

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Danielle Gonçalves Barbosa

**Gestão de Projetos em Empresas de TI: Um  
Estudo de Caso sobre a Junção dos Papéis de  
Scrum Master e Product Owner**

**Uberlândia, Brasil**

**2023**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Danielle Gonçalves Barbosa

**Gestão de Projetos em Empresas de TI: Um Estudo de  
Caso sobre a Junção dos Papéis de Scrum Master e  
Product Owner**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Ronaldo Castro de Oliveira

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2023

Danielle Gonçalves Barbosa

# **Gestão de Projetos em Empresas de TI: Um Estudo de Caso sobre a Junção dos Papéis de Scrum Master e Product Owner**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 01 de novembro de 2023:

---

**Ronaldo Castro de Oliveira**  
Ronaldo Castro de Oliveira

---

**Paulo Henrique Ribeiro Gabriel**

---

**Fabiano Azevedo Dorça**

Uberlândia, Brasil  
2023

*Este trabalho é dedicado à minha família e à minha namorada por sempre acreditarem em mim e me apoiarem em todas as minhas escolhas.*

# Agradecimentos

Gostaria de agradecer aos meu pais que sempre foram a base para que eu me torna-se o que sou hoje. Obrigada por sempre estarem ao meu lado em tudo.

Agradeço também minha namorada que me incentiva a ser melhor e acredita em todas as minhas conquistas.

E agradeço também o meu orientador Ronaldo por todo conhecimento compartilhado e por ter me ajudado tanto nessa última fase da minha graduação.

# Resumo

O presente trabalho dedica-se a investigar a junção dos papéis de Scrum Master (Facilitador do projeto) e Product Owner (Dono do produto) dentro do processo de desenvolvimento de software em equipes que adotam a metodologia Scrum. O propósito principal é avaliar como essa junção afeta a eficiência na gestão de projetos e no dia a dia da equipe. Para isso, realiza-se uma análise da aplicação de metodologias tradicionais e ágeis de gerenciamento de projetos, como PMBOK, AgilePM, RUP e Scrum, além de explorar as funções mais comuns encontradas na maioria das equipes de desenvolvimento, e os papéis e responsabilidades específicas de cada metodologia estudada. Através de um estudo de caso, esse trabalho visa investigar a fusão dos papéis de Scrum Master e Product Owner, utilizando de percepções de um profissional que desempenha ambas as funções, e com base em dados coletados de uma pesquisa quantitativa envolvendo 38 profissionais da área. Ao final, espera-se elucidar as vantagens e desvantagens dessa junção de papéis do Scrum Master e Product Owner, enfatizando que a efetividade depende das características específicas de cada projeto e de uma implementação cuidadosa e adaptativa.

**Palavras-chave:** Scrum, Junção, Papéis, Gestão, Metodologias.

# Lista de ilustrações

Figura 1 – Processos PMOK . . . . .	17
Figura 2 – Métodos tradicionais x Métodos ágeis . . . . .	20
Figura 3 – Fluxos de trabalho do RUP . . . . .	26
Figura 4 – Metodologia Scrum . . . . .	27
Figura 5 – Metodologia Kanban . . . . .	31
Figura 6 – Papeis no AgilePM . . . . .	42
Figura 7 – Na sua opinião, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza das definições do projeto? . . . . .	56
Figura 8 – Para você, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza nas prioridades do projeto? . . . . .	57
Figura 9 – Na sua opinião, a junção desses papéis do Scrum Master e do Product Owner influencia negativamente na disponibilidade do Scrum Master para auxiliar a equipe no dia a dia do projeto? . . . . .	57
Figura 10 – Você acredita que a junção dos papéis do Scrum Master e do Product Owner afeta negativamente a qualidade das entregas da equipe? . . . . .	58
Figura 11 – Na sua percepção, a junção dos papéis de Scrum Master e Product Owner afeta negativamente a comunicação dentro da equipe? . . . . .	59
Figura 12 – Resposta 1 - Qual é sua área de formação? . . . . .	70
Figura 13 – Resposta 2 - Qual é o seu cargo atual? . . . . .	71
Figura 14 – Resposta 3 - Quanto tempo você tem atuado na área de TI? . . . . .	71
Figura 15 – Resposta 4 - Em sua opinião somente o Scrum Master é responsável por orientar a equipe e remover os impedimentos do processo de desenvolvimento? . . . . .	72
Figura 16 – Resposta 5 - Para você, é tarefa somente do Scrum Master definir as prioridades do projeto? . . . . .	72
Figura 17 – Resposta 6 - Na sua visão, é de responsabilidade somente do Scrum Master facilitar a comunicação entre a equipe e o cliente dentro do projeto? . . . . .	73
Figura 18 – Resposta 7 - Para você apenas o Scrum Master tem a autoridade final para tomar decisões sobre as prioridades do backlog do projeto? . . . . .	73
Figura 19 – Resposta 8 - Para você apenas o Scrum Master deve atuar como coach da equipe, apoiando e removendo obstáculos quando necessário? . . . . .	74
Figura 20 – Resposta 9 - Na sua opinião somente o Product Owner é responsável por representar os interesses e necessidades do cliente dentro do projeto? . . . . .	74
Figura 21 – Resposta 10 - Para você, é tarefa somente do Product Owner definir as prioridades do projeto? . . . . .	75

Figura 22 – Resposta 11 - Você acredita que é papel exclusivamente do Product Owner a definição dos itens do Product Backlog do projeto? . . . . .	75
Figura 23 – Resposta 12 - Na sua visão, é de responsabilidade somente do Product Owner facilitar a comunicação entre a equipe e o cliente dentro do projeto? . . . . .	76
Figura 24 – Resposta 13 - Para você, a presença de um Product Owner que seja do cliente e não membro do time, é benéfico para o projeto? . . . . .	76
Figura 25 – Resposta 14 - Caso esse Product Owner do cliente não tenha conhecimento sobre o processo de desenvolvimento de software, isso pode atrapalhar o andamento do projeto na sua visão? . . . . .	77
Figura 26 – Resposta 15 - Na sua opinião, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza das definições do projeto? . . . . .	77
Figura 27 – Resposta 16 - Para você, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza nas prioridades do projeto? . . . . .	78
Figura 28 – Resposta 17 - A capacidade da equipe de receber orientações claras sobre as tarefas a serem desenvolvidas é afetada negativamente pela junção de papéis do Scrum Master e do Product Owner na sua visão? . . . . .	78
Figura 29 – Resposta 18 - Você acredita que a junção dos papéis do Scrum Master e do Product Owner afeta negativamente a qualidade das entregas da equipe? . . . . .	79
Figura 30 – Resposta 19 - Para você, essa junção de papéis do Scrum Master e do Product Owner influencia negativamente a capacidade da equipe de lidar com mudanças nos requisitos do projeto? . . . . .	79
Figura 31 – Resposta 20 - Na sua percepção, a junção dos papéis de Scrum Master e Product Owner afeta negativamente a comunicação dentro da equipe? . . . . .	80
Figura 32 – Resposta 21 - Na sua opinião, a junção desses papéis do Scrum Master e do Product Owner influencia negativamente na disponibilidade do Scrum Master para auxiliar a equipe no dia a dia do projeto? . . . . .	80
Figura 33 – Resposta 22 - Na sua visão, com a junção desses papéis de Scrum Master e Product Owner o time pode fazer entregas mais alinhadas com a expectativa do cliente? . . . . .	81

# Lista de tabelas

Tabela 1 – Responsabilidades do Scrum Master x Product Owner . . . . .	51
Tabela 2 – Possível junção de papéis em equipes de desenvolvimento de software .	52
Tabela 3 – Vantagens e desvantagens da junção de papéis do Scrum Master e Product Owner . . . . .	55

# Lista de abreviaturas e siglas

SM	Scrum Master
PO	Product Owner
XP	Extreme Programming
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
AgilePM	Agile Project Management
UX/UI	User Experience/ User Interface
RUP	Rational Unified Process

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	Contextualização	12
1.2	Justificativa	13
1.3	Objetivos gerais e específicos	13
1.4	Trabalhos Correlatos	14
1.5	Organização do Trabalho	14
<b>2</b>	<b>GESTÃO DE PROJETOS EM TI</b>	<b>16</b>
2.1	Gestão de projetos com PMBOK	16
2.2	Métodos ágeis	19
2.2.1	Gestão de projetos com Agile Project Management (AgilePM)	21
2.2.2	Gestão de projetos com RUP (Rational Unified Process)	24
2.2.2.1	Organização do Trabalho no RUP	24
2.2.3	Gestão de projetos com Scrum	27
2.2.3.1	Papéis do Scrum	28
2.2.3.2	Artefatos do Scrum	28
2.2.3.3	Ritos do Scrum	29
2.2.4	Gestão de projetos com Kanban	30
<b>3</b>	<b>PAPÉIS DENTRO DAS METODOLOGIAS DE GERENCIAMENTO DE PROJETOS</b>	<b>33</b>
3.1	Papéis comuns em equipes de desenvolvimento de software	33
3.2	Papéis dentro do PMBOK	37
3.3	Papéis dentro do AgilePM (Agile Project Management)	39
3.4	Papéis dentro RUP (Rational Unified Process)	41
3.5	Papéis dentro do método Scrum	44
<b>4</b>	<b>ESTUDO DE CASO</b>	<b>48</b>
4.1	Contexto	48
4.2	A Junção de Papéis	50
4.3	Perspectiva do Scrum Master	53
4.4	Pesquisa	56
<b>5</b>	<b>CONCLUSÃO</b>	<b>60</b>
	<b>REFERÊNCIAS</b>	<b>62</b>

<b>ANEXOS</b>	<b>66</b>
<b>ANEXO A – PESQUISA PARA AVALIAÇÃO DA JUNÇÃO DE PA- PÉIS DO SCRUM MASTER E DO PRODUCT OW- NER . . . . .</b>	<b>67</b>
<b>ANEXO B – RESPOSTAS DA PESQUISA PARA AVALIAÇÃO DA JUNÇÃO DE PAPÉIS DO SCRUM MASTER E DO PRODUCT OWNER . . . . .</b>	<b>70</b>

# 1 Introdução

## 1.1 Contextualização

Em um mundo onde a tecnologia e as necessidades empresariais estão em constante evolução, a gestão de projetos tornou-se uma disciplina essencial para garantir a entrega bem-sucedida de soluções eficazes. Os últimos anos têm testemunhado uma profunda transformação na maneira como os projetos são gerenciados, com uma clara divisão entre abordagens tradicionais e ágeis de gerenciamento, onde cada método possui suas próprias características, vantagens e desvantagens, com diferentes papéis e responsabilidades para os profissionais envolvidos. Apesar das características específicas de cada metodologia, a gestão dentro das equipes é uma parte essencial desse processo, principalmente em um ambiente ágil (COHN, 2005), onde a colaboração, a comunicação e a adaptação são fundamentais para o sucesso do projeto.

Dessa forma, as metodologias ágeis vêm ganhando cada vez mais espaço no mercado, sendo amplamente praticados por empresas de diversos tamanhos e setores (SUTHERLAND, 2014)(VERSIONONE.COM, 2017). Com isso o Scrum emerge como uma das metodologias ágeis mais adotadas para enfrentar os desafios inerentes a esses projetos (RAMOS; JUNIOR, 2017), já que esse possui uma estrutura simples e eficaz para gerenciar projetos de desenvolvimento de software.

Uma questão importante que precisa ser explorada dentro da gestão de projetos sob a perspectiva de diferentes metodologias, tanto tradicionais quanto ágeis, são os papéis e responsabilidades dos membros do time, com o intuito de minimizar possíveis conflitos e maximizar a eficácia da equipe (KNIBERG, 2007).

Um fenômeno observado em algumas equipes que utilizam o método Scrum é a fusão de papéis, mais especificamente do Scrum Master e do Product Owner. Embora esses papéis tenham sido originalmente concebidos com responsabilidades distintas, a realidade prática em muitas organizações levou a uma sobreposição de funções. O Scrum Master é originalmente o facilitador da equipe (SABBAGH, 2014), ele é responsável por garantir que a equipe adote os valores e princípios ágeis e que o processo de desenvolvimento ocorra sem problemas. Já o Product Owner é responsável por definir e priorizar o backlog do produto e representar e defender os interesses do cliente dentro da equipe(SVERRISDOTTIR; INGASON; JONASSON, 2014). A junção ou sobreposição desses papéis pode trazer impactos significativos para a dinâmica da equipe e para o desenvolvimento do projeto. Assim, uma investigação aprofundada sobre as implicações dessa combinação é crucial para entender suas potenciais consequências, positivas ou negativas, e como elas podem

afetar a eficácia da equipe e a entrega do projeto.

Este trabalho visa aprofundar-se na gestão de projetos de desenvolvimento de software, com foco nos papéis e responsabilidades dentro das equipes e especificamente na metodologia Scrum. Será dada particular atenção à fusão dos papéis do Scrum Master e do Product Owner, visando examinar os impactos inerentes a essa abordagem. Para isso, será construído uma base teórica sobre o assunto de gestão de projetos, metodologias tradicionais e ágeis e os papéis encontrados em cada um delas. A pesquisa não se limitará à teoria, o estudo de caso detalhará uma vivência real da autora em um ambiente onde essa junção de papéis foi adotada, explorando o cenário do projeto, desafios e benefícios encontrados, tal como os resultados alcançados. Adicionalmente, será conduzida uma pesquisa quantitativa onde profissionais da área serão consultados sobre suas perspectivas e experiências relativas ao tema.

A expectativa é que, ao término deste trabalho, seja possível proporcionar uma compreensão aprofundada acerca da viabilidade da junção desses papéis no Scrum e oferecer recomendações práticas para equipes e organizações que possam estar considerando ou já estejam enfrentando essa realidade.

## 1.2 Justificativa

A realização deste estudo é justificada pela necessidade de compreender as implicações da junção dos papéis de Scrum Master e Product Owner nas equipes de desenvolvimento de software, uma realidade cotidiana vivenciada pela autora nos projetos sob sua gestão. Esta prática de fundir os dois papéis é uma estratégia adotada pela empresa onde os projetos são executados, tornando-se comum nessa organização devido às suas peculiaridades estruturais e operacionais. Espera-se que o estudo contribua para um melhor entendimento de como essa sobreposição de responsabilidades pode impactar a dinâmica das equipes, a eficácia na gestão dos projetos, bem como a qualidade final dos produtos entregues.

## 1.3 Objetivos gerais e específicos

Esse trabalho tem como objetivo explorar as implicações práticas, desafios e benefícios da junção dos papéis de Scrum Master e Product Owner em equipes de desenvolvimento de software, além de avaliar a viabilidade dessa fusão e fornecer orientações estratégicas para otimizar a execução deste papel unificado.

Ademais, este trabalho almeja enriquecer o entendimento das melhores práticas no gerenciamento ágil, oferecendo percepções valiosas para profissionais e organizações que aspiram a otimizar seus processos e estruturas de equipe no desenvolvimento de software.

## 1.4 Trabalhos Correlatos

Analisar as metodologias de gestão de projetos e os papéis envolvidos dentro das equipes de desenvolvimento de software é de grande importância. Esta seção explora trabalhos relacionados com temas ou abordagens semelhantes as que são analisadas neste trabalho.

Como no artigo (LIMA; AMARAL, 2022), que focou em identificar quais membros de equipes ágeis assumem responsabilidades do Product Owner em projetos onde este papel está ausente ou tem participação reduzida. Através de uma revisão bibliográfica sistemática, esse estudo compilou uma lista abrangente de tarefas atribuídas ao Product Owner, que serviu de base para um estudo de caso em uma empresa, onde foram analisadas três equipes de projeto. Duas destas equipes operavam sem um PO designado, enquanto a terceira possuía um Product Owner nomeado, porém inativo. O estudo em questão forneceu percepções valiosas ao revelar quais profissionais absorveram as responsabilidades do PO e quais tarefas foram negligenciadas em decorrência de sua ausência.

Outro trabalho que realizou uma análise semelhante foi (EREIZ; MUSIC, 2019), que em seu artigo exploraram um estudo de caso de um projeto implementado com o método Scrum, com todos os ritos, práticas e ferramentas, mas sem a presença de um Scrum Master dedicado. Essa pesquisa revelou que projetos desprovidos de um Scrum Master tendem a enfrentar maiores riscos de falha. No entanto, eles também observaram que, em certas circunstâncias, uma equipe suficientemente madura pode se auto-organizar, permitindo que um de seus membros assuma as funções de um Scrum.

E por fim podemos relacionar o trabalho de (KOLESNIKOVA KATERYNA E LUKIANOV, 2020), que buscou analisar as implicações da combinação dos papéis de Scrum Master e Product Owner. Essa pesquisa chegou a conclusão de que tal combinação equivale à criação de um "soldado universal", uma figura capaz de desempenhar simultaneamente múltiplos papéis complexos, o que segundo o estudo, é impraticável na prática.

O presente trabalho possui algumas semelhanças com os citados nessa seção, focando na análise dos impactos gerados pela junção de papéis do Scrum Master e do Product Owner em equipes de desenvolvimento de software.

## 1.5 Organização do Trabalho

Este trabalho é organizado em cinco capítulos estruturados e oferece uma visão geral sobre o tema de gestão de pessoas e métodos ágeis em empresas de desenvolvimento de software, com foco especial na metodologia Scrum e na junção dos papéis de Scrum Master e Product Owner. O primeiro capítulo é dedicado a fazer uma visão geral do trabalho, incluindo uma breve descrição do tema, sua importância no contexto atual,

como também os objetivos gerais e específicos do estudo.

Já o segundo capítulo apresentará uma revisão bibliográfica sobre os métodos ágeis, abordando os princípios básicos, suas implementações em empresas de TI e as características específicas que os tornam particularmente adequados para este tipo de ambiente.

O terceiro capítulo será dedicado a uma revisão bibliográfica dos papéis que podem estar presentes em equipes de desenvolvimento de software, com um foco especial nos papéis específicos de cada metodologia abordada no capítulo 2. Essa análise se concentrará em entender e comparar as responsabilidades e expectativas associadas a cada papel dentro desses métodos, fornecendo uma base teórica sólida para a discussão subsequente sobre a junção de papéis entre o Scrum Master e o Product Owner que será analisada no capítulo seguinte.

Assim, o quarto capítulo será dedicado a análise do caso em questão, explorando a experiência de um profissional que assume simultaneamente os papéis de Scrum Master e Product Owner, além de apresentar os resultados de uma pesquisa submetida a profissionais da área da tecnologia, para que fosse possível avaliar os conhecimentos sobre os papéis de Scrum Master e Product Owner e a percepção desses profissionais quanto aos impactos dessa junção de papéis.

Finalmente, o quinto capítulo resumirá as principais descobertas do estudo e discutirá as consequências da junção dos papéis de Scrum Master e Product Owner. Nesse capítulo, será abordado as vantagens e desvantagens dessa prática, baseando-se tanto na revisão bibliográfica feita quanto nos resultados do estudo de caso.

Como resultado, este estudo visa fornecer informações úteis sobre gestão de projetos de desenvolvimento de software, com um foco particular nos papéis e responsabilidades dentro das equipes, especialmente as que adotam a metodologia Scrum. A expectativa é que, ao término deste trabalho, seja possível proporcionar uma compreensão aprofundada acerca da viabilidade da junção desses papéis de Scrum Master e Product Owner e oferecer recomendações práticas para equipes e organizações que possivelmente estão considerando ou já estejam enfrentando essa realidade.

## 2 Gestão de projetos em TI

Em razão da natureza complexa e em constante mudança da indústria de tecnologia, a gestão de projetos é fundamental para garantir que os produtos de software sejam entregues dentro do prazo, orçamento e com a qualidade esperada. Essa gestão pode ser desempenhada utilizando métodos tradicionais ou ágeis e está presente nos projetos de desenvolvimento para auxiliar no planejamento, gerenciamento de mudanças e na gestão das equipes. Esse capítulo irá aprofundar neste tema, analisando diversos métodos e práticas que podem ser aplicadas em projetos de desenvolvimento de software em empresas de TI.

### 2.1 Gestão de projetos com PMBOK

O Project Management Body of Knowledge (PMBOK), desenvolvido pelo Project Management Institute (PMI), consolida-se como um conjunto abrangente de práticas, padrões e normas amplamente aceitas em gerenciamento de projetos, aplicáveis a variados contextos e durações de projetos (CRUZ, 2013).

Enquanto guia de melhores práticas, e não um método ou metodologia específica, o PMBOK fornece uma estrutura para auxiliar os gestores de projetos no planejamento, execução e controle, abordando elementos-chave como cronograma, custos, qualidade, recursos e riscos. Ele incentiva a personalização das práticas sugeridas de acordo com as particularidades de cada projeto. Organiza-se em cinco grupos de processos e dez áreas de conhecimento (PMI, 2017), sendo cada um essencial na condução de um projeto do início ao fim.

Os 5 Grupos de Processos especificam as etapas sequenciais e interligadas que formam o ciclo de vida de um projeto segundo o PMBOK. Eles englobam:

- Iniciação:

Este processo inicial é fundamental na definição e implementação de um novo projeto ou na instauração de uma nova fase de um projeto preexistente. Durante esta fase, são determinadas a viabilidade, as principais partes interessadas e é produzido um documento de iniciação do projeto para que o mesmo seja oficialmente autorizado. Conforme destacado por Vargas (VARGAS, 2016), em projetos que aderem ao guia PMBOK, a documentação é um aspecto crucial, cada etapa precisa ser cuidadosamente documentada antes de se avançar para a próxima, assegurando assim a integridade e a rastreabilidade do processo de gestão do projeto.

- Planejamento:

No processo de planejamento são estabelecidos e refinados os objetivos e o trabalho necessário para concluir o projeto. Um plano de gerenciamento do projeto completo é necessário para isso, incluindo escopo, cronograma, custo, qualidade, recursos humanos, comunicação, riscos e aquisições.

- Execução:

Para que os objetivos sejam alcançados, essa fase envolve a realização dos trabalhos especificados no plano de gerenciamento do projeto. Segundo o PMBOK (PMI, 2017), o gerenciamento da equipe do projeto, a execução das entregas do projeto, o gerenciamento da qualidade e a comunicação com as partes interessadas podem estar entre as tarefas que serão realizadas neste estágio.

- Monitoramento e Controle:

Este conjunto de procedimentos inclui monitorar, avaliar e regulamentar o progresso e o desempenho do projeto. Isso inclui garantir que o projeto esteja de acordo com o plano de gerenciamento, no escopo, no orçamento e no prazo. Essa fase também envolve a identificação rápida de possíveis problemas e a implementação de ações corretivas quando forem necessárias.

- Encerramento:

Finaliza todas as atividades para concluir formalmente o projeto, a fase do projeto, ou o contrato. Algumas tarefas podem fazer parte dessa etapa, como a realização de uma auditoria após o término do projeto, a documentação das lições aprendidas, a liberação dos recursos do projeto e também a celebração do sucesso do projeto.

