





# Um survey sobre a utilização das Metodologias Ágeis no Desenvolvimento de Software Brasileiro

Frederico A.S.Gentil , *Student Member, IEEE*, Gustavo S.Melo , *Student Member, IEEE*, *Student Member, IEEE*, Yago S.Oliveira , *Student Member, IEEE* and Michel S.Soares , *Senior Member, IEEE*

**Abstract**—Agile methodologies were applied in software development to provide flexibility and adaptability to frequent requests for scope changes and the need to respond to customer requests quickly. Understanding how agile methodologies were applied and their impacts on software development in Brazil is important so that planning is more efficient and methodological definitions are adapted to the reality of the national development environment. This work was developed under the aegis of General Systems Theory, which was applied using intentional non-probabilistic sampling to collect and analyze information about agile methodologies in the software development process. The goal was to understand the current reality and provide support for the creation of procedures for the use and organization of agile methodologies in software development. A descriptive analysis and a quantitative/qualitative research method were used to collect professionals' experiences through the application of an online questionnaire. The results presented a variety of agile methodologies, with an emphasis on Scrum and Kanban, and tools being applied at different stages of the software development cycle. The conclusions showed the reality of companies operating in Brazil and how they are relating software development with agile methodologies, understanding the tools, results, and difficulties encountered. This information will be used to define future ways of using agile methodologies in software development

**Index Terms**—Agile methodologies, software development, software architecture.

## I. INTRODUÇÃO

A necessidade por entregas rápidas de produtos de software, sem que a qualidade fosse afetada, impulsionou uma crescente demanda por eficiência e agilidade, abrindo grande espaço para a popularização das metodologias ágeis [8]. Essas metodologias trouxeram uma nova visão focada na flexibilidade e na adaptação a alterações frequentes de demandas e necessidades. O presente trabalho tem como objetivo entender, através da aplicação de um *survey*, a situação atual da utilização das metodologias ágeis no desenvolvimento de software brasileiro e a sua relação com as arquiteturas de software. Entender essa realidade, seus impactos e dificuldades cria uma base consistente de entendimento do cenário nacional que pode ser utilizada como alicerce para a melhoria do desenvolvimento de software brasileiro. O *survey* coletou informações com o objetivo de analisar a utilização dessas metodologias nas etapas que envolvem o processo de desenvolvimento de software no Brasil. Compreender o desempenho dessas abordagens dentro das organizações, os resultados obtidos e as possíveis vulnerabilidades pode ajudar a obter aumento nos índices de êxito no desenvolvimento de sistemas.

Esse trabalho está estruturado em seções. A primeira seção traz breve apresentação do estudo realizado, enquanto a se-

gunda apresenta fundamentação teórica e artigos relacionados sobre os temas tratados durante a elaboração do estudo. A terceira seção apresenta a metodologia de pesquisa aplicada e, na seção quatro, é feita a análise dos dados coletados e são apresentados os resultados. O estudo termina com a seção cinco apresentando as conclusões do trabalho.

## II. ANTECEDENTES E TRABALHOS RELACIONADOS

Apresentar as fundamentações teóricas das metodologias ágeis, da arquitetura e do desenvolvimento de software vai servir como alicerce conceitual para o desenvolvimento do presente trabalho e são corroborados pelos trabalhos relacionados à utilização das metodologias ágeis no desenvolvimento de software.

### A. Arquitetura de Software

Desde a concepção até sua construção, a Arquitetura de Software detém um grau de importância inestimável. Diante da evolução e do aprendizado sobre Arquitetura de Software, os profissionais puderam vislumbrar que acertar, na Arquitetura de Software, é um fator crítico, podendo acarretar o sucesso ou o fracasso de um projeto [7] [27]. A Arquitetura de Software representa a estrutura fundamental, a organização e as dependências dos componentes de software, bem como os relacionamentos entre esses componentes e o ambiente externo. A Arquitetura de Software é um elemento importante do processo de desenvolvimento de software, fornecendo uma visão global do sistema e auxiliando na tomada de decisões [7] [11] [10] [9].

Tomando como referência o início do termo Arquitetura de Software a partir da década de 1960 [17] como um subcampo da Engenharia de Software, é perceptível a evolução desde a fase em que o investimento em hardware era maior que em software, chegando aos dias atuais com arquiteturas complexas baseadas em microsserviços [4]. Devido ao acelerado desenvolvimento das tecnologias, há grande investimento em ferramentas, metodologias e linguagens de programação que visam entrar no mercado de software rapidamente e aperfeiçoar o desenvolvimento de software, gerando competições entre as empresas [4] [5].

A construção de uma Arquitetura de Software não é um serviço simples. Projetos arquiteturais são, em sua grande maioria, difíceis de serem desenvolvidos e, quando relacionados com práticas ágeis, torna-os ainda mais complexos. Uma construção arquitetural não pode ser facilmente decomposta

em pequenas funcionalidades durante o desenvolvimento, tornando primordial a definição do design inicial de como deve funcionar a arquitetura em todo o seu ciclo de vida [13]. A utilização de *sprints* iterativas entre os responsáveis por planejar a arquitetura, devido à coleta de *feedbacks* constantes durante o desenvolvimento, é uma prática que agrega valor e ajuda a tomada de decisões. Durante o processo, os desenvolvedores e arquitetos de software planejam gradativamente as mudanças e as melhorias necessárias em cada arquitetura para obter os resultados esperados [13] [1] [8].

### B. Metodologias Ágeis

As metodologias ágeis dão ênfase à flexibilidade, à colaboração e à entrega contínua de software [25]. Elas contrastam com as metodologias tradicionais de desenvolvimento de software, como o Modelo em Cascata [2], que seguem um processo linear e rígido. As Metodologias Ágeis, como XP e Scrum, são baseadas em um conjunto de princípios que promovem a agilidade e a adaptação [6] e, dentre os seus benefícios, estão as entregas mais rápidas, maior satisfação dos clientes e maior qualidade. Quanto aos desafios enfrentados, são identificadas a adoção cultural, a comunicação eficiente e o gerenciamento de mudanças [2].

Existem várias metodologias ágeis presentes na literatura, dentre elas:

- Scrum - é uma das metodologias ágeis mais populares. Ele divide o trabalho em iterações chamadas *sprints* e utiliza papéis bem definidos, como *Scrum Master* e *Product Owner* [21]. O Scrum enfatiza a colaboração da equipe, a entrega incremental e a adaptação contínua.
- Extreme Programming (XP) - é conhecida por suas práticas de desenvolvimento, como programação em pares, testes automatizados e integração contínua. Tal metodologia valoriza a comunicação constante com o cliente, mudanças frequentes nos requisitos e entrega de software de alta qualidade.
- Kanban - visualiza o fluxo de trabalho em um quadro, usando cartões para representar tarefas. O objetivo é otimizar o fluxo, limitando o trabalho em andamento e tornando as restrições visíveis. Kanban é altamente flexível e se concentra em melhorar continuamente o processo.
- Lean Software Development - inspirado no Lean Manufacturing, o Lean Software Development busca eliminar o desperdício, aumentar a eficiência e fornecer valor ao cliente. Ele enfatiza a entrega rápida, a simplificação e a melhoria contínua.
- Dynamic Systems Development Method (DSDM) - é uma abordagem ágil com forte foco na colaboração e na entrega de produtos de alta qualidade. Ele fornece um framework flexível para o desenvolvimento de sistemas e é amplamente utilizado em projetos empresariais.
- Feature Driven Development (FDD) - concentra-se na identificação e na implementação de funcionalidades específicas do software. Ele divide o projeto em etapas de design e construção de funcionalidades. O FDD é especialmente adequado para projetos complexos.

