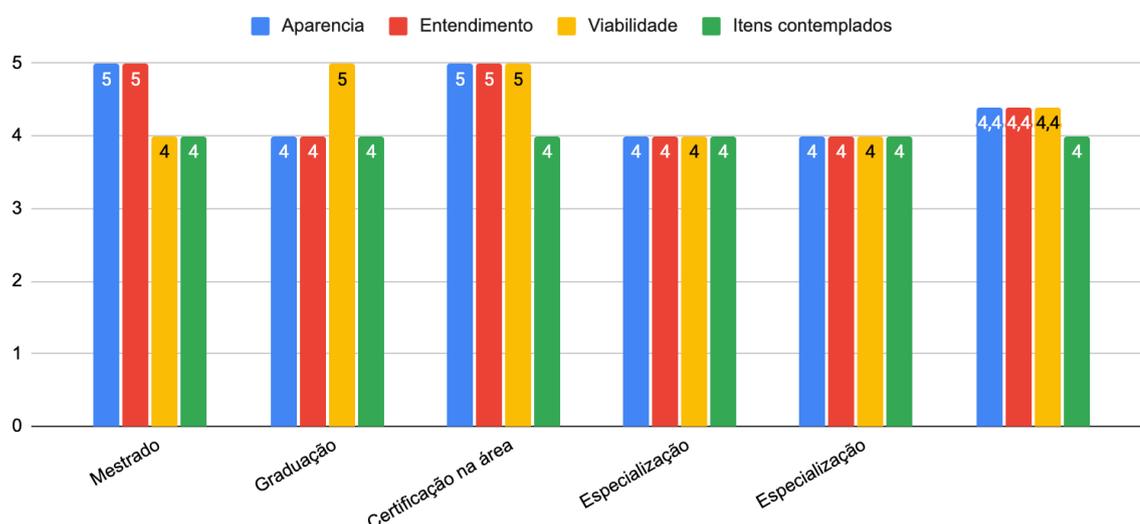


Figura 11 - Avaliação dos níveis de maturidade do modelo proposto.  
Fonte: Elaborado pelo autor.

Para os níveis de maturidade, a avaliação acerca dos itens contemplados para todos os profissionais foi que excede o esperado. O profissional com certificação na área apresentou como excelente, os quesitos de aparência, entendimento e viabilidade.

O profissional com mestrado, classificou como excelente, os itens de aparência e entendimento. Profissionais com graduação e certificação na área, avaliaram a viabilidade do modelo, em relação aos níveis de maturidade, como excelente.

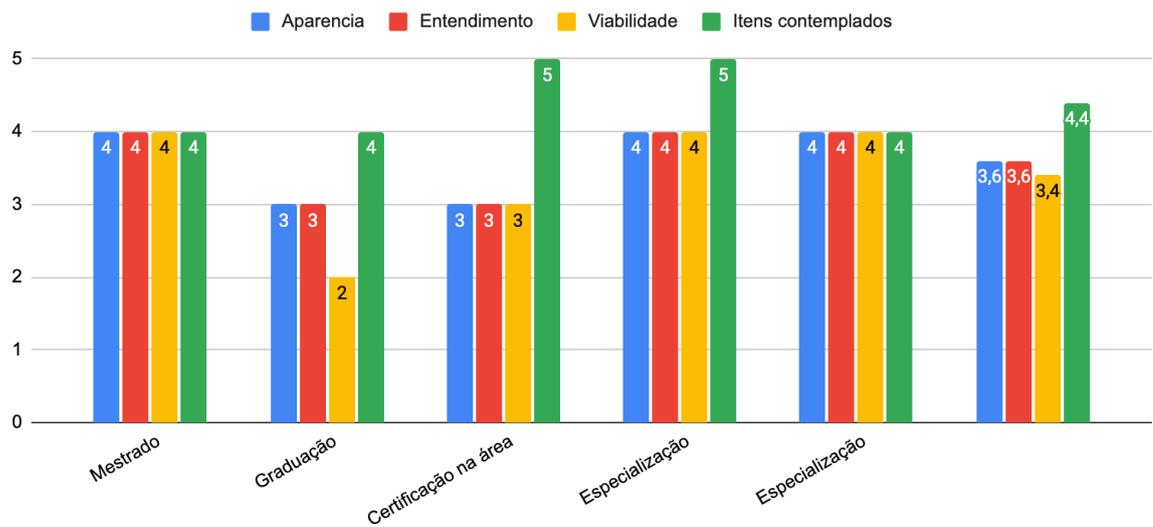


Figura 12 - Avaliação dos componentes dos domínios do modelo de maturidade proposto  
 Fonte: Elaborado pelo autor

A estrutura dos itens e a sequência lógica que foram correlacionados foram avaliados como excelente, para o profissional com certificação na área e para um dos especialistas. O profissional certificado, no entanto, classificou as variáveis aparência, facilidade de entendimento e viabilidade como atende o esperado. O profissional com graduação, considerou os itens aparência e entendimento, como atende o esperado, porém classificou como abaixo do esperado a viabilidade dos componentes para o modelo proposto.

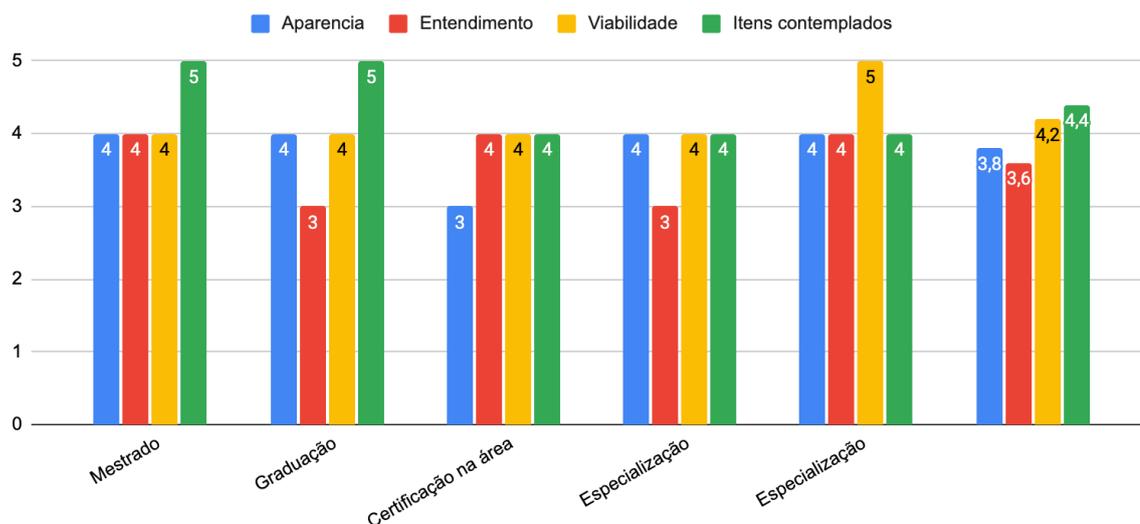


Figura 13 - Avaliação do mecanismo de autoavaliação para modelo de maturidade proposto.  
 Fonte: Elaborado pelo autor

Para os profissionais com mestrado e graduação, os itens contemplados no mecanismo de autoavaliação foram excelentes, e para todos os demais especialistas atenderam a expectativa. Em relação ao entendimento da ferramenta de autoavaliação, os profissionais com graduação e um profissional com especialização avaliaram que atendem a expectativa. Ainda, o profissional certificado na área, avaliou a aparência da ferramenta como atende a expectativa.