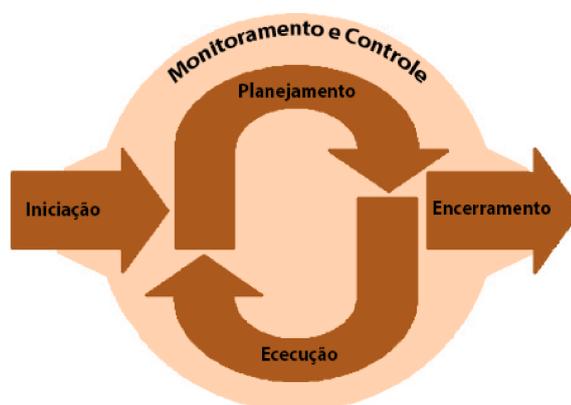


Figura 1 – Processos PMOK

(REIS, 2014) <https://www.profissionaisti.com.br/voce-conhece-o-guia-pmbok/>

O PMBOK também identifica 10 áreas de conhecimento distintas que abrangem vários aspectos da gestão de projetos, cada uma delas é inter-relacionada e trabalha em

conjunto com as outras para garantir que o projeto seja executado de maneira eficiente, cumprindo os requisitos e entregando valor ao cliente (PMI, 2017). Elas incluem:

- Gerenciamento da Integração:  
Foca na organização dos diferentes componentes do projeto. Inclui a criação do termo de abertura, gerenciamento e controle do plano e encerramento do projeto.
- Gerenciamento do Escopo:  
Engloba processos que garantem que o projeto inclua todo o trabalho necessário, e apenas o necessário, para ser concluído com sucesso.
- Gerenciamento de Cronograma:  
Foca na criação de um cronograma apropriado para a conclusão do projeto e assegura que ele seja cumprido.
- Gerenciamento de Custos:  
Inclui processos como planejamento, estimativa, orçamentação e controle de custos, garantindo que o projeto seja concluído dentro do orçamento aprovado inicialmente.
- Gerenciamento da Qualidade:  
Se concentra em garantir que os padrões e requisitos de qualidade sejam atendidos no projeto.
- Gerenciamento dos Recursos:  
Envolve a identificação, aquisição e gestão dos recursos necessários para que o projeto seja concluído com sucesso.
- Gerenciamento de Comunicações:  
Garante que as informações do projeto sejam geradas, coletadas, transmitidas, armazenadas e recuperadas de forma adequada e em tempo hábil.
- Gerenciamento de Riscos:  
Inclui o planejamento do gerenciamento de riscos, identificação, análise quantitativa e qualitativa dos riscos, planejamento de respostas e monitoramento e controle de riscos em um projeto.
- Gerenciamento de Aquisições:  
Envolve os processos de compra ou aquisição de produtos, serviços ou resultados, bem como a contratação de terceiros para a realização de parte do projeto.

- Gerenciamento das Partes Interessadas:

É centrado em processos que identificam pessoas, grupos ou organizações que podem afetar ou serem afetados pelo projeto, e assim gerenciam suas expectativas e sua participação no projeto.

O uso dos grupos de processos e das áreas de conhecimento definidos pelo PMBOK podem ser adaptados de acordo com as necessidades de cada projeto, como pode ser visto no estudo feito por Andrade ([ANDRADE; TAIT, 2012](#)), onde foram usadas somente as gerências de recursos humanos, custos e riscos. Em seu estudo, Andrade analisou um projeto de menor escala e complexidade, onde também existia a preocupação com o prazo no desenvolvimento por parte dos usuários, dessa forma ele notou que não seria necessário o uso de todas as gerências e processos propostos pelo PMBOK.

Por fim, o PMBOK segue sendo uma ferramenta amplamente utilizadas na gestão de projetos, oferecendo uma abordagem completa, adaptável e flexível para o planejamento, execução e controle de projetos, sendo uma referência valiosa para aqueles que buscam obter sucesso nessa área. Ele é considerado uma ferramenta de gestão de projetos mais tradicional, oferecendo uma abordagem sequencial e estruturada ([PMI, 2017](#)), diferentemente dos métodos ágeis que serão discutidos a seguir.

## 2.2 Métodos ágeis

Os métodos ágeis são uma abordagem para a gestão e o desenvolvimento de projetos que enfatizam a importância dos princípios de flexibilidade, colaboração, entregas contínuas e adaptação constante ([DYBÅ; DINGSØYR, 2008](#)). Essa nova abordagem teve origem em 2001, quando um grupo de especialistas em desenvolvimento de software criou o Manifesto Ágil ([PRIKLADNICKI; WILLI; MILANI, 2014](#)), com o objetivo de superar as limitações dos métodos tradicionais já existentes de gestão de projetos.

Essas novas metodologias para desenvolvimento de software surgiram como uma resposta a abordagem tradicional, conhecida por seus processos longos, falta de flexibilidade e comunicação ineficaz entre equipes de desenvolvimento e clientes. A partir de então, as metodologias ágeis vem se tornando cada vez mais populares em todo o mundo, sendo aplicadas e adaptadas por empresas de diferentes setores e tamanhos para desenvolver softwares com mais eficiência, qualidade e transparência ([COHN, 2005](#)).

Na figura 2 é possível perceber as diferenças da metodologia tradicional em contraponto da metodologia ágil. Na primeira é possível perceber uma estrutura com processos bem definidos, onde uma etapa só começa quando a anterior termina, além da entrega de valor ao cliente acontecer somente ao final de todas as etapas o que gera um alto risco devido a falta de visibilidade por parte do cliente ao longo do processo de desen-

volvimento. Já a segunda, como pode-se notar, é mais adaptável e flexível, com pequenas entregas de valor ao cliente ao longo do processo, diminuindo os riscos pois o produto é constantemente avaliado pelo cliente.

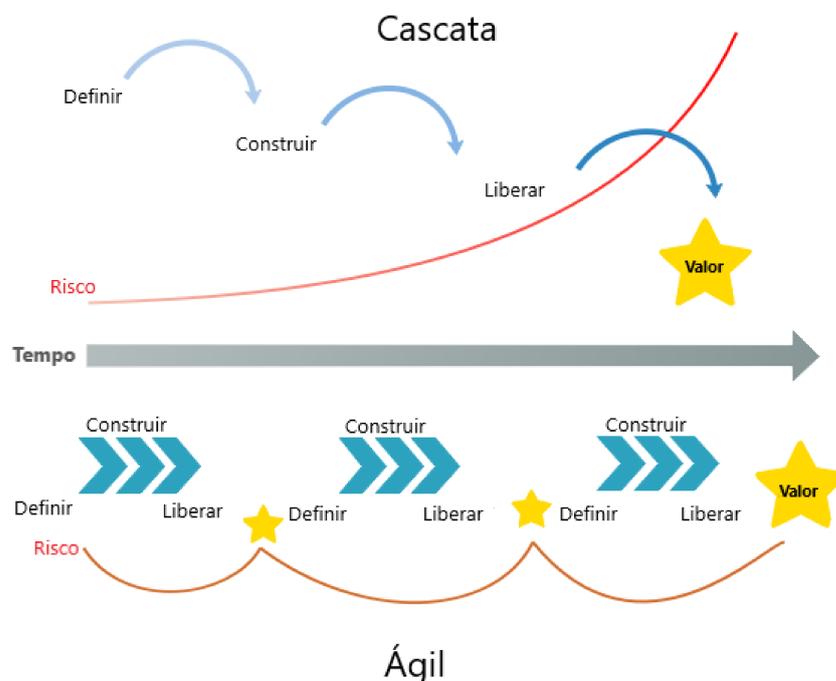


Figura 2 – Métodos tradicionais x Métodos ágeis

(SOPHIA, 2021) Adaptado pela autora de  
 «<https://www.edrawsoft.com/pt/agile-vs-waterfall.html>»

O Manifesto Ágil possui quatro conceitos-chave que são fundamentais para o desenvolvimento ágil de projetos. São eles:

- Indivíduos e interações mais que processos e ferramentas;

Martin Fowler (FOWLER, 2005), um dos autores do Manifesto Ágil, destacou que é importante valorizar a interação humana no desenvolvimento de software, visto que ele é criado por pessoas para pessoas. Ao priorizar pessoas e interações, a abordagem ágil reconhece que os seres humanos, com sua criatividade, colaboração e habilidades, são mais cruciais para o sucesso do projeto do que seguir rigidamente um conjunto de processos ou usar determinadas ferramentas.

- Software funcionando mais que documentação abrangente;

O segundo conceito-chave destaca a importância de fornecer um software em funcionamento ao invés de gastar muito tempo fazendo uma documentação detalhada. Esta abordagem é adotada devido à natureza dinâmica dos requisitos do projeto, que estão sujeitos a alterações frequentes, tornando a documentação de requisitos

propensa à ficar obsoleta ou imprecisa. Sendo assim, é mais importante focar no desenvolvimento de um produto funcional que possa ser testado e aprimorado com base no feedback do cliente (COCKBURN, 2006).

- Colaboração com o cliente mais que negociação de contratos;

Outro autor do manifesto ágil, Jim Highsmith (HIGHSMITH, 2004), enfatizou a importância da colaboração com o cliente para garantir que o software atenda às suas necessidades e expectativas. Ele explica que as equipes de desenvolvimento ágil devem trabalhar em colaboração constante com os clientes, envolvendo-os em todas as fases do processo e incorporando feedbacks contínuos ao produto.

- Responder a mudanças mais que seguir um plano.

O quarto conceito-chave destaca a importância da adaptabilidade e da flexibilidade para responder às mudanças em vez de seguir um plano rígido (BECK, 1999). Isso significa que o planejamento é importante, mas a equipe de desenvolvimento deve estar preparada e aberta para mudar de direção, ajustando o plano do projeto com base no feedback e nas necessidades do cliente. Os métodos ágeis reconhecem que as mudanças são inevitáveis e, portanto, adota uma abordagem que permite a adaptação a novas informações e circunstâncias ao longo do ciclo de vida do projeto (COBB, 2011).

Além desses quatro conceitos-chave descritos acima, o Manifesto Ágil possui doze princípios ágeis que reforçam os mesmos pontos (DINGSØYR et al., 2012). Dentre eles destacam-se a satisfação do cliente pela entrega contínua, as mudanças de requisitos serem bem vindas no projeto e as pessoas de negócios e desenvolvedores trabalhando juntas.

A seguir serão tratadas algumas metodologias ágeis que seguem os valores e princípios do Manifesto Ágil, enfatizando a entrega contínua de software funcionando, a colaboração entre indivíduos, a adaptação a mudanças e a melhoria contínua do processo (BECK et al., 2001) permitindo que as equipes sejam mais eficazes em ambientes de desenvolvimento de software que estão em constante evolução.

### 2.2.1 Gestão de projetos com Agile Project Management (AgilePM)

O AgilePM, também conhecido como Agile Project Management, é uma abordagem de gerenciamento de projetos baseada nos valores e princípios do manifesto ágil. Esse método busca oferecer soluções mais flexíveis e adaptativas, em comparação com métodos tradicionais, permitindo que as equipes se adaptem mais rapidamente às mudanças nas demandas do projeto. Ele foi desenvolvido originalmente a partir do Dynamic Systems Development Method (DSDM, ou, em português, Metodologia de Desenvolvimento de

Sistemas Dinâmicos), que é uma das abordagens ágeis mais antigas para o desenvolvimento de software (STAPLETON, 1997). O AgilePM prioriza a colaboração, a adaptação e a entrega incremental, ao contrário das abordagens tradicionais de gerenciamento de projetos, que tendem a ser mais rígidas e previsíveis (CONSORTIUM, 2017).

O AgilePM usa um conjunto de princípios e práticas específicos para garantir que os projetos sejam entregues dentro do escopo, tempo e custo necessários (BASILI; LARMAN, 2003). Esses princípios são a espinha dorsal do método e servem como guias para garantir que um projeto seja conduzido de maneira verdadeiramente ágil.

Os principais conceitos do AgilePM incluem (CONSORTIUM, 2017):

- Foco no Negócio:

O AgilePM não se concentra apenas nos aspectos técnicos, em vez disso enfatiza a importância de alinhar o projeto com objetivos comerciais claros. Isso garante que o projeto forneça valor real e atenda às necessidades dos clientes.

- Entregar no prazo:

A capacidade de entregar valor rapidamente é uma das maiores vantagens da abordagem ágil. No AgilePM isso é alcançado por meio de entregas incrementais e iterações chamadas de timeboxes, que possuem durações fixas.

- Colaboração:

Esse princípio enfatiza a importância do trabalho conjunto entre desenvolvedores, usuários e qualquer outra parte interessada. A colaboração eficaz entre todos garante que a solução final seja satisfatória e aceitável para todos os envolvidos.

- Nunca comprometer a qualidade:

Para não sacrificar a qualidade em virtude de cumprir prazos, o AgilePM insiste em padrões de qualidade consistentes, o que significa que todas as entregas devem atender a critérios definidos de aceitabilidade.

- Desenvolver de forma incremental:

Esse ponto envolve a entrega de valor de forma iterativa e incremental, permitindo que as partes interessadas vejam o progresso real e obtenham valor mais cedo no ciclo de vida do projeto. Além disso, a equipe consegue fazer revisões e adaptações contínuas, reagindo adequadamente às mudanças.

- Comunicar de forma clara e contínua:

A comunicação eficaz é vital para garantir que todos os envolvidos estejam na mesma página, por isso, o AgilePM destaca a importância de técnicas de comunicação face a face, sempre que possível.

- Demonstrar controle:

Esse princípio fala sobre o uso de técnicas e métricas para demonstrar de forma clara o progresso do projeto, permitindo que as partes interessadas tenham confiança na direção em que o projeto está caminhando.

- Ser flexível, mas não comprometer:

Embora o AgilePM valorize a flexibilidade, ele também reconhece que aspectos específicos de um projeto, como qualidade e prazos, não devem ser comprometidos. É um equilíbrio entre adaptabilidade e manutenção dos padrões.

Embora o AgilePM e o PMBOK tenham o objetivo de orientar as equipes na entrega de sucesso dos projetos, eles oferecem abordagens distintas quando o assunto é gerenciamento de projetos. Uma das principais diferenças dessas duas abordagens é quanto a flexibilidade e adaptação às mudanças, pois enquanto o AgilePM acolhe as mudanças como parte natural e esperada do gerenciamento de projetos (COBB, 2011), o PMBOK tende a defender uma abordagem mais linear, com um planejamento detalhado no início do projeto e um maior controle as mudanças (PMI, 2017).

Outra diferença é quanto a estrutura do projeto, que como estudado anteriormente o PMBOK possui cinco grupos de processos bem definidos (PMI, 2017), já o AgilePM não contém uma estrutura clara, se concentrando em ciclos de desenvolvimento ágeis e iterativos. Os papéis e responsabilidades também se diferem nesses métodos, visto que no PMBOK a figura central é o gerente de projetos e no AgilePM a ênfase é na equipe colaborativa e auto-gerida (CONSORTIUM, 2017).

A entrega de valor é outro ponto que diferencia essas duas abordagens, sendo que no AgilePM a entrega é frequente e constante, onde o cliente pode ver valor e usar os resultados ao longo do processo (CONSORTIUM, 2017). E no PMBOK a grande entrega de valor só pode ser observada ao final do projeto. E finalmente, a comunicação também distingue o AgilePM do PMBOK, no primeiro a comunicação é clara e dinâmica e tem a colaboração tanto entre a equipe como com o cliente, já no segundo apesar de valorizar a comunicação, esta é feita de forma mais formal (PMI, 2017).

Por fim, métodos ágeis populares como o método Scrum e o Kanban, são exemplos de abordagens de Agile Project Management. Eles oferecem estruturas específicas e particulares de cada método para que as equipes de desenvolvimento de software possam implementar os princípios ágeis na gestão de projetos de acordo com a necessidade de cada time. Esses métodos ágeis serão aprofundados ao decorrer desse capítulo.

## 2.2.2 Gestão de projetos com RUP (Rational Unified Process)

O Rational Unified Process (RUP), é um framework de desenvolvimento de software que visa guiar equipes na produção de software de alta qualidade de maneira eficiente e consistente, oferecendo uma abordagem disciplinada para atribuir tarefas e responsabilidades durante o processo de desenvolvimento. Ao contrário de modelos sequenciais tradicionais, o RUP é iterativo e incremental. Isso significa que o projeto é dividido em iterações menores, permitindo feedbacks frequentes e ajustes conforme necessário, onde iteração produz um incremento do produto final (KRUCHTEN, 2003).

Como uma metodologia robusta e flexível, o RUP oferece uma estrutura que pode ser adaptada para atender às necessidades específicas de diferentes projetos e organizações. Sua natureza iterativa e o foco em práticas de engenharia sólidas o tornam uma escolha atraente para muitas equipes de desenvolvimento. Porém, como dito por (GONÇALVES, 2016) o RUP não é um framework de fácil entendimento e aprendizado, se faz necessário um alto entendimento sobre seu funcionamento e a curva de aprendizado tende a ser longa, devido a sua complexidade.

O RUP é estruturado em quatro fases principais (RATIONAL, 1998):

- **Concepção:**

Esta fase tem o objetivo de estabelecer o escopo do projeto e os recursos necessários. É quando a ideia do projeto é concebida e os stakeholders são identificados.

- **Elaboração:**

Durante esta fase, os requisitos do sistema são mais claramente definidos e analisados, com isso os desenvolvedores da equipe começam a compreender o produto que será desenvolvido. Além disso, os riscos associados ao projeto são identificados e a arquitetura do sistema é estabelecida.

- **Construção:**

Nesta fase, o software é realmente construído. O foco é na codificação, teste e integração de componentes do sistema.

- **Transição:**

Esta fase é dedicada a mover o software do desenvolvimento para a produção, envolvendo a entrega do software aos usuários, treinamentos, implantação e aceitação do sistema.

### 2.2.2.1 Organização do Trabalho no RUP

O RUP é um framework abrangente e adaptável, projetado para ser personalizado de acordo com as necessidades específicas de um projeto ou organização. Para isso ele

possui fluxos de trabalho, que são uma sequência de atividades que produzem um resultado visível e significativo para o projeto (RATIONAL, 1998). Esses fluxos de trabalho destacam a abordagem holística do RUP ao desenvolvimento de software, abrangendo não apenas atividades técnicas, mas também aspectos gerenciais e de infraestrutura que são fundamentais para o sucesso de um projeto. São eles (BORGES, 2008):

1. Modelagem de Negócios:

Este fluxo se concentra na definição, documentação e refinamento dos processos operacionais da organização. Isso inclui a criação de modelos que representem os processos de negócios, pessoas envolvidos e as interações entre eles. Esse fluxo é muito importante pois ao entender claramente os processos de negócios, as equipes de desenvolvimento podem garantir que o software seja alinhado com os objetivos e necessidades da organização, proporcionando soluções verdadeiramente eficazes.

2. Requisitos:

A coleta, análise, documentação e validação das necessidades e requisitos do sistema estão centradas neste fluxo. Isso significa trabalhar em conjunto com os stakeholders para entender suas necessidades e transformá-las em requisitos de sistema claros e acessíveis. Qualquer projeto de software bem-sucedido depende da captura adequada dos requisitos, evitando erros e retrabalho futuro.

3. Análise e Design:

Aborda a arquitetura do sistema, criando modelos de design que detalham como os requisitos serão implementados. Além disso, define padrões e práticas de codificação, considera questões de desempenho, segurança e integração. Esse fluxo garante que as soluções técnicas escolhidas estejam alinhadas com os requisitos e as necessidades do negócio.

4. Implementação:

Esse fluxo envolve a codificação real do software com base nos designs e arquiteturas definidos. Também inclui testes unitários e integração contínua. É nesta fase que o objetivo e as necessidades do projeto são transformados em um produto que possa ser usado.

5. Teste:

Engloba o design, implementação e execução de testes para garantir que o software atenda aos requisitos especificados e funcione corretamente em todos os cenários previstos.

6. Implantação:

Trata da entrega, instalação e aceitação do sistema no ambiente do usuário. Uma implantação bem-sucedida garante que o software seja instalado, configurado e usado corretamente pelos usuários finais, maximizando o valor que ele traz para a organização.

#### 7. Ambiente:

O foco deste fluxo de trabalho é criar e manter o ambiente de desenvolvimento. Isso inclui configurar ferramentas, técnicas e qualquer infraestrutura necessária para apoiar o projeto.

#### 8. Configuração e Gerência de Mudanças:

Concentra-se no gerenciamento de versões e mudanças no software. Isso inclui rastrear versões diferentes do software, controlar e documentar mudanças e garantir que alterações sejam adequadamente testadas e integradas.

#### 9. Gerência de Projetos:

Envolve o planejamento, monitoramento e controle do projeto. Isso inclui definir e rastrear atividades, recursos e marcos, bem como monitorar o progresso e fazer ajustes conforme necessário.

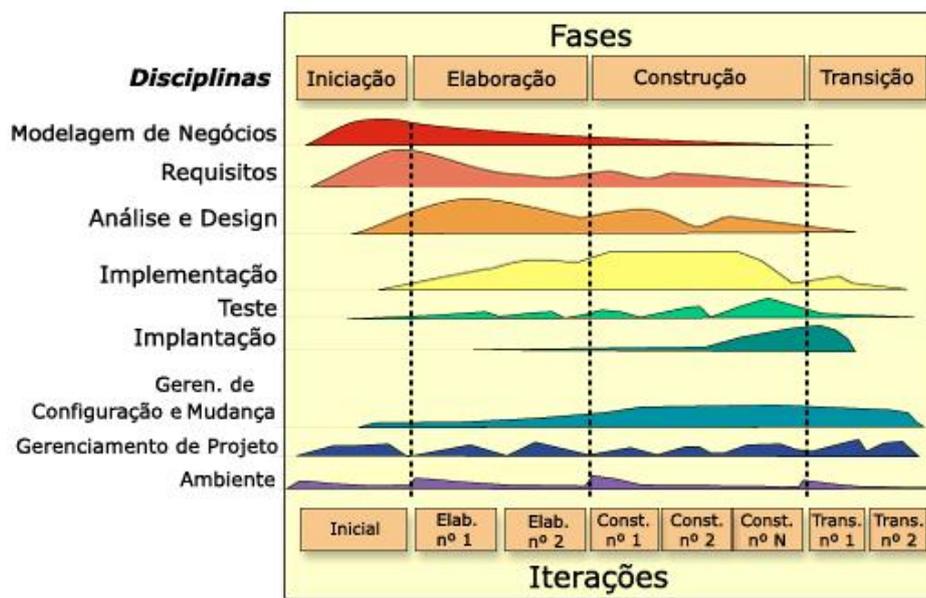


Figura 3 – Fluxos de trabalho do RUP

(WIKIPÉDIA, 2022) [https://pt.wikipedia.org/wiki/IBM\\_Rational\\_Unified\\_Process](https://pt.wikipedia.org/wiki/IBM_Rational_Unified_Process)

Cada fluxo de trabalho do RUP é fundamental para o ciclo de vida do desenvolvimento de software, cada um com um foco de trabalho específico que são essenciais para a conclusão do projeto com sucesso (BORGES, 2008). Eles garantem que todas as

atividades necessárias, desde a definição de requisitos até a implantação e suporte, sejam executadas de forma organizada e eficaz. Para isso o RUP define vários papéis, que são agrupamentos de atividades e responsabilidades, e que serão tratados no próximo capítulo.

### 2.2.3 Gestão de projetos com Scrum

O Scrum é um framework ágil para gestão e desenvolvimento de projetos de software muito utilizado nas empresas atualmente, para (SILVA; LOVATO, 2016) o Scrum é um conjunto de práticas incrementais e iterativas que tem como foco aumentar a entrega de valor para o cliente. Nessa metodologia o produto é desenvolvido em pequenas partes, entregando valor ao cliente a cada ciclo de desenvolvimento, chamados de Sprint, dessa forma permite uma maior flexibilidade e adaptação às mudanças, dado que o produto é frequentemente avaliado e melhorado ao longo do processo de desenvolvimento.