- Adaptive Software Development (ASD) - é uma metodologia com grande foco na adaptação às mudanças à medida que ocorrem. Ele envolve ciclos curtos de planejamento, desenvolvimento e revisão, permitindo uma rápida resposta às necessidades do cliente.

Existem outras metodologias ágeis disponíveis e muitas equipes adaptam e combinam elementos delas para atender às necessidades específicas de seus projetos. Cada uma dessas abordagens valoriza princípios semelhantes, como colaboração, entrega contínua e adaptação a mudanças. Tais metodologias trouxeram uma revolução ao desenvolvimento de software, promovendo uma abordagem centrada no cliente, flexível e eficiente [3]. Elas propiciam às equipes entregar software de alta qualidade de maneira rápida e adaptável, alinhando-se às necessidades em constante evolução dos clientes [2].

### C. Trabalhos relacionados

Um dos desafios na atualidade, tanto para o desenvolvimento de software quanto para as metodologias ágeis, é buscar maior integração entre essas duas áreas. Muito se tem estudado e pesquisado sobre o tema de arquitetura e desenvolvimento de software, estudando o desenvolvimento de projetos globais [20], aplicações gerais em arquitetura [22], aplicações em fases específicas da arquitetura como Engenharia de Requisitos [16], identificação de tendências [26] e degradação da arquitetura com o passar do tempo [14].

No âmbito das metodologias ágeis, estudos exploram os efeitos das melhorias nos procedimentos para otimizar a resolução de processos, visando alcançar a qualidade desejada [23] por meio da aplicação, no desenvolvimento de software, de metodologias ágeis específicas como Scrum [20] [12] [18] [19] ou XP (Extreme Programming) [22].

Um importante estudo foi feito buscando identificar os desafios e práticas de design arquitetural em projetos de desenvolvimento global [20]. Este modelo, caracterizado pela dispersão geográfica das equipes, tem sido uma realidade na indústria nos últimos 10 anos, enfrentando desafios como diferenças culturais, fusos horários e obstáculos na comunicação. O artigo destaca questões como comunicação, colaboração, controle e compreensão compartilhada, propondo a aplicação de práticas ágeis, tais como ferramentas colaborativas, documentação clara e concisa, processos e padrões bem definidos, visando mitigar os impactos da complexidade no desenvolvimento global de software.

Outro artigo relevante aborda a interação entre a Arquitetura de Software e as metodologias ágeis com o uso da metodologia XP em *startups* [22]. Destaca-se a análise dos potenciais conflitos entre abordagens ágeis e arquitetura, observando uma diminuição do interesse nas práticas de Engenharia e Arquitetura de Software com o crescimento de *startups*. O desafio central é encontrar um equilíbrio entre a flexibilidade ágil e a estabilidade arquitetural, sendo esse equilíbrio influenciado pelo tamanho e tipo do projeto, qualificação da equipe e envolvimento do cliente.

O estudo da relação mais específica entre a Engenharia de Requisitos e o desenvolvimento de software [16] explora a

aplicação de técnicas e práticas ágeis nessa engenharia. Neste contexto estão incluídas técnicas de elicitação e documentação de requisitos, gerenciamento de mudanças, colaboração entre equipes e integração de requisitos com práticas ágeis. Um dos grandes desafios é buscar o alinhamento entre a gestão de documentos, as mudanças frequentes de requisitos e a manutenção do foco na entrega de valor ao cliente por meio de uma abordagem adaptativa e flexível na Engenharia de Requisitos em projetos ágeis.

Identificar tendências, desafios e práticas da aplicação das metodologias ágeis no desenvolvimento de sistemas [26] é considerado num estudo sistemático que ratifica crescente interesse na indústria de desenvolvimento em integrar eficazmente esses dois enfoques. A busca pelo equilíbrio entre flexibilidade ágil e estabilidade arquitetural, bem como a comunicação e a colaboração entre equipes técnicas e não técnicas, é um dos desafios apresentados pelo estudo. Eles apresentam várias práticas e abordagens utilizadas em projetos que combinam Arquitetura de Software e desenvolvimento ágil, como revisões arquiteturais regulares, histórias de usuário arquiteturalmente significativas e prototipagem, com o intuito de abordar essa relação de maneira eficaz para melhorar a qualidade e a eficácia do desenvolvimento de software em contextos ágeis.

*Smell architecture* representa estruturas e arquiteturas abaixo do padrão em sistemas, como, por exemplo, troca impossível de componentes, modularização descuidada, arquitetura não escalável, dentre outros. O artigo [23] busca identificar os impactos da utilização da metodologia ágil na redução desses problemas, principalmente por meio de fatores como transparência e responsabilidade. O estudo evidencia que equipes mais integradas experimentam uma diminuição de problemas ao empregar processos e ferramentas que facilitam a comunicação e colaboração entre elas.

A utilização de *frameworks* e bibliotecas, a evolução contínua, os testes automatizados, os padrões bem definidos, a comunicação e a colaboração [20] [23] [24] são características comuns quando se pensa na aplicação e nos benefícios das metodologias ágeis no desenvolvimento de sistemas. Apesar de persistirem significativos desafios para a integração ideal entre essas duas áreas [23] [1], principalmente na integração dos conceitos aparentemente conflitantes [1] [22], resultados positivos estão sendo relatados provenientes da implementação de metodologias ágeis no âmbito da arquitetura e do desenvolvimento de software. [18] [22] [12] [24].

### III. METODOLOGIA DE PESQUISA

Esse trabalho tem como objetivo buscar informações acerca da utilização das metodologias ágeis no processo de desenvolvimento de sistemas, fazendo uma relação dessa utilização com as etapas da Arquitetura de Software. Foi aplicado um *survey*, com a utilização do *google forms*, que ficou disponível para acesso durante 20 dias entre os meses de outubro e novembro de 2023.

O questionário possui 19 questões, divididas em 4 subgrupos, que têm o objetivo de coletar informações sobre a relação entre as metodologias ágeis e o desenvolvimento de software, por meio de resposta às seguintes questões de pesquisa (QPs):

- QP1: Qual o perfil dos participantes? Busca identificar os participantes da pesquisa quanto a sua formação, cargo que exerce, experiência e área de atuação.
- QP2: Qual o nível de conhecimento dos profissionais sobre Metodologias Ágeis? Nesta questão de pesquisa, o objetivo é identificar o nível de conhecimento que os profissionais entrevistados possuem com relação às metodologias ágeis que estão usando, identificando-os como iniciantes, intermediários ou avançados.
- QP3: Quais as metodologias e ferramentas de gestão de projetos ágeis mais utilizadas no desenvolvimento de software? Essa questão tem como objetivo identificar quais as metodologias ágeis ou combinações delas estão em uso pelos profissionais da área de desenvolvimento, bem como as ferramentas utilizadas para a gestão dos projetos.
- QP4: Quais os estilos arquiteturais, as fases do desenvolvimento e os tipos de produtos que são desenvolvidos onde os participantes trabalham? Esta questão visa buscar relacionar os tipos de produtos de software com os estilos arquiteturais e as metodologias ágeis que estão sendo usadas.
- QP5: Quais os benefícios e as dificuldades da utilização das metodologias ágeis no desenvolvimento de software? Esta questão visa identificar quais os resultados alcançados, na prática, pela utilização das metodologias ágeis em projetos reais, bem como elencar as principais dificuldades encontradas na aplicação dessas metodologias.