Foi fomentado aos profissionais, uma pergunta aberta, conforme abaixo, onde o avaliador pode ter a liberdade para responder levando em consideração a sua percepção pessoal.

Pergunta aberto: de acordo com o modelo de maturidade avaliado, descreva a partir de sua percepção, os pontos fortes e os pontos fracos.

Adiante, no Quadro 32, apresenta as respostas e as percepções individuais dos avaliadores.

**Quadro 32:**  
 Respostas dos avaliadores para a questão aberta proposta

De acordo com o modelo de maturidade avaliado, descreva os pontos fortes e os pontos
--------------------------------------------------------------------------------------

fracos	
Profissional	Respostas
Graduação	Ponto forte: Genérico e aplicável em vários segmentos. Pontos fracos: Termos de administração científica complexos.
Especialização 1	Acrescentaria uma estrutura tradicional para confrontar as metodologias no modelo.
Especialização 2	Pontos fortes: Acredito ter grande adesão nas organizações. Escalável. Pontos fracos: A figura pode ser melhor trabalhada.
Mestrado	Pontos fortes: Reconheço aplicabilidade e adaptabilidade empresarial no modelo. Pontos fracos: Explorar autoavaliação com outras estruturas de desempenho.
Certificação na área	Pontos fortes: Interessante, organizado. Pontos fracos: Acrescentaria componentes de habilidades e metodologias de outros <i>frameworks</i> .

Fonte: Elaborado pelo autor.

## 5 Considerações Finais

Este trabalho propôs a construção de um modelo de maturidade que possa ser aplicado em organizações desenvolvedoras de software ambidestras, ou seja, aplicam metodologias ágeis concomitantes com as metodologias tradicionais. Como toda organização, independentemente a área de atuação, são influenciadas por dualidade estratégicas, conflitantes entre si, e influenciadas por variáveis de incerteza mercadológica e técnicas.

Para isso, foi realizada uma revisão sistemática da literatura, cobrindo os efeitos das dimensões *exploitation-exploration*, nos princípios da ambidestria organizacional, particularmente na adoção de metodologias ágeis e na utilização de modelos de maturidade como balizador, aprimorando o resultado como aconselhado por Bruin *et al.* (2005).

O problema de pesquisa proposto - quais as dimensões que devem ser consideradas para assim, elaborar uma escala de maturidade conceitual para organizações desenvolvedoras de software enquanto simultaneamente influenciadas pelas dualidades estratégicas: *exploration* e *exploitation* e na apropriação de metodologias

ágeis, dadas as condições ambidestras naturais destas organizações? - foi respondido neste estudo a partir dos construtos derivados das dualidades *exploitation-exploration* na administração de atividades analíticas e intuitivas, englobando as capacidades necessárias para competir em novos mercados e tecnologias. Ao mesmo tempo, outros construtos foram direcionados através do desdobramento das práticas e processos definidos pela metodologia ágil *Scrum*, como abordagem predominantemente influenciada em times de desenvolvimento de software.

Os objetivos específicos também foram alcançados:

- Foram identificados e caracterizados os elementos necessários a serem avaliados no modelo de maturidade, a partir das simultaneidades estratégicas *exploration-exploitation*, bem como suas aplicações, características e associações.
- Foram identificadas as dimensões da ambidestria metodológica, derivadas do *Scrum* e definidas as práticas específicas, que associadas a um modelo tradicional de desenvolvimento de software, puderam estabelecer marcos de desempenho organizacional.
- Foram utilizados os processos de conhecimento e assim, incluídos no modelo de maturidade para promover estruturalmente a ferramenta, com os respectivos níveis representados por escalas de maturidade.
- O modelo de maturidade constituído, verificado por cinco profissionais do mercado brasileiro, atuantes em empresas desenvolvedoras de software com formação e tempo de experiência consolidado. Os profissionais avaliaram os níveis de maturidade, os itens a serem avaliados, o *framework* e o mecanismo de autoavaliação, numa escala de 1 - insuficiente; 2- abaixo do esperado; 3- atende o esperado; 4- excede o esperado; 5- excelente. Para cada participante, foi ofertado a opinião individual acerca dos pontos fortes e pontos fracos do modelo de maturidade apresentado.

Conclui-se que o modelo de maturidade excede o esperado, com uma média global de 4,03. Foi descrito pelos profissionais como aplicável, interessante e escalável. Uma observação que vale ser destacada foi a avaliação da viabilidade por um profissional, sendo o único tópico com alguma classificação “abaixo do esperado”.

Com o modelo de maturidade ofertado aos profissionais e organizações o intuito é oferecer visibilidade, transparência em todos os ciclos, bem como a capacidade de adaptabilidade, herança das metodologias ágeis, por isso se insiste em um planejamento iterativo e não somente inicial, que atenda às mudanças, caso a caso. Valoriza-se o *feedback* contínuo, que associado ao planejamento contribuirá no atingimento dos objetivos, ajudando a focar nas entregas, contrapondo-se ao mero cumprimento do plano.

Portanto, essa extensa base teórica substancia a /eficácia e aplicabilidade do modelo de maturidade proposto, oferecendo uma ferramenta robusta e adaptável para profissionais e organizações.

## **5.2 Contribuições gerenciais**

Foi proposto um modelo conceitual para a avaliação da maturidade organizacional em empresas desenvolvedoras de software, utilizando metodologias ágeis. Foram levantados doze componentes desdobrados das dualidades estratégicas *exploitation-exploration* e doze práticas relevantes incorporadas ao modelo, na perspectiva de práticas ágeis do *framework Scrum*. Posteriormente, os conhecimentos foram reagrupados nas dimensões: *Exploration*, *Exploitation*, Orientação Estratégica e Agilidade. Foi delimitado quatro níveis de classificação de maturidade com a descrição destes níveis. Foi elaborado também o *framework* do modelo estudado com a inserção dos processos cíclicos de conhecimento, bem como a ferramenta de autoavaliação, caminho primordial para o diagnóstico. Além disso, o modelo pesquisado não foi encontrado em estudos anteriores relacionados a ambidestria organizacional. A partir desses resultados, espera-se agregar conhecimento ao campo empresarial, a organizações inovadoras e sua avaliação de maturidade.

Ainda, espera-se que os gestores possam extrair significativas contribuições para o campo empresarial, especialmente para organizações inovadoras, aprimorando assim sua capacidade de avaliação de maturidade e, por consequência, otimizando suas práticas estratégicas e operacionais.