Com papéis, artefatos e ritos bem definidos, que serão tratados ao decorrer dessa seção, o Scrum garante uma abordagem colaborativa e adaptativa ao trabalho, permitindo que o time responda mais rápido as mudanças e seja possível entregar valor ao cliente de forma contínua (MACHADO; MEDINA, 2009). Nos dias de hoje, empresas de todas as áreas estão adotando o Scrum como um método eficaz de gerenciar projetos, incentivando a colaboração entre equipes e alcançando resultados mais significativos em um ambiente cada vez mais dinâmico e competitivo.

Na figura 4 é possível ter uma visão geral de como é o ciclo de vida do projeto dentro da metodologia Scrum. Esse trabalho irá apresentar mais detalhes sobre cada papel, artefato e rito que está presente nesse método.

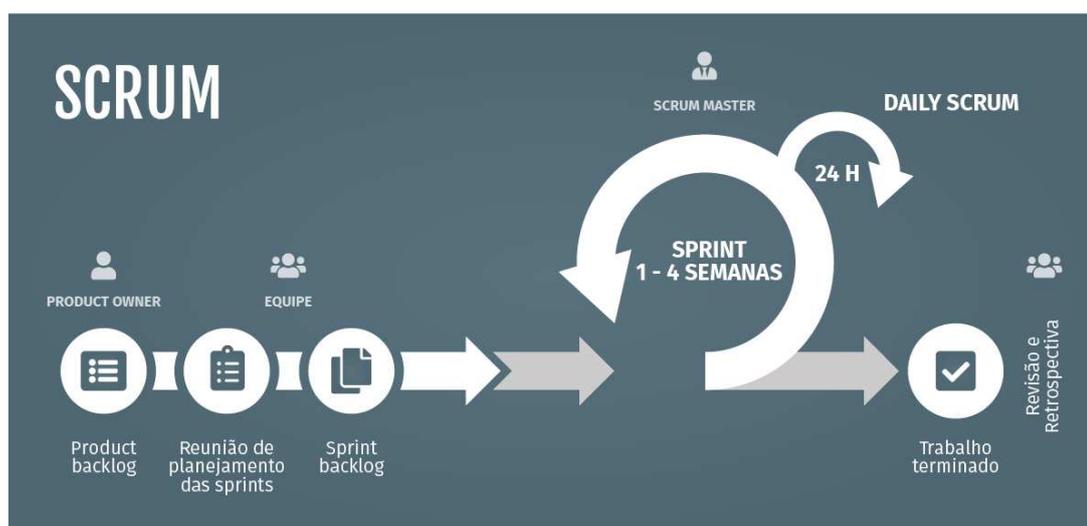


Figura 4 – Metodologia Scrum

(AMARAL, 2022)

<https://www.tabnews.com.br/LucasAmaral/o-que-sao-metodologias-ageis>

### 2.2.3.1 Papéis do Scrum

Na metodologia Scrum existem três papéis principais e bem definidos (SCHWABER; BEEDLE, 2001) : o Scrum Master, o Product Owner e a Equipe de Desenvolvimento, papéis esses que são essenciais para o sucesso do projeto, cada um com suas responsabilidades específicas. Esse estudo irá aprofundar em cada um desses papéis no próximo capítulo.

### 2.2.3.2 Artefatos do Scrum

Os artefatos do Scrum são documentos e informações fundamentais para manter a transparência, comunicação eficaz e foco na entrega de valor contínuo (SCHWABER; SUTHERLAND, 2016). Existem três artefatos principais no Scrum:

- Product Backlog:

O Product Backlog é uma lista priorizada de itens, que podem ser histórias de usuário, bugs, melhorias ou qualquer outra demanda, que são necessários para a entrega do produto final. O Product Owner é responsável por manter e atualizar o Product Backlog, garantindo que ele esteja refletindo as necessidades e prioridades do cliente e do negócio.

- Sprint Backlog:

O Sprint Backlog é uma lista detalhada dos itens do Product Backlog que foram escolhidos para serem desenvolvidos durante o Sprint atual. É um artefato dinâmico, podendo ter tarefas adicionadas ou removidas caso novas informações apareçam ou as prioridades mudem. O Sprint Backlog fornece clareza sobre o trabalho que deve ser realizado na Sprint.

- Incremento do Produto:

O Incremento do Produto é o resultado concreto de um Sprint, sendo funcional e podendo ser entregue ao cliente. O objetivo é que o Incremento seja sempre um produto utilizável e de alta qualidade. A equipe adiciona novos recursos e melhorias ao Incremento do Produto a cada Sprint, construindo gradativamente o produto final.

Estes artefatos foram criados para manter a transparência, permitir que todos os envolvidos no projeto entendam o que está sendo feito e o que está por vir, e fornecer uma visão clara do progresso do trabalho. Eles são essenciais para a gestão ágil de projetos e são usados para garantir que a equipe e os stakeholders estejam alinhados e focados na entrega de valor contínua.

### 2.2.3.3 Ritos do Scrum

Os ritos do Scrum são práticas importantes que permitem que o Time de Desenvolvimento trabalhe de forma eficiente e eficaz, eles são projetados para criar transparência, inspecionar e adaptar o trabalho à medida que o mesmo avança. Segue abaixo uma explicação sobre os principais (SCHWABER; SUTHERLAND, 2016):

- **Sprint:**

O Sprint é o ciclo de desenvolvimento que o Scrum possui, que geralmente tem a duração de 2 a 4 semanas. Um Sprint começa com uma reunião de planejamento do Sprint, onde o Product Owner em conjunto com a equipe seleciona um conjunto de itens do Product Backlog para trabalhar durante a Sprint e o objetivo é criar um Incremento funcional e potencialmente entregável até o final dessa Sprint.

- **Sprint Planning:**

No começo de cada Sprint, toda a equipe se reúne para planejar o trabalho que será realizado durante a Sprint. Durante a Sprint Planning, o Product Owner apresenta os itens do Product Backlog priorizados e a equipe seleciona as tarefas a serem realizadas durante a Sprint, envolvendo estimativas de tempo e discussões sobre como as funcionalidades serão desenvolvidas. Essa cooperação entre a equipe de desenvolvimento, o Scrum Master e o Product Owner na definição e priorização dos itens da Sprint contribuem no engajamento da equipe ao poder contribuir e participar das definições de trabalho, e no alinhamento das expectativas entre todos os envolvidos, evitando mal-entendidos e conflitos futuros.

- **Daily:**

Outro rito muito importante do Scrum é a Daily, uma reunião de curta duração, geralmente de 15 minutos, realizada diariamente onde o Time de Desenvolvimento discute suas atividades, o que foi feito no dia anterior e o que será feito no dia seguinte, além de identificar possíveis impedimentos. Para (KARDEC, 2012) a função dessas reuniões diárias é a interação entre a equipe, com o foco em encontrar soluções e alternativas para atingir o objetivo do Sprint.

- **Sprint Review:**

O Sprint Review é a reunião que acontece ao final de cada Sprint em que o Time de Desenvolvimento apresenta as funcionalidades desenvolvidas durante o Sprint para o Product Owner e para os Stakeholders. Sendo assim, nessa reunião é possível avaliar se o Sprint evoluiu com o que foi priorizado no Sprint Planning e sejam dados feedbacks sobre o que foi entregue, que pode ser uma forma de reconhecimento e celebração das conquistas do time, além de ser uma fonte de aprendizado e transparência em relação do progresso do projeto.

- Retrospectiva:

A retrospectiva é uma reunião que acontece ao final de cada Sprint onde o Time de Desenvolvimento avalia e reflete o que funcionou bem e o que pode ser melhorado no processo, para os próximos Sprints, promovendo o crescimento e a aprendizagem contínuos. É uma oportunidade de avaliar o seu próprio trabalho e as entregas do time como um todo, enaltecendo as qualidades e criando um plano de ação para as dificuldades da equipe. Ademais, o time pode propor mudanças e ajustes no processo, aumentando o sentimento de controle e responsabilidade de todos os envolvidos.

Os ritos da metodologia Scrum descritos acima desempenham um papel fundamental na eficácia do framework ágil. Eles são projetados para criar um ambiente de trabalho estruturado e colaborativo, fornecer visibilidade do progresso do projeto e promover a adaptação contínua, além de manter o foco na entrega de valor de cada Sprint. O Scrum cria um ambiente em que o crescimento individual e da equipe são valorizados, aumenta o engajamento, o alinhamento das expectativas dos envolvidos, o apoio e a responsabilização dos membros da equipe e a resolução de conflitos (SUTHERLAND, 2014).

#### 2.2.4 Gestão de projetos com Kanban

Kanban é um método de gestão visual e uma ferramenta amplamente utilizada no gerenciamento de projetos, desenvolvimento de software, e em vários outros contextos de trabalho, ele enfatiza a entrega contínua, just-in-time, gerenciamento visual e aprimoramento contínuo do processo (KNIBERG; SKARIN, 2010). A palavra "Kanban" significa "cartão visual" em japonês, e esse método se originou do sistema de produção da Toyota, onde era usado para gerenciar a produção de automóveis.

A ideia central do Kanban é limitar a quantidade de trabalho em andamento, minimizando o desperdício, melhorando a eficiência e permitindo maior flexibilidade. Para isso é usado um quadro que normalmente é dividido em colunas, como "A fazer", "Em andamento" e "Concluído", contendo uma representação visual do processo de trabalho. Cada etapa do processo é representada por cartões ou post-its que ilustram os itens necessários para concluir cada etapa. A passagem desses cartões pelas colunas mostra o progresso do trabalho, o que facilita o acompanhamento em tempo real.

O Kanban se assemelha muito ao Scrum pois também incentiva a constante melhoria e a participação e comprometimento da equipe (ANDERSON, 2010). Além de permitir uma maior flexibilidade e adaptação às mudanças, já que o processo de trabalho é regularmente avaliado e ajustado conforme as necessidades do cliente.

A figura 5 é um exemplo de um quadro Kanban, com cada coluna representando uma fase e cada cartão simbolizando uma tarefa. Assim toda a equipe consegue visualizar os itens que precisam ser desenvolvidos e os que já tiveram atuação.



Figura 5 – Metodologia Kanban

(EMPRESA, 2021) Adaptado pela autora de  
«<https://blog.delogic.com.br/quais-sao-os-tipos-de-kanban-e-como-utilizar/>»

Esse método se baseia em cinco princípios (KNIBERG; SKARIN, 2010) que serão aprofundados a seguir:

- Visualização do trabalho:

O coração do sistema Kanban é o quadro Kanban, que é uma representação visual do fluxo de trabalho e do status de cada tarefa (KNIBERG; SKARIN, 2010). A equipe tem acesso a uma visão geral clara e abrangente do trabalho em andamento, o que permite a identificação de gargalos e oportunidades de melhoria que possam existir no processo ou até mesmo no time.

- Limitação do trabalho em andamento:

Outra característica distintiva do Kanban é a definição de limites para a quantidade de trabalho que pode estar em andamento em cada etapa do processo, evitando o acúmulo de tarefas incompletas. Esse princípio é importante porque garante que o time esteja concentrado nas tarefas mais importantes e que o trabalho seja entregue dentro do prazo, sem gerar sobrecarga ou estresse da equipe (ANDERSON, 2010).

- Gerenciamento do fluxo de trabalho:

Esse princípio se refere ao acompanhamento e monitoramento do fluxo de trabalho para garantir que o mesmo flua de forma contínua. Isso inclui a identificação de gar-

galos, desequilíbrios e ineficiências do processo, para que ações possam ser tomadas e o processo siga de forma otimizada e eficiente (KNIBERG; SKARIN, 2010).

- Fazer o processo explícito:

Este princípio defende que os fluxos de trabalho sejam claramente definidos, documentados e compreendidos por todos os membros da equipe (KNIBERG; SKARIN, 2010). Isso significa ter a definição de políticas, regras e procedimentos para cada fase do fluxo de trabalho, facilitando a colaboração e as tomadas de decisões.

- Melhorar colaborativamente:

E por fim, o princípio de melhoria colaborativa incentiva a busca contínua pela melhoria do processo e do fluxo de trabalho, contando com a participação e comprometimento de todos do time (KNIBERG; SKARIN, 2010). Sendo feito através do compartilhamento de ideias, feedbacks contínuos e sugestões para melhorar o processo de trabalho.

Apesar de terem objetivos semelhantes de promover entregas e melhorias contínuas, o Kanban e o Scrum se diferenciam bastante em sua abordagem para alcançar esses objetivos. Uma das diferenças mais visíveis é quanto a estrutura, o Scrum tem seus papéis e ritos bem definidos enquanto o Kanban não possui uma pré-definição, sendo mais flexível e adaptável às necessidades da equipe e do cliente. Outro ponto onde é possível notar a diferença é quanto ao planejamento das entregas, enquanto o Scrum tem planejamentos definidos por Sprint, com itens de backlog planejados para serem entregues dentro daquele prazo estipulado, no Kanban o planejamento é contínuo, com a equipe trabalhando em itens conforme estão prontos para serem realizados, sem compromisso de entrega em um período fixo.

Tanto o Scrum como o Kanban já se mostraram eficazes na gestão de projetos e no desenvolvimento de produtos e devem ser escolhidos de acordo com a cultura e características da equipe e do projeto, sendo possível até adotar elementos dos dois métodos, combinando suas práticas para atender às suas necessidades específicas do projeto em questão.

## 3 Papéis dentro das metodologias de gerenciamento de projetos

A divisão de papéis é uma prática essencial que tem um grande impacto na eficiência, produtividade e qualidade do trabalho dentro das equipes de desenvolvimento de software. Cada membro da equipe tem um papel específico e deve cumprir tarefas bem definidas, o que ajuda a manter a equipe organizada e focada em suas tarefas-chave. Essa divisão de papéis permite que os membros da equipe se especializem em suas respectivas áreas de competência. Por exemplo, desenvolvedores se concentram em escrever código, testadores se concentram em garantir a qualidade do software e designers se concentram na experiência do usuário. Isso resulta em trabalhos de maior qualidade em suas respectivas áreas, além de também agilizar o processo de desenvolvimento dentro do projeto.

### 3.1 Papéis comuns em equipes de desenvolvimento de software

Dentro de uma equipe de desenvolvimento de software, diversos papéis podem ser identificados, cada um com suas responsabilidades específicas. Embora a presença e a nomenclatura de alguns destes papéis possam variar conforme a metodologia ou o contexto da organização, eles geralmente realizam tarefas semelhantes em diferentes ambientes. A seguir, será apresentado uma lista com os papéis mais comuns em equipes de desenvolvimento de software, que podem ser encontrados em diversas metodologias e podem variar de acordo com o tamanho e complexidade do projeto:

- Gerente de Projeto:

O Gerente de projetos é responsável por planejar, organizar e controlar os recursos para alcançar os objetivos específicos dentro dos projetos. Ele garante que o projeto seja concluído no prazo, orçamento e com qualidade. Ele está presente em diversas metodologias como PMBOK, Scrum e RUP que veremos mais especificamente a seguir.

Esse papel é muito importante pois ele atua na promoção da comunicação constante, feedback contínuo e na criação de oportunidades de aprendizado e crescimento para melhorar a agilidade e eficiência da equipe (MOE; DINGSØYR; DYBÅ, 2010)(APPELO, 2013; ASHBACHER, 2010). Isso ajuda a manter a equipe alinhada com as expectativas do cliente e a melhorar a agilidade e eficiência. Os projetos que possuem práticas ágeis ressaltam a importância de ambientes colaborativos e iterativos, o que requer uma liderança adaptável e voltada para as pessoas, promovendo uma

comunicação e aprendizado constante e equipes auto-organizadas ([COCKBURN; HIGHSMITH, 2001](#)).

- Desenvolvedor Back-End:

Os desenvolvedores back-end se concentram na lógica do servidor, bancos de dados e infraestrutura. Algumas das tarefas comuns desse papel são a criação e manutenção do banco de dados, desenvolvimento de API's para comunicação entre servidor e front-end, garantia da segurança do sistema, entre muitas outras.

Eles são muito importantes pois desenvolvem as funcionalidades em si, a lógica e todas as interações por trás do sistema ([SOMMERVILLE, 2016](#)).

- Desenvolvedor Front-End:

Esse papel se responsabiliza pela criação e design de interfaces de usuário, ou seja, a parte visual e interativa de um site ou aplicativo web. Os desenvolvedores front-end traduzem os layouts criados pelos UX/UI em código, garantem a responsividade e adaptação do sistema em diversos tipos de computadores, focam na interface e na experiência do usuário que irá usá-la, e muitas outras funções.

É com o trabalho dos desenvolvedores front-end que o cliente final consegue usufruir de todas as funcionalidades que o sistema oferece e são tratadas no back-end ([SOMMERVILLE, 2016](#)).

- Desenvolvedor Full-Stack:

Esse papel combina as habilidades de um desenvolvedor front-end e back-end, realizando todas as tarefas listadas acima para os mesmos ([SOMMERVILLE, 2016](#)). É versátil e pode trabalhar em ambos os lados da aplicação, do design da interface ao banco de dados.

- Desenvolvedor Mobile:

O desenvolvedor mobile é especializado em criar aplicativos para dispositivos móveis, seja para Android, iOS ou ambas as plataformas. Ele se parece com o papel do desenvolvedor front-end, mas voltado para o desenvolvimento em aparelhos mobile ([SOMMERVILLE, 2016](#)).

Tem como tarefas implementar interfaces de usuário para dispositivos móveis, integrar APIs e serviços de back-end, garantir que os aplicativos tenham desempenho otimizado, testar aplicativos em diferentes dispositivos e versões de sistema operacional, além de publicar aplicativos nas lojas de aplicativos (Google Play, App Store, etc.).

- Analista de Sistemas:

O analista de sistemas atua na ponte entre os requisitos do negócio e a implementação técnica, se concentrando em entender as necessidades tecnológicas e de sistema de uma organização. Ele transforma requisitos de alto nível em especificações técnicas detalhadas (WIEGERS; BEATTY, 2013).

Entre as funções de um analista de sistemas é possível citar a análise e documentação de processos e sistemas existentes e que serão criados pelo projeto, auxílio na integração de diferentes sistemas e trabalho próximo aos desenvolvedores para garantir que os sistemas sejam construídos conforme especificado.

- **Projetistas de Sistemas ou Designers de Sistemas:**

Tem a função de transformar os requisitos do sistema em especificações detalhadas que os desenvolvedores podem usar para criar o software. Eles criam modelos e diagramas para representar a estrutura do sistema, tomam decisões sobre aspectos específicos do design, como a escolha de padrões de design, algoritmos ou estruturas de dados, além de trabalhar em estreita colaboração com analistas de sistemas para garantir que os requisitos sejam claramente entendidos e traduzidos corretamente em design (WIEGERS; BEATTY, 2013).

- **Analista de Negócios:**

Foca na identificação e análise dos requisitos do negócio e traduzi-los em termos compreensíveis para a equipe de desenvolvimento, garantindo que os sistemas de TI implementados suportem essas necessidades apontadas pelo negócio (WIEGERS; BEATTY, 2013). O analista de negócios em determinadas equipes pode ter funções parecidas com a de um Product Owner, que será estudado nos papéis da metodologia Scrum.

Entre as responsabilidades comuns pode-se mencionar a interação e colaboração com as partes interessadas para entender as suas necessidades, documentação de requisitos de negócios e transformá-los em especificações claras para a equipe de TI (COCKBURN, 2001), facilitar a comunicação entre as equipes de negócio e de TI e monitorar a implementação para garantir que atendam aos requisitos de negócios.

- **Analista de qualidade/Tester/QA:**

Os profissionais que atuam nesse papel tem como responsabilidade garantir a qualidade do produto entregue ao cliente, identificando e documentando defeitos e garantindo que eles sejam corrigidos (DUSTING, 2002). Eles são a ponta final entre o desenvolvimento e a entrega do software e devem se certificar que o mesmo atende as necessidades dos usuários finais.

As funções mais comuns desse papel são a elaboração e execução de testes no sistema, ou em funcionalidades específicas que foram alteradas, reporte de bugs e erros encontrados nas execuções e garantia que o sistema atende aos requisitos.

- Analista de Automação de Testes:

Esse profissional tem a função semelhante ao analista de qualidade, ele deve garantir a qualidade e funcionamento do sistema entregue. Para isso ele projeta, desenvolve e executa testes automatizados para garantir a qualidade do software, aumentando a eficiência e a cobertura dos testes em relação aos métodos de teste manual (DUSTING, 2002).

A automação de testes é um tipo de codificação que visa garantir a qualidade do software desenvolvido, quando realizado por analistas de automação de testes o foco é validar a integração e funcionamento das funcionalidades totais do sistema.

- Designer UI/UX:

Esse papel foca na experiência do usuário, projetando interfaces que sejam esteticamente agradáveis e intuitivas para o usuário (KRUG, 2014). Eles estudam como melhorar a navegação e utilização do sistema, pensam em melhorias de layout que fariam que as funcionalidades do sistema fossem mais facilmente acessadas e entendidas pelos usuários finais. Esse papel pode inclusive fazer entrevistas com as pessoas que usam ou irão usar o sistema para entender suas dificuldades e exigências.

- Arquiteto:

O arquiteto de software tem como responsabilidade projetar a estrutura global de um sistema de software, definindo as principais decisões de design técnico e garantindo que todos os componentes do software interajam de maneira harmoniosa. Ele que decide sobre quais padrões de arquitetura e frameworks mais se encaixam no projeto, além de tomar decisões sobre tecnologias, plataformas e ferramentas a serem utilizadas (BASS; CLEMENTS; KAZMAN, 2013).

- Profissionais de Implantação e Treinamento:

Eles garantem que o software seja corretamente instalado, configurado e que os usuários finais sejam treinados para usar o software efetivamente (SOMMERVILLE, 2016). Entre as funções que eles podem desempenhar estão o planejamento, execução e instalação do software em ambiente de produção, o treinamento dos usuários finais sobre como usar cada funcionalidade da ferramenta e a documentação sobre a instalação e configuração do sistema.

- Profissionais de Suporte e Manutenção:

Os profissionais que trabalham nesse papel tem como responsabilidade garantir que o software continue funcionando conforme o esperado após o lançamento, corrigindo bugs e fazendo atualizações conforme necessário (SOMMERVILLE, 2016). Ele trabalham diagnosticando e corrigindo bugs ou problemas relatados pelos usuários e atualizando o software para adicionar mais recursos ou melhorias.

- Auditores:

Os auditores asseguram que o software e os processos associados atendam a padrões específicos, regulamentos ou melhores práticas. Eles revisam a documentação, código e processos para garantir a conformidade, identificando áreas de risco ou não conformidade, além de fazer recomendações para melhorar e diminuir riscos (DAVIS; SCHILLER; WHEELER, 2011). Eles também são responsáveis por garantir que as melhores práticas e padrões da indústria sejam seguidos.

- Partes Interessadas:

Esse papel se refere a todas as pessoas ou grupos que podem ser afetados pelo projeto ou que têm interesse em seu resultado (FREEMAN, 2015). Isso pode incluir fornecedores, reguladores, clientes e outros. As responsabilidades desse papel incluem o fornecimento de informações sobre os requisitos e expectativas do projeto, a participação de revisões e aprovações, além de conceder feedbacks sobre o progresso e os resultados do projeto.

Esses papéis descritos acima são comuns em vários tipos de projetos, e como explicado podem variar de nomenclatura dependendo da metodologia escolhida pela equipe. Além disso alguns papéis podem ser aglutinados ou até excluídos dependendo das necessidades e dimensões do projeto. A seguir neste capítulo será apresentado como são implementados os papéis e funções profissionais específicos dentro do contexto dos métodos de gestão de projetos já estudados no capítulo anterior, com suas diferenças e particularidades.