A Tabela II apresenta a relação entre as QPs e as perguntas presentes no questionário aplicado.

#### A. Público-Alvo

Foram definidos como público-alvo da pesquisa os profissionais que trabalham com desenvolvimento de software no Brasil e que utilizam metodologias ágeis durante qualquer etapa do ciclo de vida do desenvolvimento, independentemente da área ou do nível de formação. Foram buscados profissionais de empresas de diversos portes que possuíssem equipes de desenvolvimento.

#### B. Amostragem

Para realização desse estudo e levando em consideração a dificuldade de alcançar um público-alvo distribuído por todo o país, optou-se pela adoção de uma amostragem intencional não probabilística [15], fazendo a busca dos participantes por meio de contatos pessoais dos pesquisadores, em grupos de interesse do tema foco da pesquisa, em listas de discussão de ambientes acadêmicos e através da rede social profissional *LinkedIn*. Considerando que a maioria dos contatos foi realizada de forma pessoal pelos autores do estudo, também foi solicitado aos participantes que, além do preenchimento do formulário, fizessem o compartilhamento da pesquisa com seus contatos que tivessem relação com o tema do presente estudo. Para preservar a privacidade e facilitar o acesso a informações que pudessem ser mais sensíveis, o preenchimento do formulário ocorreu de forma anônima, sem que houvesse necessidade de qualquer registro ou login dos participantes.

TABLE I: Itens do Questionário x Tipo de Pergunta

Questão de Pesquisa	Perguntas	Tipo de Resposta			
		Escolha Única	Múltipla Escolha	Texto Livre “outros”	Grade de Seleção Múltipla
QP1	1.1. Qual é a sua formação acadêmica?	X		X	
	1.2. Qual o seu maior grau de Pós-graduação?	X		X	
	1.3. Qual o seu cargo/função atual?		X	X	
	1.4. Qual a sua forma de trabalho?	X			
	1.5. Qual o setor de atuação da empresa que você trabalha?		X	X	
	1.6. Qual a sua faixa etária?	X			
	1.7. Qual o seu tempo de experiência com Desenvolvimento de Sistemas?	X			
	1.8. Qual o seu tempo de experiência com Metodologias Ágeis?	X			
QP2	2.1. Como você classificaria seu conhecimento em Metodologias Ágeis?	X			
QP3	2.2. Quais das seguintes metodologias ágeis você utiliza em seus projetos?		X	X	
	2.4. Quais ferramentas de gestão de projetos ágeis você utiliza com mais frequência?		X	X	
QP4	2.3. Em que fase do desenvolvimento de software as metodologias ágeis estão sendo aplicadas no seu dia a dia?		X	X	
	3.1. Quais os tipos de produtos ou projetos de software são desenvolvidos em seu ambiente de trabalho?		X	X	
	3.2. Quais estilos/padrões arquiteturais você utiliza no desenvolvimento de software?		X	X	
	3.3. Faça uma relação entre as metodologias ágeis e os estilos arquiteturais que você utiliza no desenvolvimento de software?				X
QP5	4.1. Quais resultados positivos você observou na parte GERENCIAL com a utilização de metodologias ágeis em projetos reais?		X	X	
	4.2. Quais resultados positivos você observou na parte TÉCNICA com a utilização de metodologias ágeis em projetos reais?		X	X	
	4.3. Quais são as principais dificuldades e desafios que você enfrentou na parte GERENCIAL ao aplicar metodologias ágeis em projetos de software?		X	X	
	4.4. Quais são as principais dificuldades e desafios que você enfrentou na parte TÉCNICA ao aplicar metodologias ágeis em projetos de software?		X	X	

TABLE II: QPs x Itens do Questionário

Questões de Pesquisa	Itens do Questionário
QP1	[1.1] [1.2] [1.3] [1.4]
	[1.5] [1.6] [1.7] [1.8]
QP2	[2.1]
QP3	[2.2] [2.4]
QP4	[2.3] [3.1] [3.2] [3.3]
QP5	[4.1] [4.2] [4.3] [4.4]

### C. Pesquisa Piloto

Para validar o questionário aplicado, foi conduzida uma pesquisa piloto com sete pessoas pertencentes ao público-alvo da pesquisa, com a aplicação do questionário completo e o mesmo procedimento a ser utilizado na pesquisa final. Esse teste tem por objetivo verificar os resultados e o *feedback* com a aplicação em um universo menor de participantes [15]. Esse primeiro questionário possui 20 questões e um tempo médio de preenchimento de 7 minutos. Os participantes foram escolhidos dentro de empresas da área de desenvolvimento, que ocupavam funções diferentes, bem como doutores da área em estudo.

Dentre os participantes da pesquisa piloto, foi identificado que 7 são da área de tecnologia, sendo 4 com graduação plena, 2 com curso tecnólogo e 1 ainda cursando o nível superior. Ao considerar a pós-graduação, 1 possui MBA e 1 tem doutorado. No item relativo à experiência, os participantes apresentaram um equilíbrio no nível de conhecimento das metodologias

ágeis e do desenvolvimento de software, com 2 participantes tendo mais de 10 anos de experiência nas respectivas áreas, mas identifica-se, entre os que possuem menos de 10 anos de experiência, um grupo bem maior com pouca vivência nas metodologias ágeis, com 5 participantes com menos de 5 anos, contra 2 relacionados ao desenvolvimento de software. Todos os respondentes trabalham de forma remota.

Após a aplicação, foi solicitado *feedback* dos participantes acerca do questionário aplicado. Com base nesses *feedbacks* recebidos, foram realizados ajustes no questionário. Uma questão que versava sobre o local onde o participante morava foi retirada por dar margem a dúvidas posteriores, considerando que uma das características da área de desenvolvimento é o trabalho remoto e essa questão poderia gerar um desvio de interpretação sobre a distribuição geográfica dos participantes e sua real alocação nos centros regionais de desenvolvimento.

As questões relativas a estilos arquiteturais e ferramentas de gestão de projetos ágeis tiveram itens adicionados à lista de opções, e a questão sobre os tipos de metodologias ágeis teve um item retirado por não se tratar, essencialmente, de uma metodologia ágil. Duas observações foram analisadas, mas não acatadas, pois as sugestões poderiam resultar na alteração do foco buscado nas QPs.

As questões sobre benefícios e dificuldades tiveram a opção “nenhum” acrescentada após os *feedbacks* da pesquisa piloto.