### 5.3 Propostas para estudos futuros

Como proposta para estudos futuros, há inúmeros caminhos a serem seguidos, especialmente em campos de dualidades estratégicas contemporâneas. Após a oferta do modelo de maturidade, pode-se seguir os passos seguintes descritos por De Bruin *et al.* (2005): (iv) Teste, (v) Implementação e (vi) Manutenção.

A aplicação de um *framework* com priorização de ações pode ser feita para contribuir operacionalmente com os objetivos organizacionais. Pode-se empregar também um método quantitativo para aumentar o poder de generalização dos resultados. Finalmente, uma proposta interessante seria a aplicabilidade do modelo em organizações estrangeiras, de modo a conhecer as variáveis multiculturais que permeiam e eventualmente, são decisórias e influenciadoras aos modelos de maturidade. Acrescentar novas variáveis específicas a organização, como: número de clientes, quantidade de funcionários, tornam o modelo de maturidade mais competitivo para as organizações. Acrescentar então um tópico de avaliação e resultado da aplicabilidade do modelo em alguma organização pode ser um diferencial.

## Referências

- Adler, P. S., Goldoftas, B., & Levine, D. I. (1999). Flexibility versus efficiency? A case study of model changeovers in the Toyota production system. *Organization Science*, 10(1), 43-68. Recuperado de: <https://doi.org/10.1287/orsc.10.1.43>
- AJAGOPALAN, N. FINKELSTEIN, S. Effects of Strategic Orientation and Environmental Change of Senior Management Reward Systems. *Strategic Management Journal*; vol.13, p.127-142, 1992.
- ALAMI, O. M.; BOUKSOUR, O.; BEIDOURI, Z. An Intelligent Project Management Maturity Model for Moroccan Engineering Companies. *VIKALPA*, v.40, n. 2, p. 191– 208, 2015
- ALDAY, H.C.E. O Planejamento Estratégico dentro do Conceito de Administração Estratégica. *Revista FAE*, v.3, n.2, p.9-16, 2000.
- ALLENBACH, R. L.; HUFFMAN, J. E. Improving simulation engineering practices II - A management-by-deliverable approach for simulation project management. *International Journal of Industrial Engineering - Theory Applications and Practice*, v. 7, n. 2, p. 115-122, 2000
- ANDERSEN, E. S.; JESSEN, S. A. Project maturity in organizations. *International Journal of Project Management*, v. 21, p. 457-461. 2003
- AXELOS. Introduction to P3M3. 2016. Disponível em: . Acesso em: 13 jan. 2023
- BABBIE, E. *The Practice of Social Research*, 14. Ed, Boston: Cengage Learning, 2014.
- BACKLUND, F.; CHRONÉER, D.; SUNDQVIST, E. Project management maturity models – a critical review: a case study within Swedish engineering and construction organization. *Procedia-Social and Behavioral Sciences*, v. 119, p. 837–846, 2014.
- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). O Manifesto Ágil. Recuperado de: <http://agilemanifesto.org/iso/ptbr/principles.html>.

- BECKER, J.; KNACKSTEDT, R.; PÖPPELBUß, J. Developing Maturity Models for IT Management - A Procedure Model and its Application. Journal Business & Information Systems Engineering, Gabler Verlag, v. 1, n. 3, 2009.
- Beedle, M., Devos, M., Sharon, Y., Schwaber, K., & Sutherland, J. (1999). Scrum: An extension pattern language for hyperproductive software development. Pattern Languages of Program Design, 4, 1–18. Recuperado de : <http://www.torak.com/site/files/SCRUM>
- Benner, M. J., & Tushman, M. L. (2003). Exploitation, exploration, and process management: The productivity dilemma revisited. Academy of Management Review, 28(2), 238-256. Recuperado de: <https://doi.org/10.2307/30040711>
- Benner, M. J., & Tushman, M. L. (2015). Reflections on the 2013 decade award - "Exploitation, exploration, and process management: The productivity dilemma revisited" ten years later. Academy of Management Review, 40(4), 497-514.
- Birkinshaw J; Gibson, C. Building Ambidexterity Into an organization. MIT Sloan Management Review, p. 47-55, Summer 2004
- Birkinshaw, J., & Gupta, K. (2013). Clarifying the distinctive contribution of ambidexterity to the field of organization studies. Academy of Management Perspectives, 27(4), 287-298. Recuperado de: <https://doi.org/10.5465/amp.2012.0167>
- Boehm, B.; Turner, R. Balancing agility and discipline: A guide for the perplexed. Addison-Wesley, 2004
- Burgelman, R. A. Intraorganizational ecology of strategy making and organization adaption: theory and field research Organizations Science. V. 2, p.239-262, 1991
- CLELAND, D. I.; IRELAND, L. R. Project management: strategic design and implementation. 5th ed. New York, McGraw-Hill, 2007.
- Cockburn, A. Selecting a Project Methodology, IEEE Software, p. 64-71, July/August 2002
- Cordeiro, J. H. D. O. (2014). Ambidestria em empresas desenvolvedoras de software: Barreiras para adoção de metodologias ágeis e seu impacto na escolha do modelo organizacional (Tese de doutorado). Universidade de São Paulo, São Paulo, SP, Brasil.
- Cusumano, M. A. Japan software factories: a challenge to U.S management. Oxford University Press, 1991

- BRUIN, T. ROSEMANN, M. *et al.* Understanding the main phases of developing a maturity assessment model. Proceedings of the 16th Australasian conference on information systems (ACIS). Anais..., 2005. Sidney.
- DAY, G. S. The capabilities of market-driven organizations. *Journal of Marketing*, vol. 58, p.31-44, 1994.
- De Bruin, T., M. Rosemann, D. Bartmann, F. Rajola, J. Kallinikos, D. Avison, R. Winter, P. Ein-Dor, J. Becker, F. Bodendorf and C. Weinhardt (2005). "Towards a iSys: Revista Brasileira de Sistemas de Informação (iSys: Brazilian Journal of Information Systems) Recuperado de: <http://seer.unirio.br/index.php/isys/>
- Figueira, F., L. (2020). Proposta de um modelo de maturidade de gestão de projetos, que incorpora processos sistêmicos de gestão de conhecimento. Universidade Federal de Santa Catarina.
- FINCHER, A.; GINGER, L. The project management maturity model. Project Management Institute 28th Annual Seminar/Symposium, Chicago, IL, p. 48-55, 1997.
- Fleury, A. C. C; Vargas, N. N. Organização do Trabalho: Uma abordagem Interdisciplinar - Sete casos brasileiros para estudo. Editora Atlas, 1993
- Gibson, C. B., & Birkinshaw, J. (2004). The antecedents, consequences, and mediating role of organizational ambidexterity. *Academy of Management Journal*, 47(2), 209-226. Recuperado de: <https://doi.org/10.5465/20159573>
- Gupta, A. K., Smith, K. G., & Shalley, C. E. (2006). The interplay between exploration and exploitation. *Academy of Management Journal*, 49(4), 693-706. Recuperado de: <https://doi.org/10.2307/20159793>
- He, Z. L., & Wong, P. K. (2004). Exploration vs. exploitation: An empirical test of the ambidexterity hypothesis. *Organization Science*, 15(4), 481-494. Recuperado de: <https://doi.org/10.1287/orsc.1040.0078>
- Henri, J. F. (2006a). Management control systems and strategy: A resource-based perspective. *Accounting, Organizations and Society*, 31(6), 529-558. Recuperado de: <https://doi.org/10.1016/j.aos.2005.07.001>
- Hill, S. A., & Birkinshaw, J. (2014). Ambidexterity and survival in corporate venture units. *Journal of Management*, 40(7), 1899-1931. Recuperado de: <https://doi.org/10.1177/0149206312445925>
- HOFER, C. W.; SCHENDEL, D. Strategy Formulation: Analytic Concepts. St.Paul: West, 1978.

