UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Rodrigo Gonçalves Guedes

Um guia de boas práticas em Desenvolvimento Global de Software Ágil

Recife

2023

Rodrigo Gonçalves Guedes

Um guia de boas práticas em Desenvolvimento Global de Software Ágil

Monografia apresentada ao Curso de Licenciatura em Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Licenciado em Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Computação

Curso de Licenciatura em Computação

Orientador: Marcelo Marinho

Recife

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

G924g Guedes, Rodrigo Gonçalves
Um guia de boas práticas em Desenvolvimento Global de Software Ágil / Rodrigo Gonçalves Guedes. - 2023.
86 f. : il.

Orientador: Marcelo Marinho.
Inclui referências e apêndice(s).

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Licenciatura em Computação , Recife, 2023.

1. Desenvolvimento de Software Global. 2. Desenvolvimento Ágil de Software. 3. Portfolio. 4. Entrevistas. 5. Engenharia de Software. I. Marinho, Marcelo, orient. II. Título

CDD 004

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ–REITORIA DE ENSINO DE GRADUAÇÃO
COORDENAÇÃO DO CURSO DE LICENCIATURA EM COMPUTAÇÃO
DEPARTAMENTO DE COMPUTAÇÃO

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por **Rodrigo Gonçalves Guedes** às 17:00 o dia 15 de setembro de 2023, como requisito para conclusão do curso de Licenciatura em Computação da Universidade Federal Rural de Pernambuco, intitulado “Um Guia de Boas Práticas em Desenvolvimento Global de Software Ágil”, orientado pelo professor **Marcelo Luiz Monteiro Marinho** e aprovado pela seguinte banca examinadora:

Recife, 15 de setembro de 2023

_____

Marcelo Luiz Monteiro Marinho
DC/UFRPE
Presidente da banca

_____

Taciana Pontual
DC/UFRPE
Avaliadora

À minha familia, minha namorada, meus amigos, professores e todos que me deram suporte durante minha trajetória …

Agradecimentos

Gostaria de dedicar este trabalho às pessoas que fizeram parte da minha trajetória durante a graduação. Pessoas que, de inúmeras formas possíveis, dentro ou fora da universidade, conseguiram agregar algum valor para mim, ajudando-me a aprender e crescer como aluno e ser humano.

Agradeço primeiramente à minha família, meus pais Dora e Sandro, e ao meu irmão Davi, por sempre acreditarem em mim e desejarem o melhor para mim. Sem dúvida, eles foram meu alicerce desde o início da minha formação como pessoa.

Também expresso profunda gratidão à minha namorada, Natalia. Durante a minha jornada, cheguei à beira de desistir inúmeras vezes do curso, e seu apoio emocional foi essencial e fundamental para que eu pudesse escrever este trabalho. Além disso, ela me ajudou muito nas tomadas de decisões e no suporte aos estudos em algumas disciplinas.

Meus agradecimentos também vão para Lucas, meu psicólogo, que me guiou e ajudou no processo de persistência no curso, oferecendo tanto apoio emocional quanto estratégias para lidar com diversas situações acadêmicas. Lucas também foi fundamental para eu entender e focar no meu objetivo.

Agradeço ao professor Marcelo, com quem preferi cursar algumas disciplinas devido à sua didática e empenho em sala de aula. Ele se esforçou para ensinar bem e garantir que os alunos pudessem acompanhar, mesmo passando por momentos difíceis. Marcelo possui amplo conhecimento e experiência em pesquisas científicas, o que certamente fez a diferença no meu desenvolvimento neste trabalho.

Gostaria também de agradecer a outros professores que deixaram sua marca na minha jornada durante todo o curso. Portanto, expresso minha imensa gratidão a Robson Medeiros pela excelente introdução ao curso, a Péricles Barbosa por me mostrar meu potencial pela primeira vez, a Marcelo Pedro pela excelente didática e flexibilidade no curso de cálculo, no qual eu enfrentava muitas dificuldades, a Paulo Anselmo por me aperfeiçoar na programação orientada a objetos, a Rafael Barbosa pelo suporte nas minhas dificuldades em álgebra vetorial, mesmo fora do horário de aula, a Danilo Araujo pelo excelente período em que foi coordenador, a Marcos Cardoso pelo incentivo à criação do jogo de tabuleiro ”Afrociberdelia”no pensamento computacional, a Carlos Julian também pelo excelente período em que esteve na coordenação, e a Rozelma Soares e Rodrigo Lins nas disciplinas de estágio obrigatório, que me deram muito suporte, considerando a dificuldade de conciliar o estágio com o meu trabalho.

Agradeço aos colegas que enfrentaram todas as dificuldades junto comigo, incluindo Daniel Lemos, Alan Marinho, Huan Cristofer e Vanessa Oliveira. As experiências que compartilhamos desde o início, junto com o Daniel, e posteriormente nos estágios obrigatórios com Alan, Huan e Vanessa, foram enriquecedoras e gratificantes.

Também gostaria de agradecer aos meus colegas de trabalho da Accenture, que sempre me apoiaram e incentivaram a concluir o curso, mesmo diante de todas as demandas pesadas que temos. Com certeza, aprendi muito com vocês.

Por último, mas não menos importante, agradeço aos meus primos Raphael e Silvano por estarem ao meu lado nos momentos difíceis e por sempre tentarem elevar meu astral. Também agradeço aos meus melhores amigos Bruno, Gabriel, Guilherme e José, por estarem sempre presentes em minha vida e por me incentivarem a me formar.

Também gostaria de fazer menção a todos os outros que também fizeram parte desta jornada, seja professores, amigos do colégio, do futebol entre outros, mesmo que seus nomes não estejam presentes aqui. A todos vocês, meu muito obrigado!

“A mudança é inevitável, o progresso é opcional. Adotar metodologias ágeis é uma escolha estratégica para transformar a inevitável mudança em progresso sustentável.”

(Mary Poppendieck)

Resumo

O Desenvolvimento Ágil de Software (ASD) desempenha um papel crucial no cenário global de desenvolvimento de software. A pandemia da COVID-19 intensificou a necessidade de adaptação das práticas ágeis para ambientes remotos e distribuídos, enfatizando ainda mais a eficácia na colaboração. Este estudo expande pesquisas anteriores que identificaram 48 práticas ágeis em projetos globais, refletindo um esforço abrangente para analisar e sintetizar essas práticas. Conduzimos uma nova pesquisa (survey) para obter um conjunto mais amplo de respostas e realizamos entrevistas aprofundadas para correlacionar dados com observações construtivas. Como resultado, foram identificadas 13 práticas ágeis que emergiram como as mais aceitas nesse contexto. Essas práticas, como Planejamento, Comunicação e Auto Gerenciamento, provaram ser fundamentais para o sucesso de projetos AGSD. Com base nessas descobertas, desenvolvemos um portfólio abrangente que serve como um guia de referência para profissionais que buscam implementar práticas ágeis em contextos globais. Esta contribuição visa aprimorar o desenvolvimento ágil de software em cenários globais, promovendo resultados e eficiência superiores. À medida que as organizações se adaptam às demandas atuais e futuras, o AGSD se destaca como uma abordagem crucial para o sucesso em um mundo cada vez mais conectado.

Palavras-chave: Desenvolvimento de Software Global, Desenvolvimento Ágil de Software, Entrevistas, Portfolio, Engenharia de Software.

Abstract

The Agile Software Development (ASD) plays a pivotal role in the global software development landscape. The COVID-19 pandemic has further underscored the need to adapt agile practices for remote and distributed environments, emphasizing collaboration effectiveness. This study extends prior research that identified 48 agile practices in global projects, reflecting a comprehensive endeavor to analyze and synthesize these practices. We conducted a new survey to gather a broader set of responses and carried out in-depth interviews to correlate data with constructive observations. As a result, 13 agile practices have been identified as the most widely accepted in this context. These practices, such as Planning, Communication, and Self-Management, have proven to be fundamental for the success of AGSD projects. Based on these findings, we have developed an extensive portfolio serving as a reference guide for professionals seeking to implement agile practices in global contexts. This contribution aims to enhance agile software development in global scenarios, fostering superior results and efficiency. As organizations adapt to current and future demands, AGSD stands out as a crucial approach for success in an increasingly interconnected world.

Keywords: Global Software Development, Agile Software Development, Interviews, Portfolio, Software Engineering.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

GSD	Global Software Development
ASD	Agile Software Development
AGSD	Agile Global Software Development
GSE	Global Software Engineering
ASDT	Agile Software Development Team
SAFe	Scaled Agile Framework
SLR	Systematic Literature Review
TDD	Test Driven Development
CQRS	Command Query Responsibility Segregation
PO	Product Owner
PM	Product Manager
HPQC	HP Quality Center
CICD	Continuous Integration/Continuous Delivery

Sumário

1 Introdução

Nos últimos anos, o cenário do desenvolvimento de software tem sido marcado por transformações significativas, impulsionadas por uma demanda crescente por agilidade, eficiência e inovação. Nesse contexto, o Agile Software Development (ASD) emergiu como uma abordagem que visa atender a essas necessidades, ao mesmo tempo em que responde às demandas por flexibilidade e adaptabilidade nos processos de desenvolvimento(HIGHSMITH; COCKBURN, 2001).

A agilidade do ASD é particularmente relevante no contexto do Global Software Development (GSD), no qual equipes de desenvolvimento estão geograficamente dispersas. Esse cenário introduz desafios adicionais, como diferenças de fusos horários, barreiras culturais e dificuldades na comunicação. Para lidar com essas complexidades, abordagens ágeis são frequentemente adaptadas para a realidade do GSD, buscando manter a colaboração eficiente e a entrega de valor contínua . Como destacam (MISHRA; ALZOUBI, 2023), ”É crucial lembrar que o ágil é uma metodologia muito flexível, que pode ser adaptada às necessidades específicas da empresa, enquanto a metodologia em cascata é bastante rígida e formal, mas também pode ser alterada conforme necessário.”

Além disso, à medida que a globalização continua a moldar o cenário empresarial, o Agile Global Software Development (AGSD) ganhou destaque. Essa abordagem envolve equipes de desenvolvimento distribuídas em diferentes locais geográficos, muitas vezes pertencentes a diferentes culturas e fusos horários. A pandemia de COVID-19 também trouxe à tona a importância de adaptar práticas ágeis para cenários globais e distribuídos. Com sua chegada, o trabalho remoto foi amplamente adotado e a necessidade da comunicação ficou evidente para o bom andamento dos projetos (ÅGREN; KNOPH; SVENSSON, 2022), algo que refoça o valor ágil ”indivíduos e interações acima de processos e ferramentas”(BECK; AL., 2001). Isso também é confirmado no estudo de (MAREK; WIŃSKA; DĄBROWSKI, 2021), onde é comprovado que ”as equipes de Desenvolvimento Ágil de Software se adaptaram bem às circunstâncias da Covid-19, com a mudança mais comum sendo a transição para o trabalho remoto.”Essa adaptação eficaz foi favorecida pelo uso prévio de ferramentas que já suportavam o trabalho distribuído, onde a comunicação é essencial, demonstrando a agilidade inerente a essas equipes.

Além disso, (CAMARA et al., 2020) e (ALVES, 2021) identificaram 48 práticas frequentemente citadas em contextos AGSD. Partindo dessas práticas, o presente trabalho de conclusão de curso busca contribuir para o entendimento das práticas ágeis

mais utilizadas no contexto GSD. Além de identificar as mais aceitas, o estudo também propõe uma análise narrativa, por meio de entrevistas realizadas com profissionais da área, com o objetivo de entender como essas práticas funcionam em diferentes contextos. As informações adquiridas durante essas entrevistas foram utilizadas para compilar um portfólio que documenta a aplicação dessas práticas em situações reais. Esse documento pode ser encontrado no Apêndice A

1.1 Objetivos

1.1.1 Objetivo Geral

A presente pesquisa tem como objetivo entender como funcionam as práticas ágeis em projetos distribuídos. Identificar as práticas mais utilizadas e catalogá-las num documento de auxílio para a utilização das mesmas em ambientes GSD.

1.1.2 Objetivos Específicos

1. Analisar as práticas ágeis mais utilizadas atualmente em projetos globais.
2. Realizar uma pesquisa quantitativa utilizando um questionário (survey) para compreender como funcionam os projetos globais de desenvolvimento de software.
3. Extrair conclusões qualitativas sobre como profissionais experientes na área utilizam tais práticas e quais suas dificuldades ou beneficios no momento da execução no seu ambiente.
4. Catalogar e demonstrar as práticas ágeis mais utilizadas em ambientes GSD.

1.2 Estrutura

A estrutura deste trabalho de conclusão de curso é a seguinte: No Capítulo 2, apresentamos o embasamento teórico dos temas envolvidos na pesquisa. No Capítulo 3, descrevemos os métodos de pesquisa utilizados. No Capítulo 4, apresentamos os resultados de uma pesquisa do tipo survey (SCHEUREN, 2004) que visa ampliar o alcance dos resultados obtidos nos trabalhos utilizados como referência. O Capítulo 5 detalha as informações coletadas durante as entrevistas semi-estruturadas (LUO; WILDEMUTH, 2009) com profissionais da área. Em seguida, no Capítulo 6, realizamos discussões e abordamos as limitações dos resultados obtidos. Finalmente, no Capítulo 7, apresentamos as conclusões e sugerimos direções para futuras pesquisas. Este trabalho busca contribuir para o entendimento e aplicação eficaz de práticas ágeis no

contexto AGSD, proporcionando insights valiosos para profissionais e pesquisadores da área.

2 Referencial Teórico

Esta seção tem como objetivo apresentar o referencial teórico, também conhecido como revisão de literatura. Ele consiste na análise crítica de estudos e teorias relacionadas aos assuntos do presente projeto, fornecendo embasamento teórico para a pesquisa. O referencial teórico contextualiza o problema para o leitor, identifica lacunas no conhecimento, embasa a fundamentação teórica e contribui para o avanço da área de estudo.

2.1 Agile Software Development (ASD)

O ASD é uma abordagem iterativa e incremental que busca mitigar os desafios do desenvolvimento de software tradicional. Diferente do modelo cascata, o ASD enfatiza a colaboração, entrega incremental, comunicação efetiva e adaptação contínua aos requisitos em evolução. O Manifesto Ágil (BECK; AL., 2001) é a base comum para os métodos ágeis, como Scrum (SCHWABER; SUTHERLAND, 2020), Extreme Programming (BECK, 1999) e Lean Software Development (POPPENDIECK; POPPENDIECK, 2003). Esses métodos promovem práticas-chave, como programação em pares, testes automatizados e entrega contínua (ERDOGMUS; MORISIO; TORCHIANO, 2005), resultando em maior eficiência, satisfação do cliente e qualidade do software.

A adoção do ASD traz benefícios significativos, incluindo entrega incremental de valor ao cliente e maior adaptabilidade a mudanças nos requisitos (SERRADOR; PINTO, 2015). No entanto, equipes podem enfrentar desafios ao adotar a abordagem ágil em ambientes tradicionais (HAJJDIAB; TALEB, 2011). Além disso, no livro ”Agile Software Development: The Cooperative Game”, Alistair Cockburn (COCKBURN, 2017) explora como o ASD pode ser visto como um jogo cooperativo entre as equipes de desenvolvimento e os stakeholders, contribuindo para o sucesso de projetos ágeis.

Contudo, críticas ao ASD apontam para a necessidade de equilibrar a agilidade com preocupações de arquitetura e documentação (ABRAHAMSSON; BABAR; KRUCHTEN, 2010). Torna-se essencial considerar aspectos arquiteturais e a documentação apropriada, especialmente em projetos de maior complexidade, para garantir a sustentabilidade e a manutenibilidade do software desenvolvido.

2.2 Global Software Development (GSD)

O GSD é uma estratégia cada vez mais adotada pelas empresas para enfrentar os desafios da globalização e do mercado competitivo (MARINHO; NOLL; BEECHAM, 2018). Essa abordagem envolve equipes de desenvolvimento distribuídas geograficamente, que colaboram remotamente para criar produtos de software inovadores. De acordo com (CARMEL, 1999), o GSD oferece benefícios significativos, como acesso a talentos globais, redução de custos e maior disponibilidade de recursos. A capacidade de operar em fusos horários diferentes permite uma entrega contínua e rápida do software (CARMEL, 2011).

No entanto, o GSD também apresenta desafios únicos que precisam ser abordados. Diferenças culturais, barreiras linguísticas e comunicação assíncrona podem afetar a colaboração entre as equipes (MOCKUS; HERBSLEB, 2001) (MARINHO; NOLL; BEECHAM, 2018). A falta de comunicação cara a cara pode levar a mal-entendidos e problemas na interpretação dos requisitos (CARMEL; AGARWAL, 2001). Além disso, questões de segurança e proteção de dados devem ser consideradas ao compartilhar informações entre equipes globais (SINDRE; OPDAHL, 2005).

Para superar esses desafios, várias práticas e abordagens têm sido propostas. A adoção de ferramentas de colaboração em tempo real e plataformas de gerenciamento de projetos pode melhorar a comunicação e a coordenação entre as equipes (ZAFAR; ALI; SHAHZAD, 2011). Além disso, criar uma cultura de trabalho colaborativa e inclusiva é essencial para o sucesso do GSD (CARMEL, 1999).

2.3 Agile Global Software Development (AGSD)

AGSD é uma abordagem que combina os princípios ágeis do ASD com o contexto do GSD. Essa abordagem visa permitir que equipes distribuídas geograficamente colaborem de forma ágil, eficiente e colaborativa para desenvolver produtos de software de alta qualidade (VALLON et al., 2018).

Estudos têm destacado os benefícios do AGSD. A adoção de práticas ágeis em ambientes distribuídos pode levar a maior flexibilidade e capacidade de resposta às mudanças de requisitos, bem como à entrega contínua e incremental de software (PODARI et al., 2020). A colaboração ágil também pode superar barreiras culturais e linguísticas, permitindo uma comunicação mais efetiva entre as equipes (VALLON et al., 2018).