Ao final, o questionário ficou com um total de 19 perguntas,

sendo 7 perguntas de escolha única, 11 de escolhas múltiplas e 1 questão de relacionamento através de grade de seleção múltipla, conforme Tabela I. 13 questões possuem espaço para texto livre por meio do item “outros”.

#### D. Exclusões e Validação dos Dados

Considerando que o objetivo do presente trabalho é entender a relação entre o desenvolvimento de software, os modelos arquiteturais e as metodologias ágeis, após a aplicação do formulário, foi realizada uma pré-análise para buscar respostas que não estivessem inseridas nesse contexto e que representassem uma possibilidade de desvio do foco da pesquisa. Respostas de participantes que não utilizam metodologias ágeis ou não estivessem diretamente relacionadas à função dentro do ciclo de vida do desenvolvimento de software foram consideradas inválidas e excluídas dos dados que foram submetidos à análise final. Ao final da pré-análise, foram localizadas 6 respostas que não se enquadram no escopo desejado da pesquisa e que foram excluídas da fase de análise final dos dados.

### IV. RESULTADOS DA PESQUISA

Com base na análise dos dados das 164 respostas validadas na etapa de pré-análise, serão apresentados os resultados obtidos com este estudo.

#### A. Perfil dos Participantes (QP1)

O primeiro grupo de 8 perguntas visa traçar um perfil dos participantes da pesquisa, identificando formação e características relativas a sua atuação profissional. A primeira pergunta visava identificar a formação acadêmica dos participantes.

Das 164 respostas, a grande maioria é formada por profissionais com curso superior completo (86,50%), sendo que desse total 57,06% são graduados na área de tecnologia, 24,54% possuem curso tecnólogo e 4,91% são graduados em outras áreas. 11,04% dos participantes possuem curso superior incompleto e 2,45% possuem formação técnica de nível médio. Entre os graduados na área de tecnologia, 27,44% são graduados em Sistema de Informação, 23,17% em Ciência da Computação, 2,44% em Engenharia da Computação e 1,83% em Engenharia de Software, conforme Figura 1.

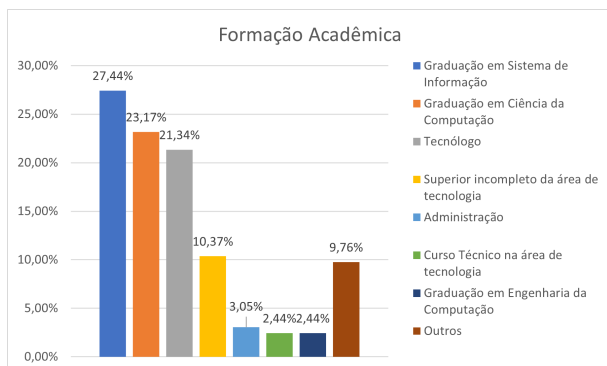


Fig. 1: Formação Acadêmica

A segunda pergunta busca identificar o nível de pós-graduação. 56,71% dos respondentes não possuem nenhum tipo de pós-graduação, 26,83% possuem especialização/MBA, 10,98% têm mestrado na área de tecnologia, 1,83% com mestrado em outras áreas de formação, 1,22% possuem doutorado em outras áreas de formação, 0,61% possuem doutorado na área de tecnologia. 1,83% das respostas ficaram em branco.

Com relação ao cargo/função exercidos pelos participantes, a Figura 2 apresenta que 36,80% são desenvolvedores de software, 18,61% analistas de sistemas, 10,82% gerentes de projetos, 7,36% se definiram como gerentes de sistemas, 6,49% são arquitetos de software, 5,19% são analistas de negócios e 3,46% responderam como sendo DBA. Como não existe uma padronização nacional relacionada aos cargos da área de desenvolvimento, foi percebido um percentual de 11,26% de participantes que se enquadram na opção “outros” e descreveram uma quantidade de 19 cargos/funções diversas às opções apresentadas na questão 3 como: instrutor, Scrum Master, head de QA, diretor, UX/UI Designer e outras.

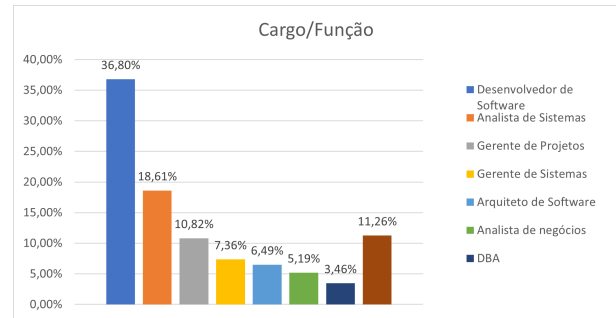


Fig. 2: Cargo/Função Ocupada

Quanto à informação do tipo de trabalho apresentada na Figura 3, fica evidente, dentre a amostra da pesquisa, uma maioria de respondentes trabalhando de forma remota (54,88%) contra 24,39% do trabalho presencial e 20,12% de pessoas que trabalham em ambiente híbrido. 1 pessoa deixou a sua resposta em branco.

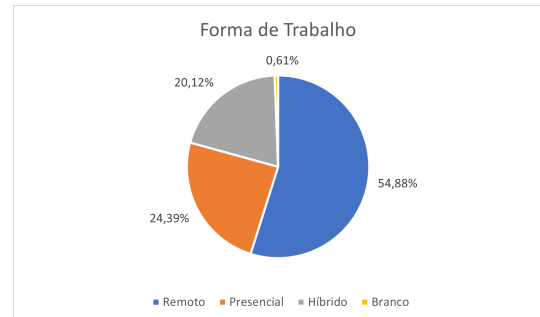


Fig. 3: Tipo de Trabalho

A pergunta 1.5 busca identificar a área de atuação da empresa em que o pesquisado trabalha. Das respostas coletadas, 47,96% trabalham em empresas da área de Tecnologia da Informação, 14,03% na área financeira, 10,41% comercial, 7,24% na área de educação. A área de saúde representa 6,33%

e a industrial, 5,43% das respostas, enquanto 2,71% foram da área de telecomunicações. Outras respostas representam 5,88% dos respondentes.

Referente à faixa etária dos participantes (Figura 4), a maioria está na faixa de 26 a 35 anos, representando 43,90%. A faixa etária de 18 a 25 representa 25,61% e a de 36 a 45 anos, 18,29% das respostas. 10,98% estão na faixa de 46 a 55 anos e apenas 1,22% tem mais de 55 anos de idade.

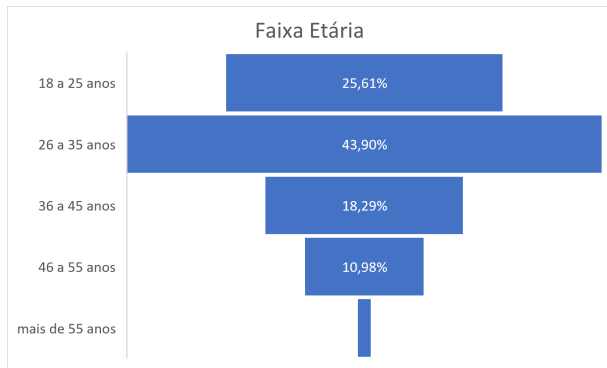


Fig. 4: Faixa Etária

As questões 1.7 e 1.8 têm por objetivo identificar a experiência dos participantes com desenvolvimento de software e com metodologias ágeis, respectivamente (Figuras 5 e 6). Quanto ao desenvolvimento de software, 43,29% responderam que têm menos de 5 anos de experiência e 21,34%, entre 5 e 10 anos. As faixas etárias com experiência de mais de 20 anos e experiência entre 11 e 15 anos representam 13,41%, e as que possuem experiência entre 16 e 20 anos e as que não tem experiência com desenvolvimento de software atingiram um percentual de 4,27% cada.