- Jeunon, E. Ester (2014). Indicadores de Desempenho na Gestão de Projetos Sociais Sustentáveis: Proposição de Modelo para os Centros Vocacionais Tecnológicos. *Revista Gestão & Tecnologia*, p. 232. Recuperado de: <http://www.spell.org.br/documentos/ver/32195/indicadores-de-desempenho-na-gestao-de-projetos-sociais-sustentaveis--proposicao-de-modelo-para-os-centros-vocacionais-tecnologico>
- Jørgensen, B., & Messner, M. (2009). Management control in new product development: The dynamics of managing flexibility and efficiency. *Journal of Management Accounting Research*, 21(1), 99-124. Recuperado de: <https://doi.org/10.2308/jmar.2009.21.1.99>
- Katila, R., & Ahuja, G. (2002). Something old, something new: A longitudinal study of search behavior and new product introduction. *Academy of Management Journal*, 45(6), 1183-1194. Recuperado de: <https://doi.org/10.2307/3069433>
- KERZNER, H. *Project management: a system approach to planning, scheduling and controlling*. New Jersey: John Wiley & Sons, 2009.
- KULKARNI, U.; LOUIS, R. S. Organizational self-assessment of knowledge management maturity. In: *Proceedings of the Americas Conference on Information Systems, AMCIS, 2003*.
- LAPPONI, J. C. *Estatística usando Excel*. São Paulo: Lapponi Treinamento e Editora, 2000.
- Larman, C., & Basili, V. R. (2003). IID History. *Computer*, 36(6), 47–56. Recuperado de: <http://ieeexplore.ieee.org/document/1204375/>
- LEE, H.; CHOI, B. Knowledge management enablers, processes and organizational performance: An integrative view and empirical examination. *Journal of Management Information Systems*, v. 20, n. 1, p. 179–228, 2003
- LINDVALL, M. *et al.* Agile software development in large organizations. *Computer*, v. 37, n. 12, p. 26+, Dec 2004.
- LLEWELLYN, S.; NORTHCOTT, D. The “singular view” in management case studies qualitative research in organizations and management. *An International Journal*, v. 2, n. 3, p. 194-207, 2007
- LYCETT, M. *et al.* Migrating agile methods to standardized development practice. *Computer*, v. 36, n. 6, p. 79+, Jun 2003
- Mafakheriab, F; Nasirib, F. Mousavi M. (2008). Project agility assessment: an integrated decision analysis approach. *Production Planning & Control Vol. 19*. p. 567-576.

- MALHOTRA, Naresh *et al.* Introdução à pesquisa de marketing. São Paulo: Prentice Hall, 2005
- March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, 2(1), 71-87. <https://doi.org/10.1287/orsc.2.1.71>
- Martini, A. (2015). *Ambidexterity in large-scale software engineering* (Doctoral dissertation), Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Swedish.
- MATTAR, F. N. Pesquisa de marketing: metodologia, planejamento. 6. ed. São Paulo: Atlas, 2005.
- MIKULENAS, G.; BUTLERIS, R.; NEMURAITĖ, L. An approach for the metamodel of the framework for a partial agile method adaptation. *Information Technology and Control*, v. 40, n. 1, p. 71-82, 2011.
- MILES, M.; HUBERMAN, A.; SALDAÑA, J. *Qualitative Data Analysis: a methods sourcebook*. 3 ed. Thousand Oaks: Sage Publication, 2013
- MORGAN, R.E; STRONG, C. A. Business Performance and Dimensions of Strategic Orientation. *Journal of Business Research*, vol.56, p.163-176, 2003.
- MORTENSEN, N. H.; HARLOU, U.; HAUG, A. Improving decision making in the early phases of configuration projects. *International Journal of Industrial Engineering Theory Applications and Practice*, v. 15, n. 2, p. 185-194, 2008.
- Mundy, J. (2010). Creating dynamic tensions through a balanced use of management control systems. *Accounting, Organizations and Society*, 35(5), 499-523. Recuperado de: <https://doi.org/10.1016/j.aos.2009.10.005>
- Napier, N. P., Mathiassen, L., & Robey, D. (2011). Building contextual ambidexterity in a software company to improve firm-level coordination. *European Journal of Information Systems*, 20(6), 674-690. Recuperado de: <https://doi.org/10.1057/ejis.2011.32>
- NEGASH, S.; GRAY, P. Business intelligence. In: BURSTEIN, F.; HOLSAPPLE, C. W. (Eds). *Handbook on decision support systems 2*. Berlin: Springer, p. 175-193, 2008.
- Nerur, S.; Mahapatra, R. & Mangalaraj, G. Challenges of migrating to Agile Methodologies. *Communication of the ACM*, v. 48, p. 73-78, 2005
- O'Reilly, C. A., & Tushman, M. L. (2013). Organizational ambidexterity: Past, present, and future. *Academy of Management Perspectives*, 27(4), 324-338. Recuperado de: <https://doi.org/10.5465/amp.2013.0025>

- O'Reilly, C. A., & Tushman, M. L. (2008). Ambidexterity as a dynamic capability: Resolving the innovator's dilemma. *Research in Organizational Behavior*, 28, 185-206. Recuperado de: <https://doi.org/10.1016/j.riob.2008.06.002>
- OGC. *Managing successful projects with Prince2* (Fifth ed.). Stationery Office Books, London, 2009
- PAULK, M. C.; CHARLES V. W.; MARY, B. C. *The Capability Maturity Model for Software*. Software Engineering Institute, 1996.
- PEREIRA, J. C. R. *Análise de Dados Qualitativos: Estratégias Metodológicas para as Ciências da Saúde, Humanas e Sociais*. 2. ed. São Paulo: Editora da Universidade de São Paulo, 1999.
- PINTO, J. A. C. *OPM3 Portugal: Assessing project management maturity on Portuguese organizations*. 2016. Recuperado de: [http://recipp.ipp.pt/bitstream/10400.22/8568/1/PTE\\_PintoJosé\\_2016.pdf](http://recipp.ipp.pt/bitstream/10400.22/8568/1/PTE_PintoJosé_2016.pdf)
- PMI. *The Standard for Portfolio Management*. 2. ed. Project Management Institute: Estados Unidos, 2013.
- Popadiuk, S., & Bido, D. D. S. (2016). Exploration, exploitation, and organizational coordination mechanisms. *Revista de Administração Contemporânea*, 20(2), 238-260.
- Porter, M.E. (1986). *Estratégia competitiva: Técnicas para analisar indústrias e competitividade* Rio de Janeiro: Campus.
- Porter, M. E. (1996). O que é estratégia. *Harvard Business Review*, 74(6), 61-78.
- Pressman, R. *Engenharia de Software* McGraw Hill, 6 edição, 2010
- Raisch, S., & Birkinshaw, J. (2008). Organizational ambidexterity: Antecedents, outcomes, and moderators. *Journal of Management*, 34(3), 375-381. Recuperado de: <https://doi.org/10.1177/0149206308316058>
- SEI - Software Engineering Institute. *CMMI® para Desenvolvimento – Versão 1.2*”, Carnegie Mellon University, 2006
- Senai. *Modelo Senai de Prospecção*, 2004.
- SILVA, H; MARCUS, V ; DORNELAS, J . Determinantes da não utilização de frameworks de gestão e/ou governança de TI. *Revista Gestão e Tecnologia*, v.