## 3.2 Papéis dentro do PMBOK

A divisão de papéis em equipes que seguem o PMBOK (Project Management Body of Knowledge) é essencial para o planejamento, execução e controle eficazes de projetos, e é projetada para garantir uma alocação clara de responsabilidades e a promoção da comunicação. Além de definir a divisão de papéis e tarefas atribuídas a cada um, o PMBOK (PMI, 2017) fornece diretrizes e orientações gerais sobre como esses diferentes papéis devem interagir ao longo do ciclo de vida do projeto.

Os principais papéis que geralmente estão presentes em equipes que seguem o guia PMBOK são:

- Gerente de Projetos (Project Manager):

O papel central em uma equipe que segue o PMBOK é o do Gerente de Projetos e suas responsabilidades padrões já foram descritas no começo desse capítulo. Dentro da metodologia PMBOK sua função inclui a criação e atualização do plano de

projeto, comunicação com as partes interessadas, gerenciamento de riscos, monitoramento do progresso e tomada de decisões (VARGAS, 2016).

A atuação do gerente de projetos já começa desde o início do mesmo, onde ele lidera a equipe na elaboração do Plano do Projeto. Isso envolve a colaboração com a equipe do projeto, partes interessadas e o patrocinador para definir objetivos, escopo, cronograma, orçamento e recursos. As informações fornecidas pelas partes interessadas são fundamentais para a definição dos requisitos e do escopo do projeto.

- Patrocinador do Projeto (Project Sponsor):

A pessoa ou entidade que fornece apoio financeiro, recursos e autoridade para o projeto é conhecido como patrocinador do projeto. Eles são essenciais para determinar os objetivos do projeto e garantir que ele esteja em conformidade com os objetivos estratégicos da organização. Eles são responsáveis por tomar decisões, fornecer orientações estratégicas e manter uma comunicação eficaz com as partes interessadas de alto nível.

- Equipe do Projeto (Project Team):

A equipe do projeto é formada por profissionais que desempenham papéis específicos, como desenvolvedores, arquitetos e analistas. Cada membro da equipe é responsável por realizar as tarefas relacionadas ao projeto e contribuir para a entrega bem-sucedida, como também já foi descrito acima. O PMBOK destaca a importância de algumas tarefas desempenhadas pela equipe do projeto em conjunto com especialistas relevantes e partes interessadas, como as revisões do progresso e qualidade do trabalho e a avaliação de possíveis riscos ao longo do desenvolvimento do produto.

- Partes Interessadas (Stakeholders):

No PMBOK as partes interessadas possui processos específicos dedicados à identificação, gerenciamento e engajamento dos stakeholders ao longo do ciclo de vida do projeto. A comunicação bem-sucedida com os stakeholders é um componente essencial do gerenciamento de projetos nessa metodologia. O processo de Identificação das Partes Interessadas (Stakeholder Identification) é um passo importante no PMBOK que envolve a colaboração entre o Gerente de Projetos e outros membros da equipe para identificar todas as partes que podem influenciar ou serem afetadas pelo projeto.

As definições claras de papéis e funções dentro da equipe de um projeto que segue o PMBOK é uma prática essencial para garantir que o projeto seja executado de forma eficaz, cumprindo seus objetivos e padrões de qualidade. Isso promove uma colaboração eficaz, comunicação clara e uma melhor alocação de recursos, contribuindo para o sucesso do projeto. Porém é importante lembrar que a estrutura organizacional do projeto e as

atribuições de papéis podem variar de acordo com o tamanho, a complexidade e a natureza do projeto. Além disso, o PMBOK fornece diretrizes gerais, mas as organizações podem adaptar esses papéis às suas necessidades específicas.

Em situações excepcionais, é possível que uma organização opte por combinar ou ajustar papéis de acordo com suas necessidades. No entanto, isso deve ser feito com consideração cuidadosa para garantir que as responsabilidades e tarefas não se sobrepõem e que a estrutura organizacional e de papéis seja eficaz para atender aos objetivos do projeto.

A abordagem do PMBOK é mais tradicional e estruturada em comparação com as metodologias ágeis, como o Scrum, que oferecem maior flexibilidade na adaptação dos papéis. Portanto, a ênfase no PMBOK é fornecer uma estrutura de gerenciamento de projetos consistente e bem definida, mas ainda há margem para ajustar essa estrutura em conformidade com as necessidades específicas de um projeto ou organização.

### 3.3 Papéis dentro do AgilePM (Agile Project Management)

Para garantir o sucesso de um projeto dentro de um ambiente ágil, o AgilePM estabelece uma série de papéis essenciais, que representam os interesses do processo, comerciais, de gerenciamento e técnicos e soluções. Estes papéis foram definidos e estruturados para trazer clareza, eficiência e responsabilidade ao processo de desenvolvimento, mantendo a flexibilidade característica das abordagens ágeis. São eles ([CONSORTIUM, 2014a](#)):

- Patrocinador Comercial (Business Sponsor):

O Business Sponsor é a força motriz em termos de suporte e recursos no mais alto nível organizacional para o projeto. Ele é o campeão do projeto, responsável pelo Business Case e pelo orçamento, e desempenha um papel vital nas decisões estratégicas, atuando frequentemente como a última instância para decisões que ultrapassam a esfera de influência do Gerente de Projeto.

- Visionário Empresarial (Business Visionary):

O Business Visionary define e comunica a visão do projeto, garantindo que a equipe esteja alinhada com as expectativas e os objetivos do patrocinador e do negócio. Este papel fornece orientação contínua em termos de prioridades de negócios e expectativas, ajudando a guiar as decisões de desenvolvimento e design para manter o projeto alinhado com os objetivos estratégicos.

- Coordenador Técnico (Technical Coordinator):

O Coordenador Técnico é uma figura-chave na equipe do AgilePM, garantindo que as questões técnicas sejam tratadas de forma eficaz e que a solução técnica desenvolvida seja sólida, sustentável e alinhada com as necessidades do negócio.

- Gerente de Projeto (Project Manager):

Como visto no começo do capítulo, no AgilePM o Gerente de Projeto também é responsável pelo planejamento detalhado, execução e conclusão do projeto. Além de coordenar atividades e gerenciar riscos, ele assegura uma comunicação clara e efetiva entre todas as partes interessadas.

- Analista de Negócios (Business Analyst):

No AgilePM o Analista de Negócios facilita a interação entre o desenvolvimento e o negócio. Ele é responsável por analisar e modelar os requisitos de negócios, garantindo que a solução entregue esteja em conformidade com as necessidades do negócio e os objetivos do projeto.

- Líder de Equipe (Team Leader):

No AgilePM, o Líder de Equipe, semelhante ao papel de um Scrum Master na metodologia Scrum, é responsável por coordenar o projeto e as entregas, mantendo a equipe focada e produtiva. Este papel não é tipicamente gerencial e pode ser preenchido por diferentes membros da equipe em fases distintas do projeto.

- Embaixador de Negócios (Business Ambassador):

Esse papel se assemelha ao de um Product Owner da metodologia Scrum, e atua como um representante do cliente ou negócio dentro da equipe, definindo e priorizando requisitos e oferecendo apoio contínuo durante o desenvolvimento do projeto.

- Desenvolvedor de Soluções (Solution Developer):

Os Desenvolvedores de Soluções são os artesãos técnicos, transformando requisitos de negócios em incrementos de software funcionais, aderindo a práticas de desenvolvimento ágil e melhorias contínuas.

- Testador de Soluções (Solution Tester):

Essa função já foi definida no início desse capítulo e tem como responsabilidade assegurar a qualidade, realizando testes apropriados e mantendo os padrões desde a concepção dos requisitos até a entrega final do produto.

- Consultor de Negócios (Business Advisor):

O papel de Business Advisor é concebido para envolver os especialistas e partes interessadas do negócio no projeto. Ele é fundamental para fornecer informações detalhadas sobre o contexto de negócios e requisitos, além de assegurar que a solução desenvolvida esteja alinhada com as necessidades do negócio.

- Consultor Técnico (Technical Advisor):  
O Technical Advisor oferece conhecimento técnico especializado e aconselhamento, garantindo que a equipe de projeto tenha acesso a informações e práticas técnicas atualizadas.
- Facilitador de Workshop (Workshop Facilitator):  
O Workshop Facilitator é essencial para o sucesso dos workshops no AgilePM, garantindo que estes sejam conduzidos de forma eficaz e contribuam para o progresso do projeto.
- Treinador DSDM (DSDM Coach):  
O DSDM Coach é um mentor para equipes que são novas na metodologia AgilePM, facilitando a adaptação dos membros do projeto e otimizando a adoção da metodologia dentro dos parâmetros do mesmo.

Cada um destes papéis contribui de maneira específica para o ambiente colaborativo e adaptativo proposto pelo AgilePM (CONSORTIUM, 2017), assegurando que todos os aspectos do projeto sejam adequadamente abordados e alinhados com os objetivos do negócio.

### 3.4 Papéis dentro RUP (Rational Unified Process)

Como visto no capítulo anterior, o RUP é uma metodologia estruturada que divide o desenvolvimento de software em uma série de atividades e tarefas. No modelo base do RUP, são identificados aproximadamente 40 papéis distintos, cada um com suas características e responsabilidades específicas (KRUCHTEN, 2003)(RATIONAL, 1998). Esses papéis são cruciais para assegurar que todas as atividades sejam realizadas de maneira correta e eficiente.

No entanto, dado que o RUP é um framework flexível e adaptável, em projetos menores e de menor complexidade, o número de papéis pode ser diminuído, conforme evidenciado por Borges (BORGES, 2008). Em tais situações, as tarefas dos papéis omitidos podem ser realocadas entre os papéis remanescentes na equipe ou, em alguns casos, podem até não ser executadas. Combinar múltiplos papéis em um único profissional pode diminuir a produtividade e ser prejudicial ao projeto se feito imprudentemente (GONÇALVES, 2016).

Essa metodologia pode apresentar papéis comuns como os que foram mencionados no início do capítulo, o grande diferencial é como esses profissionais atuam e em qual fase tem mais participação. O RUP possui quatro fases principais, Iniciação (ou Concepção), Elaboração, Construção e Transição, e cada uma com objetivos específicos e com diferentes

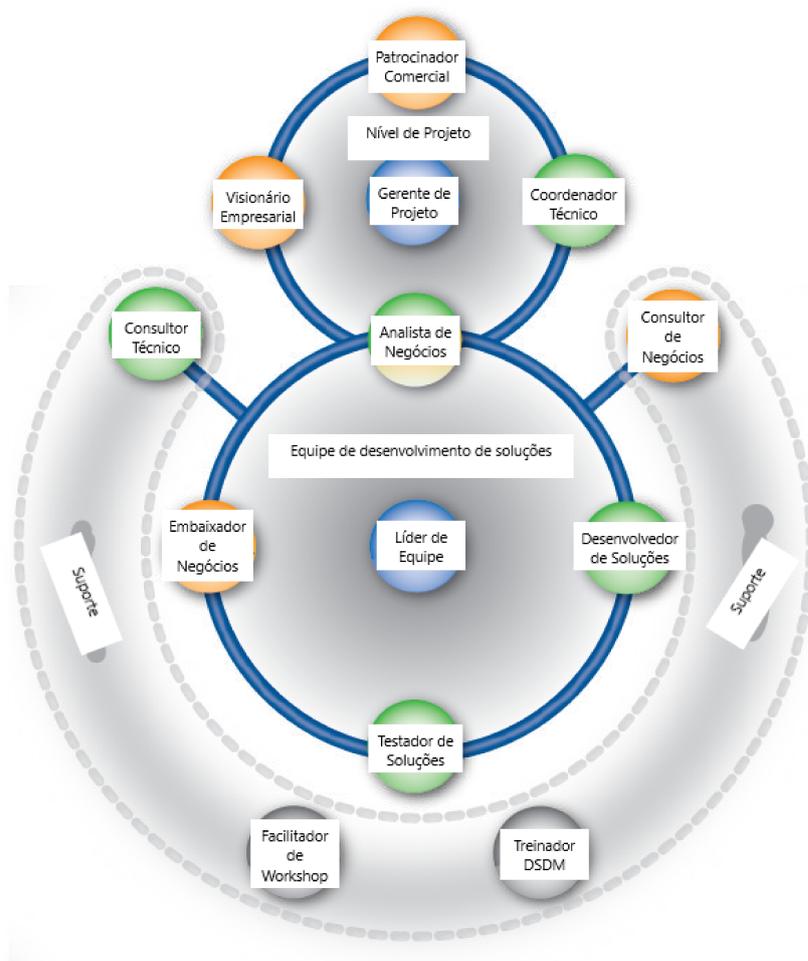


Figura 6 – Papéis no AgilePM

(CONSORTIUM, 2014b) Adaptado pela autora de  
«<https://www.agilebusiness.org/dsdm-project-framework/roles-and-responsibilities.html>»

papéis desempenhando tarefas-chave. Será representada a atuação desses papéis em cada fase abaixo:

#### 1. Arquiteto de Software:

- Iniciação: Começa a estabelecer uma visão de alto nível da arquitetura do sistema.
- Elaboração: Refina a arquitetura, resolve os riscos arquiteturais mais significativos e define a maioria das interfaces e componentes do sistema.
- Construção: Garante que a arquitetura está sendo seguida e faz ajustes conforme necessário.

#### 2. Analista de Negócios:

- Iniciação: Identifica os principais stakeholders e requisitos iniciais do sistema.
- Elaboração: Desenvolve o modelo de casos de uso e garante que os requisitos estão bem definidos e entendidos.

3. Gerente de Projetos:

- Iniciação: Define o escopo do projeto e o caso de negócio.
- Elaboração: Monitora riscos e ajusta planos conforme necessário, e continua a coordenação com os stakeholders.
- Construção: Gerencia recursos e cronogramas para garantir a entrega do produto.
- Transição: Prepara-se para a entrega do produto e o fechamento do projeto.

4. Desenvolvedor de Software:

- Elaboração: Prototipa componentes-chave e ajuda a validar a arquitetura.
- Construção: Foca na implementação de todas as funcionalidades do sistema.

5. Analista de qualidade:

- Elaboração: Começa a planejar os testes com base nos casos de uso e requisitos.
- Construção: Realiza testes de integração, sistema e aceitação.
- Transição: Executa testes de aceitação do usuário e garante que o sistema atende aos requisitos.

6. Gerente de configuração:

- Elaboração: Estabelece a infraestrutura para controle de versão e gestão de mudanças.
- Construção: Gerencia as versões de lançamento e assegura a qualidade das builds.

7. Profissional de implantação e treinamento:

- Transição: Trabalha na implantação do sistema e no treinamento dos usuários finais.

8. Profissional de suporte e manutenção:

- Transição: Prepara-se para fornecer suporte contínuo após a implantação.

Essa metodologia tem papéis projetados para abordar todos os aspectos do desenvolvimento de software, desde a identificação de requisitos até a entrega e manutenção do sistema. Cada papel tem uma função específica, e a colaboração eficaz entre eles é essencial para o sucesso do projeto. Por ter uma estrutura de papéis, atividades e artefatos bem definidos, o RUP é frequentemente considerado adequado para grandes projetos com equipes que podem estar geograficamente separadas. No entanto, quando aplicado a projetos menores, algumas das formalidades e papéis do RUP podem ser aglutinados ou simplificados para tornar o processo mais ágil e menos oneroso.

### 3.5 Papéis dentro do método Scrum

Como estudado no capítulo anterior, a metodologia Scrum possui papéis bem definidos dentro das equipes, cada um desses papéis com suas funções e responsabilidades específicas, onde o foco é o sucesso do projeto. Essa definição clara de papéis contribui para uma organização eficiente e uma gestão bem presente junto ao time de desenvolvimento.

O Scrum tem como foco a criação um ambiente que incentive a cooperação, a responsabilidade compartilhada e a autonomia da equipe, criando assim equipes auto-organizadas, onde cada membro do time tem liberdade e responsabilidade para tomar decisões e contribuir ativamente para o sucesso do projeto (WATTS, 2012).

Nessa metodologia pode-se observar também uma liderança servidora (ADKINS, 2010), ou seja, líderes que apoiam suas equipes removendo obstáculos, dando suporte e criando um ambiente saudável e dinâmico. Essa habilidade de facilitação se mostra no Scrum com o papel do Scrum Master. Além do Scrum Master, outro papel importante é o do Product Owner que é responsável por estabelecer os requisitos e prioridades do produto. Seu trabalho em conjunto com a equipe de desenvolvimento ajuda a garantir que o produto atenda às necessidades do cliente. As responsabilidades de cada papel dentro do Scrum são:

#### 1. Scrum Master

O Scrum Master é essencialmente um facilitador da equipe (DAVIDSON; KLEMME, 2016), certificando-se que a equipe está seguindo as práticas e os valores do Scrum. Ele é responsável por garantir que o Scrum seja compreendido e utilizado de forma correta, além de facilitar o trabalho do time de desenvolvimento, ajudando em qualquer ponto que esteja impedindo o progresso da equipe e suas tarefas. Além disso, outro papel do Scrum Master de acordo com (SILVA; LOVATO, 2016) é o de treinar os membros da equipe para que os mesmos sejam auto-gerenciáveis, isto é, sejam capazes e tenham autonomia para gerenciar suas próprias atividades.

O Scrum Master desempenha um papel semelhante a um Gerente de Projetos dentro das equipes, ele atua na remoção de impedimentos e na criação de um ambiente em que a equipe possa ser produtiva (SCHWABER; BEEDLE, 2001). Sua atuação promove a agilidade e a colaboração eficaz da equipe, o que é essencial para o sucesso do projeto. Ele está envolvido em todas as fases do ciclo de vida do desenvolvimento ágil, trabalhando com a equipe desde a definição do Product Backlog até a entrega das funcionalidades no final de cada Sprint. E também tem a função de facilitar os ritos do Scrum, como a reunião diária (Daily Scrum), o planejamento da Sprint (Sprint Planning) e a revisão da sprint (Sprint Review).

## 2. Product Owner:

O Product Owner é o representante do cliente dentro do projeto (SVERRISDOTIR; INGASON; JONASSON, 2014), e tem o papel de definir, detalhar e priorizar as funcionalidades do produto (SABBAGH, 2014), garantindo que o Time de Desenvolvimento esteja trabalhando nas tarefas que entregam mais valor para o cliente, além de assegurar que o item desenvolvido esteja de acordo com os padrões esperados (SCHWABER; BEEDLE, 2001).

O Product Owner desempenha um papel central na definição de direção e prioridades para o projeto. Sua participação garante que o projeto esteja progredindo alinhado com as expectativas dos stakeholders e cliente. Ele também está envolvido desde o início do ciclo de vida do desenvolvimento, definindo o Product Backlog, até o final de cada sprint, revisando e aceitando o trabalho concluído.

## 3. Equipe de Desenvolvimento:

O Time de Desenvolvimento, por sua vez, é responsável por desenvolver e entregar um produto capaz de ser utilizado a cada Sprint (SCHWABER; BEEDLE, 2001). Ele pode ser formado por vários perfis de profissionais diferentes, como estudado no começo desse capítulo. Como desenvolvedores, analistas, arquitetos e muitos outros papéis diferentes, cada qual com sua função específica dentro do projeto. Essa equipe é multifuncional e auto-gerenciada, ou seja, ela se organiza e gerencia seu próprio trabalho (SHRIVASTAVA; RATHOD, 2014). Segundo (KARDEC, 2012) o time possui autoridade e autonomia para definir como cada tarefa será desenvolvida, além de estimar o tempo que será gasto na mesma. Sendo assim a equipe se torna responsável por cada resultado alcançado.

A equipe de desenvolvimento é a força motriz por trás da entrega de valor ao cliente. Sua colaboração e responsabilidade compartilhada garantem que o produto seja desenvolvido de forma eficaz e com alta qualidade, garantindo que o mesmo atende aos critérios de aceitação do cliente. A equipe de desenvolvimento está envolvida em

todas as fases do ciclo de vida do desenvolvimento, desde a criação do incremento do produto até a revisão e retrospectiva da sprint.

Ao longo do ciclo de vida do desenvolvimento ágil, esses papéis trabalham juntos de maneira colaborativa e interdependente. A colaboração entre Scrum Master, Product Owner e equipe de desenvolvimento é essencial para alcançar os princípios ágeis, como a entrega contínua de valor, a adaptação às mudanças e a colaboração eficaz. O Scrum se torna uma metodologia poderosa para o sucesso de projetos de desenvolvimento ágil quando esses papéis funcionam de forma eficiente.

Apesar de ter os papéis definidos de forma clara, em algumas situações, principalmente em equipes menores ou em projetos menos complexos, pode ocorrer uma junção de papéis. Isso significa que uma pessoa pode desempenhar mais de um papel ao mesmo tempo. Serão exploradas algumas situações em que a junção de papéis pode ocorrer:

- Scrum Master e Membro da Equipe de Desenvolvimento:

Em equipes muito pequenas, um membro da equipe de desenvolvimento pode assumir o papel de Scrum Master, desde que tenha o conhecimento necessário para facilitar o processo Scrum e remover impedimentos. No entanto, essa combinação deve ser feita com cuidado, pois o Scrum Master deve, idealmente, ser um facilitador neutro que não está diretamente envolvido no trabalho de desenvolvimento.

- Product Owner e Membro da Equipe de Desenvolvimento:

Em projetos extremamente pequenos ou em startups, um membro da equipe de desenvolvimento pode desempenhar o papel de Product Owner, tomando decisões sobre as prioridades e requisitos do produto. Isso pode funcionar bem quando as necessidades do cliente são relativamente simples.

- Scrum Master e Product Owner:

Apesar de serem papéis que possuem responsabilidades e interesses diferentes e, muitas vezes, conflitantes em situações muito específicas ou em equipes extremamente pequenas, pode haver casos em que uma pessoa desempenha ambos os papéis, devido a escassez de recursos ou de profissionais.

Quando esses papéis são desempenhados por pessoas diferentes, há um equilíbrio de poder e perspectivas, o que é saudável para o sucesso do projeto. Caso haja uma situação em que uma única pessoa precise assumir ambos os papéis, é essencial que ela esteja ciente dos potenciais conflitos e encontre maneiras de mitigá-los, tomando decisões que beneficiem o projeto como um todo.

Embora a junção de papéis seja possível em algumas circunstâncias, é importante ter em mente que o Scrum foi projetado com papéis distintos para promover a colaboração

e a divisão de responsabilidades. Idealmente, a equipe deve trabalhar para manter esses papéis separados para garantir a transparência, a responsabilidade e a eficácia do processo Scrum.

No estudo de caso do próximo capítulo, esse trabalho irá aprofundar na análise da integração dos papéis de Scrum Master e Product Owner. Será Abordado como essa combinação de responsabilidades pode impactar o processo ágil e o sucesso do projeto.

## 4 Estudo de caso

O presente estudo de caso tem como foco examinar dois projetos de desenvolvimento de software realizado por uma empresa de tecnologia, que será chamada de 'Empresa X' nesse trabalho a fim de de manter o anonimato da mesma. A 'Empresa X' que está envolvida em dois projetos significativos com um cliente do ramo da construção civil, que também terá seu nome preservado e será chamada de 'Construtora Y'. A peculiaridade desses projetos é que o Scrum Master desempenha o papel de Scrum Master tradicional e também assume as responsabilidades do Product Owner.

Este capítulo irá explorar a dinâmica de trabalho nos projetos desenvolvidos pela 'Empresa X' para atender as necessidades da 'Construtora Y'. Além disso, investigaremos como a junção dos papéis de Scrum Master e Product Owner impacta o dia a dia da equipe de desenvolvimento e dos projetos em si.