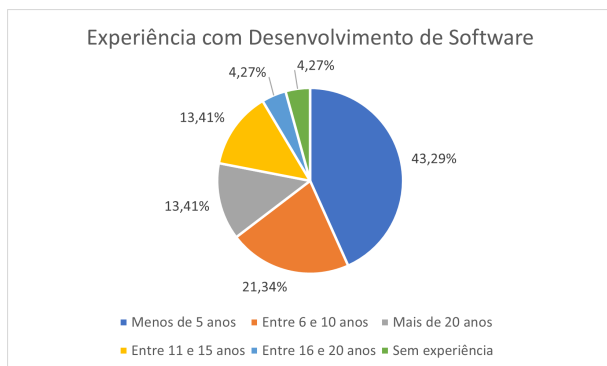


Fig. 5: Experiência com Desenvolvimento de Software

Com relação às metodologias ágeis, percebe-se que a maioria dos respondentes possui menos de 5 anos de experiência, perfazendo um total de 61,18%. Os que responderam ter entre 6 e 10 anos somam 25,29%, 5,29% entre 11 a 15 anos, 2,94% com mais de 20 anos de experiência e 1,18% possui experiência entre 16 e 20 anos. 4,12% responderam não possuir experiência com metodologias ágeis.

#### B. Nível de Conhecimento sobre Metodologias Ágeis (QP2)

A pergunta 2.1. do questionário buscava identificar qual nível de conhecimento os participantes possuíam com relação

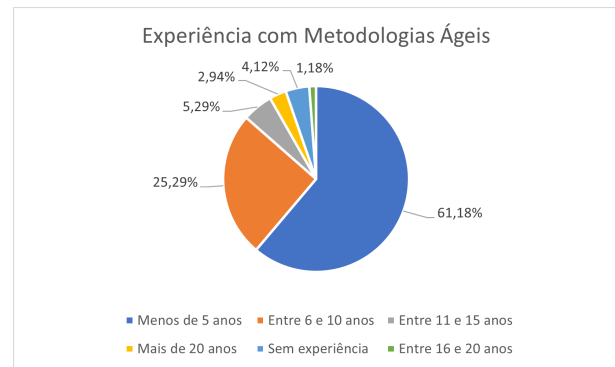


Fig. 6: Experiência com Metodologias Ágeis

às metodologias ágeis (Figura 7). Segundo os dados coletados, mais da metade dos participantes (53,66%) declararam possuir conhecimento intermediário sobre as metodologias ágeis, 25,00% possuem conhecimento avançado e 21,34% têm conhecimento iniciante.

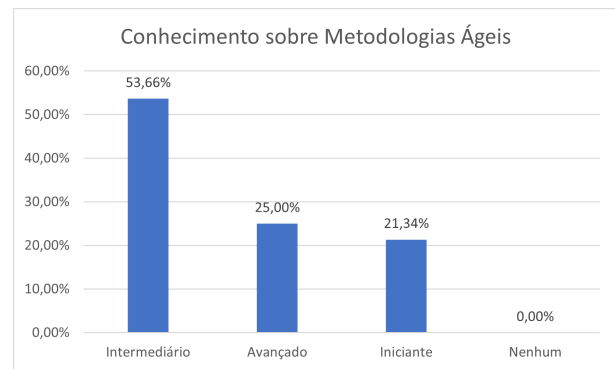


Fig. 7: Conhecimento sobre Metodologias Ágeis

#### C. Metodologias e Ferramentas da Gestão de Projetos Ágeis Utilizadas no Desenvolvimento de Software (QP3)

Quando a pergunta se refere às metodologias ágeis mais usadas no cotidiano dos participantes, fica clara uma predominância do Scrum e Kanban, que, juntas, representam 72,21% do total de respostas, 41,42% e 30,79% de cada uma, respectivamente. A metodologia XP representa 7,36% das respostas, Lean, 4,63%, SAFe, 4,09%, Scrum of Scrums tem 3,54%, e outras 11 metodologias tiveram um percentual menor que 2% cada.

A pergunta 2.4 busca identificar quais ferramentas de gestão de projeto ágil são mais utilizadas nas empresas. Como apresentado na Figura 9, as duas ferramentas mais citadas foram o Jira e o Trello com 20,95% e 16,46% respectivamente. Após essas duas aparece o Figma com 13,47%, o GitLab com 11,72%, o Microsoft Azure DevOps, 10,97% e o Miro com 10,47%. O Clickup recebeu 2,74% das respostas e o Redmine, 2,49%. Após esse primeiro grupo, as respostas ficam bastante pulverizadas, apresentando 21 ferramentas diferentes que perfazem um total de 10,72% das respostas.



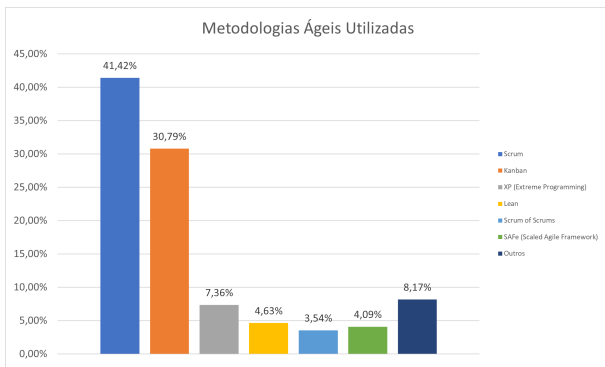


Fig. 8: Metodologias Ágeis Utilizadas

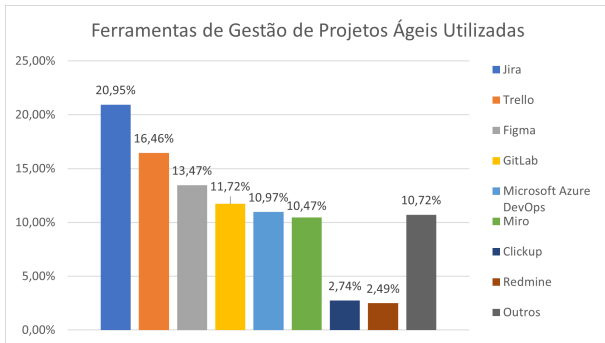


Fig. 9: Ferramentas de Gestão de Projetos Ágeis Utilizadas

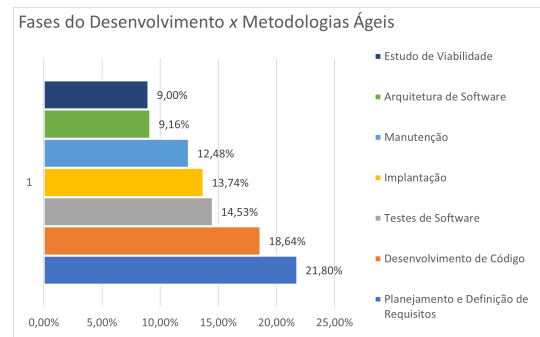


Fig. 10: Fases do Desenvolvimento x Metodologias Ágeis

intensiva de dados, extração de dados, sistema de sistemas, e sistemas Daemon e microsserviços.