18, p. 294, 2018. Recuperado de: <https://doi.org/10.20397/2177-6652/2018.v18i2.1161>

Simsek, Z. (2009). Organizational ambidexterity: Towards a multilevel understanding. *Journal of Management Studies*, 46(4), 597-624. Recuperado de: <https://doi.org/10.1111/j.1467-6486.2009.00828.x>

SOLER, A. M. OPM3: A contribuição PMI para a maturidade em gestão de projetos. Curitiba: Revista Mundo PM, n. 3, 2005

Takeuchi, H., & Nonaka, I. (1986). The new product development game. *Harvard Business Review*, 64(1), 137-146 f.

Tonelli, A. O., Zambalde, A. L., Brito, M. J., & Bermejo, P. H. de S. (2016). A teoria da prática e o desenvolvimento de ambidesteridades no processo de inovação em software. *Brazilian Business Review*, 13(3), 26-49. Recuperado de: <http://doi.org/10.15728/bbr.2016.13.3.2>

TURNER, N.; MAYLOR, H.; SWART, J. Ambidexterity in projects: An intellectual capital perspective. *International Journal of Project Management*, v. 33, n. 1, p. 177–188, 2015.

Tushman, M. L., & O'Reilly, C. A. (1996). The ambidextrous organizations: Managing evolutionary and revolutionary change. *California Management Review*, 38(4), 8-30. Recuperado de: <http://doi.org/10.2307/41165852>

Venkatraman, N., Lee, C. H., & Iyer, B. (2006). Strategic ambidexterity and sales growth: A longitudinal test in the software sector. *Proceedings of the Annual Meetings of the Academy of Management Meetings*, Honolulu, Hawaii.

Vinekar, V., Slinkman, C. W., & Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management*, 23(3), 31-42. Recuperado de: <https://doi.org/10.1201/1078.10580530/46108.23.3.20060601/93705.4>

Whiteley, A., Pollack, J., & Matous, P. (2021) The origins of agile and iterative methods. *J Modern Project Management*, 8(3), 20-29.

WILLIAMS, T. Post project reviews to gain effective lessons learned. Newton Square, Pennsylvania: Project Management Institute, 2007

YAZICI, H. J. The Role of Project Management Maturity and Organizational Culture in Perceived Performance. *Project Management Journal*, v. 40, n. 3, p. 14-33, 2009.

Zimmermann, A., Raisch, S., & Birkinshaw, J. (2015). How is ambidexterity initiated? The emergent charter definition process. *Organization Science*, 26(4), 1119-1139. Recuperado de: <https://doi.org/10.1287/orsc.2015.0971>

ZANELLI, J. C. Pesquisa qualitativa em estudos da gestão de pessoas. *Estudos da Psicologia*, n. 7, p. 79-88, 2002.

ZWIKAEL, O.; LEVIN, G.; RAD, P. Top management support: the project friendly organization. *Cost Engineering*, v. 50, n. 9, p. 22-30, 2008.

[www.agilealliance.org](http://www.agilealliance.org) - Consultado em Dezembro de 2022

## Apêndices

### Apêndice A: Roteiro de entrevista - Termo de Compromisso de Confidencialidade

Convidamos a participar da pesquisa acadêmica que faz parte da dissertação de mestrado realizada na Fundação Pedro Leopoldo Educacional, intitulada: **Maturidade organizacional sob a ótica dos efeitos *exploration-exploitation* e ambidestria: Proposta de um Modelo conceitual aplicado a organizações desenvolvedoras de software**, desenvolvida pelo mestrando Henrique Rodrigues, sob a orientação da professora Dra . Ester Eliane Jeunon. A pesquisa tem por objetivo de definir, através de práticas específicas e genéricas, em escalas representativas, a maturidade das organizações desenvolvedoras de software, enquanto influenciadas pelas dualidades estratégicas: *exploration-exploitation* e na adoção de metodologias ágeis no processo de desenvolvimento de softwares.

Esta pesquisa não oferece qualquer risco ao participante. Além disso, não apresenta benefício individual direto aos participantes. As informações coletadas poderão contribuir para o meio científico.

Os resultados da pesquisa serão analisados e publicados, mas sua identidade não será divulgada, pois os pesquisadores garantem o sigilo e a confidencialidade dos dados coletados. Sua participação na pesquisa possui caráter voluntário e não lhe trará qualquer tipo de ônus ou remuneração.

Para qualquer outra informação, o(a) Sr.(a) poderá entrar em contato com o pesquisador Henrique Augusto Reis Rodrigues - Mestrando - [henrireirod@gmail.com](mailto:henrireirod@gmail.com)  
Agradecemos sua valiosa contribuição com informações para este estudo.

Eu fui informado sobre o que os pesquisadores querem fazer e porque precisam da minha colaboração. Estou ciente de que a minha contribuição não será remunerada e posso sair quando quiser.

( ) Aceito

## **Apêndice B - Instrumento de Coleta de Dados**

Objetivo: Analisar, através de práticas específicas e genéricas, em escalas representativas, a maturidade das organizações desenvolvedoras de software, enquanto influenciadas pelas dualidades estratégicas: *exploration-exploitation* e na adoção de metodologias ágeis no processo de desenvolvimento de softwares.

O instrumento é composto por três blocos com um total de 23 questões fechadas. O primeiro bloco solicita informações do respondente, sem a necessidade de identificação. O segundo se refere ao *framework* do modelo. O terceiro aos níveis de maturidade. O quarto se refere aos componentes do domínio, itens avaliados dentro do modelo de maturidade. O quinto é a ferramenta de autoavaliação e uma pergunta aberta para um melhor entendimento e julgamento da opinião do profissional.