Este estudo de caso também incluirá uma pesquisa aplicada aos funcionários da 'Empresa X', buscando obter percepções valiosas sobre a eficácia dessa junção de papéis e como ela influencia o sucesso do projeto e o bem-estar da equipe. O objetivo é entender as implicações e os desafios dessa prática, bem como a visão dos profissionais envolvidos, visando aprimorar ainda mais o uso do Scrum e adaptar as melhores práticas ao contexto específico da empresa.

### 4.1 Contexto

- A Empresa:

A empresa que será estudada nesse trabalho é a 'Empresa X', mencionada anteriormente, ela já está no mercado a mais de 15 anos e conta com cerca de 200 profissionais da área da tecnologia. Possui sua sede localizada em Belo Horizonte - Minas Gerais, mesma localidade do cliente que será estudado.

É uma empresa prestadora de serviços, que foca na entrega de qualidade e conta com uma vasta lista de clientes em todo o Brasil. Além de ter equipes de desenvolvimento de software também possui produtos próprios e trabalha com a terceirização de serviços voltados para a área de qualidade de software.

- O Cliente:

O cliente dono e idealizador dos projetos que serão discutidos nesse capítulo é a 'Construtora Y', uma empresa com mais de 40 anos de história no mercado nacional, fundada em Belo Horizonte e hoje presente em 13 estados brasileiros. Possui mais

de 6 mil profissionais desde engenheiros a desenvolvedores, além de já ter entregue mais de 180 mil unidades habitacionais.

- Os Projetos:

Esse estudo de caso discutirá dois projetos que foram desenvolvidos pela 'Empresa X' em parceria com a 'Construtora Y'.

1. Projeto 1: O primeiro projeto envolve a criação de uma plataforma digital, web e mobile, que tem como objetivo melhorar a experiência dos clientes no pós venda. Ele será usado para que o cliente que compre um imóvel da 'Construtora Y' tenha uma plataforma para acompanhar o andamento da sua obra, documentos importantes, os fluxos financeiros dessa aquisição e muitas outras informações.

Esse projeto está sendo desenvolvido a cerca de 2 anos e meio e tem a previsão de ser totalmente entregue em 6 meses, e grandes entregas já foram feitas durante esse período. A primeira entrega aconteceu 1 ano após o início da análise e negociação do produto, concentrando-se nas características mais relevantes e valiosas para o cliente, entregando um produto funcional que vem sendo evoluído desde então. A equipe por trás do desenvolvimento desse projeto segue a metodologia Scrum, sem a presença de um Product Owner (PO) devido a escassez de recursos no projeto. Com a ausência do PO, a equipe possui 8 membros com a seguinte divisão:

- Um Scrum Master;
  - Um Analista de Sistemas;
  - Um Analista de Qualidade;
  - Dois Desenvolvedores Frontend;
  - Um Desenvolvedor Backend;
  - Um Profissional de UX/UI;
  - Um Arquiteto.
2. Projeto 2: O segundo projeto foca na criação de uma plataforma digital que será usada para a contratação de novos colaboradores para a 'Construtora Y'. Ele engloba a abertura da vaga, o envio de documentos por meio de um membro do RH ou até mesmo pelo próprio funcionário que está sendo contratado e também a validação e aprovação desses documentos por parte do RH da empresa.

Esse projeto está em desenvolvimento a cerca de 1 ano e meio e tem previsão de entrega em 1 ano, sendo que a primeira grande entrega foi feita por volta de 8 meses após seu início e novas funcionalidades e melhorias seguem sendo desenvolvidas.

Esse segundo projeto também é desenvolvido por uma equipe ágil, que utiliza a metodologia Scrum para a gestão no dia a dia. Também não existe o papel do PO e possui 8 membros, sendo que a equipe tem a seguinte divisão:

- Um Scrum Master;
- Um Analista de Sistemas;
- Um Analista de Qualidade;
- Um Desenvolvedor Frontend;
- Dois Desenvolvedores Backend;
- Um Profissional de UX/UI;
- Um Arquiteto.

Ao longo de todo esse tempo de desenvolvimento dos projetos alguns profissionais foram mudando nas equipes, isso é, alguns membros migraram para outros projetos dentro da empresa, outros mudaram de cargo dentro do projeto e alguns novos colaboradores foram adicionados a medida que o projeto demandou.

## 4.2 A Junção de Papéis

No cenário estudado, o Scrum Master assume também o papel do Product Owner desde o começo de ambos os projetos, somando todas as responsabilidades desses dois papéis já descritas nesse trabalho e resumidas na tabela 1. Esse estudo de caso explorará os pontos positivos e negativos dessa junção de papéis através de uma perspectiva de vivência real, em conjunto com a revisão bibliográfica feita nos capítulos anteriores.

Os times que estão por trás do desenvolvimento desses produtos seguem desde o começo a metodologia Scrum, sendo feitos todos os ritos citados anteriormente, como Daily, Sprint Planning, Sprint Review e Srint Retrospective. A adaptação de todos da equipe com o Scrum se deu de forma tranquila e natural, e também foi facilmente aceito pelo cliente que não compartilhava dessa metodologia nos processos internos.

A ausência de um Product Owner em projetos geridos com a metodologia Scrum não possui muita referência nas bases teóricas, como citado pela A. Lima 2022 em seu artigo (LIMA; AMARAL, 2022), a falta de um profissional com esse papel dentro dos projetos foge das definições da metodologia, onde todos os papéis e ritos devem ser mantidos para que o projeto realmente seja ágil e siga a metodologia a risca. Sendo assim, não existe um manual a ser seguido de como prosseguir quando temos uma equipe onde o PO não esteja presente.

Na prática, alguns papéis são mais suscetíveis à aglutinação, especialmente em equipes menores ou em organizações com recursos limitados. Papéis como o Analista de

<b>Scrum Master</b>	<b>Product Owner</b>
Identifica e remove obstáculos que impedem o progresso da equipe.	Define e prioriza os itens do Product Backlog.
Facilita os ritos do Scrum, como Daily, Sprint Planning, Sprint Review e Retrospectiva.	Mantém uma comunicação contínua com os stakeholders, entendendo e representando suas necessidades.
Protege o time de distrações externas e interrupções durante o Sprint.	Gerencia e comunica as expectativas de prazos e metas do projeto.
Garante que a equipe esteja seguindo os princípios e praticas do Scrum.	Representa os interesses e necessidades do cliente e define os requisitos do produto.
Atua como coach para a equipe, isso significa ouvir ativamente, apoiar, remover obstáculos e fornecer orientação quando necessário.	Mantém o Product Backlog transparente, visível e claro para todos da equipe.
Promove a colaboração entre a equipe de desenvolvimento e o Product Owner.	Participa das reuniões da equipe e esclarece dúvidas sobre os itens do Product Backlog.
Contribui para o planejamento, fornecendo ideias sobre os itens do Product Backlog.	Decide quais funcionalidades serão desenvolvidas e a ordem, baseado no valor de negócio.
Auxilia a equipe na Definição de Pronto dos incrementos.	Tem a autoridade final para tomar decisões sobre as prioridades e o conteúdo do Product Backlog.
Ajuda a equipe a monitorar métricas relevantes para melhorar o desempenho.	

Tabela 1 – Responsabilidades do Scrum Master x Product Owner

Elaborado pela autora (2023)

Negócios e o Product Owner, ou o Desenvolvedor e o Testador de Software, podem ser combinados com sucesso se forem gerenciados de forma a evitar conflitos de interesse e sobrecarga. Entretanto, a pesquisa e o estudo de caso que serão estudados na próxima seção indicam que a fusão de Scrum Master e Product Owner deve ser abordada com cautela. A fim de manter a eficácia do processo ágil, é essencial buscar um equilíbrio, preservar a clareza das responsabilidades e assegurar que a qualidade e a comunicação dentro da equipe não sejam comprometidas.

Com isso, os papéis que são mais facilmente agrupados em um único profissional, segundo a análise feita sobre os papéis e responsabilidades dentro dos projetos de desenvolvimento de software durante esse trabalho está representada na tabela 2. É importante estar ciente que a junção de papéis pode ou não ser uma boa estratégia para determinados projetos, irá depender de vários fatores como a complexidade, duração, pressão do projeto, e até do tipo do profissional que assumirá esses papéis. Segue uma análise dessas possíveis junções em equipes de desenvolvimento:

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[1] Scrum Master	X			X					
[2] Product Owner		X	X	X					
[3] Analista de Negócios		X	X	X	X				
[4] Analista de Sistemas	X	X	X	X	X				
[5] Analista de Qualidade			X	X	X	X			
[6] Analista de Automação de Testes					X	X	X	X	
[7] Desenvolvedor Backend						X	X	X	X
[8] Desenvolvedor Frontend						X	X	X	X
[9] Arquiteto							X	X	X

Tabela 2 – Possível junção de papéis em equipes de desenvolvimento de software

Elaborado pela autora (2023)

- Junção de Scrum Master e Analista de Sistemas:

A união dos papéis de Scrum Master e Analista de Sistemas em alguns projetos pode ser considerada uma estratégia eficaz, já que o Scrum Master já possui uma compreensão profunda do fluxo de trabalho da equipe e das dinâmicas de projeto, e pode fornecer uma visões técnicas e análises aprofundadas, garantindo que os requisitos do negócio sejam compreendidos e implementados eficientemente.

- Junção de Product Owner, Analista de Negócios e Analista de Sistemas:

Essa fusão dos papéis pode ser uma escolha prática, visto que o Analista de Negócios ou Analista de Sistemas, com sua habilidade de analisar requisitos de negócios e transformá-los em especificações técnicas, complementa naturalmente as responsabilidades do Product Owner. Essa combinação permite uma comunicação mais direta e eficiente entre as necessidades do negócio e a implementação técnica, facilitando a tradução de requisitos complexos em soluções práticas e viáveis.

- Junção de Analista de Negócios, Analista de Sistemas e Analista de Qualidade:

O Analista de Negócios ou de Sistemas já possui um entendimento aprofundado dos requisitos e objetivos do negócio, bem como das especificações técnicas necessárias para a implementação do projeto. Ao adicionar as responsabilidades de um Analista de Qualidade, garante que o produto final atenda aos padrões de qualidade esta-

belecidos. Isso permite uma verificação e validação mais eficientes e integradas dos requisitos durante todo o ciclo de vida do desenvolvimento do software.

- Desenvolvedores Backend, Frontend e Arquiteto:

Essa união aproveita a expertise técnica dos desenvolvedores, que já estão intimamente envolvidos na codificação e implementação do software, e a amplia com as responsabilidades de um arquiteto, que se concentra na estruturação e planejamento geral do sistema. Essa junção pode oferecer percepções valiosas sobre a viabilidade e eficácia de certas abordagens arquiteturais, tendo em vista a realidade do desenvolvimento cotidiano.

- Desenvolvedores Backend, Frontend e Analista de Automação de Testes:

Em certos tipos de projetos pode ser vantajoso combinar os papéis de Desenvolvedores Backend ou Frontend com o de Analista de Automação de Testes. Esta combinação alavanca a experiência técnica dos desenvolvedores, que já estão imersos na lógica e estrutura do software, para implementar testes automatizados mais eficazes. Essa união de papéis encoraja a adoção de práticas de desenvolvimento orientadas a testes, onde os testes são escritos antes do código, ajudando a garantir um software de maior qualidade desde o início.

Nos projetos descritos a solução encontrada foi que o Scrum Master assumisse as responsabilidades do Product Owner, o que gerou benefícios e desafios para o profissional em questão e para a equipe. Nas seções seguintes será explorado as percepções dessa junção pela visão do Scrum Master que liderou essas equipes e também haverá a aplicação de uma pesquisa para os demais profissionais desses times para que seja possível avaliar os impactos desse formato adotado pela equipe no desenvolvimento desses projetos.

### 4.3 Perspectiva do Scrum Master

Baseando-se na experiência e opiniões de um profissional que atua simultaneamente como Scrum Master e Product Owner, essa seção irá discorrer sobre as nuances de desempenhar ambas as funções em projetos de desenvolvimento de software. Entre as funções mais importantes de um Product Owner está a definição e priorização do Product Backlog, e nesse contexto onde essa função também é responsabilidade do Scrum Master, foi possível observar que houve ganho na agilidade e flexibilidade na tomada de decisões, visto que o Scrum Master está mais envolvido com o projeto e possui uma relação bem próxima do time de desenvolvimento e nesse caso, do cliente. Sendo assim, ele é capaz de identificar quais itens são mais valiosos, e prioriza-los de acordo com a capacidade do time em cada Sprint de maneira ágil e eficiente. Esse alinhamento entre o processo ágil

e as necessidades do cliente levou a uma entrega de valor mais rápida e alinhada com as expectativas do cliente.

Para conseguir desempenhar a função de priorização do backlog, o Scrum Master em questão adotou a prática de realizar reuniões mensais com os stakeholders. Esses encontros eram dedicados a refinar a lista do Product Backlog e definir as atividades a serem priorizadas nas duas Sprints subsequentes. Diferentemente de alguns times de desenvolvimento que projetam suas atividades para períodos de até seis meses, a grande mudança das necessidades do cliente nesses projetos específicos exigia uma abordagem mais ágil e adaptativa. Assim, as sessões de priorização focavam em entregas de curto prazo, com a flexibilidade para reajustes frequentes conforme as demandas evoluíam.

Entretanto, a constante alteração nas prioridades levantou desafios, como a falta de objetividade e imparcialidade do Scrum Master, sendo um ponto amplamente discutido e pontuado pela equipe de desenvolvimento, gerando conflitos e estresse devido às pressões do cliente e às necessidades da equipe.

Durante os projetos, principalmente o primeiro citado, aconteceram mudanças de itens que já estavam priorizados pelo cliente e sobre os quais a equipe já havia começado a trabalhar. Essas mudanças ocorreram porque esse item em questão já não tinha mais valor algum para o cliente, ou por impedimentos técnicos não estudados anteriormente por parte da equipe técnica do cliente, ou até mesmo pelo surgimento de uma demanda urgente que deveria ser priorizada a frente de qualquer outra. Nesse cenário o Scrum Master se viu em um conflito entre representar o cliente e garantir que as equipes estejam trabalhando de forma focada e sem sobrecarga gerou grande estresse e desgaste no Scrum Master.

Outro conflito de interesses que foi possível notar com a junção desses papéis ocorreu por conta de problemas e atrasos no desenvolvimento. Isso aconteceu pois na área de desenvolvimento de software principalmente em ambientes ágeis é de conhecimento que nem sempre o código funciona como o esperado, bugs e erros podem aparecer e causar retrabalho e muitas vezes atraso nas entregas. No papel de Scrum Master já é esperado que problemas possam surgir, porém quando o profissional também atua como Product Owner e está em contato direto com o cliente, pode-se perceber que existiu uma dificuldade em não ceder a pressão do prazo e da urgência da demanda. Esse item em específico causou impaciência do lado dos stakeholders e também insatisfação nas equipes de desenvolvimento ao serem pressionadas.

O acúmulo desses papéis também levou a uma sobrecarga de trabalho para o Scrum Master, afetando a qualidade do trabalho e o tempo que ele tinha disponível para o apoio às equipes. Por ter uma relação de Product Owner com o cliente, o Scrum Master foi muito requisitado durante todo o processo de desenvolvimento, seja para relatar algum problema no produto, ou para inclusão de novas demandas no product backlog. Com

tantas responsabilidades, foi possível perceber uma queda na produtividade quanto as tarefas originalmente de responsabilidade do Scrum Master, como o apoio a equipe.

Contudo, foi possível observar uma vantagem bem expressiva pela proximidade desse profissional com o cliente e com as equipes, criando assim uma ligação e interação entre todos, trabalhando juntos em prol de entregar produtos com qualidade e atendendo as necessidades e expectativas do cliente. Os times de desenvolvimento foram profundamente envolvidos nas tomadas de decisões e tiveram a liberdade de opinar sobre cada item a ser priorizado, para que todos entendessem seu valor e os desafios que haveriam na construção dessa tarefa do backlog. Esse envolvimento do time nas decisões e também na construção do projeto em si, respeitando e ouvindo as opiniões e sugestões deles criou um ambiente de trabalho colaborativo e adaptativo. O mesmo ocorreu com os stakeholders e o cliente, por conta da proximidade maior com os profissionais técnicos, eles passaram a entender melhor o processo de desenvolvimento e a complexidade que alguns itens do backlog representavam.

Após dois anos e meio da implementação desta fusão de papéis, as equipes se adaptaram ao ritmo de trabalho, superando os desafios iniciais. O projeto, apesar dos percalços, é considerado um sucesso, gerando benefícios tangíveis para o cliente e experiências enriquecedoras para a equipe. As vantagens e desvantagens encontradas na gestão desses projetos devido a junção de papéis podem ser sumarizadas na tabela 3.

<b>Vantagens</b>	<b>Desvantagens</b>
Eficiência na comunicação entre a equipe do projeto, os stakeholders e o cliente.	Conflito de interesses devido a divergência de foco dos dois papéis.
Uma visão unificada sobre o projeto, combinando a visão estratégica do produto com a facilitação eficaz do processo.	Sobrecarga de trabalho em um profissional, o que pode afetar a qualidade da gestão do projeto e do produto.
A fusão dos papéis pode agilizar o processo de tomada de decisão, já que a mesma pessoa compreende as necessidades do cliente e as limitações da equipe.	Risco de que as necessidades da equipe sejam negligenciadas em favor dos requisitos do produto, ou vice-versa, devido à parcialidade inerente a um papel duplo.
Entrega de valor mais rápida e alinhada com as expectativas do cliente.	Falta de objetividade e imparcialidade pelo Scrum Master devido a constante alteração das prioridades.
	Dificuldade em lidar com a pressão do cliente e da equipe.

Tabela 3 – Vantagens e desvantagens da junção de papéis do Scrum Master e Product Owner

Elaborado pela autora (2023)

## 4.4 Pesquisa

Para fim de entender os impactos que essa junção de papéis teve, foi submetida uma pesquisa para todos os membros das duas equipes e para alguns outros profissionais da área de TI da mesma Empresa X que já tiveram alguma vivência parecida em seus próprios projetos. Essa pesquisa teve como foco avaliar pontos importantes da percepção desses profissionais e do andamento de projetos quando se tem essa organização da equipe onde o Scrum Master desempenha a função do Product Owner também. As perguntas desse questionário e as respostas para cada uma delas estão em anexo no final desse trabalho.

Essa pesquisa foi realizada com 38 profissionais, dos quais a maioria possui formação na área e mais de cinco anos de experiência, e revelou percepções valiosas sobre os papéis no âmbito do Scrum. O questionário completo e as respostas obtidas em cada pergunta podem ser encontrados ao final desse trabalho, nessa seção serão explorados apenas os dados mais relevantes sobre a percepção desses profissionais sobre a junção de papéis do Scrum Master e do Product Owner.

Sendo assim, quanto à junção dos papéis de Scrum Master e Product Owner, 73,7% (28 respostas) dos respondentes acreditam que essa sobreposição de funções pode prejudicar a clareza nas prioridades e definições do projeto, como representado nas figuras 7 e 8.

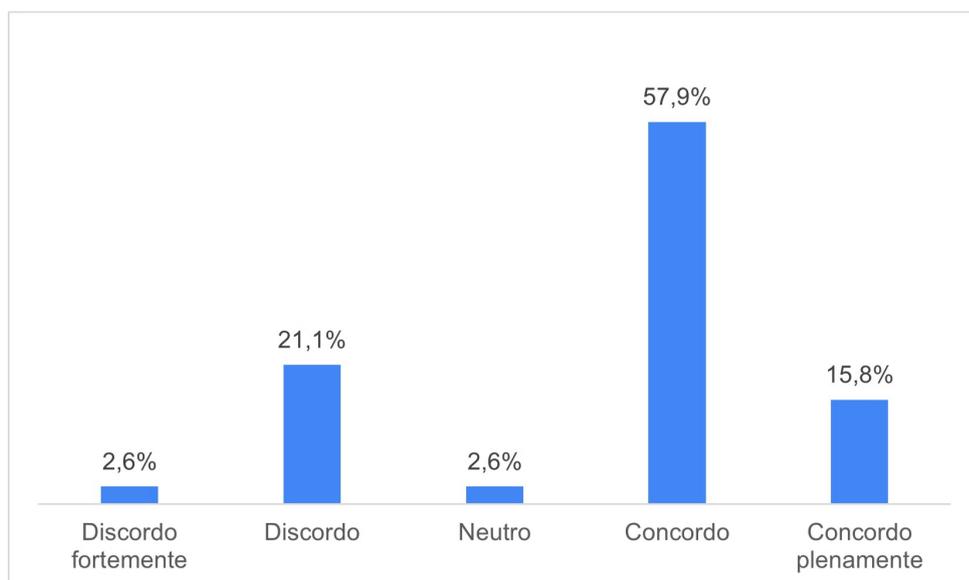


Figura 7 – Na sua opinião, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza das definições do projeto?

análise feita pela autora através das respostas da pesquisa

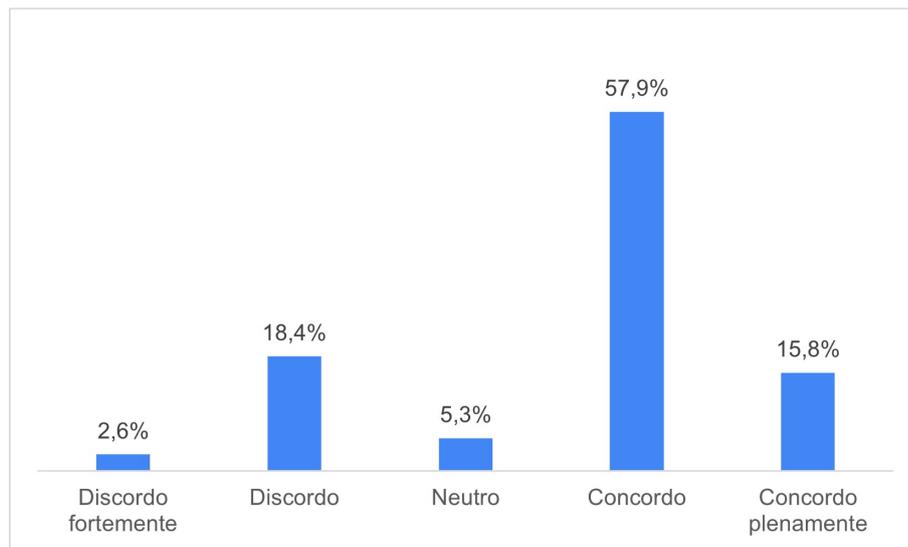


Figura 8 – Para você, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza nas prioridades do projeto?

análise feita pela autora através das respostas da pesquisa

Além disso, de acordo com a figura 9 pode-se afirmar que 86,9% (33 respostas) expressam preocupação de que tal junção comprometa a disponibilidade do Scrum Master para prestar apoio para a equipe no dia a dia do projeto.