No entanto, o AGSD também enfrenta desafios únicos. A coordenação entre equipes distribuídas pode ser complexa, com diferentes fusos horários e culturas afetando a sincronização das atividades no GSD (VALLON et al., 2018). Dessa forma as

práticas ágeis se tornam viáveis visto que elas estão sendo cada vez mais utilizadas no desenvolvimento de software de forma global (MARINHO et al., 2019). Além disso, a comunicação efetiva é crucial para evitar mal-entendidos e garantir a colaboração bem-sucedida no ASD (CALEFATO; DAMIAN; LANUBILE, 2012). Questões relacionadas à segurança e proteção de dados também devem ser consideradas para garantir o cumprimento de regulamentações e padrões no GSD (HOSSAIN; ALI; BABAR, 2020).

As tendências futuras no AGSD envolvem a integração de tecnologias emergentes. A aplicação de práticas DevOps pode melhorar a colaboração e a integração contínua entre as equipes distribuídas no ASD (HOSSAIN; ALI; BABAR, 2020). Além disso, o uso de ferramentas avançadas de comunicação e colaboração, como videoconferências e plataformas de trabalho em tempo real, tem o potencial de facilitar a interação entre membros de equipes globais no GSD (CALEFATO; DAMIAN; LANUBILE, 2012). A incorporação de inteligência artificial e automação de testes também promete aumentar a eficiência e a qualidade dos processos de desenvolvimento no AGSD (KHANNA; POPLI; CHAUHAN, 2021).

2.4 Trabalhos Relacionados

Em (HOSSAIN; BABAR; PAIK, 2009), um trabalho significativo aborda o crescente interesse na aplicação de práticas ágeis em projetos GSD, com foco no popular método Scrum. Para alcançar um panorama abrangente, os pesquisadores realizaram uma revisão sistemática de 366 artigos, identificando 20 estudos relevantes que forneceram insights valiosos sobre os desafios e estratégias associados ao uso do Scrum em GSD. Os resultados enfatizaram a importância de compreender os desafios que impactam a comunicação, colaboração e coordenação em projetos GSD, bem como as estratégias adotadas para enfrentá-los. No entanto, é importante mencionar que a força das evidências em relação a essas estratégias foi considerada baixa, o que apontou para a necessidade de mais pesquisas e estudos empíricos nessa área. A relevância dos achados não se limita apenas ao campo acadêmico, pois profissionais de GSD podem se beneficiar ao conhecer os obstáculos e encontrar maneiras de superá-los, tornando o uso do Scrum mais eficaz em suas práticas. Essas descobertas ressaltaram a importância de fortalecer o entendimento sobre a adoção das práticas Scrum em projetos GSD e servem como um guia útil para aqueles envolvidos em tais projetos.

Já o trabalho de (AVRITZER; BRONSARD; MATOS, 2010) discute as vantagens de utilizar práticas ágeis no desenvolvimento global, baseado em duas experiências significativas em projetos distribuídos. O primeiro projeto teve duração de três anos, envolveu cerca de cem desenvolvedores em dez equipes distribuídas nos EUA, Europa e Índia. O objetivo era construir uma plataforma para uma linha de produtos de

sistemas de controle e monitoramento baseados em PC, substituindo sistemas legados que apoiavam diferentes linhas de negócio. Desafios principais foram a adoção de várias tecnologias novas e a criação de uma plataforma genérica. O projeto ágil global enfrentou riscos significativos, incluindo desafios de comunicação, com equipes distantes (EUA e Índia) tendo apenas uma hora de sobreposição de trabalho. Devido à duração prevista do projeto, não era prático pedir que as equipes ajustassem seus horários de trabalho para aumentar a sobreposição. O risco mais sério foi a dependência de múltiplas tecnologias não testadas ou muito imaturas. No entanto, o projeto adotou processos ágeis, em parte, devido à sua capacidade de lidar com tecnologias voláteis. O trabalho apresenta reflexões sobre como lidar com essas dificuldades e as melhorias que a abordagem ágil trouxe para o desenvolvimento global.

(JALALI; WOHLIN, 2010) realizaram uma revisão sistemática da literatura sobre o uso de práticas ágeis e Lean Software Development em Global Software Engineering (GSE). O estudo buscou identificar as circunstâncias em que essas práticas foram aplicadas com eficiência. Foram considerados termos relacionados a práticas ágeis e GSE em busca de artigos revisados por pares, publicados entre 1999 e 2009. A análise revelou que, na maioria dos casos, as práticas ágeis foram adaptadas conforme o contexto e requisitos dos projetos de GSE. A necessidade de integrar experiências e práticas para apoiar projetos ágeis não co-localizados foi identificada como um ponto importante para futuras pesquisas. O desenvolvimento de software global, com equipes distribuídas e culturas diversas, apresenta desafios de comunicação e coordenação. Os métodos ágeis, conhecidos por sua flexibilidade e colaboração, enfrentam complexidades adicionais nesse contexto. Embora alguns estudos tenham relatado sucesso na integração de ágeis e GSE, a aplicabilidade dessas experiências em diferentes contextos e demandas de projeto ainda precisa ser explorada. O estudo buscou resumir a literatura e investigar quais práticas ágeis foram efetivamente utilizadas em projetos de GSE. O artigo conclui apresentando os resultados, metodologia e apontando direções para futuras pesquisas nessa área.

O trabalho de (MAREK; WIŃSKA; DĄBROWSKI, 2021) relata uma pesquisa através de survey para entender como se comportaram os Agile Software Development Teams (ASDT) durante a pandemia do COVID-19. Visto que o virus é bastante letal e contagioso, medidas protetivas de isolamento social foram tomadas no mundo todo. Sendo assim o trabalho remoto foi uma alternativa aderida pela maioria das empresas de desenvolvimento de software, com isso um survey de 22 questões foi disponibilizado num período de 10 dias e foram coletadas 120 respostas. Os resultados mostraram que a maioria das equipes de ASDT conseguiu fazer a mudança para o trabalho remoto sem grandes problemas, devido à familiaridade prévia com o trabalho distribuído. A implementação de novas ferramentas foi limitada, facilitando a transição. Além disso, o trabalho remoto melhorou a comunicação e reduziu reuniões desnecessárias. As

equipes continuaram utilizando métricas e ferramentas de cooperação, como Jira e Confluence, mesmo em ambientes co-localizados.

Em (CAMARA et al., 2020) foi feita uma revisão sistemática da literatura (SLR) que teve como objetivo entender quais práticas ágeis podem ser usadas em projetos GSD. Foram encontradas e descritas 48 práticas que de alguma foram são úteis e funcionam em projetos GSD. O trabalho teve continuidade em (FIGUEREDO, 2020) que seguiu a mesma linha porém dando foco as práticas escaláveis que adotam as práticas do "Scaled Agile Framework" (SAFe). Nessa pesquisa das 48 práticas encontradas, foram classificadas 18 como práticas que adotam o SAFe. Em (ALVES, 2021) que também deu sequência ao trabalho, fez um survey com engenheiros de software e foi identificado que essas práticas implementáveis fornecem soluções para gerenciar equipes GSD e assim, abraçar a agilidade.

O presente trabalho se apresenta como uma continuação de (CAMARA et al., 2020) e (ALVES, 2021) pois é baseado nas 48 práticas citadas neles. Além disso serve de complemento de informações para (FIGUEREDO, 2020) visto que algumas práticas que utilizam SAFe são consideradas como mais aceitas. Além disso o trabalho de (HOSSAIN; BABAR; PAIK, 2009) foca em estudar o Scrum, framework que também é abordada no presente trabalho. (AVRITZER; BRONSARD; MATOS, 2010) aborda dificuldades encontradas em projetos distribuídos. Na presente pesquisa podem ser encontradas recomendações que possam ajudar a mitigar tais dificuldades. Por fim a pesquisa serve de complemento para os trabalhos de (JALALI; WOHLIN, 2010) e (MAREK; WIŃSKA; DĄBROWSKI, 2021) visto que trazem informações importantes sobre práticas ágeis em projetos distribuídos.

3 Metodologia

Esta seção tem como objetivo mostrar a metodologia utilizada para a construção do presente trabalho, que é continuidade do trabalho de (CAMARA et al., 2020). Ela descreverá como foi feita a coleta de dados, cálculos para seleção das práticas mais utilizadas, decisão das pesquisas semi-estruturadas para que seja possível a confecção do documento de apoio das metodologias.

Para que fique mais claro, o projeto foi dividido em 8 etapas, sendo elas:

1. Revisão bibliográfica de carater exploratória.
2. Identificação das 48 práticas ágeis baseadas em (CAMARA et al., 2020).
3. Confecção e coleta do survey baseado em (ALVES, 2021). O mesmo pode ser encontrado no Apêndice B.
4. Identificação das práticas ágeis mais utilizadas utilizando desvio-padrão das respostas do survey.
5. Identificação dos desenvolvedores candidatos a entrevistas.
6. Confecção e realização da entrevistas (ANZALDÚA, 2020) semi-estruturadas (LUO; WILDEMUTH, 2009) nos desenvolvedores. O roteiro das entrevistas pode ser encontrado no Apêndice C
7. Elaboração do portifólio de auxílio contendo as práticas ágeis definidas no item 4. O mesmo pode ser encontrado no Apêndice A.
8. Análise e conclusão dos resultados obtidos em comparação com (CAMARA et al., 2020) e (ALVES, 2021).

Para eludicar o processo observe a Figura 1.

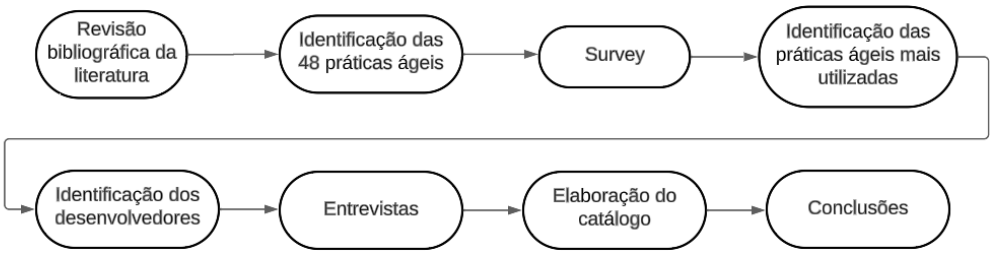

Figura 1 – Organograma da metodologia

Sendo assim, na primeira etapa é realizada uma revisão extensiva da literatura relacionada ao tema do artigo, com o objetivo de compreender o contexto em que o trabalho se insere. A revisão bibliográfica serve para explorar as práticas ágeis, seus conceitos fundamentais, evolução ao longo do tempo (como se comportou durante a pandemia) e as contribuições já feitas por outros pesquisadores, como no caso do trabalho de (CAMARA et al., 2020), estudo ao qual o presente documento dá seguimento. Isso permite estabelecer uma base sólida para a continuidade do seu estudo. Dessa forma foi possível definir bem os conceitos de GSD, ASD e AGSD.

Na segunda etapa, ocorre a identificação das 48 práticas ágeis mencionadas na obra de referência (CAMARA et al., 2020), citadas também em (FIGUEREDO, 2020) e (ALVES, 2021). Tal procedimento requer uma análise minuciosa do trabalho original, a fim de catalogar e listar essas práticas.

Na terceira fase do processo, é desenvolvido um questionário ou survey, conforme abordado por (SCHEUREN, 2004), seguindo a metodologia adaptada de (ALVES, 2021) como base. Isso implica em realizar certas adaptações para melhor adequação ao escopo do estudo.

O requisito para participar da survey é: trabalhar com desenvolvimento de software em projetos distribuidos. O questionário é estruturado para adquirir informações relevantes sobre a adoção das práticas ágeis por parte dos profissionais em atividade. O questionário possui 3 seções com questões a fim de obter informações de:

- Abordagens de desenvolvimento de software:

  - Entender como a população da pesquisa trabalha, se trabalha em projedos ASD, GSD e/ou AGSD.

- Práticas de desenvolvimento ágeis em projetos de software distribuído

  - Para identificar quais são as práticas são mais utilizadas pela amostra.

- Nessa seção estão listadas as 48 práticas abordadas em (CAMARA et al., 2020).

- Dados demográficos:

  - Para poder identificar a população da presente pesquisa.

Após a coleta dos dados do survey, é conduzida uma análise estatística dos resultados. A utilização do desvio-padrão das respostas viabiliza a avaliação da dispersão dos dados em relação à média, permitindo a identificação das práticas ágeis que são consistentemente adotadas pelos profissionais e aquelas que podem apresentar variações significativas. Isso proporciona a priorização das práticas mais pertinentes para inclusão no portfólio de auxílio.

Na quinta etapa, critérios são definidos para a identificação de profissionais de desenvolvimento distribuído para realização de pesquisas semi-estruturadas. Esses profissionais devem ter pelo menos 4 anos de experiência na área. A partir da entrevista, será possível entender na prática relatos de como funcionam as práticas ágeis utilizadas pelos mesmos, quais seus pontos fortes e como elas combinam com outras práticas.

Já na sexta etapa é desenvolvido um roteiro de perguntas semi-estruturadas, com base em abordagens como a de (LUO; WILDEMUTH, 2009), para guiar as entrevistas com os profissionais selecionados. A pesquisa semi-estruturada é uma abordagem que combina flexibilidade e organização, permitindo uma exploração aberta das opiniões e experiências dos participantes. Essa metodologia é especialmente vantajosa em pesquisas qualitativas, como no estudo das práticas ágeis, pois permite capturar nuances e insights ricos. De acordo com (FLICK, 2015), as entrevistas semi-estruturadas oferecem a oportunidade de aprofundar perspectivas dos entrevistados enquanto mantêm um certo grau de consistência nas questões abordadas, proporcionando um equilíbrio entre padronização e flexibilidade.

Na sétima etapa, alicerçada nos desdobramentos da análise de desvio-padrão a partir das respostas do survey, concebe-se um portfólio abrangente que incorpora as práticas ágeis mais frequentemente adotadas, tal como identificado na etapa 4. Posterior à seleção destas práticas, o portfólio é elaborado, utilizando como base os relatos obtidos na etapa 6, a partir dos quais é possível identificar os pontos positivos e negativos das práticas ágeis. Essa análise propicia a explicação de como, por que e quando empregar tais práticas, visando aprimorar a experiência em contextos de ambientes ASD ou AGSD. Este documento se apresenta como uma ferramenta prática destinada a profissionais que almejam a implementação eficaz de práticas ágeis em seus projetos, conferindo-lhes um recurso substanciado e confiável.

Na etapa final, os resultados obtidos ao longo da pesquisa são reunidos e comparados com os estudos anteriores de (CAMARA et al., 2020) e (ALVES, 2021). Essa análise permite identificar as contribuições únicas do estudo em andamento e destacar áreas onde há concordância ou diferenças em relação às conclusões prévias. Com base nessa análise e na criação do portfólio mencionado anteriormente, a pesquisa chega à sua conclusão. Isso destaca as descobertas relevantes e suas implicações na prática atual e em futuras pesquisas. Essas descobertas têm um papel crucial no aprimoramento da aplicação das práticas ágeis em ambientes de desenvolvimento de software distribuídos.

4 Resultados do Survey

Nesta seção veremos os resultados estatísticos do survey realizado pelo presente trabalho. Conforme citado no capítulo 3, o survey utilizado foi baseado no mesmo survey utilizado no trabalho de (ALVES, 2021), onde foram incluídas duas questões abertas e uma fechada para coletar informações qualitativas para auxilio na confecção do portifólio demonstrado no seguinte capítulo. A ideia de utilizar o mesmo survey, como base, seria para poder reaproveitar as respostas antigas e somar com as novas respostas coletadas.

O novo survey foi distribuído de forma controlada onde só pessoas com o link poderiam participar. Através de grupos conhecidos em redes sociais ou emails, ou até de forma direta através do networking redes sociais e circulo social próximo. Essa estratégia tende a gerar um menor numero de respostas quando se compara com estratégias de distribuição aberta, no entanto garante um filtro de quem está respondendo implicando numa melhor qualidade da população da pesquisa, conforme afirmam (WAGNER et al., 2020). Sendo assim, foram coletadas um total de 35 novas respostas que somando às do antigo survey totalizam 60 respostas.

4.1 Dados Demográficos

Assim como na pesquisa anterior, a maioria dos participantes são do grupo que possui 3-5 anos de experiência totalizando 20 respostas. Da mesma forma que as pessoas entre 1-2 anos de experiência segue na segunda colocação com 19 respostas, no entanto dessa vez segue do grupo com 6-10 anos de experiência com 14 respostas. Um fato interessante, é que os demais grupos se mantiveram com a mesma quantidade da pesquisa anterior, conforme mostra a figura 2.