### **Seção 1 - Informações dos respondentes**

#### **Dados**

1. Sexo

MasculinoFeminino

2. Faixa Etária

<35 anos36 a 41 anos42 a 47 anos48 a 53 anos54 a 59 anos> 59 anos

3. Qualificação Profissional

GraduaçãoEspecializaçãoCertificaçãoMestradoDoutorado

4. Tempo de experiência

< 5 anos6 a 10 anos11 a 20 anos> 21 anos

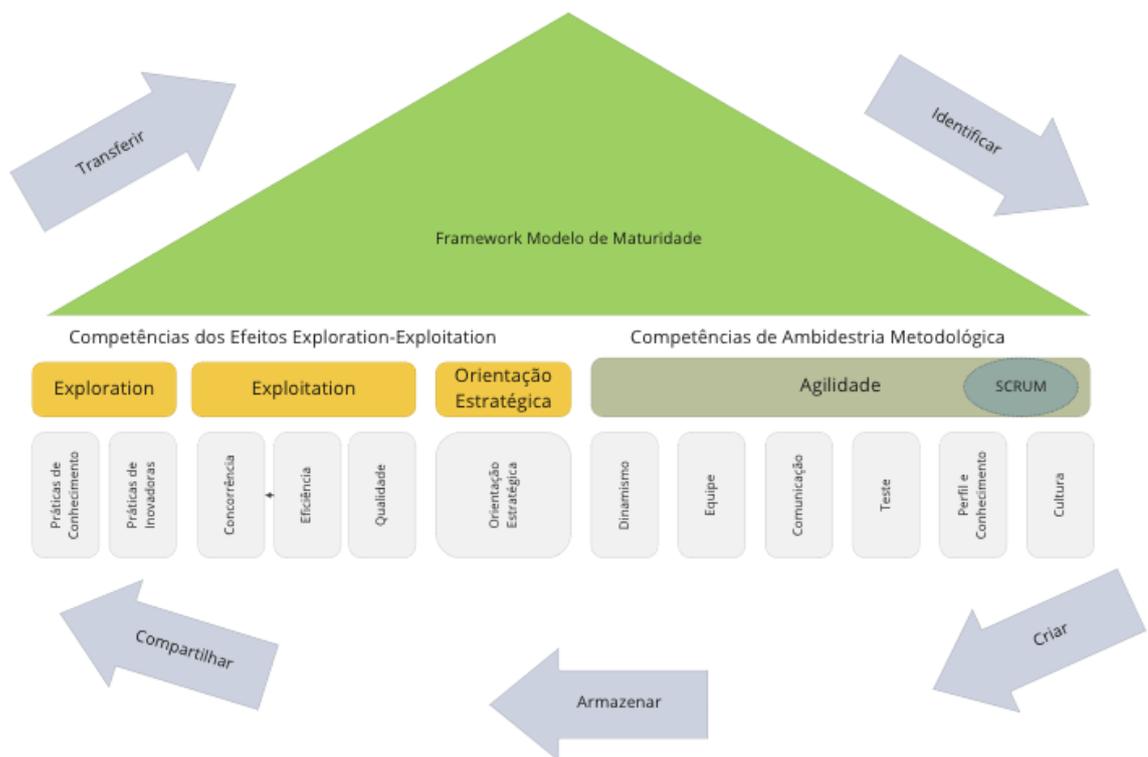
5. Diretamente envolvido com times de desenvolvimento de software?

SimNão

## Seção 2 - **Framework** do modelo de maturidade proposto

O *framework* do modelo proposto neste trabalho conta com todos os processos de conhecimento, atuando de forma perpendicular às perspectivas, gerando pontos de intersecção com as mesmas.

Em relação ao framework desenvolvido, você avalia como?



### **Escala:**

- 1 - Insuficiente,
- 2 - Abaixo do esperado
- 3 - atende o esperado
- 4 - excede o esperado,
- 5 - excelente.

### Descrição dos requisitos a serem analisados:

1 Aparência geral: Os itens estão bem estruturados, seguem uma sequência lógica.

2 Entendimento: Explicitado de forma clara, simples e inequívoca.

3 Viabilidade: É útil, permite ação de avaliação clara e precisa.

4 Itens contemplados: Cada item de identificação e avaliação é útil, não se confunde com os demais, não se repetem, permite resposta objetiva e atende ao objetivo proposto.

	1	2	3	4	5
Aparência geral	<input type="radio"/>				
Entendimento	<input type="radio"/>				
Viabilidade	<input type="radio"/>				
Itens contemplados	<input type="radio"/>				

### Seção 3 - Níveis de maturidade do modelo proposto

O modelo de maturidade proposto neste trabalho possui quatro níveis:

#### Nível - Modelo de maturidade proposto

1 - Adhoc

2 - Padronizado

3 - Gerenciado

4 - Otimizado

Cada nível de maturidade é descrito por uma série de características.

## **Nível**

### **1 - Adhoc**

A organização não tem uma abordagem estruturada para explorar novas oportunidades ou otimizar processos existentes. As práticas de *explotation* e *exploration* não são reconhecidas ou realizadas de maneira informal e não documentada. Cultura organizacional não suporta práticas ágeis

### **2 - Padronizado**

A organização começou a adotar práticas de *explotation* e *exploration*, mas elas são realizadas de maneira desorganizada e ainda não há uma estrutura formal para gerenciá-las. A organização pode estar experimentando metodologias ágeis, como o *Scrum*, em alguns projetos, mas não respeitando todos os eventos e ritos definidos pelo *framework*, nem com atribuições de papéis e responsabilidade definidos.

### **3 - Gerenciado**

A organização possui uma estrutura formal para gerenciar práticas de *explotation* e *exploration*. As metodologias ágeis, como o *Scrum*, são amplamente utilizadas na organização e existem processos documentados para apoiar a adoção dessas práticas. A organização também possui processos bem estabelecidos, mas não otimizados, para gerenciar projetos tradicionais.

### **4 - Otimizado**

A organização é altamente eficiente em gerenciar práticas de *explotation* e *exploration*. As metodologias ágeis são utilizadas em todos os projetos e a organização tem uma cultura forte de inovação e melhoria contínua. A organização também tem processos bem estabelecidos para gerenciar projetos tradicionais e é capaz de adotar novas práticas com rapidez e eficácia.

## Em relação aos níveis de maturidade, você os avalia como?

### Escala:

- 1 - Insuficiente,
- 2 - Abaixo do esperado
- 3 - atende o esperado
- 4 - excede o esperado,
- 5 - excelente.

### Descrição dos requisitos a serem analisados:

1 Aparência geral: Os itens estão bem estruturados, seguem uma sequência lógica.

2 Entendimento: Explicitado de forma clara, simples e inequívoca.

3 Viabilidade: É útil, permite ação de avaliação clara e precisa.

4 Itens contemplados: Cada item de identificação e avaliação é útil, não se confunde com os demais, não se repetem, permite resposta objetiva e atende ao objetivo proposto.

	1	2	3	4	5
Aparência geral	<input type="radio"/>				
Entendimento	<input type="radio"/>				
Viabilidade	<input type="radio"/>				
Itens contemplados	<input type="radio"/>				

## **Seção 4 - Componentes do domínio - *Exploitation, Exploration, Orientação Estratégica e Agilidade***

Os componentes do domínio são fatores avaliados dentro do modelo de maturidade, onde se procura mensurar um conjunto de características. Sendo assim, o modelo proposto então mensura para cada perspectiva questões relacionadas à organização e processos, respeitando as atividades do conhecimento para cada nível de maturidade.