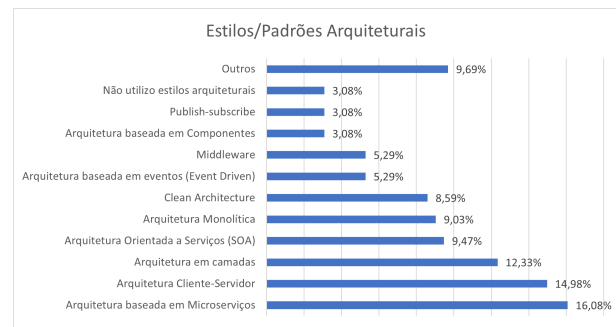


Fig. 11: Padrões/Estilos Arquiteturais Utilizados

#### D. Tipos de Produtos e Estilos Arquiteturais (QP4)

Nesse grupo de 4 perguntas, foi buscado identificar quais os estilos arquiteturais utilizados, os tipos de produtos desenvolvidos, em que fases do desenvolvimento as metodologias ágeis estão sendo utilizadas e a relação entre os estilos arquiteturais e quais as metodologias que estão sendo utilizadas para cada um deles.

A pergunta 2.3 busca identificar em quais fases do desenvolvimento de software as metodologias ágeis estão sendo usadas (Figura 10). 21,80% das respostas dizem utilizar na fase de planejamento e definição de requisitos, enquanto 18,64% informaram usar durante o desenvolvimento de código. A fase de teste de software foi citada por 14,53% dos participantes, a de Implantação por 13,74% e a de manutenção apareceu com 12,48% de uso. 9,16% informaram usar essas metodologias na fase de Arquitetura de Software e 9% na fase de estudo de viabilidade. A integração de sistemas foi citada uma vez, no grupo “outros”.

A questão seguinte pergunta quais os tipos de produtos ou projetos de software são desenvolvidos no ambiente de trabalho do pesquisado. O item mais citado foi o de sistemas Web com 31,10%, seguido pelos sistemas de informação (21,77%), aplicativos móveis (18,18%), sistemas de gestão empresarial (13,40%), aplicação de análise de dados (9,81%) e sistemas embarcados (3,11%). Entre as opções apresentadas na pergunta, o desenvolvimento de jogos foi a menos citada, com apenas 0,72% de respostas. Dois pesquisados deixaram esse item em branco, e a opção “outros” teve 1,44% de citações em itens como: sistemas para segurança cibernética, utilização

Na questão que se refere aos estilos/padrões arquiteturais utilizados, foram citados, na respectiva ordem, as arquiteturas em microsserviços (16,08%), cliente-servidor (14,98%), em camadas (12,33%), orientada a serviços (9,47%), monolítica (9,03%), Clean Architecture (8,59%), Event Driven e Middleware com 5,29%. A arquitetura baseada em componentes, o Publish-subscribe e os que responderam não utilizar estilos arquiteturais representam 3,08% das respostas. As camadas hexagonais têm 2,86% e a Remote Procedure Call, 2,42%. As respostas em branco totalizam 1,54% dos formulários e os outros itens como Pipe-and-Filter, Remote Method Invocation, Microkernel e Vertical Slice obtiveram menos de 2% do total das respostas.

A próxima questão solicitou que o participante fizesse uma relação entre os estilos arquiteturais utilizados e as metodologias ágeis aplicadas a eles. Seguindo a mesma tendência da questão que identificou as metodologias ágeis mais usadas essa informação se refletiu na presente questão ratificando a utilização do Scrum e do Kanban como as metodologias mais adotadas e citadas como aplicáveis em todos os estilos arquiteturais apresentados, conforme evidenciado na Figura 12. Além delas, a XP também apresentou citações de uso com todos os estilos. Após essas metodologias, aparecem o DevOps, Lean e Scrum of Scrums com uma quantidade significativa de citações e utilização na maioria dos estilos. As metodologias SAFe, DAD, Crystal, ASD e DSDM foram mais citadas em estilos como SOA, cliente-servidor, em camadas e microsserviços. A FDD e a LeSS tiveram uma quantidade pequena de citações. Uma característica que trouxe um dado

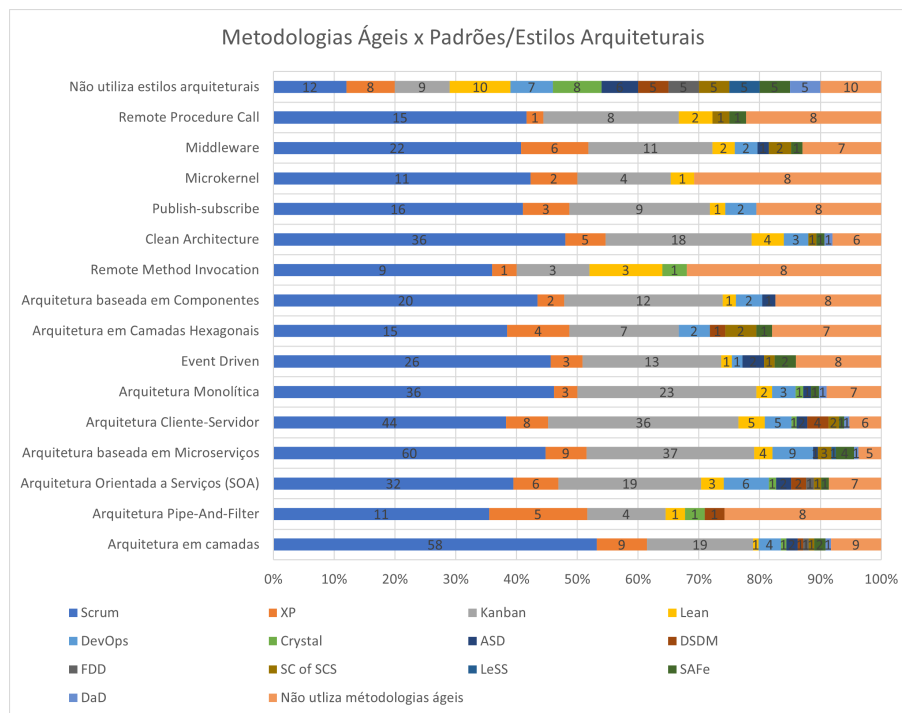


Fig. 12: Metodologias Ágeis x Padrões/Estilos Arquiteturais

importante foram as citações de 75 participantes que utilizam as metodologias ágeis, mas informaram não usá-las com estilos arquiteturais, e 109 que utilizam os estilos arquiteturais, mas sem relacioná-los às metodologias ágeis.