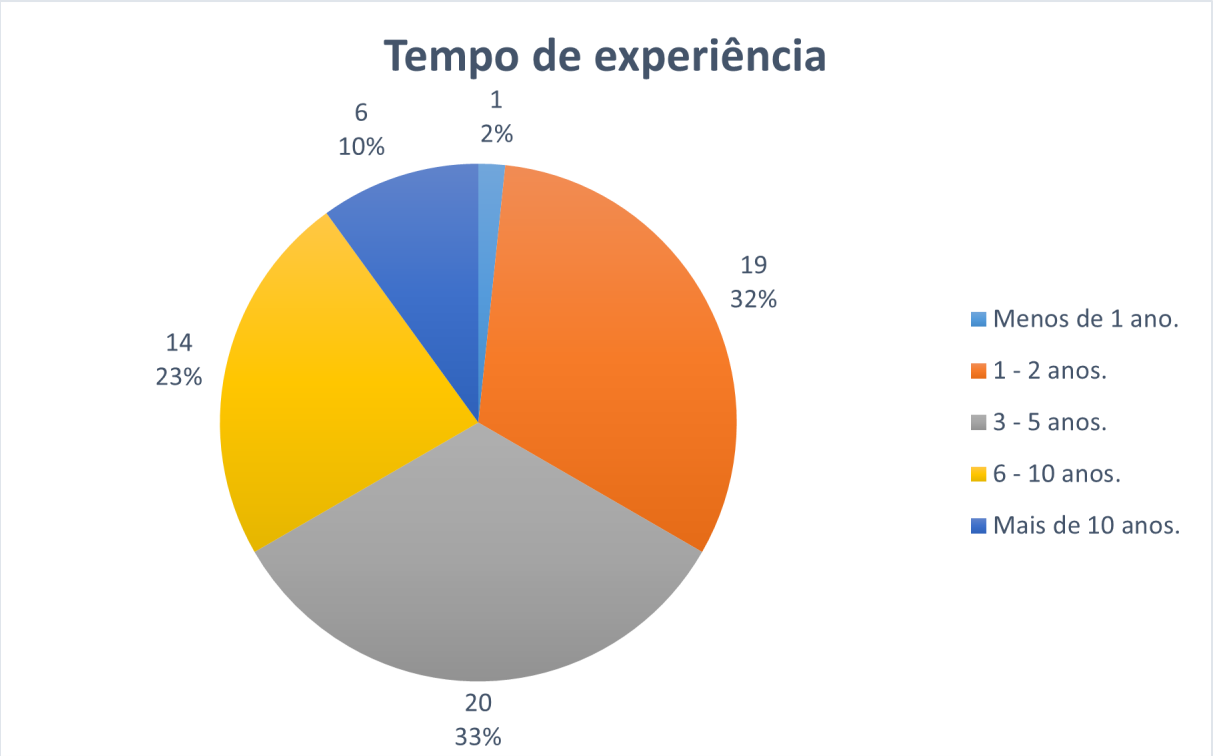

Figura 2 – Tempo de experiência com desenvolvimento de software

Como não era uma resposta obrigatória, nem todas as pessoas responderam à pergunta sobre qual empresa trabalham. No entanto, foi possível contabilizar um total de 22 empresas distintas entre as 28 pessoas que responderam. Isso indica uma variedade de experiências profissionais entre os participantes.

Além disso, observamos que houve um total de 32 pessoas que optaram por não responder a essa pergunta específica. No entanto, ao analisarmos as 60 respostas que foram fornecidas, pudemos obter insights interessantes sobre o panorama corporativo. Quanto ao tamanho das empresas em que os 60 respondentes trabalham, podemos destacar uma diversidade notável.

Em comparação com o trabalho antecessor, algo que mudou um pouco foi a distância entre as grandes das menores. As empresas abrangem desde startups e empresas de médio porte até organizações multinacionais estabelecidas, sendo em sua maioria empresas de porte muito grande (mais de 2500 funcionários) com 26 respostas seguido de empresas de grande porte (entre 251 até 2499 funcionários) com 17 respostas. (Observe figura 3)

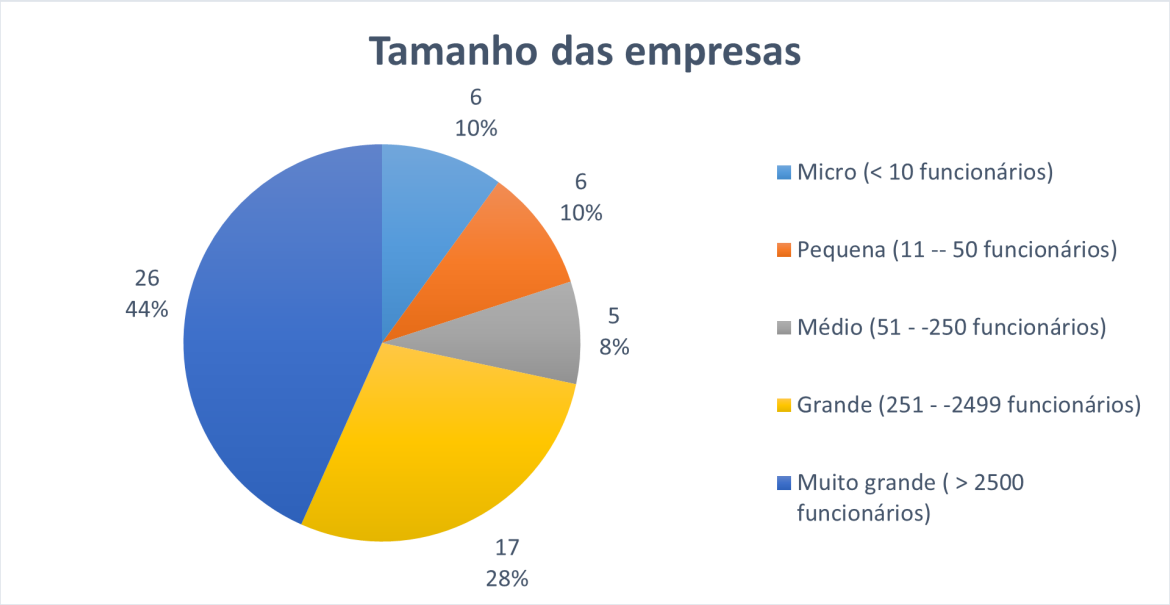

Figura 3 – Tamanho das empresas de software

Sobre o espectro de atuação dessas empresas, as observações revelam um cenário intrigante (consulte a figura 4). A área de desenvolvimento de software permanece proeminente, abarcando agora 50% das respostas, em contraste com a porcentagem anterior de 40%. Um dado de destaque é o crescimento da presença da área de consultoria, que ultrapassou a categoria de desenvolvimento de sistemas. Além disso, é interessante observar um aumento discreto na representatividade da área financeira, juntamente com o surgimento de uma nova categoria de varejistas no setor de beleza.

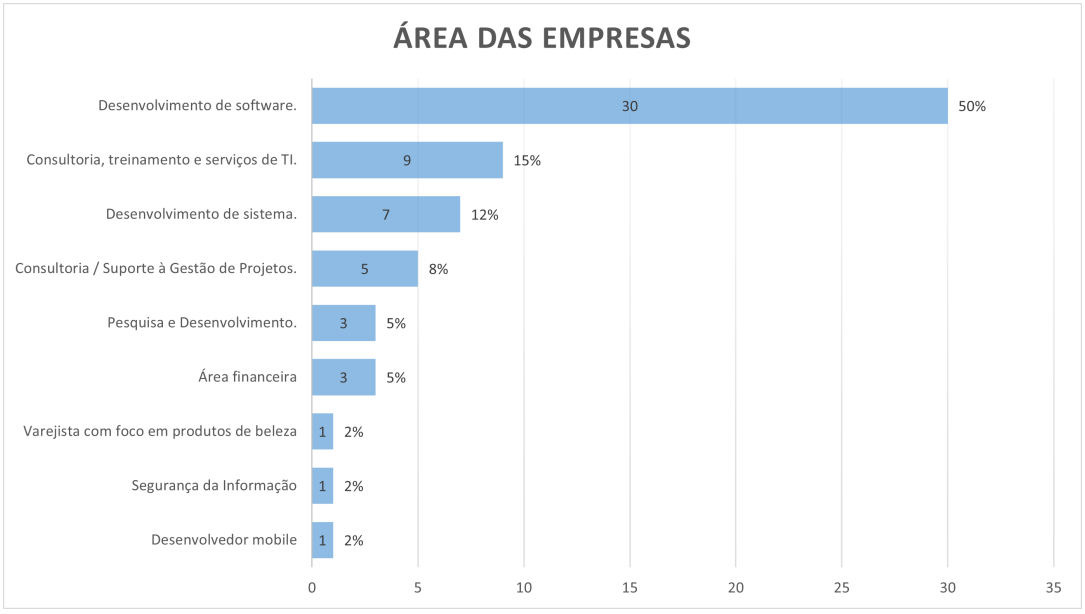

Figura 4 – Área de atuação das empresas de software

Encerrando a análise dos dados demográficos, é crucial destacar a evolução dos papéis desempenhados pelos entrevistados em suas respectivas empresas. De forma avassaladora, a categoria dominante continua sendo a dos desenvolvedores, porém agora com uma vantagem percentual muito mais substancial - de 52% anteriormente para expressivos 72%. Os demais dados mantêm-se consistentes com as proporções prévias, havendo apenas algumas diferenças pontuais. Um incremento notável é visível no número de Tech Leads, quando comparado ao levantamento anterior, embora ainda não representem uma fatia significativa do panorama geral. Ademais, novos cargos emergiram, embora em quantidades limitadas. (Consulte a figura 5 para visualizar essas distribuições).

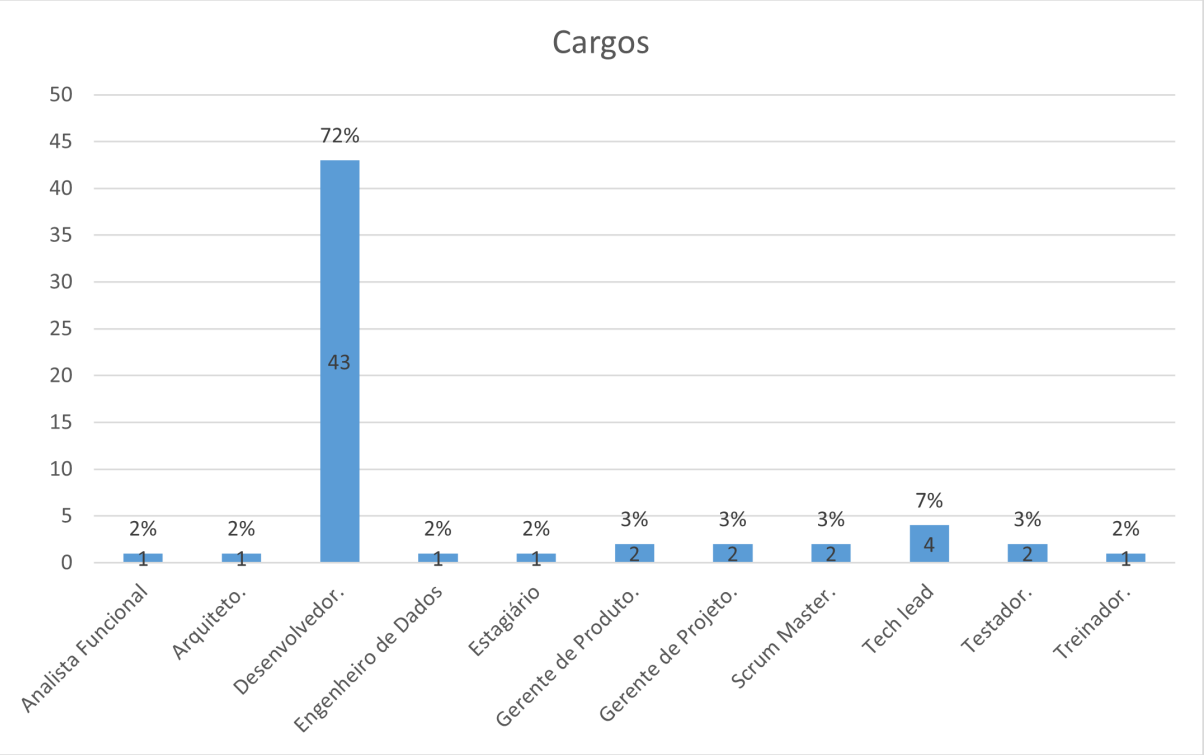

Figura 5 – Cargos

4.2 Frameworks e Abordagens de Desenvolvimento de Software

Para compreendermos melhor como os participantes abordam o tema, fizemos perguntas sobre como eles aplicam essas abordagens em suas rotinas diárias. Isso nos deu uma visão clara de como eles as utilizam, ilustrada no gráfico da Figura 6.

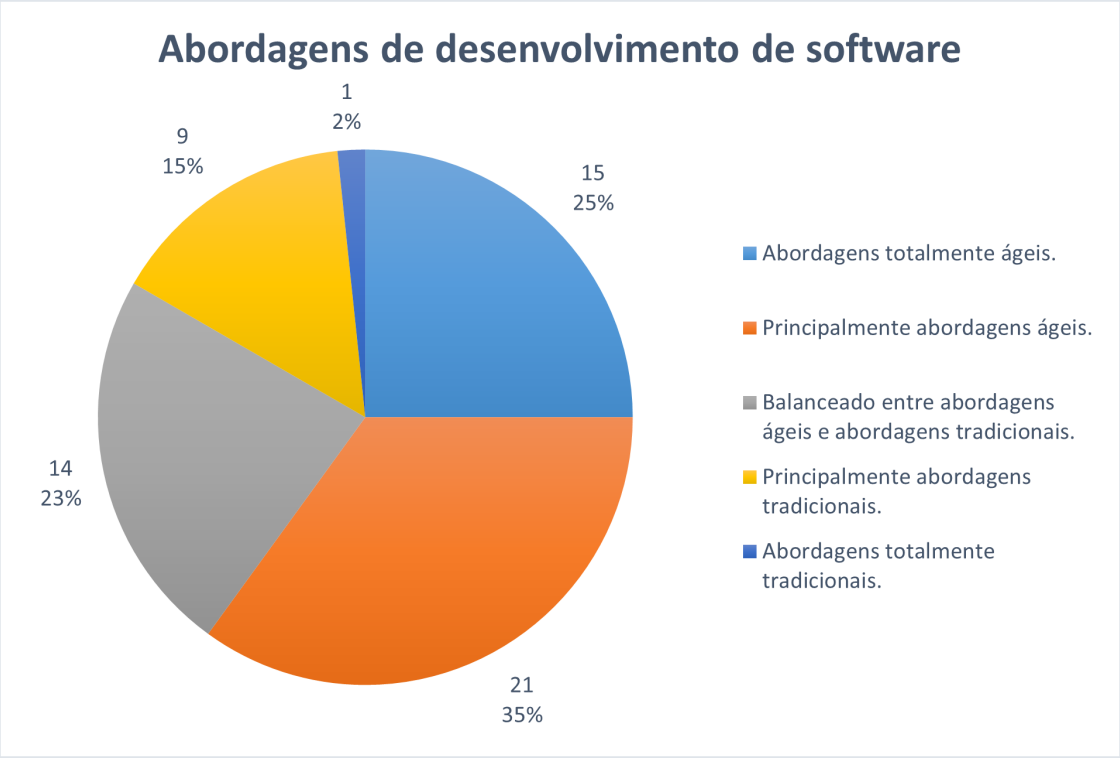

Figura 6 – Abordagens de desenvolvimento de software

Com um aumento significativo no número de respostas, é evidente que as práticas ágeis ganharam considerável popularidade entre os participantes. Cerca de 35% (21 respostas) indicaram a opção ”Principalmente abordagens ágeis”, enquanto 25% (15 respostas) escolheram ”Abordagens totalmente ágeis”. Isso reforça a clara inclinação dos participantes em direção a estratégias ágeis, o que pode ser atribuído à sua adaptabilidade e eficácia comprovada.

Também foi perguntado sobre que frameworks os participantes utilizavam em seu dia a dia. Conforme mostra a Figura 7 Scrum se destaca com 45 respostas, representando 75% do total máximo de respostas, solidificando sua popularidade. O Kanban também apresenta uma presença expressiva, com 35 respostas (58%), indicando uma aceitação significativa. O DevOps ganha importância com 19 respostas (32%), refletindo sua crescente relevância nas práticas de desenvolvimento.

É relevante ressaltar que 2 participantes responderam com ”Não sei responder”e uma pessoa mencionou ”Não utilizo”. Essas respostas podem apontar para a ausência de definição clara de frameworks no projeto ou indicar a limitação do conhecimento por parte dos participantes sobre as abordagens utilizadas.

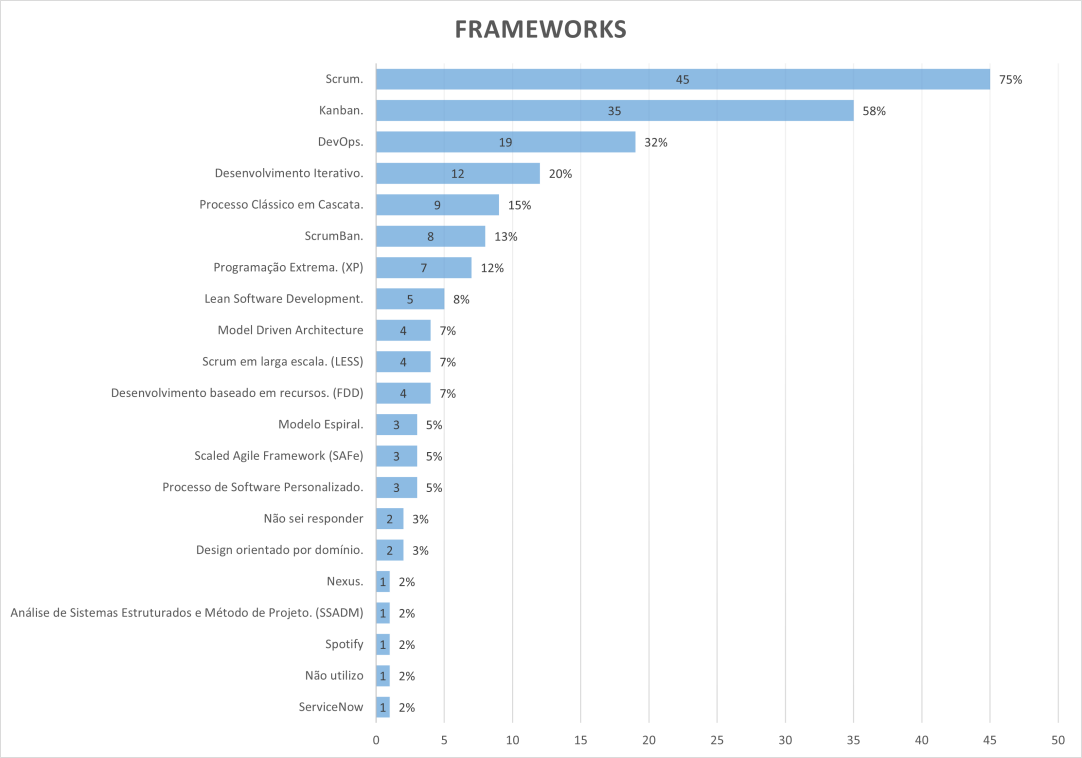

Figura 7 – Frameworks de desenvolvimento de software mais utilizados

4.3 Uso das práticas AGSD

Nesta seção, serão exibidos os resultados obtidos a partir das perguntas referentes à utilização das 48 práticas ágeis. Para essa análise, optou-se pela aplicação da Escala Likert, um método simples porém eficaz. Conforme detalhado por (BATTERTON; HALE, 2017), essa escala envolve cinco alternativas de resposta, abrangendo desde ”Discordo Totalmente” até ”Concordo totalmente”. No âmbito deste estudo, essa escala foi adaptada para incluir os termos ”Não Uso”, ”Uso Raramente”, ”Às Vezes Uso”, ”Uso Frequentemente”e ”Sempre Uso”, de modo a melhor refletir as nuances das respostas fornecidas pelos participantes.