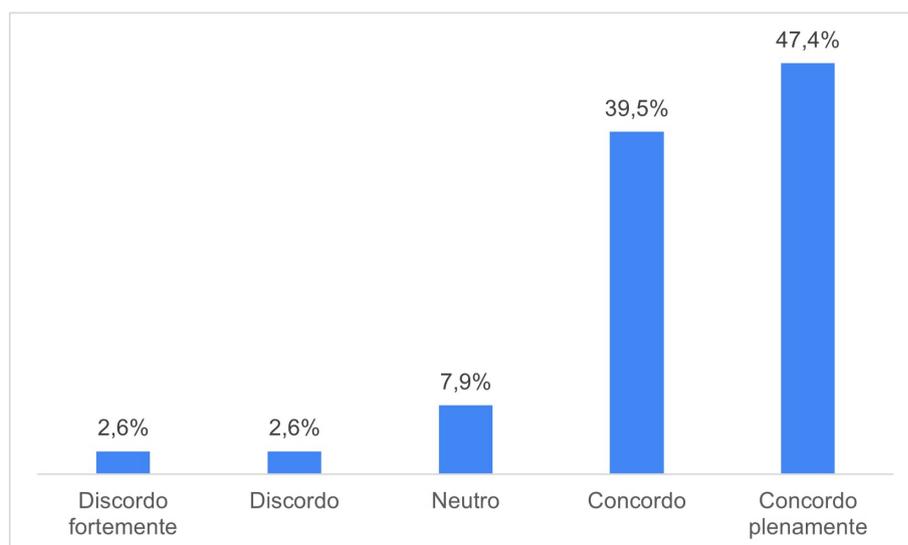


Figura 9 – Na sua opinião, a junção desses papéis do Scrum Master e do Product Owner influencia negativamente na disponibilidade do Scrum Master para auxiliar a equipe no dia a dia do projeto?

análise feita pela autora através das respostas da pesquisa

Curiosamente, apenas metade dos entrevistados (19 respostas) acredita que a qualidade do projeto é negativamente impactada por essa fusão de papéis, indicando uma divisão de opiniões sobre o impacto nesse ponto específico que está demonstrado na figura 10.

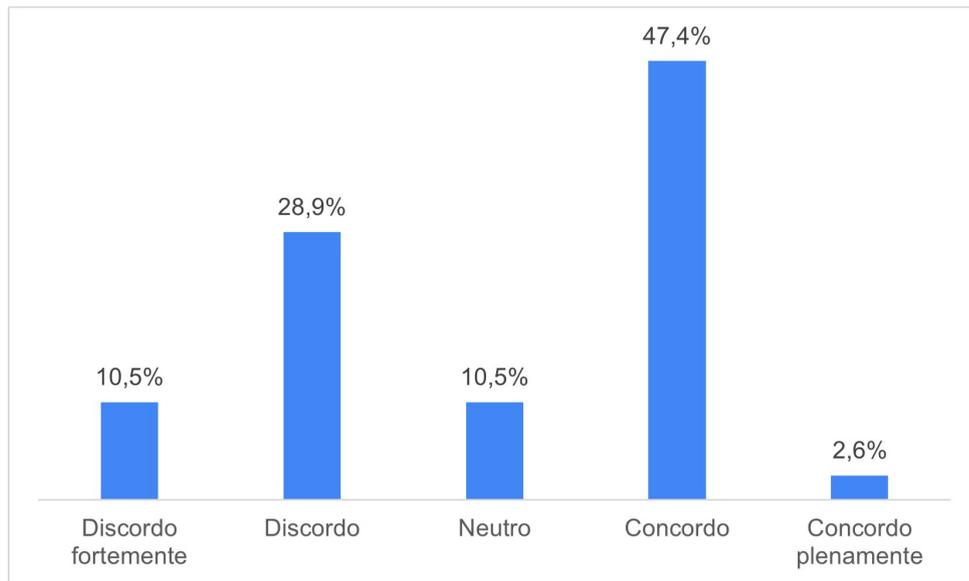


Figura 10 – Você acredita que a junção dos papéis do Scrum Master e do Product Owner afeta negativamente a qualidade das entregas da equipe?

análise feita pela autora através das respostas da pesquisa

Por outro lado, como pode ser visto na figura 11 na página seguinte, a maioria, com 65,8% (25 respostas), discorda da afirmação de que a combinação dos papéis afete a comunicação dentro da equipe.

Esses resultados evidenciam uma clara percepção dos profissionais sobre as funções e responsabilidades dentro do framework Scrum, principalmente se tratando dos papéis do Scrum Master e do Product Owner, assim como os possíveis efeitos da concentração de papéis em uma única pessoa, ressaltando a importância de manter papéis distintos para a eficácia da gestão ágil de projetos.

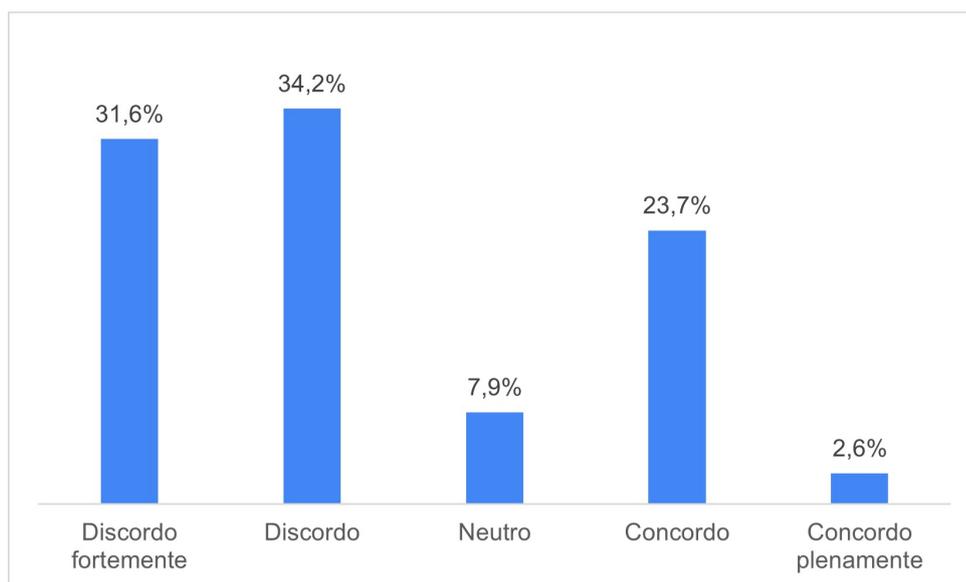


Figura 11 – Na sua percepção, a junção dos papéis de Scrum Master e Product Owner afeta negativamente a comunicação dentro da equipe?

análise feita pela autora através das respostas da pesquisa

## 5 Conclusão

Este trabalho dedicou-se a uma análise criteriosa da aplicação de diversas metodologias de gestão de projetos no contexto do desenvolvimento de software, com um enfoque especial nos papéis e responsabilidades dentro das equipes de desenvolvimento e na interação e sobreposição de funções que podem ocorrer em determinados projetos. Em particular, esse estudo se concentrou em desvendar a complexidade e os impactos emergentes quando os papéis do Scrum Master, que tradicionalmente atua como facilitador dos processos e removedor de obstáculos, e do Product Owner, encarregado de definir a visão e representar os interesses do cliente, são combinados. A união desses papéis representa uma fusão significativa de responsabilidades e prioridades, cujo impacto na eficiência da gestão de projetos foi o principal foco de estudo. Através desta investigação, procurou-se compreender como tal junção afeta a dinâmica da equipe, a tomada de decisões e o alinhamento geral com os objetivos do projeto, esperando oferecer uma visão aprofundada sobre as melhores práticas e estratégias para otimizar o gerenciamento em projetos de software ágeis.

Esse estudo analisou os benefícios e desafios inerentes à dinâmica resultante da junção de tarefas, na qual o Scrum Master desempenhou suas funções originais, bem como as de responsabilidade do Product Owner. Após o estudo de caso pode-se perceber que uma adaptação da metodologia ágil Scrum pode ser possível, apesar dos obstáculos encontrados pela equipe citada.

A pesquisa realizada com 38 profissionais da área da tecnologia forneceu dados empíricos que ajudaram a validar as observações do estudo de caso. A partir das respostas coletadas, ficou evidente que a grande maioria dos profissionais concorda com a ideia de que o Scrum Master é essencial para orientar a equipe e remover impedimentos, atuando também como coach do time. A pesquisa também destacou uma divergência considerável quanto à sua responsabilidade de definir prioridades, reiterando que o Scrum Master não é visto como o responsável por essa função, reforçando assim o estudo feito ao longo do trabalho sobre as definições e responsabilidades essenciais de cada profissional.

Além disso, a pesquisa também indicou uma forte convicção dos profissionais de que o Product Owner deve representar os interesses do cliente, definir as prioridades e os itens do Product Backlog. Essas percepções alinham-se com as funções tradicionalmente atribuídas ao Product Owner e reiteram a importância de sua posição estratégica no projeto. Quando se trata da junção dos papéis de Scrum Master e Product Owner, as respostas da pesquisa mostraram que uma porcentagem significativa de profissionais acredita que isso pode comprometer a clareza nas prioridades e definições do projeto, além

de afetar a disponibilidade do Scrum Master para apoiar a equipe cotidianamente.

Assim, a conclusão deste trabalho é que, embora a junção de papéis do Scrum Master e do Product Owner possa trazer uma tomada de decisão mais ágil e uma compreensão mais profunda das expectativas do cliente, além de aumentar a transparência e confiança por proporcionar uma conexão mais estreita entre o processo de desenvolvimento e as decisões estratégicas, ela também pode levar a conflitos de interesse e sobrecarga de trabalho, diminuindo assim a eficiência do Scrum Master nas suas funções originais. Esses desafios são potencializados quando um único profissional assume responsabilidades que naturalmente possuem objetivos e focos distintos.

A análise dessa equipe mostrou que combinar as funções do Scrum Master e do Product Owner pode ser uma estratégia viável, apesar de não ser o ideal, desde que seja executado com cuidado. A busca pelo equilíbrio entre os papéis é fundamental, ressaltando a importância de prevenir conflitos de interesses e evitar a sobrecarga do profissional que assume essas duas funções. Em suma, este trabalho contribui para uma compreensão mais aprofundada das dinâmicas entre o Scrum Master e o Product Owner, oferecendo percepções valiosas sobre como otimizar a gestão de projetos em ambientes ágeis.

Para pesquisas futuras, pode ser valioso ampliar o escopo de análise sobre a junção dos papéis de Scrum Master e Product Owner em contextos de projetos mais amplos e complexos, particularmente em equipes maiores onde as dinâmicas de trabalho e os desafios associados podem variar significativamente. Outra linha de investigação promissora poderia explorar cenários onde o Product Owner é um profissional do cliente, que não tem conhecimento aprofundado sobre o processo de desenvolvimento de software. Este arranjo oferece um terreno fértil para examinar como a falta de familiaridade desse papel com as práticas internas de desenvolvimento pode impactar a gestão do projeto e a colaboração entre a equipe e o cliente. Além disso, poderia ser interessante avaliar a fusão de um desses papéis - Scrum Master ou Product Owner - com outros papéis dentro da equipe de desenvolvimento de software. Tal abordagem poderia potencialmente otimizar a eficiência da equipe, minimizando conflitos de interesse e maximizando o aproveitamento dos recursos e habilidades disponíveis. Essas investigações podem proporcionar outras visões, e talvez até mais profundas sobre estratégias eficazes de gerenciamento de projetos em ambientes de desenvolvimento de software ágeis e complexos.

# Referências

- ADKINS, L. **Coaching Agile Teams: A Companion for ScrumMaster, Agile Coaches, and Project Managers in Transition**. 1. ed. [S.l.]: Addison-Wesley Professional, 2010. ISBN 0321660358, 9780321660350. Citado na página 44.
- AMARAL, L. **MetodologiaScrum**. 2022. Disponível em: <<https://www.tabnews.com.br/LucasAmaral/o-que-sao-metodologias-ageis>>. Acesso em: 27 out 2023. Citado na página 27.
- ANDERSON, D. J. **Kanban: Successful Evolutionary Change for Your Technology Business**. Blue book ed. [S.l.]: Blue Hole Press, 2010. ISBN 0984521402, 978-0984521401. Citado 2 vezes nas páginas 30 e 31.
- ANDRADE, S. C. de; TAIT, T. F. C. Uma aplicação do guia pmbok na gestão de projetos de software. **Revista Brasileira de Computação Aplicada**, v. 4, 2012. Citado na página 19.
- APPELO, J. Management 3.0: Leading agile developers, developing agile leaders. **Journal of Chemical Information and Modeling**, v. 53, 2013. ISSN 1098-6596. Citado na página 33.
- ASHBACHER, C. Succeeding with agile: Software development using scrum, by mike cohn. **The Journal of Object Technology**, v. 9, 2010. Citado na página 33.
- BASILI, V. R.; LARMAN, C. Iterative and incremental development: A brief history. **Computer**, v. 36, 2003. Citado na página 22.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice Second Edition Third Edition**. [S.l.: s.n.], 2013. Citado na página 36.
- BECK, K. Embracing change with extreme programming. **Computer**, v. 32, 1999. ISSN 00189162. Citado na página 21.
- BECK, K.; BEEDLE, M.; BENNEKUM, A. V.; 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. Manifesto for agile software development. **The Agile Alliance**, v. 2009, 2001. Citado na página 21.
- BORGES, P. M. P. Configuração do rup com vista à simplificação dos elencos processuais em pmes de desenvolvimento de software. **Disseração de Mestrado - <https://hdl.handle.net/1822/8062>**, 2008. Citado 3 vezes nas páginas 25, 26 e 41.
- COBB, C. G. **Making Sense of Agile Project Management: Balancing Control and Agility**. [S.l.: s.n.], 2011. Citado 2 vezes nas páginas 21 e 23.
- COCKBURN, A. **Writing effective use cases**. [S.l.]: Pearson Education India, 2001. Citado na página 35.

- \_\_\_\_\_. **Agile software development: the cooperative game**. 2. ed. [S.l.]: Pearson Education, 2006. v. 113. ISBN 032148275, 978-0321482754. Citado na página 21.
- COCKBURN, A.; HIGHSMITH, J. Agile software development: The people factor. **Computer**, v. 34, p. 131–133, 11 2001. ISSN 00189162. Citado na página 34.
- COHN, M. Agile estimating and planning. In: . [S.l.]: Pearson, 2005. ISBN 9780131479418, 978-0131479418. ISSN 03579387. Citado 2 vezes nas páginas 12 e 19.
- CONSORTIUM, A. B. The dsdm agile project framework handbook. **Agile Business Consortium**, 2014. Citado na página 39.
- \_\_\_\_\_. **PapeisAgilePM**. 2014. Disponível em: <<https://www.agilebusiness.org/dsdm-project-framework/roles-and-responsibilities.html>>. Acesso em: 10 nov 2023. Citado na página 42.
- \_\_\_\_\_. **Agile Project Management Handbook v2**. [S.l.: s.n.], 2017. Citado 3 vezes nas páginas 22, 23 e 41.
- CRUZ, F. **Scrum e PMBOK unidos no Gerenciamento de Projetos**. [S.l.]: Brasport, 2013. Citado na página 16.
- DAVIDSON, A.; KLEMME, L. **Why a ceo should think like a scrum master**. 2016. Citado na página 44.
- DAVIS, C.; SCHILLER, M.; WHEELER, K. **IT Auditing : Using Controls to Protect Information Assets**. [S.l.: s.n.], 2011. Citado na página 37.
- DINGSØYR, T.; NERUR, S.; BALIJEPALLY, V.; MOE, N. B. The journal of systems and software a decade of agile methodologies: Towards explaining agile software development. **The Journal of Systems e Software**, v. 85, 2012. Citado na página 21.
- DUSTING, E. Effective software testing. **Mind**, 2002. Citado 2 vezes nas páginas 35 e 36.
- DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: A systematic review. **Information and Software Technology**, v. 50, p. 833–859, 8 2008. ISSN 09505849. Citado na página 19.
- EMPRESA, B. A. M. **Kanban**. 2021. Disponível em: <<https://abriminhaempresa.com/kanban-o-que-e/>>. Acesso em: 20 out 2023. Citado na página 31.
- EREIZ, Z.; MUSIC, D. Scrum without a scrum master. **2019 IEEE International Conference on Computer Science and Educational Informatization, CSEI 2019**, 2019. Citado na página 14.
- FOWLER, M. The new methodology. **Software World**, v. 36, 2005. ISSN 00380652. Citado na página 20.
- FREEMAN, R. E. **Strategic management: A stakeholder approach**. [S.l.: s.n.], 2015. Citado na página 37.
- GONÇALVES, C. D. L. Ajustamento dos papéis do rup reduzido no desenvolvimento de projetos em contexto acadêmico. **Dissertação de Mestrado - <https://hdl.handle.net/1822/42600>**, 2016. Citado 2 vezes nas páginas 24 e 41.

- HIGHSMITH, J. Innovative product development jim highsmith. **Saylor.org**, 2004. Citado na página 21.
- KARDEC, M. S. Estudo de compatibilidade entre pmbok e scrum. **Revista Tecnologias em Projeção**, v. 3, 2012. ISSN 2178-6267. Citado 2 vezes nas páginas 29 e 45.
- KNIBERG, H. Scrum and xp from the trenches. **Reading**, 2007. ISSN 14747456. Citado na página 12.
- KNIBERG, H.; SKARIN, M. **Kanban and Scrum-making the most of both**. [S.l.]: Lulu Press, 2010. v. 1. ISSN 1098-6596. ISBN 0557138329, 978-0557138326. Citado 3 vezes nas páginas 30, 31 e 32.
- KOLESNIKOVA KATERYNA E LUKIANOV, D. Análise da eficácia da combinação das funções de scrum master e product owner em equipes scrum. **Herald of Advanced Information Technology**, v. 1, 2020. Citado na página 14.
- KRUCHTEN, P. **The rational Unified Process An Introduction Third Edition**. [S.l.: s.n.], 2003. Citado 2 vezes nas páginas 24 e 41.
- KRUG, S. **Don't Make Me Think Revisited: A Common Sense Approach to Web and Mobile Usability**. [S.l.: s.n.], 2014. ISSN 07338627. Citado na página 36.
- LIMA, A. P. D. R.; AMARAL, D. C. O que fazer quando não há product owner? um estudo sobre a ausência desse papel. **Revista de Gestão e Projetos**, v. 13, 2022. ISSN 22360972. Citado 2 vezes nas páginas 14 e 50.
- MACHADO, M.; MEDINA, S. G. Scrum – método Ágil: uma mudança cultural na gestão de projetos de desenvolvimento de software. **Intraciência**, 2009. Citado na página 27.
- MOE, N. B.; DINGSØYR, T.; DYBÅ, T. A teamwork model for understanding an agile team: A case study of a scrum project. **Information and Software Technology**, v. 52, 2010. ISSN 09505849. Citado na página 33.
- PMI, P. M. I. **Project Management Body of Knowledge (PMBOK®) 6th Edition**. [S.l.: s.n.], 2017. v. 40. Citado 6 vezes nas páginas 16, 17, 18, 19, 23 e 37.
- PRIKLADNICKI, R.; WILLI, R.; MILANI, F. **Métodos Ágeis para Desenvolvimento de Software**. [S.l.: s.n.], 2014. Citado na página 19.
- RAMOS, A. B.; JUNIOR, D. C. V. A influência do papel do scrum master no desenvolvimento de projetos scrum. **Revista de Gestão e Projetos**, v. 08, 2017. ISSN 22360972. Citado na página 12.
- RATIONAL. Best practices for software development teams. **A Rational Software Corporation White Paper**, 1998. Citado 3 vezes nas páginas 24, 25 e 41.
- REIS, M. **ProcessosPMBOK**. 2014. Disponível em: <<https://www.profissionaisti.com.br/voce-conhece-o-guia-pmbok/>>. Acesso em: 15 out 2023. Citado na página 17.
- SABBAGH, R. Scrum gestão Ágil para projetos de sucesso. **Casadocodigo**, 2014. ISSN 13514180. Citado 2 vezes nas páginas 12 e 45.

- SCHWABER, K.; BEEDLE, M. **Agile Software Development with Scrum**. Prentice Hall, 2001. 158 p. ISBN 978-0130676344. Disponível em: <[http://sutlib2.sut.ac.th/sut\\_contents/H129174.pdf](http://sutlib2.sut.ac.th/sut_contents/H129174.pdf)>. Citado 2 vezes nas páginas 28 e 45.
- SCHWABER, K.; SUTHERLAND, J. **Guia do SCRUM - Um guia definitivo para o Scrum: As regras do jogo**. [S.l.: s.n.], 2016. IV. Citado 2 vezes nas páginas 28 e 29.
- SHRIVASTAVA, S. V.; RATHOD, U. Risks in distributed agile development: A review. **Procedia - Social and Behavioral Sciences**, v. 133, 2014. ISSN 18770428. Citado na página 45.
- SILVA, E. C. da; LOVATO, L. A. Framework scrum: Eficiência em projetos de software. **Revista de Gestão e Projetos**, v. 07, 2016. ISSN 22360972. Citado 2 vezes nas páginas 27 e 44.
- SOMMERVILLE, I. **Software engineering (10th edition)**. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 34 e 36.
- SOPHIA, A. **Metodologias Tradicionais X Ageis**. 2021. Disponível em: <<https://www.edrawsoft.com/pt/agile-vs-waterfall.html>>. Acesso em: 15 out 2023. Citado na página 20.
- STAPLETON, J. **DSDM, Dynamic Systems Development Method: The Method in Practice**. [S.l.: s.n.], 1997. Citado na página 22.
- SUTHERLAND, J. **Scrum: A arte de fazer o dobro do trabalho na metade do tempo**. 1. ed. [S.l.]: Grupo LeYa, 2014. ISBN 9788544100882. Citado 2 vezes nas páginas 12 e 30.
- SVERRISDOTTIR, H. S.; INGASON, H. T.; JONASSON, H. I. The role of the product owner in scrum-comparison between theory and practices. **Procedia - Social and Behavioral Sciences**, v. 119, 2014. ISSN 18770428. Citado 2 vezes nas páginas 12 e 45.
- VARGAS, L. M. Gerenciamento Ágil de projetos em desenvolvimento de software: Um estudo comparativo sobre a aplicabilidade do scrum em conjunto com pmbok e/ou prince2. **Revista de Gestão e Projetos**, v. 07, 2016. ISSN 22360972. Citado 2 vezes nas páginas 16 e 38.
- VERSIONONE.COM. 11th annual state of agile report. **VersionOne Agile Annual Report**, 2017. ISSN 1473-6691. Citado na página 12.
- WATTS, G. **Scrum Mastery**. [S.l.: s.n.], 2012. Citado na página 44.
- WIEGERS, K. E.; BEATTY, J. **Software Requirements 3rd Ed.** 3. ed. [S.l.: s.n.], 2013. ISBN 978-0-7356-7966-5. Citado na página 35.
- WIKIPÉDIA, C. da. **FluxosRUP**. 2022. Disponível em: <[https://pt.wikipedia.org/w/index.php?title=IBM\\_Rational\\_Unified\\_Process&oldid=64705823](https://pt.wikipedia.org/w/index.php?title=IBM_Rational_Unified_Process&oldid=64705823)>. Acesso em: 20 out 2023. Citado na página 26.

# Anexos

# ANEXO A – Pesquisa para avaliação da junção de papéis do Scrum Master e do Product Owner

Essa pesquisa foi desenvolvida para avaliar as funções de cada um dos papéis em um projeto de desenvolvimento de software e para analisar as vantagens e desvantagens da junção das tarefas do Scrum Master e do Product Owner em um só profissional.

As respostas são anônimas e para fins acadêmicos, peço que respondam de acordo com seus conhecimentos e vivências dentro de um projeto de desenvolvimento de software que utiliza da metodologia Scrum.

As 3 primeiras questões foram elaboradas para conhecimento do público que estaria respondendo a pesquisa, visto que as respostas de todos que participaram são anônimas. O restante da pesquisa visou nos conhecimentos sobre os papéis do Scrum Master e do Product Owner, avaliando se o público abordado conhecia cada característica dos papéis, além de questionar sobre os impactos que a junção dos mesmos poderia causar em um determinado projeto. E a última questão era dissertativa e buscava coletar sugestões e comentários sobre essa junção de papéis do Scrum Master e do Product Owner.