#### E. Benefícios e Dificuldades da Utilização das Metodologias Ágeis no Desenvolvimento de Software (QP5)

As duas primeiras questões desse grupo (4.1 e 4.2) buscam identificar os benefícios da utilização das metodologias ágeis no desenvolvimento de software nos contextos gerencial e técnico, respectivamente.

Quanto aos benefícios percebidos na parte gerencial, a melhoria da comunicação da equipe foi a mais citada com 24,09%, seguida da melhoria na previsibilidade de prazos, com 18,84%. Completam essa lista a melhor interação com os clientes (15,58%), a redução de riscos (14,67%), o aumento da satisfação da equipe (14,13%) e a redução dos custos do projeto (10,69%). Em percentuais menores que 1%, na opção “outros”, foram citados o aumento da produtividade da equipe, o melhor planejamento, a execução e a transparência. 5 pesquisados não responderam a essa pergunta e 3 citaram não perceber nenhuma melhoria gerencial com a aplicação das metodologias ágeis.

Quanto aos benefícios na área técnica, a melhoria na produtividade foi citada com 20,34% das respostas, seguida da maior velocidade de entrega (19,59%), rapidez no *feedback* das demandas (18,47%), maior qualidade do software (16,60%), diminuição de bugs (12,87%) e utilização de testes contínuos (10,26%). Na opção “outros”, foi citada a melhor previsibilidade de escopo. 5 pessoas não responderam a essa questão e 4 não perceberam melhorias na parte técnica.

As duas questões finais (4.3 e 4.4) buscavam identificar os desafios e as dificuldades da utilização das metodologias ágeis no desenvolvimento de software, tanto na parte gerencial quanto na técnica.

A questão 4.3 pergunta sobre as dificuldades e os desafios da utilização das metodologias ágeis na parte gerencial e a falta de maturidade em metodologias ágeis, com 23,04%, foi a opção mais citada. A resistência a mudanças e a falta de treinamento em metodologias ágeis aparecem logo depois, com 17,65% e 17,16%, respectivamente, seguidos, com 10,78%, pela falta de disponibilidade do cliente, 9,80% da dificuldade de alocação de recursos. Com 8,09%, foi citada a dificuldade de comunicação e com 7,84%, a dificuldade em escalar metodologias ágeis para grandes projetos. 3,19% dos pesquisados informaram que não têm desafios ou dificuldades e 1,47% não responderam a essa questão. Na opção “outros”, com uma citação cada, foram mencionadas a dificuldade dos donos em respeitarem o ciclo do Scrum, a não aplicação completa da metodologia ágil quando as solicitações são urgentes, as mudanças constantes e a falta de transição para a metodologia ágil. Quando os desafios e as dificuldades são relacionados à parte técnica, 25,90% dos pesquisados citaram as mudanças frequentes de requisitos, 20% citaram a dificuldade em lidar com as mudanças constantes, 16,39% questionaram a demora do *feedback* dos clientes, 14,43% a perda de produtividade com reuniões diárias, 8,85% a utilização de testes automatizados e 6,56% citaram como dificuldade a reutilização de código. Dentre os pesquisados, a quantidade que citou não perceber dificuldades ou desafios na parte técnica foi de 4,59%, e 1,64% deixou essa questão em branco. A opção “outros” recebeu 1,64% de menções, citando a resistência e o respeito à metodologia, falta de definição precisa do escopo e



a preocupação excessiva dos desenvolvedores com o processo consumindo muito tempo. Todas tiveram uma citação cada.

## V. DISCUSSÃO

Com a análise e o entendimento dos resultados obtidos com a aplicação do questionário, pode-se perceber, quando se leva em consideração o perfil dos participantes, uma predominância de pessoas com ensino superior completo, mas com baixa incidência de pós-graduados. Os cargos/funções de desenvolvedores de software e analistas de sistemas ainda são predominantes na amostra participante, tendo 75% trabalhando em ambiente remoto ou híbrido, com a grande maioria trabalhando em empresas de tecnologia da informação. Isso mostra um cenário com possibilidades bem interessantes com relação ao acesso aos postos de trabalho, tanto para as empresas como para os profissionais. Quanto à faixa etária, percebe-se uma distribuição dos participantes de forma bem diversa, estando a minoria em faixas etárias mais altas, principalmente a partir dos 46 anos. No tocante à experiência, quando o foco é o desenvolvimento de softwares, os resultados se apresentaram bem distribuídos, tendo desde pessoas com pouco experiência até outras com muitos anos de prática. Porém, quando o assunto abordado foram as metodologias ágeis, a grande maioria tem pouco tempo de prática, com mais de 86% tendo até 10 anos de experiência com essas metodologias. Essa informação pode servir de base para estudos futuros que busquem relacionar a maturidade de cada área e a relação com a sua popularização entre os profissionais.

Apesar do pouco tempo de experiência com as metodologias ágeis, os resultados apresentam um nível de maturidade intermediário e uma utilização bem consistente das metodologias e das suas ferramentas nos contextos organizacionais. Quanto às metodologias mais utilizadas, o Scrum e o Kanban possuem uma predominância de aplicação, sendo operacionalizados através uma grande gama de ferramentas, sendo as mais citados o Jira e o Trello, apesar de não serem unanimidades, tendo várias outras ferramentas em uso.

Outra informação importante diz respeito às fases do ciclo de desenvolvimento de software em que as metodologias ágeis estão sendo usadas. Fica explícita essa utilização em todas as etapas do ciclo, tendo uma maior predominância nas fases de planejamento e desenvolvimento e menor uso nas fases de estudo de viabilidade e Arquitetura de Software. Com relação aos tipos de produto de software, os sistemas Web, os sistemas de informação e os aplicativos móveis predominam dentre o que está sendo desenvolvido nas organizações utilizando, de forma mais frequente, as arquiteturas em camadas, cliente-servidor e os microsserviços.

A validade da construção do questionário emerge como uma preocupação quanto ao alinhamento dos resultados com o tema de pesquisa. Para mitigar esse risco, realizou-se uma pré-análise por meio da aplicação de uma pesquisa-piloto com sete participantes, visando validar previamente os dados coletados, identificar possíveis desvios de objetivo e coletar *feedback* da percepção dos participantes com relação às perguntas apresentadas. Devido à quantidade limitada de respostas e à seleção dos participantes através de contatos pessoais dos

pesquisadores, ou seja, uma amostragem intencional não probabilística, os resultados não podem ser generalizados.

## VI. CONCLUSÃO

Com a utilização das metodologias ágeis, foram percebidos benefícios nas partes gerencial e operacional das organizações. Fica evidente a melhoria na comunicação entre a equipe e das equipes para com os clientes. A previsibilidade com relação aos prazos de entrega impacta na melhoria de produtividade, na satisfação das equipes e na velocidade do atendimento às demandas. Apenas um participante respondeu que não percebeu melhorias, mas corresponde a um percentual muito baixo e não impactante nos resultados obtidos.

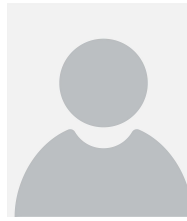
Quanto às dificuldades e aos desafios do uso das metodologias ágeis nos ambientes de desenvolvimento, uma grande maioria citou a falta de maturidade e de treinamento nas metodologias ágeis alinhadas à resistência a mudanças por parte das equipes e *stakeholders*. Na parte operacional, foram relatadas dificuldades em lidar com as mudanças frequentes de requisitos e a demora no *feedback* dos clientes, o que impacta na produtividade e na entrega das demandas nos prazos acordados.