Assim, ao analisar a Figura 8, nota-se que as práticas ”Sprint”, ”Reunião Diária”, ”Backlog do Produto”, ”Planejamento” e ”Práticas de Comunicação” receberam, respectivamente, a maior quantidade de respostas na categoria ”Sempre Uso” (todas com mais de 30 respostas). Esse padrão reflete uma significativa aceitação dessas práticas por parte dos respondentes. Em contraste, as práticas ”Visitas entre Locais de Trabalho”, ”Alternar Equipe entre Locais” e ”Coaching” obtiveram, respectivamente, a maior quantidade de respostas na categoria ”Não Uso” (todas também com mais de 30 respostas). Isso aponta para uma clara tendência de não utilização dessas práticas.

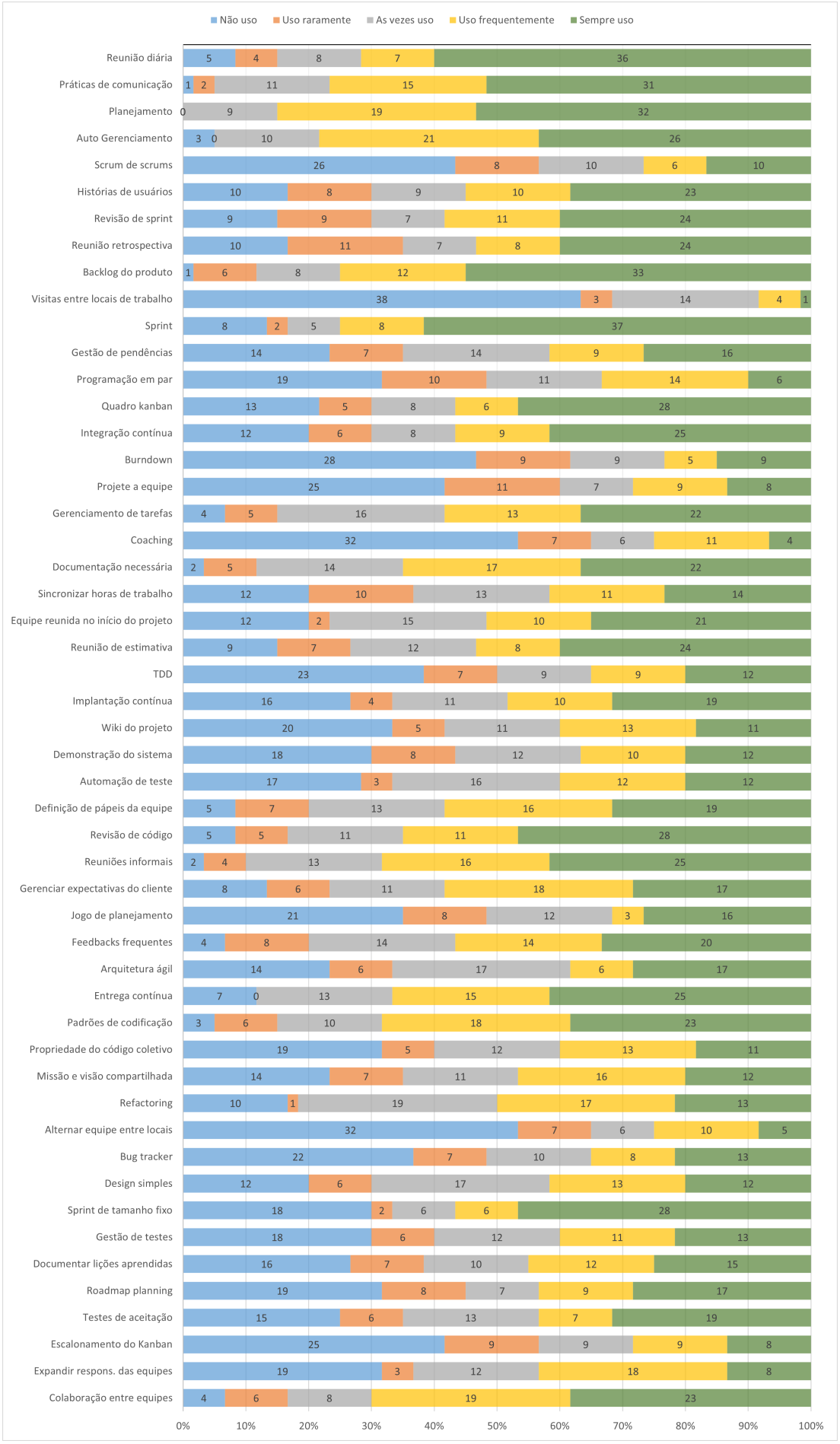

Figura 8 – Frequências das práticas ágeis

5 Construção e Resultados das Entrevistas

A realização das entrevistas foi motivada pela busca por uma compreensão mais profunda do funcionamento das práticas ágeis com maior aceitação, conforme identificadas no levantamento (Survey). O objetivo central era adquirir insights imersivos sobre como essas práticas interagem em um contexto real.

5.1 Definição das práticas AGSD mais aceitas

Para alcançar esse objetivo, o primeiro passo consistiu em definir quais práticas seriam categorizadas como ”Aplicáveis”. Essa definição direcionou a coleta de informações durante as entrevistas, as quais desempenhariam um papel fundamental na construção de um portfólio robusto e na promoção de discussões significativas à luz dos resultados obtidos no levantamento.

Para realizar essas definições, utilizaram-se as variáveis estatísticas média e desvio padrão. A média foi calculada a partir dos valores de cada resposta da escala Likert, atribuindo-se o valor 1 para ”Não Uso”, 2 para ”Uso Raramente” e assim sucessivamente, até o valor 5 para ”Sempre Uso”. A partir desse cálculo, os valores foram somados e divididos pelo número total de respostas para cada prática.

Essa abordagem permitiu identificar quais práticas apresentavam as maiores médias. No entanto, mesmo quando as médias eram consideradas ”aceitáveis”, a distribuição das respostas poderia indicar falta de concordância homogênea. Para aprimorar os resultados, o desvio padrão foi empregado para compreender como as respostas se dispersavam em relação à média, a fim de obter insights mais refinados sobre a aceitação das práticas (LIVINGSTON, 2004).

Para clarificar, imagine que possuímos um desvio padrão de 1,2 e média 4. Isso implica que os valores estão dentro da faixa de 4 - 1,2 (2,8) a 4 + 1,2 (5,2). Com base nisso, podemos inferir que um desvio padrão menor indica que as respostas estão mais concentradas em torno de uma opinião, enquanto um desvio padrão maior sugere que as respostas estão mais dispersas e heterogêneas (LIVINGSTON, 2004). Isso realça a importância do desvio padrão como indicador da variabilidade nas respostas coletadas.

Sabendo disso, foi definida uma fórmula específica para categorizar as práticas como ”Aplicáveis”. Essa fórmula é calculada como Média - Desvio Padrão (aqui referido como ”Valor Mínimo”), representando o menor valor que está próximo à média. A aplicação dessa fórmula visa selecionar as práticas com maior aceitação e, ao mesmo tempo, com respostas mais uniformes. Para que uma prática seja considerada ”Aplicá-

vel", é necessário que o "Valor Mínimo" seja igual ou superior a 2,5, além de um desvio padrão inferior a 1,5.

Essa abordagem tem como objetivo identificar não apenas práticas bem aceitas, mas também aquelas nas quais os participantes expressam opiniões mais convergentes. Os resultados obtidos através dessa fórmula aplicada nas práticas estão documentados na Tabela 1.

Frequências	Não uso	Uso raramente	As vezes uso	Uso frequentemente	Sempre uso	Média	Desvio Padrão	Valor Mínimo	Classificação
Reunião diária	5	4	8	7	36	4,08	1,33	2,75	Aplicável
Práticas de comunicação	1	2	11	15	31	4,22	0,98	3,24	Aplicável
Planejamento	0	0	9	19	32	4,38	0,74	3,64	Aplicável
Auto Gerenciamento	3	0	10	21	26	4,12	1,03	3,09	Aplicável
Scrum de scrums	26	8	10	6	10	2,43	1,53	0,90	
Histórias de usuários	10	8	9	10	23	3,47	1,52	1,94	
Revisão de sprint	9	9	7	11	24	3,53	1,51	2,02	
Reunião retrospectiva	10	11	7	8	24	3,42	1,57	1,85	
Backlog do produto	1	6	8	12	33	4,17	1,11	3,06	Aplicável
Visitas entre locais de trabalho	38	3	14	4	1	1,78	1,12	0,66	
Sprint	8	2	5	8	37	4,07	1,44	2,63	Aplicável
Gestão de pendências	14	7	14	9	16	3,10	1,51	1,59	
Programação em par	19	10	11	14	6	2,63	1,40	1,23	
Quadro kanban	13	5	8	6	28	3,52	1,64	1,88	
Integração contínua	12	6	8	9	25	3,48	1,59	1,89	
Burndown	28	9	9	5	9	2,30	1,50	0,80	
Projete a equipe	25	11	7	9	8	2,40	1,49	0,91	
Gerenciamento de tarefas	4	5	16	13	22	3,73	1,23	2,50	Aplicável
Coaching	32	7	6	11	4	2,13	1,41	0,73	
Documentação necessária	2	5	14	17	22	3,87	1,11	2,75	Aplicável
Sincronizar horas de trabalho	12	10	13	11	14	3,08	1,45	1,63	
Equipe reunida no início do projeto	12	2	15	10	21	3,43	1,50	1,93	
Reunião de estimativa	9	7	12	8	24	3,52	1,49	2,03	
TDD	23	7	9	9	12	2,67	1,59	1,08	
Implantação contínua	16	4	11	10	19	3,20	1,60	1,60	
Wiki do projeto	20	5	11	13	11	2,83	1,54	1,29	
Demonstração do sistema	18	8	12	10	12	2,83	1,52	1,31	
Automação de teste	17	3	16	12	12	2,98	1,49	1,49	
Definição de pápeis da equipe	5	7	13	16	19	3,62	1,28	2,34	
Revisão de código	5	5	11	11	28	3,87	1,32	2,55	Aplicável
Reuniões informais	2	4	13	16	25	3,97	1,10	2,86	Aplicável
Gerenciar expectativas do cliente	8	6	11	18	17	3,50	1,36	2,14	
Jogo de planejamento	21	8	12	3	16	2,75	1,62	1,13	
Feedbacks frequentes	4	8	14	14	20	3,63	1,26	2,37	
Arquitetura ágil	14	6	17	6	17	3,10	1,51	1,59	
Entrega contínua	7	0	13	15	25	3,85	1,30	2,55	Aplicável
Padrões de codificação	3	6	10	18	23	3,87	1,19	2,68	Aplicável
Propriedade do código coletivo	19	5	12	13	11	2,87	1,52	1,34	
Missão e visão compartilhada	14	7	11	16	12	3,08	1,46	1,62	
Refactoring	10	1	19	17	13	3,37	1,31	2,05	
Alternar equipe entre locais	32	7	6	10	5	2,15	1,44	0,71	
Bug tracker	22	7	10	8	13	2,72	1,60	1,12	
Design simples	12	6	17	13	12	3,12	1,39	1,73	
Sprint de tamanho fixo	18	2	6	6	28	3,40	1,76	1,64	
Gestão de testes	18	6	12	11	13	2,92	1,54	1,37	
Documentar lições aprendidas	16	7	10	12	15	3,05	1,56	1,49	
Roadmap planning	19	8	7	9	17	2,95	1,65	1,30	
Testes de aceitação	15	6	13	7	19	3,15	1,58	1,57	
Escalonamento do Kanban	25	9	9	9	8	2,43	1,49	0,94	
Expandir respons. das equipes	19	3	12	18	8	2,88	1,47	1,41	
Colaboração entre equipes	4	6	8	19	23	3,85	1,23	2,62	Aplicável

Tabela 1 – Resumo das Avaliações

Com base nos dados apresentados, as práticas que foram amplamente aceitas e entraram no critério da pesquisa foram, respectivamente: "Planejamento", "Práticas de comunicação", "Auto Gerenciamento", "Backlog do product", "Reuniões informais", "Documentação necessária", "Reunião diária", "Padrões de codificação", "Sprint", "Colaboração entre equipes", "Entrega contínua", "Revisão de código" e "Gerenciamento de tarefas". Essas serão as práticas que as entrevistas irão abordar e presentes no portifólio final do trabalho.

Em marcante contraste, práticas como "Escalonamento do Kanban", "Projete a equipe", "Scrum de scrums", "Burndown", "Coaching", "Alternar equipe entre locais" e "Visitas entre locais de trabalho" receberam menor nível de aceitação por parte dos participantes.

5.2 Realização das entrevistas

Ao efetuar a síntese das práticas discutidas na seção 5.1, essas mesmas práticas foram empregadas como diretrizes para conduzir a pesquisa semi-estruturada, visando a obtenção de uma compreensão aprofundada sobre como os engenheiros de software lidam com as práticas destacadas. Dessa forma, cada uma dessas práticas é abordada como um tópico individual durante as entrevistas, permitindo a exploração de diversas áreas e nuances envolvidas no contexto. Tal abordagem é intrínseca à natureza da entrevista semi-estruturada, como elucidado por (MANZINI, 2004).

Com o roteiro de entrevista solidificado, foi possível conduzir entrevistas com um total de 5 participantes que estão em empresas diferentes, atuando de forma distribuida e no modelo home office. Eles serão identificados neste documento como ”Entrevistado A” até ”Entrevistado E”. As entrevistas transcorreram de maneira síncrona, fazendo uso da plataforma Google Meet para a realização de videochamadas. Adicionalmente, foi empregado o software OBS Studio para criar gravações de backup das entrevistas. A única exceção a esse formato foi a situação do ”Entrevistado E”, que, devido a dificuldades de sincronização de agendas, participou por meio de trocas assíncronas de áudio via WhatsApp.

5.2.1 Entrevistado A

O entrevistado A é um Desenvolvedor com 4 anos e 6 meses de experiência que trabalha em uma grande empresa de consultoria de software diversos, num time que utiliza o Scrum como metodologia. Ele linka diretamente o **”Planejamento”** com o a prática **”Sprint”**, visto que é na ”Sprint Planning” que o planejamento ocorre. Ocorrendo a cada 15 dias (tempo de duração da sprint), o planejamento é dividido em algumas partes sendo elas: a definição de quais user storys serão relevantes para a sprint, seguindo da definição de quais times atuarão em quais tarefas, como por exemplo backend, frontend e até cenários de teste visto que em seu projeto utilizam a prática ”Test-Driven Development” (TDD).

No âmbito do seu projeto, os encontros de planejamento são fundamentados no **”Backlog do produto”**, cuja elaboração decorre de deliberações entre os gestores e o cliente. Por conseguinte, o entrevistado possui uma participação limitada no processo de definição desse backlog. Seguindo a estrutura característica do Scrum, faz-se presente a figura do ”Product Owner”, representante do cliente, que desempenha um papel ativo nas reuniões de planejamento e na definição das tarefas a serem abordadas.

Uma vez que as tarefas são estabelecidas, os times congregam-se para atribuir valores aos story points, empregados na avaliação da complexidade de cada emprei-

tada. O entrevistado explana sobre a aplicação da técnica ”Planning Poker”, realizada na plataforma Azure, na qual os membros da equipe submetem suas avaliações referentes a cada tarefa. Caso surjam discrepâncias substanciais nas avaliações, ocorre uma discussão mais detalhada a fim de dirimir as divergências. Não obstante, o entrevistado observa que, em várias ocasiões, essa etapa consome considerável tempo dos integrantes da equipe, questionando sua eficácia. Tudo isso fica documentado e visível para a equipe na plataforma da Azure que serve como ferramenta de **”Gerenciamento de tarefas”**.

Ao discutir sobre as **”Práticas de comunicação”**, o entrevistado destaca sua presença significativa no cotidiano, tanto em formatos assíncronos quanto síncronos, por meio da plataforma Microsoft Teams. Com a realização da **”Reunião diária”**, os membros da equipe compartilham, de maneira sincrônica, o andamento de suas atividades no dia anterior e suas metas para o dia atual. A regulação dessa reunião é supervisionada por um ”Scrum Master”, cujo foco reside na manutenção da brevidade, geralmente entre 15 a 20 minutos, embora possa ser excepcionalmente estendida em situações pontuais.