### **Dimensão Exploration - Práticas de Conhecimento**

- Volume de novas ideias
- Utilização de novas fontes de conhecimento
- Conhecimento existente em base de dados
- Compartilhamento de conhecimento interno
- Processo individuais de aprendizagem
- Processo de aprendizagem coletiva
- Formação de capacidades de equipe
- Intensidade do desenvolvimento pessoal
- Valorização do conhecimento individual

### **Dimensão Exploration - Práticas inovadoras**

- Concentração de produtos ou processos novos
- Desenvolvimento de protótipos
- Taxa de inovação de produto
- Inovação de técnica de Marketing
- Abertura de novos canais de distribuição
- Concentração em inovações radicais nos produtos
- Concentração em inovações tecnológicas radicais
- Busca incessante por novos mercados
- Desenvolvimento de novos produtos e serviços
- Participação agressiva em novas alianças baseadas em tecnologia

### **Dimensão *Exploitation* - Concorrência**

- Aparecimento de novos concorrentes
- Existência de produtos ou processos substitutos
- Concorrência no mercado local
- Concorrência no setor empresarial
- Existência de guerra de promoção no setor
- Concorrência abrange oferta facilmente

### **Dimensão *Exploitation* - Eficiência**

- Criação de rotinas detalhadas
- Importância da eficiência
- Foco na realização das atividades
- Preocupações com ganhos de escala
- Mecanismo de controle organizacional
- Foco nos custos
- Foco inclinado para a produção

### **Dimensão *Exploitation* - Qualidade**

- Equipe de testes dedicada
- Controle de qualidade estabelecido
- Controle de qualidade compartilhado
- Preocupação com a qualidade dos processos
- Preocupação com a qualidade dos produtos
- Foco na qualidade

### **Dimensão Orientação Estratégica**

- Visão Estratégica focada no presente
- Estratégias focadas no curto prazo

### **Dimensão Agilidade - Dinamismo**

- Habilidade de mudar requisitos

- Entregas frequentes
- Softwares funcionais
- Ritos definidos - Inspeção, Adaptação e Transparência
- Iteração contínua
- Geração de artefatos

#### **Dimensão Agilidade - Equipe**

- Definição de atribuições e papéis
- Atuação semiautônoma e autoadministrada
- Participação ativa e colaborativa

#### **Dimensão Agilidade - Comunicação**

- Decisões rápidas
- Papel do Cliente
- Redução da documentação

#### **Dimensão Agilidade - Teste**

- Frequência de testes

#### **Dimensão Agilidade - Perfil e conhecimento dos desenvolvedores**

- Senioridade da equipe

#### **Dimensão Agilidade - Cultura**

- Alto grau de liberdade para qualquer um da equipe

Em relação às dimensões e respectivas perspectivas (componentes), você os avalia como?

**Escala:**

- 1 - Insuficiente
- 2 - Abaixo do esperado
- 3 - atende o esperado
- 4 - excede o esperado,
- 5 - excelente.

**Descrição dos requisitos a serem analisados:**

1 Aparência geral: Os itens estão bem estruturados, seguem uma sequência lógica.

2 Entendimento: Explicitado de forma clara, simples e inequívoca.

3 Viabilidade: É útil, permite ação de avaliação clara e precisa.

4 Itens contemplados: Cada item de identificação e avaliação é útil, não se confunde com os demais, não se repetem, permite resposta objetiva e atende ao objetivo proposto.

	1	2	3	4	5
Aparência geral	<input type="radio"/>				
Entendimento	<input type="radio"/>				
Viabilidade	<input type="radio"/>				
Itens contemplados	<input type="radio"/>				

## Seção 5 - Ferramenta de Autoavaliação

As informações a seguir possuem as questões de autoavaliação propostas para o modelo de maturidade organizacional em todas as doze perspectivas e processo de conhecimento.

*Dimensão: Exploration*

Componente: Práticas de Conhecimento

- Participamos de um ambiente propício à geração de novas ideias, com processos e incentivos para que os colaboradores possam contribuir ativamente com sugestões de melhorias, inovações e soluções criativas. **Há cultura de valorização e um volume de ideias significativas na organização.**
- Temos um nível de concordância em relação a utilização de novas fontes de conhecimento para a tomada de decisão, embora não seja uma prática comum. **A organização tem intenção de fomentar o uso de novas fontes de conhecimento.**
- Utilizamos o uso de conhecimento existente para melhorar os processos, mas nem sempre temos as melhores informações atualizadas. **Existe uma abordagem de armazenar e usar o conhecimento, mas não há fomento na manutenção de novos processos.**
- Incentivamos o compartilhamento do conhecimento interno. **As atividades de conhecimento interno são enfim registradas e armazenadas e de certa forma disponíveis a todos.**
- Estimulamos o aprendizado individual, embora não seja possível controlar o nível de conhecimento. **Não utiliza-se métricas para o aumento da produtividade.**
- Estimulamos o aprendizado coletivo, embora não seja possível controlar o nível de conhecimento. **Não utiliza-se métricas para o aumento da produtividade.**

- Temos programas e incentivos para o desenvolvimento de habilidades e conhecimentos específicos. **Há intenção de gerenciar o conhecimento e melhoria contínua de produtividade,**

#### Componente: Práticas Inovadoras

- Atendemos melhor às necessidades dos clientes, devido a grande concentração de produtos enquanto a concentração de processos nos permite operar de maneira mais eficiente e eficaz. **Há uma consciência organizacional competitiva sobre o equilíbrio entre produtos e atendimento**
- Utilizamos ferramentas e técnicas de inovação aberta para apoiar o desenvolvimento de protótipos. **Há uma abordagem comum incorporada na organização para promover a inovação de novos produtos**
- Preocupa-se com a promoção da inovação de produtos, garantindo uma cultura de criatividade e aprendizado contínuo em todos os níveis. **Práticas estabelecidas para compartilhamento e uso do conhecimento é parte integrante da organização e continuamente melhorada.**
- Temos inovado em relação às novas tendências e técnicas de marketing. **Há consciência da necessidade de inovação e abertura de penetração no mercado**
- Temos aberto novos canais de distribuição em relação aos canais existentes. **Há intenção em desenvolver e gerenciar o equilíbrio estratégico.**
- Há um incentivo na inovação radical sem comprometer a base de clientes atual. **Reconhecimento da necessidade de inovar otimizando o desempenho na manutenibilidade dos clientes.**
- Há um incentivo na inovação tecnológica radical sem comprometer os produtos e serviços existentes. **Reconhecimento da necessidade de inovação tecnológica integrada com o portfólio de produtos atual.**

- Preocupamos em buscar novos mercados de atuação. **Necessidade de expansão e diversificação de fontes de receita e dependência de mercados únicos.**
- Preocupamos em formar novas alianças estratégicas baseadas em tecnologias. **Necessidade de maior eficiência operacional e competitividade**