As questões poderiam ser respondidas com (1) Discordo fortemente, (2) Discordo, (3) Neutro, (4) Concordo e (5) Concordo plenamente.

1. Qual é sua área de formação?
2. Qual é o seu cargo atual?
3. Quanto tempo você tem atuado na área de TI?
4. Em sua opinião somente o Scrum Master é responsável por orientar a equipe e remover os impedimentos do processo de desenvolvimento?
5. Para você, é tarefa somente do Scrum Master definir as prioridades do projeto?
6. Na sua visão, é de responsabilidade somente do Scrum Master facilitar a comunicação entre a equipe e o cliente dentro do projeto?
7. Para você apenas o Scrum Master tem a autoridade final para tomar decisões sobre as prioridades do backlog do projeto?

8. Para você apenas o Scrum Master deve atuar como coach da equipe, apoiando e removendo obstáculos quando necessário?
9. Na sua opinião somente o Product Owner é responsável por representar os interesses e necessidades do cliente dentro do projeto?
10. Para você, é tarefa somente do Product Owner definir as prioridades do projeto?
11. Você acredita que é papel exclusivamente do Product Owner a definição dos itens do Product Backlog do projeto?
12. Na sua visão, é de responsabilidade somente do Product Owner facilitar a comunicação entre a equipe e o cliente dentro do projeto?
13. Para você, a presença de um Product Owner que seja do cliente e não membro do time, é benéfico para o projeto?
14. Caso esse Product Owner do cliente não tenha conhecimento sobre o processo de desenvolvimento de software, isso pode atrapalhar o andamento do projeto na sua visão?
15. Na sua opinião, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza das definições do projeto?
16. Para você, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza nas prioridades do projeto?
17. A capacidade da equipe de receber orientações claras sobre as tarefas a serem desenvolvidas é afetada negativamente pela junção de papéis do Scrum Master e do Product Owner na sua visão?
18. Você acredita que a junção dos papéis do Scrum Master e do Product Owner afeta negativamente a qualidade das entregas da equipe?
19. Para você, essa junção de papéis do Scrum Master e do Product Owner influencia negativamente a capacidade da equipe de lidar com mudanças nos requisitos do projeto?
20. Na sua percepção, a junção dos papéis de Scrum Master e Product Owner afeta negativamente a comunicação dentro da equipe?
21. Na sua opinião, a junção desses papéis do Scrum Master e do Product Owner influencia negativamente na disponibilidade do Scrum Master para auxiliar a equipe no dia a dia do projeto?

22. Na sua visão, com a junção desses papéis de Scrum Master e Product Owner o time pode fazer entregas mais alinhadas com a expectativa do cliente?
23. Você tem alguma sugestão ou comentário adicional sobre a junção de papéis do Scrum Master e Product Owner que gostaria de compartilhar?

## ANEXO B – Respostas da pesquisa para avaliação da junção de papéis do Scrum Master e do Product Owner

A pesquisa detalhada no Anexo A, que foi elaborada com o intuito de avaliar a percepção sobre eficácia da junção dos papéis de Scrum Master e Product Owner, obteve resposta de 38 profissionais da área da tecnologia. Este grupo diversificado de respondentes incluiu não apenas membros ativos das equipes mencionadas no estudo de caso, mas também outros profissionais experientes da Empresa X. As respostas coletadas são variadas e refletem uma ampla gama de percepções e experiências, fornecendo revelações valiosas sobre a dinâmica e os impactos dessa prática de gestão de projetos. São elas:

### 1. Qual é sua área de formação?

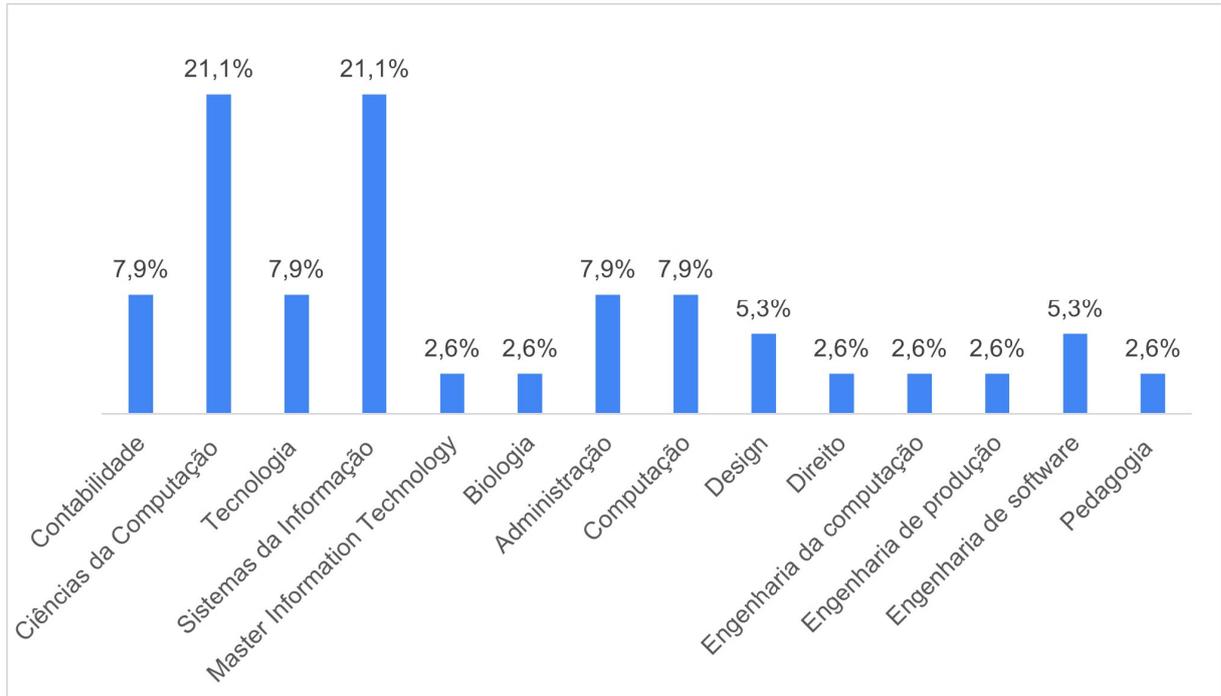


Figura 12 – Resposta 1 - Qual é sua área de formação?  
análise feita pela autora através das respostas da pesquisa

### 2. Qual é o seu cargo atual?

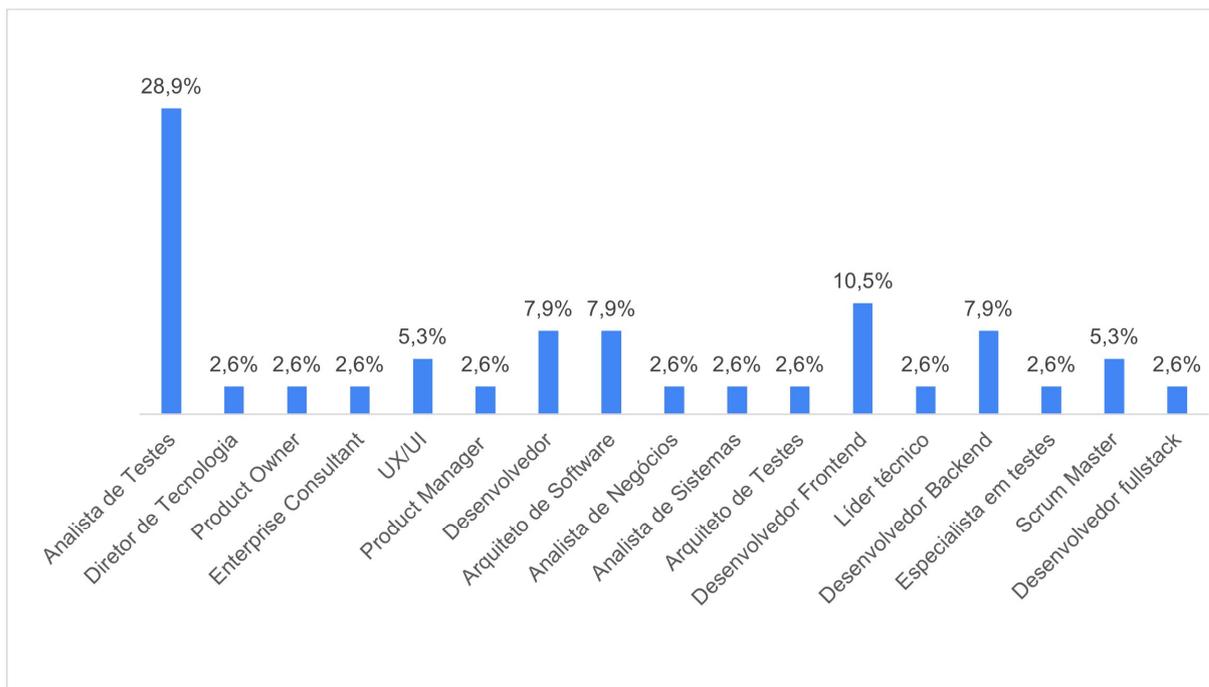


Figura 13 – Resposta 2 - Qual é o seu cargo atual?  
análise feita pela autora através das respostas da pesquisa

3. Quanto tempo você tem atuado na área de TI?

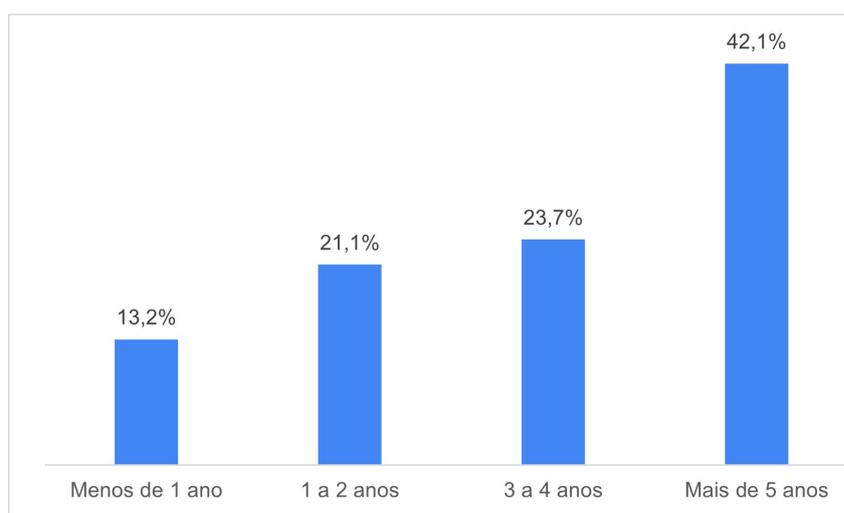


Figura 14 – Resposta 3 - Quanto tempo você tem atuado na área de TI?  
análise feita pela autora através das respostas da pesquisa

4. Em sua opinião somente o Scrum Master é responsável por orientar a equipe e remover os impedimentos do processo de desenvolvimento?

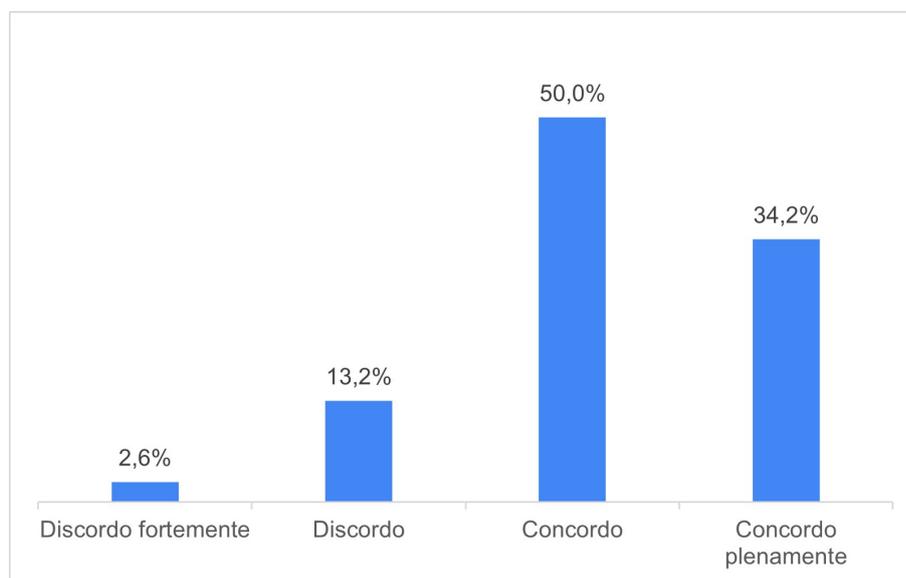


Figura 15 – Resposta 4 - Em sua opinião somente o Scrum Master é responsável por orientar a equipe e remover os impedimentos do processo de desenvolvimento?  
análise feita pela autora através das respostas da pesquisa

5. Para você, é tarefa somente do Scrum Master definir as prioridades do projeto?

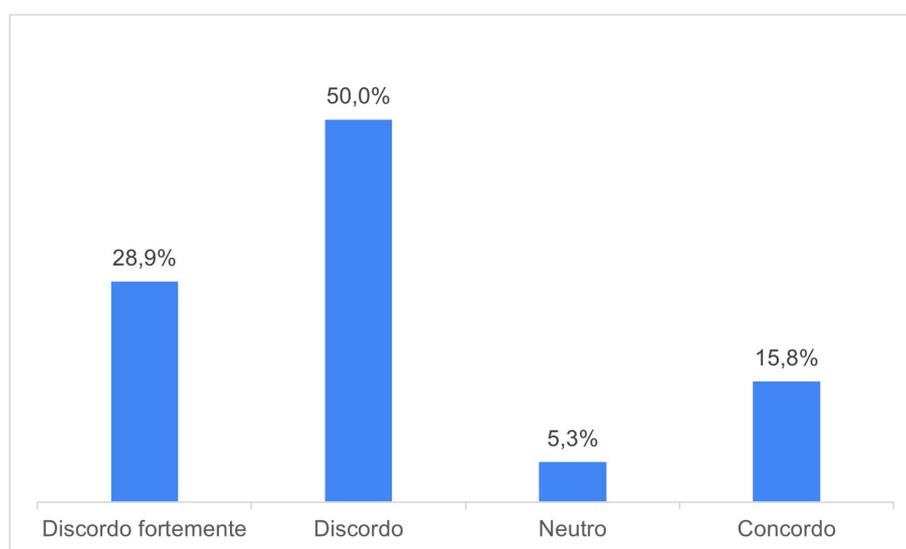


Figura 16 – Resposta 5 - Para você, é tarefa somente do Scrum Master definir as prioridades do projeto?  
análise feita pela autora através das respostas da pesquisa

6. Na sua visão, é de responsabilidade somente do Scrum Master facilitar a comunicação entre a equipe e o cliente dentro do projeto?

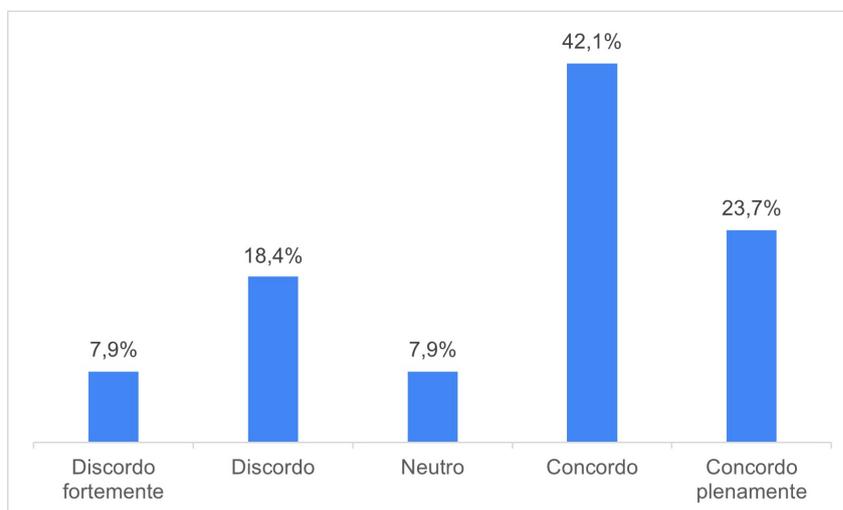


Figura 17 – Resposta 6 - Na sua visão, é de responsabilidade somente do Scrum Master facilitar a comunicação entre a equipe e o cliente dentro do projeto?

análise feita pela autora através das respostas da pesquisa

7. Para você apenas o Scrum Master tem a autoridade final para tomar decisões sobre as prioridades do backlog do projeto?

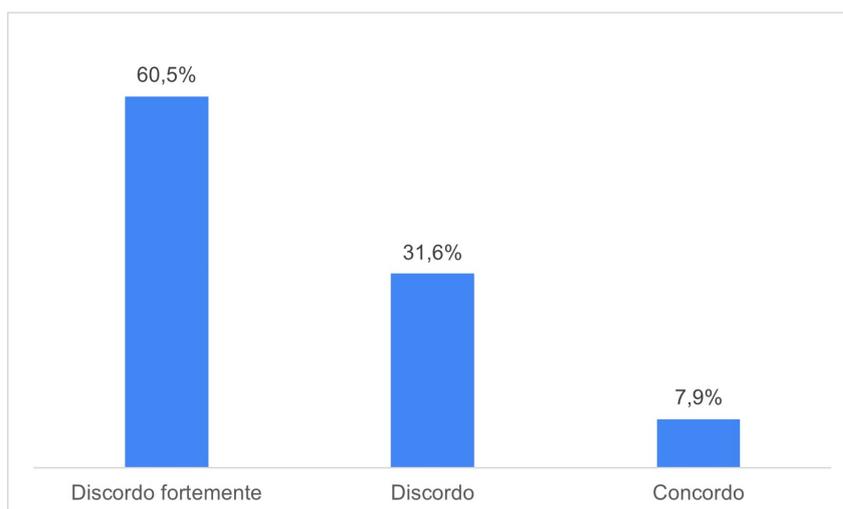


Figura 18 – Resposta 7 - Para você apenas o Scrum Master tem a autoridade final para tomar decisões sobre as prioridades do backlog do projeto?

análise feita pela autora através das respostas da pesquisa

8. Para você apenas o Scrum Master deve atuar como coach da equipe, apoiando e removendo obstáculos quando necessário?

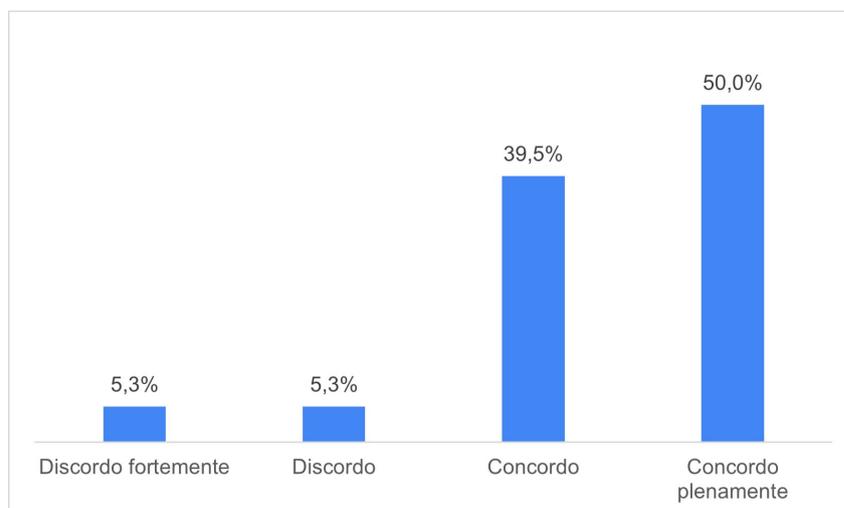


Figura 19 – Resposta 8 - Para você apenas o Scrum Master deve atuar como coach da equipe, apoiando e removendo obstáculos quando necessário?

análise feita pela autora através das respostas da pesquisa

9. Na sua opinião somente o Product Owner é responsável por representar os interesses e necessidades do cliente dentro do projeto?

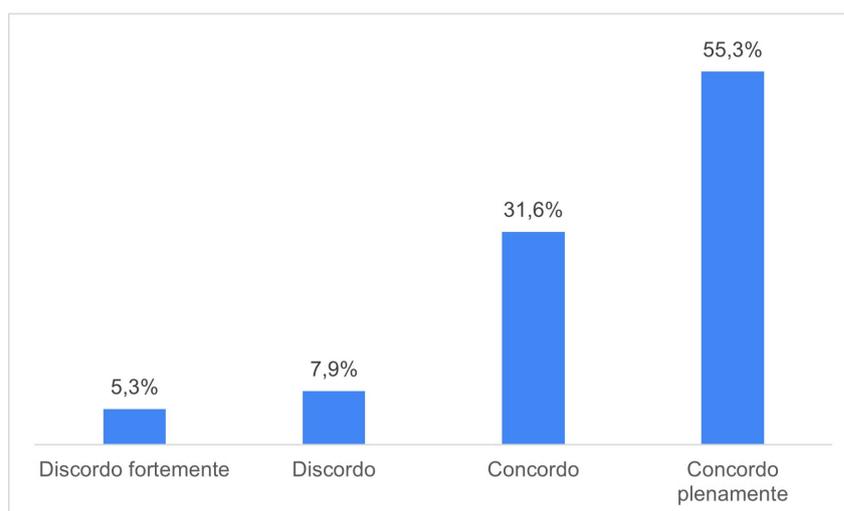


Figura 20 – Resposta 9 - Na sua opinião somente o Product Owner é responsável por representar os interesses e necessidades do cliente dentro do projeto?

análise feita pela autora através das respostas da pesquisa

10. Para você, é tarefa somente do Product Owner definir as prioridades do projeto?

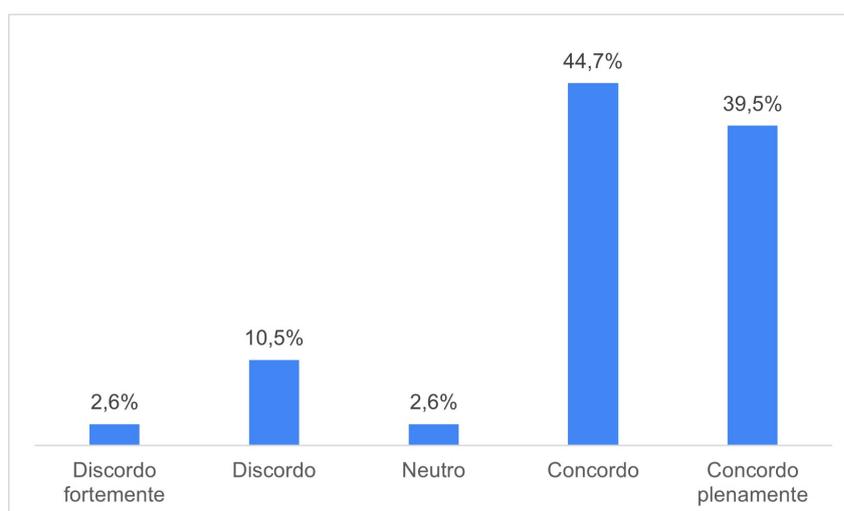


Figura 21 – Resposta 10 - Para você, é tarefa somente do Product Owner definir as prioridades do projeto?

análise feita pela autora através das respostas da pesquisa

11. Você acredita que é papel exclusivamente do Product Owner a definição dos itens do Product Backlog do projeto?

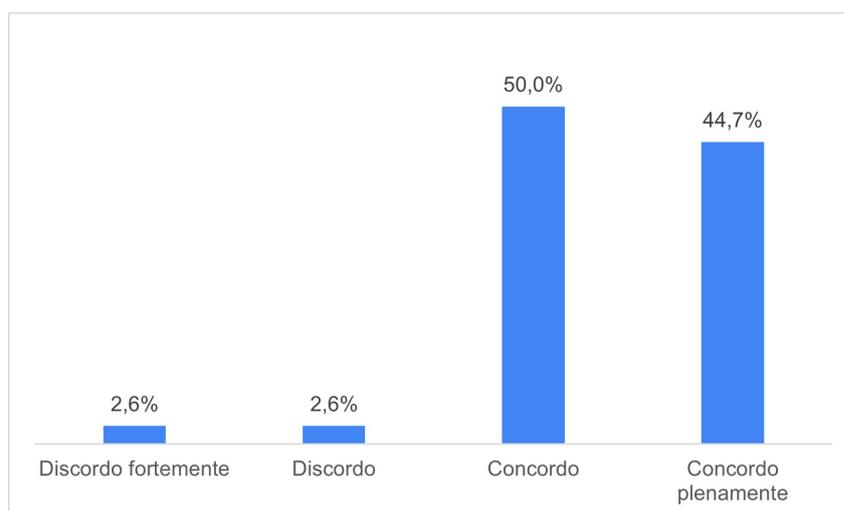


Figura 22 – Resposta 11 - Você acredita que é papel exclusivamente do Product Owner a definição dos itens do Product Backlog do projeto?

análise feita pela autora através das respostas da pesquisa

12. Na sua visão, é de responsabilidade somente do Product Owner facilitar a comunicação entre a equipe e o cliente dentro do projeto?