Além das interações diárias, a comunicação ocorre de maneira assíncrona por meio de chats. Entretanto, em algumas circunstâncias, além das reuniões programadas, emergem as **”Reuniões Informais”**. Para o entrevistado, esses encontros informais desempenham um papel crucial na resolução de eventuais obstáculos que possam ter sido sinalizados durante a reunião diária ou em outros momentos. Muitos desses obstáculos surgem devido a dependências entre as tarefas de diferentes equipes. Tais problemas podem ser comunicados tanto na reunião diária quanto de forma mais casual.

Nesse contexto, a **”Colaboração entre equipes”** se revela de suma importância. Ela facilita a comunicação interdepartamental, proporcionando um entendimento compartilhado das prioridades e estimulando ação conjunta em prol da resolução dos problemas levantados. Essa sinergia entre equipes é altamente eficaz e se torna uma peça indispensável para o entrevistado.

Indagado acerca do **”Auto Gerenciamento”**, o entrevistado compartilha que, em virtude do quadro metodológico do Scrum, o auto gerenciamento encontra certos limites. Isso se justifica pelo fato de que as demandas predefinidas para a sprint devem ser estritamente observadas. Contudo, o entrevistado ressalta que cada colaborador detém a responsabilidade de administrar suas prioridades, desde que estejam alinhadas com os acordos estabelecidos nas reuniões de planejamento. O entrevistado ainda aponta que o auto gerenciamento se torna uma tarefa mais desafiadora ao lidar com desenvolvedores em estágios mais júnior. Estes frequentemente relutam em comunicar entraves ou gargalos à equipe.

Para concluir, o entrevistado traça conexões ao abordar a **"Documentação Necessária"**. Ele atribui uma importância extrema a essa prática, uma vez que ela desempenha um papel fundamental em várias fases de seu projeto. Além de ser uma fonte de informações para consultas pontuais, evitando a necessidade de reuniões desnecessárias, a documentação no contexto Scrum também serve como um registro que garante a conformidade com o que foi acordado entre a equipe de desenvolvimento e o cliente. Isso previne lacunas ou desentendimentos que poderiam resultar em complicações ao longo da sprint.

O entrevistado também aborda os **"Padrões de Codificação"** aplicados em seu projeto, destacando a adoção da "Arquitetura Clean"e da abordagem de divisão de responsabilidades "CQRS". Após a conclusão do desenvolvimento de uma demanda, esses padrões, juntamente com a validação dos cenários implementados, passam por um processo de **"Revisão de Código"**. Utilizando a própria plataforma Azure, o entrevistado explica que esse processo envolve uma série de etapas, desde a submissão do "pull request" até a aprovação final por outros membros da equipe de desenvolvimento. Essas etapas combinam tanto ferramentas automáticas quanto avaliações manuais. Caso alguma etapa detecte um problema, é crucial realizar os ajustes prioritariamente para evitar interrupções no fluxo de trabalho da sprint.

Assim, com a documentação adequada, a aplicação de padrões de codificação bem definidos e revisões minuciosas, as tarefas se preparam para a transição ao ambiente produtivo. Esse processo é realizado ao final de cada sprint, impulsionando as **"Entregas Contínuas"** e garantindo uma evolução constante do projeto.

5.2.2 Entrevistada B

A Entrevistada B é uma desenvolvedora com 5 anos e meio de experiência que trabalha em uma empresa de consultoria multinacional. Atualmente vive em um contexto com várias práticas ágeis em seu projeto atual, porem já passou por outras experiências com metodologias mais tradicionais, sendo assim chegou a fazer comparações entre as metodologias e mostrar a importância das práticas ágeis.

Ao abordar a prática do **"Planejamento"**, a entrevistada compartilha uma diferença marcante entre sua experiência anterior e a situação atual em relação a essa atividade. Em sua empresa anterior, ela trabalhava em uma arquitetura de natureza tradicional, onde o planejamento de demandas era centralizado nas mãos do gerente de projeto, sem envolver figuras técnicas. Nesse cenário, as demandas chegavam a ela já com prazos predefinidos, limitando seu envolvimento no processo. Por outro lado, em sua experiência atual e também em outras vivências profissionais, o planejamento é conduzido em reuniões que reúnem um Product Owner e um representante técnico em nome do cliente, juntamente com os desenvolvedores e, por vezes, um gerente

representando a empresa provedora de serviços.

Nesse cenário, as discussões abrangem uma série de aspectos como prioridades, prazos e complexidades. Durante essas sessões, são abordadas as necessidades do PO, que são subdivididas em pequenas demandas para compor o **"Backlog do Produto"**. Esse backlog consiste em tarefas originadas tanto de novas necessidades quanto de correções ou melhorias em artefatos já construídos durante o projeto. Vale ressaltar que, durante essas interações, o gerente mantém um controle minucioso do **"Gerenciamento de Tarefas"** por meio de planilhas Excel. A entrevistada observa também que seu conhecimento sobre outras ferramentas de gerenciamento é limitado, uma vez que, em sua realidade, a atualização de status nessas ferramentas e o controle da visão macro são atividades centralizadas nos gerentes. Ela também menciona que, para o rastreamento de bugs, alguns membros de sua equipe utilizam o HPQC, embora essa não seja uma prática totalmente transparente para ela.

No contexto do **"Auto Gerenciamento"**, a entrevistada explora comparações entre suas experiências passadas e as atuais, destacando a centralização do controle das demandas na gestão de projetos de outrora. Nesse ambiente, a capacidade de auto gerenciamento era praticamente inexistente, ao ponto de, em caso de prazos não estabelecidos, a solução recorrente ser a realização de horas extras para tentar mitigar possíveis atrasos. Porém, em contraste, ao adotar um formato fundamentado nas práticas ágeis de planejamento, a entrevistada descreve sua função como responsável por um setor específico do produto. Isso implica que as demandas relacionadas a esse segmento são centralizadas em sua atuação, incluindo priorizações, negociações de prazo e outras decisões, sem depender diretamente da liderança.

Ela enfatiza que essa nova abordagem trouxe significativamente mais qualidade de vida, pois os prazos e priorizações passaram a ser mais sensatos desde o estágio inicial de planejamento. Além disso, a entrevistada ressalta que o Auto Gerenciamento é eficaz e deve ser incentivado até mesmo para desenvolvedores menos experientes. Isso ocorre não somente para manter os prazos alinhados com a realidade individual, mas também para estimular habilidades de gerenciamento e promover o amadurecimento dos membros da equipe.

Quando questionada sobre a aplicação da **"Sprint"**, a entrevistada menciona que essa prática não é adotada no projeto atual, uma vez que o projeto se baseia nos prazos de testes realizados pelo próprio cliente. No entanto, ela também compartilha sua experiência em situações passadas em que as Sprints foram utilizadas. Nesse contexto, ela destaca que as Sprints eram um tanto atípicas, devido às mudanças frequentes de escopo que ocorriam durante sua execução. Essas mudanças eram tão frequentes que o projeto acabou por optar por não continuar com a abordagem das Sprints. A entrevistada ressalta que essa decisão foi influenciada pelo nível de

maturidade do cliente, o que teve um grande impacto na dinâmica do projeto.

Quando abordamos as **”Práticas de Comunicação”**, nesse contexto realizada predominantemente pelo Microsoft Teams, a entrevistada destaca sua importância em diversas etapas do processo de engenharia de software. No âmbito do planejamento, por exemplo, onde a comunicação ocorre de maneira sincronizada, sua relevância é incontestável, visando evitar mal-entendidos e alinhar os pontos-chave. No entanto, essa dimensão da comunicação se estende para além desse momento. Como mencionado previamente pelo Entrevistado A (5.2.1), essas práticas também se manifestam em **”Reuniões Informais”** que emergem esporadicamente quando surgem dúvidas acerca das tarefas. Geralmente, essas reuniões se concretizam entre o cliente e o desenvolvedor ou mesmo em situações de **”Colaboração entre Equipes”**, situações que não eram tão comuns, visto que os times costumavam ser autocentrados e independentes. Porém existiam cenários onde um time precisou do outro e a colaboração foi bem sucedida. Outro relato comentado, foi a existencia de ”Intercâmbio entre equipes” onde um integrante de uma equipe era trocado temporariamente por outra equipe.

No contexto atual do relato, a entrevistada enfatiza que essas reuniões são incontornáveis e de suma importância. No entanto, ela destaca a necessidade de manter a objetividade a fim de evitar que elas consumam um tempo excessivo das partes envolvidas. No que diz respeito à comunicação, também se inserem as **”Reuniões Diárias”**. Diferentemente das que são organizadas por um ”Scrum Master”, como mencionado anteriormente, essas reuniões não possuem um controle tão preciso, o que, por vezes, resulta em uma duração excessiva. Ainda que o princípio de uma reunião diária seja a brevidade, neste relato, elas ocasionalmente se estendem por uma hora e meia a duas horas. A entrevistada acredita que essa extensão decorre de uma orquestração inadequada da reunião e também do grande número de participantes de diversos times. A entrevistada compartilha experiências anteriores em que não encontrou execução eficiente das reuniões diárias. Em alguns casos, chegou até a participar de duas reuniões no mesmo dia, algo que ela julga desnecessário, uma vez que os status não sofriam alterações significativas entre esses momentos. Essa abordagem resultava somente em um desperdício de tempo excessivo e uma falta de eficácia nas reuniões.

Quando questionada sobre a **”Documentação Necessária”**, a entrevistada compartilha sua perspectiva. Ela menciona que em experiências anteriores, a excessiva documentação frequentemente consumia muito tempo no processo de desenvolvimento, tornando até mesmo as atualizações das próprias documentações um desafio. No entanto, em sua experiência atual, há um indivíduo designado para assegurar que a documentação da equipe esteja em ordem. Isso, segundo ela, simplificou significativamente a tarefa de manter as documentações atualizadas.

No entanto, quando questionada sobre se essa documentação ajuda a reduzir

comunicações desnecessárias, a entrevistada observa que, mesmo com a documentação em dia, muitas vezes a comunicação ainda é necessária. Isso ocorre porque as documentações nem sempre são tão precisas quanto se espera, e podem não abranger todos os detalhes ou nuances das tarefas. Portanto, embora as documentações sejam úteis, elas não eliminam completamente a necessidade de comunicação direta entre os membros da equipe.

Quando abordados os tópicos de **"Padrões de Codificação"** e **"Revisão de Código"**, a entrevistada compartilha algumas nuances específicas de seu projeto. Ela observa que manter um único padrão de código não é uma tarefa simples em seu contexto, uma vez que o desenvolvimento ocorre em um sistema fechado e lida frequentemente com programas existentes que requerem manutenção ou aprimoramento. Muitas dessas partes do sistema podem não estar alinhadas com o padrão de codificação atual devido às variações históricas. Portanto, a entrevistada opta por aderir ao padrão de código já existente em cada programa, mesmo que isso signifique seguir padrões diferentes em partes diferentes do projeto. Essa escolha visa melhorar a legibilidade do código, evitando a confusão que poderia surgir ao misturar diferentes padrões.

Devido a essa abordagem, a revisão de código não é tão rigorosa para garantir um único padrão, uma vez que diferentes partes do sistema podem aderir a padrões distintos que evoluíram ao longo do tempo. A revisão de código é em grande parte automatizada, focando em identificar pontos críticos que possam resultar em problemas de execução. Quando questionada sobre a eficácia desse método para evitar bugs, a entrevistada menciona que ainda não teve experiência de suporte pós lançamento em seu projeto atual.

Ela conclui relatando que após a conclusão da revisão, o incremento do código é entregue ao cliente, sem a necessidade de esperar outro incremento estar pronto, caracterizando assim a prática de **"Entrega Contínua"**. Isso sugere que, apesar das complexidades dos padrões de codificação variados, a equipe busca manter um fluxo contínuo de entrega de valor ao cliente.

5.2.3 Entrevistado C

O Entrevistado C tem 5 anos de experiência com desenvolvimento de software e atualmente vive em uma equipe de desenvolvimento de softwares customizados de forma globalmente distribuída, atuando como desenvolvedor. Importante colocar que seu time possui fusos horários diferentes, pois se tratam de pessoas de diferentes países.

Ao iniciarmos tradicionalmente abordando o tópico de **"Planejamento"**, o entrevistado compartilha que segue a metodologia Scrum, na qual o planejamento se desen-

volve durante as reuniões de **"Sprint"**. Nessas sessões, a equipe se dedica a definir as prioridades e a relevância de cada tarefa para o próximo ciclo, que tem duração de 1 semana. Vale ressaltar que a complexidade das tarefas é definida em conjunto com toda a equipe, pois ao criada a tarefa recebe uma pontuação que é discutida dentre os desenvolvedores até chegar num concenso.

Esse processo se baseia no **"Product Backlog"**, que é meticulosamente elaborado pelo Product Manager (PM). O PM, em conjunto com o cliente, estabelece as necessidades e tarefas que o projeto deve abordar. Esse processo constitui o alicerce para a organização das atividades. O **"Gerenciamento de Tarefas"** nesse contexto é realizado pelo Jira e todos os desenvolvedores tem acesso a ferramenta, conseguindo ter a visão macro do projeto.

Dentro de seu projeto, opera-se sob o conceito de "Quarter", um período de três meses no qual a meta é concluir uma determinada quantidade de tarefas. Essas tarefas são inicialmente delineadas pelo PM e, posteriormente, têm suas prioridades debatidas durante as reuniões de planejamento. Contudo, dentro desse cenário, nota-se que o **"Auto Gerenciamento"** adota uma postura ligeiramente mais restrita, limitando-se ao que está definido na sprint. Mesmo nesse contexto, pode haver situações de interdependência entre tarefas, o que por vezes influencia nas prioridades que um desenvolvedor tem que adotar.

Explorando esse contexto, surgiram questionamentos sobre as **"Práticas de comunicação"** empregadas. No cenário do entrevistado em questão, essas práticas desempenham um papel crucial devido à natureza específica do projeto. Para facilitar a comunicação, são adotadas tanto ferramentas síncronas, como o Zoom, quanto assíncronas, como o Slack. No âmbito das práticas, durante o decorrer da sprint, são realizadas as **"Reuniões Diárias"**, agendadas de maneira a acomodar participantes de diferentes fusos horários. Conduzida em inglês, essas reuniões contam com o PM assumindo o papel de Scrum Master. Para manter a fluidez e evitar que se estendam demasiadamente, são estabelecidas algumas diretrizes, como um limite de 3 minutos para cada pessoa falar e a resolução de discussões mais complexas ao final da reunião. As reuniões duram em média 20 minutos e caso o PM não consiga participar, outra pessoa do time assume o papel de Scrum Master e conduz a reunião.

Além das reuniões diárias, existe também a possibilidade de **"Reuniões Informais"**, as quais ocorrem quando tentativas prévias de comunicação via chat não se mostram eficazes. Essas reuniões informais oferecem um espaço adicional para abordar questões que possam exigir uma discussão mais aprofundada, normalmente são para tirar dúvidas. No geral, o entrevistado considera que tanto as reuniões diárias quanto informais são bem executadas e necessárias para o bom andamento do projeto.

Ao abordarmos o tópico da **”Colaboração entre equipes”**, o entrevistado observa que não há uma reunião formal que reúna diversas equipes simultaneamente. No entanto, ele destaca que sempre que há uma convergência identificada entre as tarefas de equipes distintas, e essa convergência demanda algum nível de alinhamento, é adotada uma abordagem específica.

Nesses casos, é aberto um chamado no Jira, a ferramenta de gerenciamento utilizada, visando documentar e acompanhar a colaboração necessária. Caso a comunicação assíncrona não seja suficiente para resolver a situação de forma eficaz, é agendada uma chamada síncrona no Zoom, proporcionando um espaço para discussões mais imediatas e diretas entre as equipes envolvidas. Dessa forma, a colaboração entre equipes é gerenciada de maneira adaptável, priorizando a eficiência e a resolução de possíveis obstáculos.

Continuando, o próximo tópico discutido foi a **”Documentação Necessária”**. O entrevistado compartilha da opinião de que a documentação desempenha um papel fundamental em suas atividades diárias. No projeto, existe não apenas uma documentação referente ao produto, fornecida pelo Product Manager (PM), mas também uma documentação técnica que é atualizada de forma ativa e constante. De acordo com o entrevistado, essas documentações desempenham um papel importante na redução da necessidade de reuniões informais, proporcionando um recurso de consulta mais abrangente.

Além disso, o entrevistado expressa a opinião de que a documentação deveria ser mais valorizada e priorizada no âmbito do projeto, visto que sua presença contribui para uma comunicação mais clara e uma compreensão mais eficaz das atividades e requisitos do projeto.

Por último, foram abordados aspectos como o **”Padrão de Codificação”**. O entrevistado compartilha que, devido ao uso da linguagem Java, eles seguem rigorosamente o padrão estabelecido pela linguagem. Além disso, o tópico da **”Revisão de Código”** também foi explorado. A revisão é conduzida de forma assíncrona, usando a ferramenta ”Phabricator”, que é empregada para realizar commits e controlar as versões do código.

Quando um desenvolvimento é concluído e o commit é realizado, o desenvolvedor sinaliza ao seu time, permitindo que outros membros da equipe revisem a alteração. A aprovação do commit está sujeita a uma validação automática simples, que verifica critérios como limites de caracteres por linha, entre outros. Os desenvolvedores que revisam o código atentam-se ao cumprimento do padrão de codificação, identificam possíveis erros e avaliam oportunidades de refatoração. Caso algum problema seja identificado, é esperado que o desenvolvedor responsável faça as correções necessárias.

Após as correções, o código segue diretamente para produção, demonstrando o compromisso com a **"Entrega Contínua"**, uma prática também adotada de forma integral pelo projeto do entrevistado.

Como observação adicional, o entrevistado faz uma comparação entre sua experiência atual e experiências passadas. Ele menciona ter trabalhado em projetos com menos cerimônias das práticas ágeis, que, em certo sentido, poderiam economizar tempo. No entanto, ele destaca que, apesar da economia de tempo, a eficácia da execução do projeto ficava comprometida. Isso poderia resultar em retrabalhos e comprometimento da qualidade. Em contraste, ele descreve sua experiência atual, na qual os ritos e cerimônias ágeis são bem definidos e rigorosamente seguidos. Isso, apesar de demandar mais tempo nas cerimônias, resulta em um fluxo de trabalho mais eficiente e na garantia de qualidade.