### *Dimensão Exploitation*

Componente: Concorrência

- Monitoramos o surgimento de novos concorrentes em seu mercado-alvo. **Necessidade de identificação de possíveis ameaças e oportunidades de negócio.**
- Monitoramos continuamente o mercado em busca de possíveis produtos ou serviços mais eficientes e que potencialmente podem impactar nos produtos e processos atuais. **Adaptação às novas tendências, tecnologias e aumentar a eficiência dos produtos existentes.**
- Monitoramos continuamente a concorrência no mercado local. **Necessidade de ajustar estratégias e táticas do negócio**
- Avaliamos com frequência o resultado das ações promocionais em relação aos concorrentes. **Necessidade de ajustar estratégias de marketing e identificação de vantagens competitivas.**

Componente: Eficiência

- Buscamos garantir a conformidade com normas e regulamentações das rotinas e processos atuais. **Manutenção da confiabilidade e integridade dos negócios e processos.**
- Avaliamos através de indicadores de desempenho a eficiência da empresa. **Identificação de possíveis oportunidades de melhoria.**

- Preocupamos em equilibrar a busca pela redução de custos com a necessidade de manter ou melhorar a qualidade dos produtos/serviços oferecidos. **Necessidade manter a rentabilidade da empresa e aumentar sua capacidade de investimento em novos projetos**
- Preocupamos em buscar constantemente melhorias nos processos produtivos. **Avaliação contínua dos processos existentes para identificar oportunidades de otimização e melhoria**

Componente: Qualidade

- Existe uma equipe de testes dedicada para garantir a qualidade do produto. **Garantir a qualidade do produto e a satisfação do cliente, além de melhorar a reputação da empresa**
- Há uma preocupação em relação a conformidade nos processos e padrões de qualidade. **Garantir a qualidade do produto ou serviço oferecido, mas também a proteger a empresa de possíveis consequências legais**

Dimensão: Orientação Estratégica

Componente: Orientação Estratégica

- A visão estratégica é focada no presente. **Adaptação às mudanças do mercado e mantenha uma vantagem competitiva**
- Estratégias focadas no curto prazo. **Capacidade de lidar com questões imediatas, como a necessidade de aumentar as vendas, melhorar a eficiência operacional ou reduzir custos**

Dimensão: Agilidade

Componente: Dinamismo

- Adotamos metodologias ágeis nos processos de desenvolvimento de software, você diria que o time de desenvolvimento tem habilidades em alterar requisitos

de forma frequente. **Alta capacidade de responder às mudanças, em vez de seguir um plano rígido e inflexível.**

- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza entregas frequentes. **Entrega iterativa e incremental de funcionalidades do software, de forma que o cliente possa ter acesso a versões do produto que já tenham algum valor.**
- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza entregas funcionais de software. **Objetivo é entregar uma versão do software com funcionalidades básicas, mas que possa ser utilizada pelo cliente para obter algum benefício ou resultado.**
- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza os eventos ( Release Planning, Sprint, sprint planning, sprint review, retrospective e daily *scrum*). **Necessidade de garantir que o processo de desenvolvimento seja iterativo e incremental, com entregas frequentes de software funcional.**
- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento realiza iterações frequentes. **Durante cada sprint, o time de desenvolvimento trabalha em um conjunto de atividades previamente definidas, que incluem a análise de requisitos, o desenvolvimento de funcionalidades, testes e integração.**
- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento utiliza-se dos artefatos (product backlog, release burndown, sprint backlog, sprint burndown e o DoD) em cada sprint. **Necessidade de alinhamento em relação aos objetivos do projeto, que todas as funcionalidades estejam claramente definidas e priorizadas, e que o progresso do projeto possa ser medido e monitorado**

Componente: Equipe

- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento possui as atribuições e papéis (*Scrum Master*, *Product Owner*, Desenvolvedores de Software) estabelecidos? **Necessidade de garantir que cada membro do time de desenvolvimento tenha**

**responsabilidades claras e que as decisões sejam tomadas de forma colaborativa e transparente**

- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento possui atuação semiautônoma e auto administrada. **Liberdade de tomar decisões e gerenciar o próprio trabalho, em vez de depender de direções e ordens hierárquicas**
- Adotamos metodologias ágeis nos processos de desenvolvimento de software, o time de desenvolvimento tem participação ativa e colaborativa. **Necessidade de compartilhamento de conhecimentos e habilidades entre os membros da equipe, além da participação em reuniões diárias, revisões e retrospectivas.**

Componente: Comunicação

- Adotamos metodologias ágeis nos processos de desenvolvimento de software, embora temos alta frequência de mudanças de prioridades em projetos em andamento. **A frequência de mudanças de prioridades pode variar dependendo do projeto e das necessidades do cliente, mas não deve ser uma prática comum.**
- Adotamos metodologias ágeis nos processos de desenvolvimento de software, a equipe de desenvolvimento tem autonomia para realizar alterações no produto com base no *feedback* do cliente. **Liberdade para cada membro da equipe contribuir com ideias e soluções, é possível obter resultados melhores e mais inovadores**
- Adotamos metodologias ágeis nos processos de desenvolvimento de software, com qual frequência a equipe de desenvolvimento revisa e valida os artefatos produzidos sem documentação detalhada. **Necessidade de validação dos artefatos é contínuo e realizado de forma iterativa e incremental durante as sprints.**

Componente: Testes

- Adotamos metodologias ágeis nos processos de desenvolvimento de software, e com frequência a equipe de desenvolvimento realiza os testes dos potenciais entregáveis. **Necessidade de garantir que o software esteja sempre funcionando e atendendo aos requisitos definidos**

Componente: Perfil e Conhecimento

- Adotamos metodologias ágeis nos processos de desenvolvimento de software, e com frequência a equipe atua utilizando alta senioridade dos desenvolvedores. **Importante que a equipe tenha uma combinação adequada de habilidades e conhecimentos, com membros experientes**

Componente: Cultura

- Entendemos a importância do nível de concordância em relação ao alto grau de liberdade a todos da equipe. **Importante que haja um nível de concordância em relação aos objetivos e metas a serem alcançados, para que a equipe trabalhe de forma unida e em direção ao mesmo propósito.**

Em relação ao questionário de avaliação, você avalia como as atualizações realizadas (em negrito)?

**Escala:**

- 1 - Insuficiente,
- 2 - Abaixo do esperado
- 3 - atende o esperado
- 4 - excede o esperado,
- 5 - excelente.

**Descrição dos requisitos a serem analisados:**

1 Aparência geral: Os itens estão bem estruturados, seguem uma sequência lógica.

2 Entendimento: Explicitado de forma clara, simples e inequívoca.

3 Viabilidade: É útil, permite ação de avaliação clara e precisa.

4 Itens contemplados: Cada item de identificação e avaliação é útil, não se confunde com os demais, não se repetem, permite resposta objetiva e atende ao objetivo proposto.

	1	2	3	4	5
Aparência geral	<input type="radio"/>				
Entendimento	<input type="radio"/>				
Viabilidade	<input type="radio"/>				
Itens contemplados	<input type="radio"/>				

De acordo com o modelo de maturidade avaliado, descreva os pontos fortes e os pontos fracos.