Com base nos dados coletados fica evidente que existe, no cenário de desenvolvimento de software no Brasil, uma aplicação robusta das metodologias ágeis alinhadas a diferentes estilos arquiteturais, porém ainda sem uma uniformização de ferramentas e padrões. Evidencia-se a necessidade do aumento do nível de maturidade e treinamento das equipes para utilização das metodologias ágeis, bem como um maior envolvimento e comprometimento do cliente na agilidade do *feedback* das demandas solicitadas. A criação de uma metodologia de padronização para implantação e utilização das metodologias ágeis, em ambientes de desenvolvimento de software e alinhada à Arquitetura de Software, é uma vertente importante para estudos futuros.

## REFERENCES

- [1] Abrahamsson, Pekka and Babar, Muhammad Ali and Kruchten, Philippe, *Agility and architecture: Can they coexist?* , IEEE software. v.27, no.2, pp.16–22, 2010.
- [2] Al-Saqqa, Samar and Sawalha, Samer and AbdelNabi, Hiba , *Agile software development: Methodologies and trends* , International Journal of Interactive Mobile Technologies. v.14, no.11, 2020.
- [3] Alaidaros, Hamzah and Omar, Mazni and Romli, Rohaida, *The state of the art of agile kanban method: challenges and opportunities* , Independent Journal of Management & Production. Vol. 12, no. 8, pp.2535–2550, 2021.
- [4] Boehm, Barry, *A view of 20th and 21st century software engineering* , Proceedings of the 28th international conference on Software engineering. pp.12–29, 2006.
- [5] Booch, Grady, *The economics of architecture-first* , IEEE Software. Vol. 24, no. 5, pp.18–20, 2007.
- [6] Chan, Frank KY and Thong, James YL, *Acceptance of agile methodologies: A critical review and conceptual framework* , Decision support systems. Vol. 46, no. 4, pp.803–814, 2009.
- [7] Garlan, David, *Software architecture: a travelogue* , Future of Software Engineering Proceedings. pp.29–39, 2014.
- [8] Gothelf, J., Seiden, J., *Agile Architecture: A Practical Guide* O'Reilly Media, Inc, 2021.
- [9] IEEE *IEEE standard for architectural description of software-intensive systems* IEEE Std 1471-2019. Piscataway, NJ: IEEE, 2019.
- [10] IEEE *ISO/IEC/IEEE International Standard - Systems and software engineering – System life cycle processes*. ISO/IEC/IEEE 12207:2017(E) First edition 2017-11, 2017.

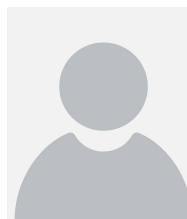
- [11] IEEE ISO/IEC/IEEE *International Standard - Systems and software engineering – System life cycle processes*. ISO/IEC/IEEE 15288 First edition 2015-05-15, 2015.
- [12] Ismat Soueid, Mohamad and Corá Martins, Antonio Felipe, *Scrum and Agility Beyond It: Evidences in the Brazilian Mining Industry*, Revista de Gestão e Projetos. Vol. 12, no. 1, 2021.
- [13] Kruchten, P., *Agile Architecture: An Introduction* Manning Publications Co, 2019.
- [14] Li, Ruiyin and Liang, Peng and Soliman, Mohamed and Avgeriou, Paris, *Understanding software architecture erosion: A systematic mapping study*, Journal of Software: Evolution and Process. Vol. 34, no. 3, pp.e2423, 2022.
- [15] Linaker, Johan and Sulaman, Sardar Muhammad and Host, Martin and de Mello, Rafael Maiani, *Guidelines for conducting surveys in software engineering v. 1.1*, Lund University. Vol. 50, 2015.
- [16] Medeiros, JDRV and Alves, DC and Vasconcelos, AML and Schuene-mann, CTLLS and Wanderley, E, *Engenharia de requisitos em projetos ágeis: uma revisão sistemática da literatura*, Revista Principia-Divulgação Científica e Tecnológica do IFPB. Vol. 28, pp.11–24, 2015.
- [17] Parnas, David, *On the criteria for decomposing systems into modules*, Communications of ACM. Vol. 15, no.12, pp.1053–1058, 1971.
- [18] Quezada-Sarmiento, Pablo Alejandro and Mengual Andrés, Santiago, *Implementación de una solución web y móvil para la gestión vehicular basada en Arquitectura de Aspectos y metodologías ágiles: Un enfoque educativo de la teoría a la práctica*, RISTI-Revista Iberica de Sistemas e Tecnologias de Informação, vol. 25, pp. 98–111, 2017.
- [19] Sganderla, Mauricio Andreazza and Lacerda, Guilherme and Ribeiro, Vinicius Gadis and Silveira, Sidnei Renato, *Aprimorando a Gerência e o Desenvolvimento de Software com Metodologias Ágeis*, RCT-Revista de Ciência e Tecnologia, vol. 2, no.2, 2016.
- [20] Sievi-Korte, Outi and Richardson, Ita and Beecham, Sarah, *Software architecture design in global software development: An empirical study*, Journal of Systems and Software, vol. 158, 2019.
- [21] Srivastava, Apoorva and Bhardwaj, Sukriti and Saraswat, Shipra, *SCRUM model for agile methodology*, International Conference on Computing, Communication and Automation (ICCCA), pp. 864–869, 2017.
- [22] Shrivastava, Anchit and Jaggi, Isha and Katoch, Nandita and Gupta, Deepali and Gupta, Sheifali, *A systematic review on extreme programming*, Journal of Physics: Conference Series, vol.1969, no.1, 2021.
- [23] Tamburri, Damian and Kazman, Rick and Van den Heuvel, Willem-Jan, *Splicing community and software architecture smells in agile teams: An industrial study*, 2019.
- [24] Vasquez, Guillermo Omar Pizarro, *Descriptive Study About Design and Architecture Practices in Startups*, Ciencia Unemi. Vol. 10, no.23, pp.125–132, 2017.
- [25] Van Casteren, Wilfred, *The Waterfall Model and the Agile Methodologies: A comparison by project characteristics*, Research Gate. Vol. 2, pp.1–6, 2017.
- [26] Yang, Chen and Liang, Peng and Avgeriou, Paris, *A systematic mapping study on the combination of software architecture and agile development*, Journal of Systems and Software. Vol. 111, pp.157–184, 2016.
- [27] Wohlrab, Rebekka and Eliasson, Ulf and Pelliccione, Patrizio and Heldal, Rogardt, *Improving the consistency and usefulness of architecture descriptions: Guidelines for architects*, IEEE International Conference on Software Architecture (ICSA). pp.151–160, 2019.



**2nd author** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis,

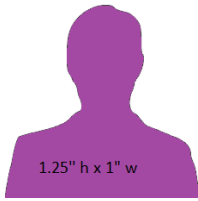
viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.



**2nd author** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis,

viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



**3rd author** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis,

viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.