Além disso, o entrevistado foi questionado sobre se o cliente respeita e adere bem aos ritos ágeis. Sua resposta foi positiva. Ele também enfatizou que essa sólida adesão por parte do cliente desempenha um papel significativo no sucesso contínuo das práticas ágeis no projeto. Esse ponto condiz com o relato compartilhado pela Entrevistada B (ver seção 5.2.2).

5.2.4 Entrevistado D

O Entrevistado D, um desenvolvedor com mais de 5 anos de experiência em desenvolvimento de software, atualmente trabalha em uma empresa que oferece um produto direcionado a grandes empresas. Sua equipe é responsável por implementar novas funcionalidades para esse produto, com o objetivo de aumentar o valor entregue e atrair mais clientes para a empresa. Vale destacar que o time do entrevistado opera em fusos horários diferentes.

Como de costume, ao começar no tópico do **"Planejamento"**, o entrevistado introduz um formato distinto dos mencionados até o momento. Em contraste com as práticas de "Sprint", o entrevistado adota o Kanban por meio da aplicação "Linear", uma ferramenta análoga ao Jira, para conduzir o **"Gerenciamento de Tarefas"** e rituais ágeis. Durante a explicação do processo de planejamento, o entrevistado estabelece conexões com várias práticas da nossa lista.

Na empresa do entrevistado, há uma notável transparência em relação ao planejamento. Esse processo tem início com uma triagem entre os Product Managers (PMs) e os stakeholders de negócio. Eles colaboram para identificar os pontos que podem se beneficiar de novas funcionalidades ou ajustes. Com esses resultados em macroescala, é elaborado um conjunto de mudanças significativas, que são categorizadas como **"Product Backlog"**. Em seguida, ocorre a alocação de tarefas para

determinar a responsabilidade de cada equipe. Cada tarefa passa por um refinamento técnico, conduzido pelos desenvolvedores mais experientes, como o próprio entrevistado. Esse processo visa garantir que as tarefas sigam o padrão estabelecido pela empresa, uma outra prática ágil chamada ”Design Simples”, que consiste em manter tarefas mais simples possível.

Uma vez que o backlog está estabelecido e refinado, ocorrem reuniões semanais, nas quais é definida uma data de entrega para todas as tarefas, com base nas prioridades definidas pela alta administração. Uma vez estabelecidas, essas tarefas são gerenciadas por meio do **”Auto Gerenciamento”** de cada desenvolvedor, reduzindo a dependência de um gerente para supervisionar constantemente. À medida que são listadas como ”a fazer” no Linear, cada desenvolvedor é encorajado a selecionar uma tarefa e trabalhar nela de maneira proativa. Essa abordagem é fomentada pela cultura da empresa, que inclusive motiva os desenvolvedores juniores a adaptarem-se e exercitarem essa prática desde cedo.

Dentro do âmbito das **”Práticas de Comunicação”**, o entrevistado destaca a importância desse aspecto para a empresa. Como uma parte cultural, a empresa promove a tentativa inicial de contatos assíncronos através do Slack. Geralmente, os desenvolvedores preferem se comunicar por meio de mensagens de texto ou, em casos mais complexos, até gravar vídeos para explicar determinados temas ou dúvidas. Embora a comunicação assíncrona seja a prioridade, a comunicação síncrona não é proibida, pois essa preferência não é uma regra fixa, mas sim uma sugestão cultural. Quando necessário, os desenvolvedores buscam horários em que todas as partes envolvidas estejam disponíveis para realizar videoconferências por meio do Google Meet, onde ocorrem as **”Reuniões Informais”**. O entrevistado, por sua vez entende que essas reuniões são necessárias para resolver prioridades e não devem ser subestimadas.

Devido às diferenças de fusos horários, as **”Reuniões Diárias”** não são amplamente utilizadas no contexto atual. Em vez disso, ocorre apenas uma reunião semanal oficial, que é o único momento em que toda a equipe precisa se sincronizar. As equipes, conhecidas como ”Verticals”, são responsáveis por diferentes partes do produto. Muitas vezes, ocorrem dependências entre as tarefas das ”Verticals”, que são discutidas principalmente de maneira assíncrona por meio de mensagens de texto ou, se necessário, de maneira síncrona, conforme mencionado no tópico de práticas de comunicação.

Adicionalmente, a **”Documentação Necessária”** desempenha um papel vital no cotidiano da empresa. Através da plataforma ”Notion”, a empresa mantém não apenas toda a documentação de todas as fases dos projetos, mas também uma documentação pública explicando suas diretrizes e fluxos de trabalho mencionados anteri-

ormente. O entrevistado enfatiza a importância da documentação e ressalta que sua criação demanda o tempo necessário. Segundo ele, a documentação reduz consideravelmente a necessidade de reuniões síncronas, o que é extremamente útil em um contexto globalmente distribuído, onde os fusos horários podem variar significativamente e nem todos estejam online ao mesmo tempo.

As documentações desempenham um papel crítico e são submetidas a validações durante as etapas de **"Revisão de Código"**. Após a conclusão do desenvolvimento, o desenvolvedor cria um "pull request"no GitLab e convida até dois outros desenvolvedores para revisarem não apenas o **"Padrão de Codificação"**, mas também a documentação, identificando possíveis oportunidades de "Refatoramento"ou bugs. No que diz respeito ao padrão de codificação, o entrevistado enfatiza que o projeto segue diretrizes tanto em termos de código quanto de arquitetura, e a revisão deve abranger todas essas regras.

O processo de desenvolvimento é acompanhado de perto por toda a equipe. Após a aprovação de um desenvolvedor, o código é incorporado imediatamente ao ambiente produtivo, alinhado com a prática de **"Entrega Contínua"**. Isso resulta em entregas diárias de valor para a empresa, uma abordagem que o entrevistado destaca como uma característica essencial do fluxo de trabalho da organização.

O entrevistado também enfatizou que a empresa realiza entregas diárias de atualizações para o software, demonstrando o compromisso contínuo em fornecer melhorias e novas funcionalidades aos clientes. Além disso, foi salientado que, caso haja necessidade de ajustes sugeridos durante a revisão de código, o desenvolvedor prioriza esses ajustes. Isso é crucial, pois a empresa está sujeita a auditorias que não permitem a implantação de código sem uma revisão prévia.

É importante destacar a eficácia desse modelo, em que a documentação, os padrões de codificação e as revisões são cuidadosamente tratados dentro do processo de desenvolvimento. Essas práticas permitem que a empresa mantenha a qualidade do código, reduza erros e bugs, bem como ofereça um fluxo contínuo de melhorias e novas funcionalidades para o software. Isso não apenas resulta em um software mais confiável, mas também contribui para a entrega contínua de valor para os clientes da empresa.

Como uma observação adicional, o entrevistado compara sua experiência atual com outras em que trabalhou com Sprint e reuniões diárias, algo que não vivencia atualmente. Ele menciona que o modelo de Sprint gerava um nível de estresse maior devido à necessidade constante de replanejamento semanal das demandas, e ele pessoalmente prefere o formato atual. No entanto, ele compreende que em uma empresa de consultoria, os clientes podem demandar um acompanhamento mais próximo do projeto, tornando a prática de Sprint mais adequada. Isso é diferente de uma em-

presa com um produto próprio, onde as decisões são exclusivamente tomadas internamente. Em relação à qualidade de vida, o entrevistado destaca que o modelo de auto-gerenciamento com Kanban cria um ambiente mais tranquilo, e ele tem funcionado bem, mesmo considerando os desafios de fusos horários e diferenças culturais no trabalho globalmente distribuído.

5.2.5 Entrevistado E

O último entrevistado possui uma experiência de 7 anos no campo do desenvolvimento de software e está atualmente engajado com duas empresas, assumindo papéis tanto como desenvolvedor quanto como líder técnico. No entanto, devido à diminuição da demanda em uma das empresas, o entrevistado informou que está atualmente desempenhando principalmente o papel de desenvolvedor de forma prática. É importante ressaltar que a condução da entrevista com este participante ocorreu através de áudios no WhatsApp, devido a questões logísticas. Embora essa abordagem tenha algumas limitações, ela ainda acrescenta valor significativo à pesquisa.

Quando questionado sobre o **"Planejamento"**, o entrevistado compartilha que nas duas empresas em que trabalha, a abordagem inclui a prática de "Histórias de Usuário". Nessas situações, as histórias já estão predefinidas e a partir delas ocorre uma análise de requisitos para a formação do **"Product Backlog"**. Além disso, ele menciona ter utilizado a sequência de Fibonacci para atribuição de Story Points. Embora ele não afirme que seja uma prática essencial, destaca que em suas experiências anteriores houve um momento de planejamento que exerceu um impacto significativo no andamento dos projetos.

No tocante ao **"Gerenciamento de Tarefas"**, o entrevistado enumera várias ferramentas que já utilizou, como o Asana, Figma e a mais conhecida delas, o Jira. Ao abordar o tópico do **"Auto Gerenciamento"**, ele menciona ter empregado o Atlassian para visualizar gráficos de velocidade, considerando esse auto gerenciamento como uma forma de obter feedback sobre o progresso de suas atividades de desenvolvimento.

No que diz respeito às **"Práticas de Comunicação"**, o entrevistado as considera comuns e essenciais, independentemente da metodologia utilizada. Ele lista várias ferramentas que já utilizou, como Microsoft Teams, Slack, Skype e até mesmo Discord. O entrevistado enfatiza que a comunicação é aplicável tanto para esclarecer dúvidas em tickets ou tarefas quanto para os ritos ágeis, como as **"Reuniões Diárias"**, as **"Sprints"** e também as **"Reuniões Informais"**. Ele destaca a importância das Reuniões Informais, considerando-as essenciais, pois muitas vezes um problema significativo pode ser resolvido com uma simples chamada entre duas pessoas.

Ao abordar as reuniões diárias, o entrevistado ressalta que elas podem ser muito benéficas ao fornecer uma visão completa do status do projeto, mas alerta que, se mal executadas, podem consumir um tempo excessivo, contrariando a natureza ágil que deveriam ter. O mesmo raciocínio se aplica à prática das sprints, que quando bem conduzidas agregam muito valor às entregas, mas, se não executadas adequadamente, podem levar a confusões e retrabalho no planejamento.

Ao passar para a **"Colaboração em Equipe"**, o entrevistado considera essa prática importante, embora não a veja como sempre indispensável. Ele ilustra essa perspectiva com uma experiência de trabalho em um projeto com duas equipes, onde um membro era introvertido. A colaboração entre as equipes era um pouco mais desafiadora quando era necessário interagir com esse indivíduo, mas, graças à sua forte habilidade técnica, as soluções eram encontradas em tempo hábil. Portanto, o entrevistado destaca a importância da colaboração para manter um ambiente saudável no projeto, mas enfatiza que não é um requisito absoluto em todas as situações.

No tópico da **"Documentação Necessária"**, o entrevistado concorda que ela é uma necessidade independente da metodologia utilizada. Ele observa que a documentação surge sempre que há alguém responsável pelas definições e outro responsável pela parte técnica. O entrevistado também menciona que já utilizou ferramentas auxiliares como Swagger e Cucumber para essa finalidade.

Quando abordamos os **"Padrões de Codificação"**, o entrevistado afirma que esse tema era menos enfatizado anteriormente, mas que agora está ganhando mais importância. Ele menciona exemplos de design patterns como Singleton, Abstract Factory, Facade e Observer. Além disso, o entrevistado destaca que muitas vezes as pessoas seguem padrões que podem não ter um nome específico, mas ainda assim são padrões que contribuem para a qualidade e legibilidade do código.

No que diz respeito à **"Revisão de Código"**, o entrevistado acredita que essa prática é subestimada e deveria ser mais valorizada. Ele enfatiza que ninguém consegue resolver um sistema sozinho e que a revisão de código não só ajuda a evitar bugs e erros, mas também fornece feedback valioso para aprendizado contínuo. Além disso, o processo de revisão de código gera dados estatísticos relacionados à escalabilidade e impacto, que podem ser analisados posteriormente no projeto. O entrevistado compartilha que, em seu projeto atual, trabalha com revisões de código e que, para que uma revisão seja aprovada, ela deve ser validada por duas pessoas, sendo pelo menos uma delas um desenvolvedor sênior.

Por fim, ao abordar a **"Entrega Contínua"**, o entrevistado a descreve como "CICD"(Integração Contínua e Entrega Contínua) e enfatiza a importância dos processos automatizados para facilitar a entrega de software. Ele menciona que ferramentas como Jenkins são comuns no auxílio dessa prática. Vale ressaltar que enquanto a

Integração Contínua foca em fornecer valores ao cliente por meio da automação de processos de entrega, a Entrega Contínua se concentra em integrar e testar o código frequentemente. Apesar de não ser diretamente o tópico discutido, a Integração Contínua pode complementar a prática de Entrega Contínua para garantir a qualidade do produto final.

6 Discussão

A pesquisa em questão teve como objetivo reavaliar os dados referentes às 48 práticas destacadas em (ALVES, 2021) quando aplicadas em projetos GSD. Com uma amostra consideravelmente maior, pudemos identificar tanto mudanças quanto semelhanças em relação aos resultados anteriores. Ao examinar essas práticas, adotamos uma abordagem diferenciada para garantir uma maior aceitação das mesmas, como detalhado na seção 5.1. Ao contrário do cenário anterior, a prática de Reunião Diária não ocupa mais o topo das práticas mais aceitas, considerando a nova abordagem aplicada. No entanto, ela ainda permanece classificada como uma das mais aceitas. Em vez disso, práticas como Planejamento, Práticas de Comunicação e Auto Gerenciamento ganharam maior destaque.

Essas práticas puderam ser identificadas durante as entrevistas detalhadas na seção 5.2. Através de um planejamento sólido, comunicação eficiente e a liberdade para o auto gerenciamento, foi possível criar um ambiente mais propício e motivador nos projetos, como indicado pelos entrevistados B (5.2.2) e D (5.2.4). Essa abordagem está alinhada com os princípios do Manifesto Ágil, conforme descrito em (BECK; AL., 2001). Assim como em (ALVES, 2021) as visitas entre locais de trabalho são cada vez menos populares entre projetos distribuidos, sendo assim no survey essa prática segue sendo a que alcançou menos popularidade dentre as demais.

Ao classificarmos as práticas mais aceitas dentre as amostras (consulte a seção 5.1), foi possível observar diversas interações entre elas nas entrevistas detalhadas na seção 5.2. Um ponto que chamou bastante atenção foi o destaque dado à utilização das reuniões diárias. Segundo os relatos dos entrevistados B (5.2.2) e E (5.2.5), a má gestão das reuniões diárias podem resultar em um desperdício de tempo para os participantes. Sendo populares no contexto do Scrum, essas reuniões devem ser conduzidas pelo Scrum Master de forma eficaz para evitar que saiam do controle. Isso foi evidenciado como um fator positivo pelos entrevistados A (5.2.1) e C (5.2.3), que possuem um Scrum Master ativo, resultando em reuniões fluidas e produtivas.

Além disso, vários entrevistados fizeram menção a práticas menos populares em seus relatos, enfatizando que essas práticas frequentemente gravitam em torno das principais listadas. Entre essas práticas menos destacadas, estão o TDD, Histórias de Usuário, Refatoração, Reunião de Retrospectiva, Design Simples e Sprint de Tamanho Fixo. Além disso, durante as entrevistas, também ficou evidente o uso de diversas ferramentas de apoio que sustentam várias atividades, especialmente as práticas relacionadas à gestão, como Jira, Asana e Azure; comunicação, como Microsoft

Teams, Slack, Skype e Discord; documentação, como Notion, Swagger e Cucumber; e código, em sua maioria por meio do GitLab.

Scrum e Kanban permaneceram entre as frameworks mais populares em relação às demais (consulte a Figura 7). Essa observação também foi reforçada pelos relatos dos entrevistados (veja a seção 5.2). No entanto, uma diferença notável se destaca quanto ao formato de trabalho adotado pelas empresas. O depoimento do entrevistado D (conforme a seção 5.2.4) revela que sua empresa, que desenvolve software próprio, escolheu o Kanban como framework, uma decisão que ele acredita se encaixar perfeitamente na dinâmica da empresa. Ele destaca que o Kanban proporcionou uma abordagem menos estressante e mais tranquila para a equipe. Entretanto, ele pondera que a abordagem do Scrum com sprints pode ser mais apropriada para empresas de consultoria, onde os clientes frequentemente desejam um acompanhamento online mais próximo de seus projetos. Isso é confirmado pelos relatos dos entrevistados A (5.2.1) e C (5.2.3), ambos atuando em empresas de consultoria que adotam o Scrum juntamente com sprints, indicando uma possível tendência.

Além disso, é importante destacar o relato da entrevistada B (5.2.2), também atuando em uma empresa de consultoria, que descreveu uma tentativa de implementar o Scrum, porém o cliente não se adaptou à prática. Isso sugere que a maturidade do cliente pode desempenhar um papel significativo na escolha da metodologia adequada. No entanto vale ressaltar que a presença de facilitadores desses rituais (como um Scrum Master) também é essencial para o funcionamento eficaz dessas abordagens (EREIZ; MUŠIĆ, 2019)(RAMOS; JUNIOR, 2017).