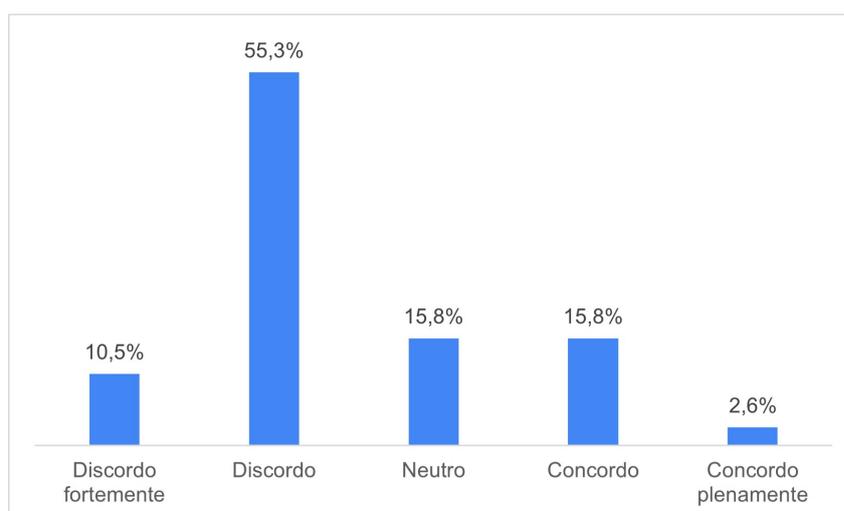


Figura 23 – Resposta 12 - Na sua visão, é de responsabilidade somente do Product Owner facilitar a comunicação entre a equipe e o cliente dentro do projeto?

análise feita pela autora através das respostas da pesquisa

13. Para você, a presença de um Product Owner que seja do cliente e não membro do time, é benéfico para o projeto?

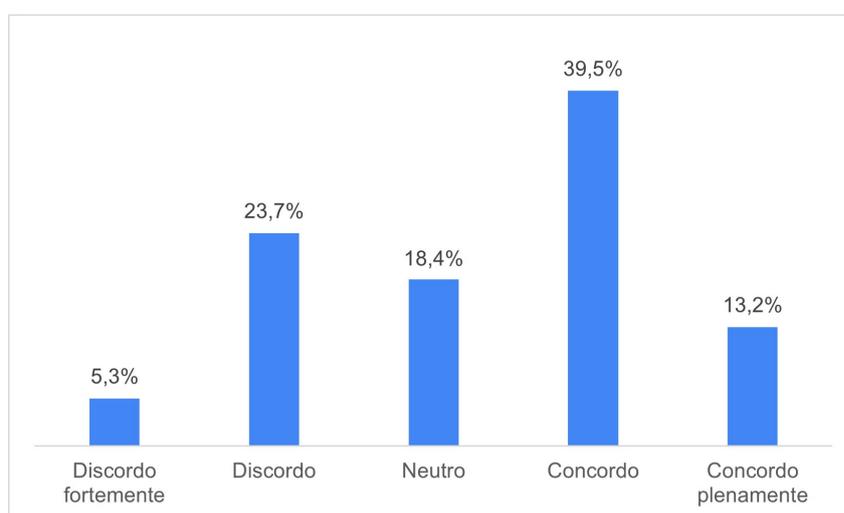


Figura 24 – Resposta 13 - Para você, a presença de um Product Owner que seja do cliente e não membro do time, é benéfico para o projeto?

análise feita pela autora através das respostas da pesquisa

14. Caso esse Product Owner do cliente não tenha conhecimento sobre o processo de desenvolvimento de software, isso pode atrapalhar o andamento do projeto na sua visão?

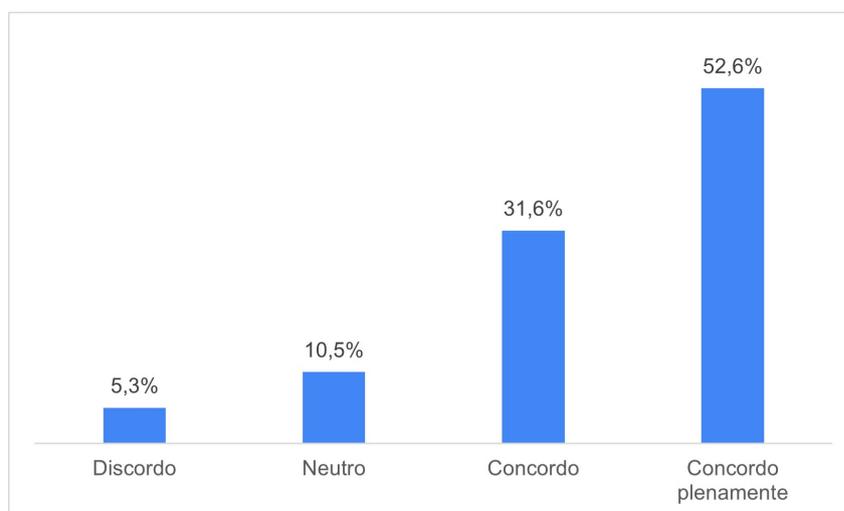


Figura 25 – Resposta 14 - Caso esse Product Owner do cliente não tenha conhecimento sobre o processo de desenvolvimento de software, isso pode atrapalhar o andamento do projeto na sua visão?

análise feita pela autora através das respostas da pesquisa

15. Na sua opinião, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza das definições do projeto?

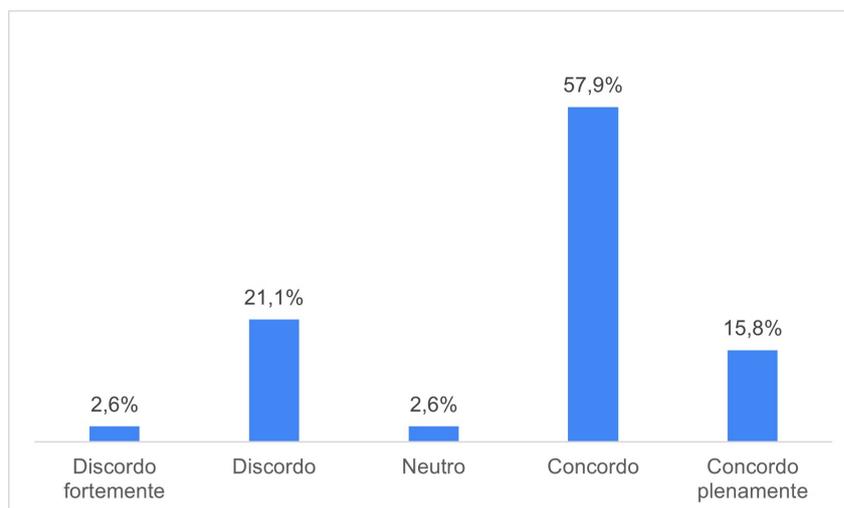


Figura 26 – Resposta 15 - Na sua opinião, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza das definições do projeto?

análise feita pela autora através das respostas da pesquisa

16. Para você, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza nas prioridades do projeto?

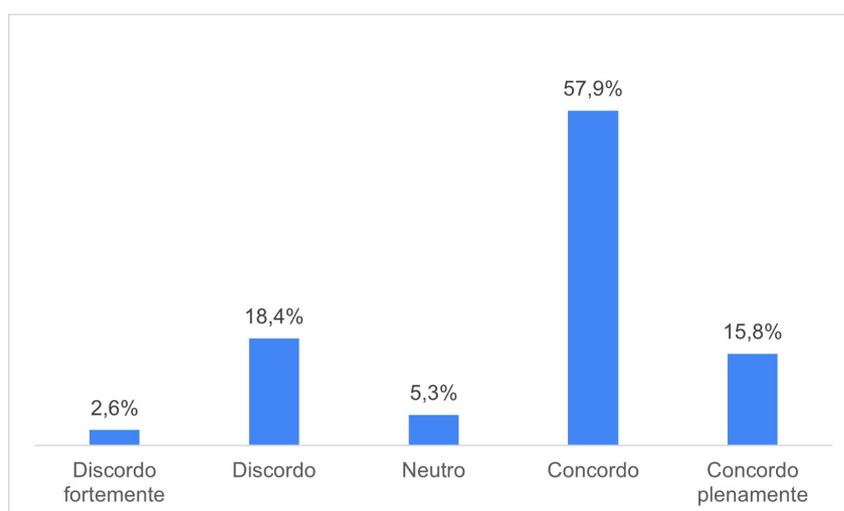


Figura 27 – Resposta 16 - Para você, a junção dos papéis de Scrum Master e Product Owner prejudica a clareza nas prioridades do projeto?

análise feita pela autora através das respostas da pesquisa

17. A capacidade da equipe de receber orientações claras sobre as tarefas a serem desenvolvidas é afetada negativamente pela junção de papéis do Scrum Master e do Product Owner na sua visão?

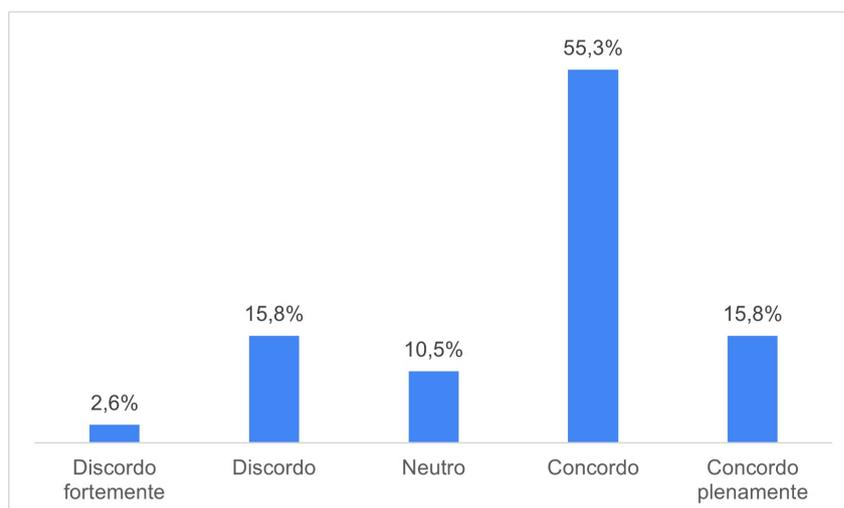


Figura 28 – Resposta 17 - A capacidade da equipe de receber orientações claras sobre as tarefas a serem desenvolvidas é afetada negativamente pela junção de papéis do Scrum Master e do Product Owner na sua visão?

análise feita pela autora através das respostas da pesquisa

18. Você acredita que a junção dos papéis do Scrum Master e do Product Owner afeta

negativamente a qualidade das entregas da equipe?

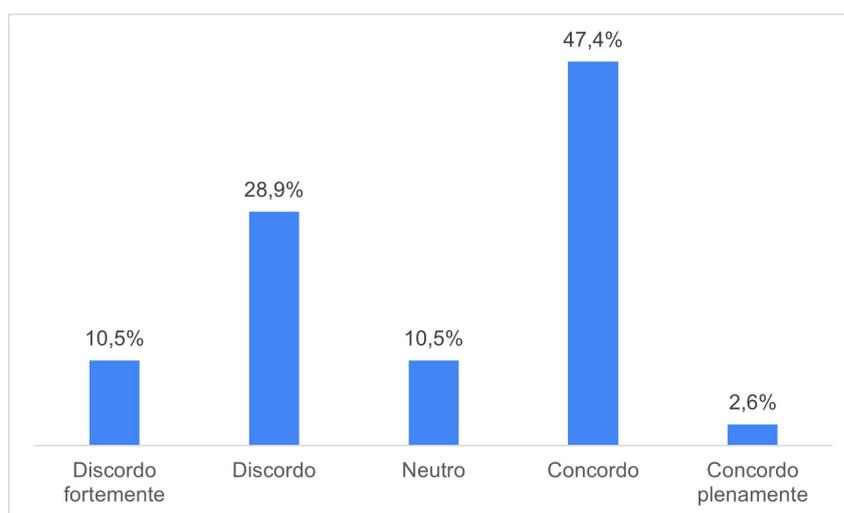


Figura 29 – Resposta 18 - Você acredita que a junção dos papéis do Scrum Master e do Product Owner afeta negativamente a qualidade das entregas da equipe?

análise feita pela autora através das respostas da pesquisa

19. Para você, essa junção de papéis do Scrum Master e do Product Owner influencia negativamente a capacidade da equipe de lidar com mudanças nos requisitos do projeto?

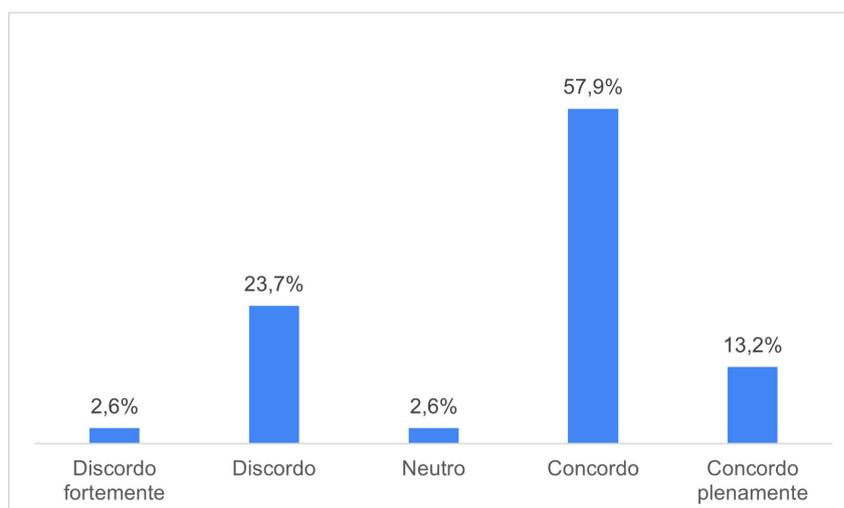


Figura 30 – Resposta 19 - Para você, essa junção de papéis do Scrum Master e do Product Owner influencia negativamente a capacidade da equipe de lidar com mudanças nos requisitos do projeto?

análise feita pela autora através das respostas da pesquisa

20. Na sua percepção, a junção dos papéis de Scrum Master e Product Owner afeta negativamente a comunicação dentro da equipe?

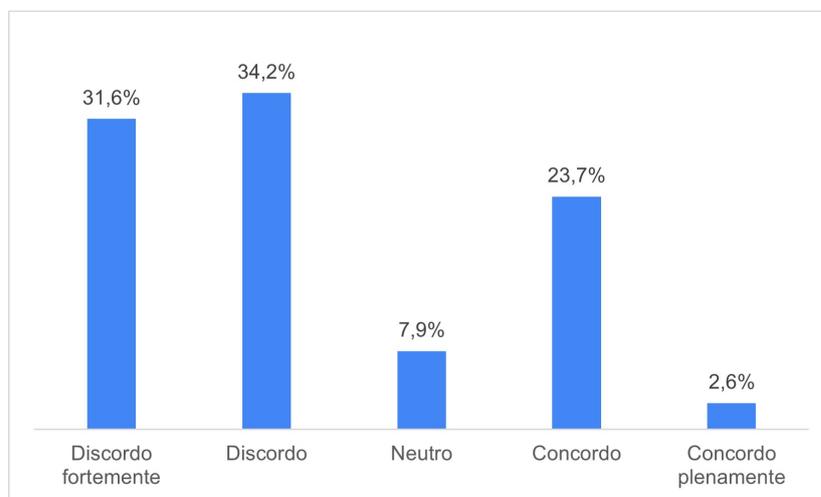


Figura 31 – Resposta 20 - Na sua percepção, a junção dos papéis de Scrum Master e Product Owner afeta negativamente a comunicação dentro da equipe?

análise feita pela autora através das respostas da pesquisa

21. Na sua opinião, a junção desses papéis do Scrum Master e do Product Owner influencia negativamente na disponibilidade do Scrum Master para auxiliar a equipe no dia a dia do projeto?

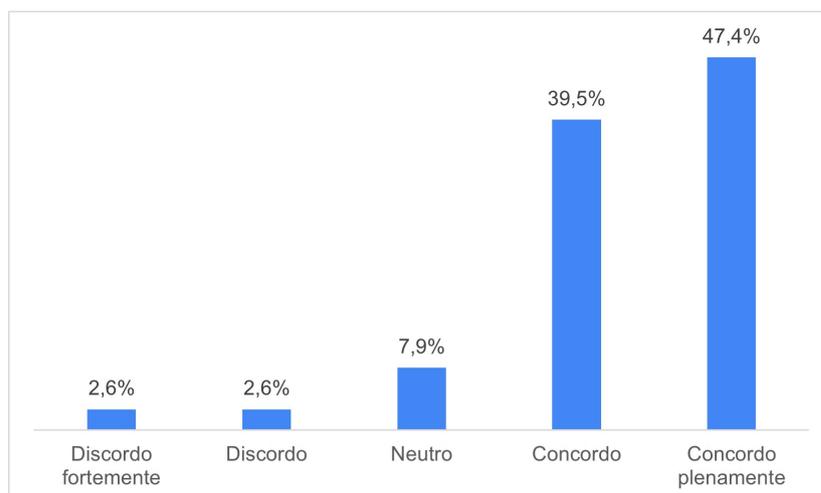


Figura 32 – Resposta 21 - Na sua opinião, a junção desses papéis do Scrum Master e do Product Owner influencia negativamente na disponibilidade do Scrum Master para auxiliar a equipe no dia a dia do projeto?

análise feita pela autora através das respostas da pesquisa

22. Na sua visão, com a junção desses papéis de Scrum Master e Product Owner o time pode fazer entregas mais alinhadas com a expectativa do cliente?

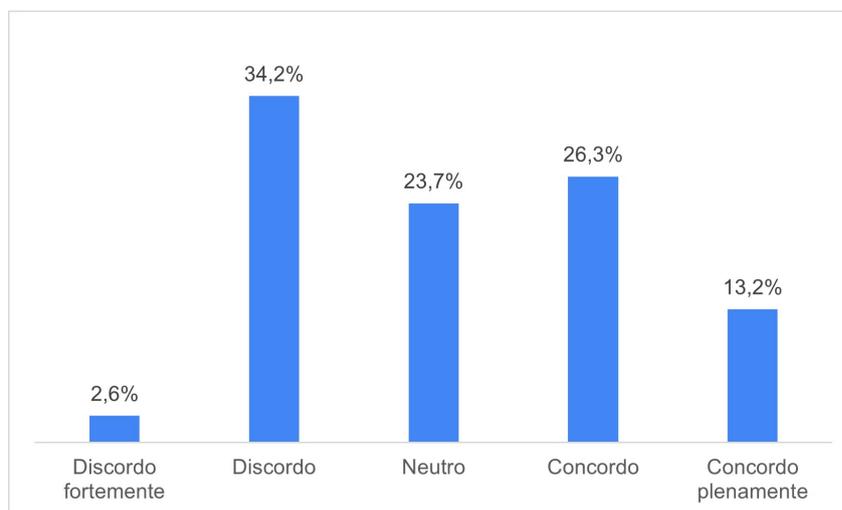


Figura 33 – Resposta 22 - Na sua visão, com a junção desses papéis de Scrum Master e Product Owner o time pode fazer entregas mais alinhadas com a expectativa do cliente?

análise feita pela autora através das respostas da pesquisa

23. Você tem alguma sugestão ou comentário adicional sobre a junção de papéis do Scrum Master e Product Owner que gostaria de compartilhar?

- Não.
- Não é aconselhável unir esses dois papéis no Scrum, pois eles têm focos diferentes. O Scrum Master concentra-se em melhorar o processo e remover obstáculos, enquanto o Product Owner concentra-se na visão do produto e nas necessidades dos clientes. Separar essas funções ajuda a equilibrar os interesses e a manter a transparência e responsabilidade no desenvolvimento do produto. O SM tem o papel fundamental de garantir que a equipe compreenda e siga os princípios e práticas do Scrum, além de remover impedimentos que possam prejudicar o progresso da equipe. O PO é responsável por definir e priorizar o backlog do produto, garantindo que a equipe esteja trabalhando nas tarefas mais valiosas para o cliente. Todo equilíbrio é importante.
- O papel do Scrum Master ficou dividido não só com o Product Owner mas tbm com o Squad Leader. O Squad Leader tbm faz atividades que eram do Scrum Master.

- A junção dos papéis de Scrum Master e Product Owner pode sobrecarregar o indivíduo que assume ambos os papéis, tornando difícil manter uma clareza de responsabilidades e prejudicando a eficácia de ambos os papéis. Embora possa ser tentador combinar os papéis em equipes menores ou em situações específicas, é crucial considerar o impacto potencial na eficiência da equipe e na qualidade do produto final.
- Misturar os papéis de Scrum Master e Product Owner pode causar conflitos de interesses. Enquanto o Scrum Master se concentra em manter o processo e ajudar a equipe, o Product Owner está mais preocupado com o valor do produto entregue ao cliente. Ter uma pessoa para ambos os papéis pode diluir a eficácia de cada papel individualmente.
- Para mim devemos ter os dois papéis em profissionais diferentes para que o scrum fique focado somente em ajudar o time e não fique refém dos pedidos do cliente.
- Como Diretor de Tecnologia, tenho visto e liderado diversas implementações ágeis e considero essencial manter a clareza e a distinção entre os papéis de Scrum Master e Product Owner. A junção desses papéis pode causar confusão e sobreposição de responsabilidades, potencialmente comprometendo a eficácia de nossa abordagem ágil. Em nossa busca contínua por eficiência e entrega de valor, é crucial que cada papel tenha um foco claro e possa se dedicar completamente às suas responsabilidades distintas. Misturar essas funções pode diluir essa clareza e comprometer nossa capacidade de entregar consistentemente produtos de alta qualidade que atendam às necessidades de nossos clientes.
- Não é o ideal, pois o scrum master fica sempre ocupado e as vezes não tem tempo para problemas do dia a dia da equipe. Mas hoje o projeto já anda melhor e conseguimos seguir o ritmo.
- Pode funcionar em projetos pequenos como o nosso, mas não é o ideal.