6.1 Limitações

Realizar este trabalho de forma independente apresentou diversos desafios, particularmente em relação à coleta de dados. Dado que a pesquisa anterior resultou em apenas 25 respostas, a expectativa para este estudo era obter pelo menos o dobro de participações, a fim de garantir uma amostra mais robusta para análises estatísticas. Apesar dos esforços para promover o novo survey em grupos de engenheiros de software, em redes sociais e através de networking, foi possível reunir apenas 34 respostas, um número maior do que na pesquisa anterior, mas ainda aquém do ideal para análises significativas. Diante desse cenário, foi tomada a decisão de combinar os dados das duas pesquisas, buscando consolidar uma amostra mais representativa. É importante destacar que, mesmo com ações de compartilhamento do survey, a maior parte das respostas foi obtida por meio de abordagens individuais, contatando pessoas uma a uma e solicitando sua participação.

Além disso, a condução das entrevistas também exigiu considerável esforço,

já que poucas pessoas estavam disponíveis para dedicar seu tempo. Inicialmente, as entrevistas estavam programadas para durar de 15 a 25 minutos, abrangendo todos os tópicos relacionados às práticas ágeis. No entanto, ao longo das conversas, o tempo médio de duração acabou se estendendo para cerca de 40 minutos a uma hora. Esse intervalo revelou-se suficiente para coletar informações valiosas e enriquecedoras para o presente projeto. Contudo, alguns imprevistos, como um erro durante a gravação da primeira entrevista e questões logísticas enfrentadas pelo Entrevistado E, acabaram gerando atrasos na conclusão do projeto.

7 Conclusão

Em suma, o presente trabalho se propôs a investigar e compreender as práticas ágeis mais utilizadas em projetos de Desenvolvimento de Software Distribuído (GSD). Através da análise de dados provenientes de survey e entrevistas com profissionais da área, foi possível reavaliar e identificar as práticas mais aceitas nesse contexto, bem como entender suas interações e como elas se relacionam entre si.

Este estudo foi embasado em pesquisas prévias que identificaram e analisaram 48 práticas ágeis através de uma revisão sistemática. A obtenção de dados contou com a participação de 60 respondentes, majoritariamente desenvolvedores de diversas empresas. Essas respostas validaram e aprofundaram as descobertas prévias, trazendo à tona contrastes e sutilezas que enriquecem a compreensão do campo. A diversidade da amostra resultou em conclusões substanciais, reforçando a predominância de práticas frequentemente citadas na literatura e, simultaneamente, destacando o papel significativo de práticas menos exploradas no contexto ágil de desenvolvimento de software. Como resultado, foi possível classificar 13 práticas como as mais aceitas entre os respondentes.

Para melhor compreensão dessas 13 práticas mais aceitas, realizamos entrevistas semi-estruturadas com 5 profissionais experientes da área. Esses indivíduos trabalham em empresas com contextos variados, oferecendo uma visão diversificada do campo. Através de seus relatos, pudemos mergulhar profundamente nas nuances das práticas, compreendendo sua aplicação em cenários distintos, explorando como elas interagem e identificando os desafios associados à sua implementação.

A análise das entrevistas evidenciou a presença significativa das 13 práticas no cotidiano dos 5 entrevistados. Adicionalmente, observou-se a incorporação de outras práticas menos populares, indicando a flexibilidade e adaptabilidade dos profissionais em relação ao uso de diferentes abordagens ágeis. Essas descobertas enriqueceram nosso entendimento sobre como as práticas ágeis podem ser aplicadas na prática, contribuindo para a construção de um portfólio informativo abrangente.

Este estudo abre caminho para investigações futuras que possam aprofundar nosso entendimento das práticas menos populares no contexto de desenvolvimento de software distribuído ágil (AGSD). Explorar essas práticas menos citadas poderia fornecer informações valiosas sobre suas aplicações, benefícios e possíveis desafios, contribuindo para um panorama mais abrangente das abordagens ágeis em ambientes distribuídos.

Referências

ABRAHAMSSON, P.; BABAR, M. A.; KRUCHTEN, P. Agility and architecture: Can they coexist? *IEEE software*, IEEE, v. 27, n. 2, p. 16–22, 2010. Citado na página 16.

ÅGREN, P.; KNOPH, E.; SVENSSON, R. B. Agile software development one year into the covid-19 pandemic. *Empirical Software Engineering*, Springer, v. 27, n. 6, p. 121, 2022. Citado na página 13.

ALVES, A. F. *Aplicação de métodos ágeis em desenvolvimento global de software*. Dissertação (B.S. thesis) — Brasil, 2021. Citado 7 vezes nas páginas 13, 20, 21, 22, 24, 25 e 48.

ANZALDÚA, G. E. *Interviews/entrevistas*. [S.l.]: Routledge, 2020. Citado na página 21.

AVRITZER, A.; BRONSARD, F.; MATOS, G. Improving global development using agile. In: *Agility Across Time and Space*. [s.n.], 2010. Disponível em: <https://api.semanticscholar.org/CorpusID:60022438>. Citado 2 vezes nas páginas 18 e 20.

BATTERTON, K. A.; HALE, K. N. The likert scale what it is and how to use it. *Phalanx*, JSTOR, v. 50, n. 2, p. 32–39, 2017. Citado na página 30.

BECK, K. Extreme programming explained: Embrace change. Addison-Wesley Professional, 1999. Citado na página 16.

BECK, K.; AL. et. *Manifesto for Agile Software Development*. 2001. Retrieved from <http://agilemanifesto.org/>. Citado 3 vezes nas páginas 13, 16 e 48.

CALEFATO, F.; DAMIAN, D.; LANUBILE, F. Computer-mediated communication to support distributed requirements elicitations and negotiations tasks. *Empirical Software Engineering*, v. 17, n. 6, p. 640–674, 2012. ISSN 1573-7616. Journal article. Disponível em: <https://doi.org/10.1007/s10664-011-9179-3>. Citado na página 18.

CAMARA, R. et al. Agile global software development: A systematic literature review. In: *Proceedings of the XXXIV Brazilian Symposium on Software Engineering*. [S.l.: s.n.], 2020. p. 31–40. Citado 6 vezes nas páginas 13, 20, 21, 22, 23 e 24.

CARMEL, E. *Global software teams: collaborating across borders and time zones*. [S.l.]: Prentice Hall PTR, 1999. Citado na página 17.

CARMEL, E. *Global Software Teams*. [S.l.]: Prentice Hall PTR, 2011. Citado na página 17.

CARMEL, E.; AGARWAL, R. Tactical approaches for alleviating distance in global software development. *IEEE software*, IEEE, v. 18, n. 2, p. 22–29, 2001. Citado na página 17.

COCKBURN, A. *Agile Software Development: The Cooperative Game*. 2. ed. [S.l.]: Pearson, 2017. Citado na página 16.

ERDOGMUS, H.; MORISIO, M.; TORCHIANO, M. On the effectiveness of the test-first approach to programming. *IEEE Transactions on Software Engineering*, v. 31, n. 3, p. 226–237, 2005. Citado na página 16.

EREIZ, Z.; MUŠIĆ, D. Scrum without a scrum master. In: IEEE. *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*. [S.l.], 2019. p. 325–328. Citado na página 49.

FIGUEREDO, R. d. C. *Scaling agile methods in global software projects: Is it possible with SAFe?* Dissertação (B.S. thesis) — Brasil, 2020. Citado 2 vezes nas páginas 20 e 22.

FLICK, U. *Introducing research methodology: A beginner's guide to doing a research project*. [S.l.]: Sage, 2015. Citado na página 23.

HAJJDIAB, H.; TALEB, A. Adopting agile software development: Issues and challenges. *International Journal of Managing Value and Supply Chains*, v. 2, p. 1–10, 09 2011. Citado na página 16.

HIGHSMITH, J.; COCKBURN, A. Agile software development: The business of innovation. *Computer*, IEEE, v. 34, n. 9, p. 120–127, 2001. Citado na página 13.

HOSSAIN, E.; ALI, N. B. M.; BABAR, M. A. A systematic literature review on the devops in global software development. *Journal of Systems and Software*, Elsevier, v. 165, p. 110578, 2020. Citado na página 18.

HOSSAIN, E.; BABAR, M. A.; PAIK, H. young. Using scrum in global software development: A systematic literature review. *2009 Fourth IEEE International Conference on Global Software Engineering*, p. 175–184, 2009. Disponível em: <https://api.semanticscholar.org/CorpusID:11139156>. Citado 2 vezes nas páginas 18 e 20.

JALALI, S.; WOHLIN, C. Agile practices in global software engineering - a systematic map. *2010 5th IEEE International Conference on Global Software Engineering*, p. 45–54, 2010. Disponível em: <https://api.semanticscholar.org/CorpusID:918074>. Citado 2 vezes nas páginas 19 e 20.

KHANNA, E.; POPLI, R.; CHAUHAN, N. Artificial intelligence based risk management framework for distributed agile software development. In: *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*. [S.l.: s.n.], 2021. p. 657–660. Citado na página 18.

LIVINGSTON, E. H. The mean and standard deviation: what does it all mean? *Journal of Surgical Research*, Elsevier, v. 119, n. 2, p. 117–123, 2004. Citado na página 32.

LUO, L.; WILDEMUTH, B. M. Semistructured interviews. *Applications of social research methods to questions in information and library science*, Libraries Unlimited Westport, v. 232, 2009. Citado 3 vezes nas páginas 14, 21 e 23.

MANZINI, E. J. Entrevista semi-estruturada: análise de objetivos e de roteiros. *Seminário internacional sobre pesquisa e estudos qualitativos*, v. 2, p. 58–59, 2004. Citado na página 34.

MAREK, K.; WIŃSKA, E.; DĄBROWSKI, W. The state of agile software development teams during the covid-19 pandemic. In: PRZYBYŁEK, A. et al. (Ed.). *Lean and Agile Software Development*. Cham: Springer International Publishing, 2021. p. 24–39. ISBN 978-3-030-67084-9. Citado 3 vezes nas páginas 13, 19 e 20.

MARINHO, M.; NOLL, J.; BEECHAM, S. Uncertainty management for global software development teams. In: IEEE. *2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC)*. [S.l.], 2018. p. 238–246. Citado na página 17.

MARINHO, M. et al. Plan-driven approaches are alive and kicking in agile global software development. In: IEEE. *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. [S.l.], 2019. p. 1–11. Citado na página 18.

MISHRA, A.; ALZOUBI, Y. I. Structured software development versus agile software development: a comparative analysis. *International Journal of System Assurance Engineering and Management*, Springer, p. 1–19, 2023. Citado na página 13.

MOCKUS, A.; HERBSLEB, J. Challenges of global software development. In: IEEE. *Proceedings seventh international software metrics symposium*. [S.l.], 2001. p. 182–184. Citado na página 17.

PODARI, Z. et al. Systematic literature review on global software development risks in agile methodology. In: *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*. [S.l.: s.n.], 2020. p. 231–236. Citado na página 17.

POPPENDIECK, M.; POPPENDIECK, T. *Lean Software Development: An Agile Toolkit*. [S.l.]: Addison-Wesley Professional, 2003. Citado na página 16.

RAMOS, A. B.; JUNIOR, D. C. V. A influencia do papel do scrum master no desenvolvimento de projetos scrum. *Revista de Gestão e Projetos*, v. 8, n. 3, p. 80–99, 2017. Citado na página 49.

SCHEUREN, F. What is a survey? In: AMERICAN STATISTICAL ASSOCIATION ALEXANDRIA. [S.l.], 2004. Citado 2 vezes nas páginas 14 e 22.

SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide*. [S.l.], 2020. Citado na página 16.

SERRADOR, P.; PINTO, J. K. Does agile work?—a quantitative analysis of agile project success. *International journal of project management*, Elsevier, v. 33, n. 5, p. 1040–1051, 2015. Citado na página 16.

SINDRE, G.; OPDAHL, A. L. Eliciting security requirements with misuse cases. *Requirements engineering*, Springer, v. 10, p. 34–44, 2005. Citado na página 17.

VALLON, R. et al. Systematic literature review on agile practices in global software development. *Information and Software Technology*, Elsevier, v. 96, p. 161–180, 2018. Citado na página 17.

WAGNER, S. et al. Challenges in survey research. In: ____. *Contemporary Empirical Methods in Software Engineering*. Cham: Springer International Publishing, 2020. p. 93–125. ISBN 978-3-030-32489-6. Disponível em: <https://doi.org/10.1007/978-3-030-32489-6_4>. Citado na página 25.

ZAFAR, A.; ALI, S.; SHAHZAD, R. K. Investigating integration challenges and solutions in global software development. *2011 Frontiers of Information Technology*, IEEE, p. 291–297, 2011. Citado na página 17.

A Apêndice A - Um catálogo das Práticas adotadas em Desenvolvimento Global de Software Ágil

Baseado nos resultados do presente trabalho, através da coleta de dados e entrevistas foi gerado documento a seguir. O portfolio abrange boas práticas em Desenvolvimento de Software Ágil, mostrando como essas práticas podem funcionar em contextos e situações diferentes e indicando ferramentas que auxiliam em seu funcionamento.

Rodrigo Guedes

UM CATÁLOGO DAS PRÁTICAS ADOTADAS EM **DESENVOLVIMENTO GLOBAL DE SOFTWARE ÁGIL**

Universidade Federal Rural de Pernambuco

Introdução

O presente portfólio foi criado com a intenção de auxiliar profissionais a compreender a aplicação de práticas ágeis amplamente utilizadas em seus processos de desenvolvimento de software. Este recurso oferece uma visão abrangente das práticas essenciais em diferentes esferas, proporcionando insights sobre como integrá-las de maneira eficaz para promover um ambiente de desenvolvimento coeso.

Para sua elaboração, foram conduzidas pesquisas abrangentes, incluindo análise de artigos científicos, aplicação de surveys e realização de entrevistas com profissionais da área de engenharia de software. Essa abordagem permitiu observar e compreender as práticas ágeis mais adotadas por equipes de desenvolvimento de software distribuídas. Além disso, foi possível identificar os contextos nos quais essas práticas se destacam, como elas interagem com diversos frameworks e, por fim, destacar as ferramentas que podem potencializar sua implementação eficaz.

Lista de Práticas Ágeis

1. Backlog do Produto
2. Planejamento
3. Gerenciamento de Tarefas
4. Auto Gerenciamento
5. Sprint
6. Documentação Necessária
7. Práticas de Comunicação
8. Reuniões Diárias
9. Reuniões Informais
10. Colaboração entre Equipes
11. Padrão de Codificação
12. Revisão de Código
13. Entrega Contínua

Backlog do Produto

Definição

O Product Backlog é fundamental no desenvolvimento ágil de software, concentrando recursos de negócios e arquitetônicos para equipes distribuídas. Proprietário do produto e líder da equipe são responsáveis por priorizar funcionalidades e melhorias com base nas necessidades do cliente e feedback. Esse registro dinâmico passa por refinamentos regulares, permitindo adaptações locais em projetos globais. Sua flexibilidade centraliza o foco nas demandas do cliente e acomoda mudanças ao longo do desenvolvimento.

Contextos e frameworks

É aplicado em vários contextos e frameworks de desenvolvimento de software. Ele deve ser utilizado em frameworks como Kanban ou Scrum e em empresas que fornecem qualquer tipo de produto ou serviço. Em consultorias, por exemplo deve ser identificado as necessidades do cliente e sintetiza-las em histórias de usuário, por exemplo. Em um contexto de empresas de produto autoral, deve ser identificado inovações ou melhorias que gerem valor para o produto em questão.

Ferramentas

Várias ferramentas de gerenciamento auxiliam no backlog, sendo exemplos: Jira, Asana, Asure, Linear ou Phabricator.

Planejamento

Definição

Normalmente baseado no backlog do produto, Planejamento é um processo geralmente realizado em reuniões, no qual as equipes de desenvolvimento de software, juntamente com o product owner, scrum master e stakeholders, têm a oportunidade de esclarecer dúvidas, alinhar o escopo de acordo com as expectativas do cliente e reduzir mal-entendidos em relação às tarefas propostas. Esse processo visa garantir uma compreensão clara das metas e requisitos do projeto. A partir das prioridades definidas, o time pode seguir com o desenvolvimento.

Contextos e frameworks

Deve ser configurado em qualquer situação, mudando apenas sua forma de abordagem. No KanBan, por exemplo é sugerido que seja estipulado uma meta de entregas em uma data especifica, dois meses por exemplo. A partir daí pode ser realizada uma reunião de planejamento para identificar as entregas que geram maior valor naquele momento e segui-las como prioridades. Já no Scrum, o planejamento esta bem alinhado com a Sprint, pois ao inicio de cada uma, deve ser realizado um planejamento para a mesma. Esse rito é chamado de Sprint Planning.

Ferramentas

Comunicação e Gerenciamento estão ativos no planejamento, então ferramentas como: Jira, Asana, Asure, Linear ou Phabricator são essenciais, assim como: Slack, e Microsoft teams.

Gerenciamento de Tarefas

Definição

O Gerenciamento de Tarefas, nas práticas ágeis, tem como objetivo facilitar a colaboração e coordenação da equipe durante a execução das atividades do projeto. As equipes, juntamente com o proprietário do produto e o scrum master, utilizam ferramentas de compartilhamento online para acessar o backlog do produto, storyboard, taskboard, gráficos de burndown e outros artefatos ágeis. Além disso, um repositório central de código é frequentemente utilizado para rastrear o progresso das tarefas. Essa prática promove uma gestão eficaz das atividades e mantém a equipe alinhada com as metas do projeto.

Contextos e frameworks

Essa prática pode ser inserida em todos os contextos, qualquer framework ou metodologia necessita de um controle de tarefas e para isso é fundamental utilizar uma ferramenta poderosa. Essa prática se relaciona como várias outras práticas, como backlog, planejamento, auto gerenciamento e tudo que envolva algum tipo de status.

Ferramentas

Ferramentas de gerenciamento: Jira, Azure, Linear, Trello e Sharepoint estão entre as mais famosas.

Auto Gerenciamento

Definição

Auto Gerenciamento é uma prática ágil que visa alinhar prioridades, planejamento e manter a equipe focada de forma responsável. Essa abordagem envolve equipes autogerenciadas que têm a liberdade de escolher suas próprias tarefas, com ênfase na aprendizagem contínua, compartilhamento de conhecimento e formação de equipes. Os membros da equipe também são capacitados a compartilhar e definir problemas e atividades quando possível, além de atribuir responsabilidades exclusivas. Essa prática promove um ambiente de trabalho colaborativo e empoderado, onde cada membro contribui ativamente para o sucesso do projeto.

Contextos e frameworks

Auto gerenciamento pode variar seu nível de liberdade dependendo de que framework utilize. No Scrum, por exemplo o auto gerenciamento deve ser feito conforme os alinhamentos da Sprint. Onde você pode definir a ordem de execução das tarefas da sprint, desde que esteja alinhado com o planejamento. Já no Kanban, ao estipular uma data relativamente maior para todas as entregas, cada funcionário deve identificar suas prioridades no decorrer daquele período. Essa prática se torna mais dificil num contexto com funcionários menos experientes, requerendo observação.

Ferramentas

Comunicação e Gerenciamento estão ativos no planejamento, então ferramentas como: Jira, Asana e Asure são essenciais, assim como: Slack, e Microsoft teams.

Sprint

Definição

Sprint é uma prática ágil com o objetivo de desenvolver requisitos pré-definidos em um período de tempo limitado, conhecido como time-boxed. Durante esse período, nenhuma mudança nos requisitos é aceita, e a equipe trabalha para produzir um incremento de software funcional até o final da sprint. Essa abordagem envolve equipes de desenvolvimento e é implementada através de práticas ágeis, como planejamento de sprint, revisão de sprint e retrospectiva de sprint. Além disso, a recomendação é manter ciclos de sprints curtos para entregar valor constantemente ao cliente. Essa prática promove um desenvolvimento ágil e eficiente, focado na entrega de valor em intervalos regulares.

Contextos e frameworks

Sprint é muito utilizado no Scrum. Normalmente em situações onde o cliente prefere trabalhar de perto com o time de desenvolvimento, a Sprint é interessante para melhor controle e comunicação. Pode ser combinada com reuniões diárias ou semanais. Importante enfatizar a necessidade de um scrum master para que seja garantido os ritos da sprint conforme devem ser. Caso o contrário a Sprint pode ser executada de forma desorganizada e ineficaz.

Ferramentas

Para a Sprint é predominante a necessidade de ferramentas de comunicação. Por exemplo: Slack ou Microsoft Teams.

Documentação Necessária

Definição

Documentação Necessária é uma prática ágil que visa melhorar a comunicação e fornecer informações detalhadas sobre o projeto para todas as equipes envolvidas. Isso é facilitado por meio da criação de documentação, o que reduz a necessidade constante de comunicação entre os membros da equipe, promovendo uma colaboração eficaz.

Contextos e frameworks

Documentações são relevantes em qualquer framework ágil, no entanto são mais eficazes em projetos que possuem algum tipo de dificuldade na comunicação. Geralmente projetos onde os fuzo horários são diferentes, a priorização nas documentações são bem vindas, visto que minimiza as necessidades de contatos síncronos que acabam sendo mais difíceis de acontecer nessas situações.

Ferramentas

Repositórios onde estejam dispostos para todo o projeto ver são bem vindos. Podendo ser: Notion, Swagger, Cucumber ou até mesmo o Jira.

Práticas de Comunicação

Definição

Práticas de Comunicação são métodos empregados para reduzir mal-entendidos, fortalecer os laços entre equipes distribuídas e compartilhar informações comuns. Isso pode envolver a adoção de "modos de comunicação múltiplos", combinando abordagens síncronas e assíncronas para uma comunicação eficaz. Além disso, a sincronização dos horários de trabalho entre equipes distribuídas é sugerida para melhorar os níveis de comunicação.

Contextos e frameworks

Comunicação é necessária em toda e qualquer situação, Geralmente em projetos onde os horários de seus funcionários não destoam tanto, a comunicação síncrona tende a ser popular e em fuso horários distintos assíncrona.

Ferramentas

Ferramentas populares: Slack ou Microsoft Teams.

Reuniões Diárias

Definição

Reuniões Diárias têm o propósito de compartilhar o progresso do trabalho, identificar bloqueios e definir as tarefas do dia. Envolvem equipes e scrum master, sendo realizadas por meio de diversas ferramentas de comunicação, como telefone, videoconferência, webcam, e-mails e chats na internet.

Contextos e frameworks

Essas reuniões devem ser utilizadas de maneira estratégica e bem executada. Frameworks que utilizam sprint, normalmente utilizam reuniões diárias síncronas através de conferências. No entanto é recomendado impor regras durante a reunião para que ela não se expanda demais. Regras como por exemplo: cada funcionário tem apenas 3 minutos para falar, limitam um pouco o tempo impedindo de extrapolar. Caso o time seja muito grande, uma comunicação apenas via chat compartilhado pode ser mais efetivo, onde cada pessoa escreve seu status.

Ferramentas

Para as Reuniões Diárias é predominante a necessidade de ferramentas de comunicação. Por exemplo: Slack ou Microsoft Teams.

Reuniões Informais

Definição

Reuniões informais no contexto ágil são encontros não planejados entre membros da equipe, destinados a promover comunicação, esclarecer dúvidas e discutir questões do projeto. Geralmente breves e espontâneas, ocorrem presencialmente, por chats, chamadas de vídeo ou outras ferramentas de comunicação, contribuindo para manter a equipe alinhada e promovendo uma cultura de colaboração.

Contextos e frameworks

Essas reuniões tendem a ajudar bastante a resolver gargalos ou impedimentos de forma efetiva e rápida. Uma simples chamada pode resolver um problema complicado. As reuniões informais também podem ser utilizadas por qualquer framework, no entanto tendem a acontecer menos em projetos com fuso horários distintos.

Ferramentas

Para as Reuniões Informais é predominante a necessidade de ferramentas de comunicação. Por exemplo: Slack ou Microsoft Teams.

Colaboração entre Equipes

Definição

A colaboração entre equipes no contexto ágil tem como objetivo fortalecer a cooperação, construir confiança e promover um ambiente harmonioso. Realizada por meio de sessões prévias às reuniões oficiais ou até mesmo em reuniões informais, essa prática permite esclarecer dúvidas, compartilhar desafios e até mesmo promover interações sociais entre os membros das equipes.

Contextos e frameworks

As equipe num ambiente ágil tem o mesmo propósito: gerar valor para o cliente. Sendo assim, eventualmente elas podem ter tarefas que sejam codependentes entre si, tanto em frameworks como Scrum ou Kanban. Sendo assim, importante manter comunicação para alinhar necessidades, tirar duvidas, ou simplesmente manter um bom relacionamento. Esse também é um fator chave para um bom ambiente corporativo.

Ferramentas

Para a colaboração entre equipes é predominante a necessidade de ferramentas de comunicação. Por exemplo: Slack ou Microsoft Teams.

Padrão de Codificação

Definição

O padrão de código tem como objetivo apoiar os desenvolvedores na criação de diretrizes comuns para escrever código legível e sustentável. Essa prática envolve líderes de equipe e equipes de desenvolvimento. Para implementá-la, é recomendado que os padrões de codificação sejam estabelecidos no início do processo de desenvolvimento, preferencialmente discutidos por membros seniores do projeto para garantir o entendimento de todas as regras entre os membros da equipe. Além disso, é fundamental que esses padrões sejam seguidos por todas as equipes distribuídas .

Contextos e frameworks

Sendo uma prática essencial em qualquer contexto. Algo bem comum entre os padrões de codificação, equipes costumam utilizar workbooks, guidelines e design patterns, todos visam garantir uma melhor analise do código e uma melhor definição de arquitetura. São exemplos de design patterns: Arquitetura Clean, CQRS, Singleton, Abstract Factory, Facade, Observer, entre outros...

Ferramentas

Não existe exatamente uma ferramenta para auxilio nos Design Patterns, no entanto uma boa documentação é essencial para guiar o desenvolvedor a seguir um padrão adequado.

Revisão de Código

Definição

A revisão de código tem como objetivo aprimorar a qualidade do código, identificar bugs precocemente, compartilhar conhecimento, reduzir testes e garantir a qualidade geral da solução. Esta prática envolve equipes de desenvolvimento e pode ser realizada com o auxílio de recursos sêniores, que possuem habilidades específicas para revisar o código. Além disso, sugere-se que desenvolvedores sêniores conduzam revisões em conjunto com desenvolvedores juniores, com o propósito de treinar os membros mais novos para escreverem código de acordo com os padrões desejados.

Contextos e frameworks

A revisão de código também é uma prática essencial em qualquer contexto, pois evita retrabalhos ou gargalos desnecessários. Algumas equipes sugerem que o code review deve ser feito por pelo menos 2 pessoas, e não pode ser a mesma que construiu o código.

Ferramentas

Ferramentas de auxilio de bugs ou até controladores de versões como o GitLab e alguns sistemas ja vem com code review automatico embutido.

Além disso, pode ser utilizado ferramentas para contabilizar erros como HPQC ou até algumas de gerenciamento como Jira ou Phabricator possuem features que podem trazer algum apoio.

Entrega Contínua

Definição

A entrega contínua tem como objetivo fornecer incrementos de software ao cliente de forma frequente. Essa prática é realizada pelas equipes de desenvolvimento. As entregas de código são uma atividade recorrente, com um cronograma estabelecido com base nas necessidades do projeto e devem ocorrer desde as fases iniciais do projeto.

Contextos e frameworks

As entregas contínuas podem ser realizadas a qualquer momento do projeto. Em frameworks como Scrum, normalmente é utilizado o as entregas ao final da sprint. Em frameworks como Kanban, a entrega pode ser realizada imediatamente após o termino da tarefa.
Uma prática que pode combinar com a entrega contínua, seria a própria Integração Contínua, onde essa visa utilizar ferramentas de automação para agilizar as entregas. Essa combinação de práticas é chamada de CI/CD.

Ferramentas

Ferramentas como Jenkins e GitLab podem auxiliar na automação da entrega.

B Apêndice B - Survey sobre práticas ágeis em ambientes AGSD

Este survey foi criado a partir da ferramenta Google Forms e distribuído para profissionais AGSD.

Pesquisa sobre Praticas Ágeis em Desenvolvimento de Software em Projetos Distribuídos.

Gostaríamos do seu apoio nesta pesquisa que objetiva compreender como as empresas de softwares utilizam as práticas de desenvolvimento ágeis em projetos distribuídos. O questionário foi desenvolvido no âmbito de projeto de Trabalho de Conclusão de Curso de Licenciatura em Computação da UFRPE.

Se você faz parte de um time ágil e de um projeto de software distribuído, ou seja, composto por pessoas que atuam em diferentes localidades, gostaria de pedir a disponibilidade de 5 a 10 minutos do seu tempo para nos ajudar com essa pesquisa.

Caso, no decorrer do questionário, não se sinta confortável em responder as perguntas, você é livre para interromper sua participação em qualquer momento. A pesquisa é anônima e ao respondê-la, você autoriza o uso de suas respostas para fins acadêmicos. Por favor, leia atentamente as questões e responda-as com sinceridade.

Caso tenha alguma dúvida entre em contato pelo e-mail:

Agradecemos sua participação!

1. Declaro que concordo em participar dessa pesquisa e autorizo o uso dos resultados deste estudo para atividades academicas e científica. *

   Marcar apenas uma oval.

   - ◯ Sim
   - ◯ Não

Abordagens de desenvolvimento de software

Nessa seção temos algumas perguntas básicas que visam compreender as abordagens ou frameworks utilizados por você dentro do projeto.

2. Qual dos seguintes frameworks você geralmente usa? (Marcar todas que se aplicam) *

- [ ] Processo Clássico em Cascata.
- [ ] Família Crystal.
- [ ] DevOps.
- [ ] Design orientado por domínio.
- [ ] Método de Desenvolvimento de Sistemas Dinâmicos. (DSDM)
- [ ] Programação Extrema. (XP)
- [ ] Desenvolvimento baseado em recursos. (FDD)
- [ ] Desenvolvimento Iterativo.
- [ ] Kanban.
- [ ] Scrum em larga escala. (LESS)
- [ ] Lean Software Development.
- [ ] Model Driven Architecture
- [ ] Nexus.
- [ ] Processo de Software Personalizado.
- [ ] Stage-gate model
- [ ] PRINCE2.
- [ ] Processo Unificado da Racional (RUP).
- [ ] Scaled Agile Framework (SAFe)
- [ ] Scrum.
- [ ] ScrumBan.
- [ ] Modelo Espiral.
- [ ] Análise de Sistemas Estruturados e Método de Projeto. (SSADM)
- [ ] Vshaped Process. (VModel)
- [ ] Outro: _____

3. Quais abordagens de desenvolvimento de software você utiliza no seus projetos? *

Marcar apenas uma oval.

- ◯ Abordagens totalmente ágeis.
- ◯ Principalmente abordagens ágeis.
- ◯ Balanceado entre abordagens ágeis e abordagens tradicionais.
- ◯ Principalmente abordagens tradicionais.
- ◯ Abordagens totalmente tradicionais.

Práticas de desenvolvimento ágeis em projetos de software distribuído

Com base na sua experiência pessoal, verifique as práticas listadas e informe a frequência de utilização da prática no seu projeto. Pense no seu dia-a-dia de trabalho e classifique-os de 1 a 5, com 5 sendo para uso sempre a prática e 1 sendo para não utilização da prática.

1 = Não uso
2 = Uso raramente
3 = As vezes uso
4 = Uso frequentemente
5 = Sempre uso

4. Com que frequência você utiliza as seguintes práticas? *

	Não uso	Uso raramente	As vezes uso	Uso frequentemente	Sempre uso
Reunião diária	◯	◯	◯	◯	◯
Práticas de comunicação	◯	◯	◯	◯	◯
Planejamento	◯	◯	◯	◯	◯
Auto Gerenciamento	◯	◯	◯	◯	◯
Scrum de scrums	◯	◯	◯	◯	◯
Histórias de usuários	◯	◯	◯	◯	◯
Revisão de sprint	◯	◯	◯	◯	◯
Reunião retrospectiva	◯	◯	◯	◯	◯
Backlog do produto	◯	◯	◯	◯	◯
Visitas entre locais de trabalho	◯	◯	◯	◯	◯
Sprint	◯	◯	◯	◯	◯
Gestão de pendências	◯	◯	◯	◯	◯
Programação em par	◯	◯	◯	◯	◯
Quadro kanban	◯	◯	◯	◯	◯
Integração contínua	◯	◯	◯	◯	◯
Burndown	◯	◯	◯	◯	◯
Projete a equipe	◯	◯	◯	◯	◯
Gerenciamento de tarefas	◯	◯	◯	◯	◯

Coaching					
Documentação necessária					
Sincronizar horas de trabalho					
Equipe reunida no início do projeto					
Reunião de estimativa					
TDD					
Implantação contínua					
Wiki do projeto					
Demonstração do sistema					
Automação de teste					
Definição de pápeis da equipe					
Revisão de código					
Reuniões informais					
Gerenciar expectativas do cliente					
Jogo de planejamento					
Feedbacks frequentes					
Arquitetura ágil					
Entrega contínua					
Padrões de codificação					

Propriedade do código coletivo	⬭	⬭	⬭	⬭	⬭
Missão e visão compartilhada	⬭	⬭	⬭	⬭	⬭
Refactoring	⬭	⬭	⬭	⬭	⬭
Alternar equipe entre locais	⬭	⬭	⬭	⬭	⬭
Bug tracker	⬭	⬭	⬭	⬭	⬭
Design simples	⬭	⬭	⬭	⬭	⬭
Sprint de tamanho fixo	⬭	⬭	⬭	⬭	⬭
Gestão de testes	⬭	⬭	⬭	⬭	⬭
Documentar lições aprendidas	⬭	⬭	⬭	⬭	⬭
Roadmap planning	⬭	⬭	⬭	⬭	⬭
Testes de aceitação	⬭	⬭	⬭	⬭	⬭
Escalonamento do Kanban	⬭	⬭	⬭	⬭	⬭
Expandir respons. das equipes	⬭	⬭	⬭	⬭	⬭
Colaboração entre equipes	⬭	⬭	⬭	⬭	⬭

5. Você utiliza outras práticas no seu projeto?

_____

6. Em sua experiência, as práticas ágeis utilizadas facilitam o desenvolvimento de seu projeto? *

1 - Definitivamente Não Facilitam 5 - Facilitam Bastante

7. Alguma prática ágil que você não pratica, poderia melhorar algum processo de desenvolvimento de seu projeto distribuído? (Comunicação, definição de prioridades, outros aspectos)

Marcar apenas uma oval.

- ◯ Não
- ◯ Sim

8. Em caso de resposta positiva na questão acima, cite as práticas ágeis que você acredita que possa melhorar o processo de desenvolvimento.

Dados demográficos

9. Quantos anos de experiência você tem em desenvolvimento de software e sistemas? *

Marcar apenas uma oval.

- ◯ Menos de 1 ano.
- ◯ 1 - 2 anos.
- ◯ 3 - 5 anos.
- ◯ 6 - 10 anos.
- ◯ Mais de 10 anos.

10. Qual é o tamanho da sua empresa em quantidade de funcionários? *

Marcar apenas uma oval.

- ◯ Micro (< 10 funcionários)
- ◯ Pequena (11 - 50 funcionários)
- ◯ Médio (51 - 250 funcionários)
- ◯ Grande (251 - 2499 funcionários)
- ◯ Muito grande (> 2500 funcionários)

11. Qual é a principal área de negócios da sua empresa? *

Marcar apenas uma oval.

- ◯ Área de negócios: opção residual. (negativa) ou número de opções selecionadas
- ◯ Desenvolvimento de software. (software personalizado, ou seja, soluções individuais)
- ◯ Desenvolvimento de software. (software padrão, por exemplo, SAP, Office)
- ◯ Desenvolvimento de sistema. (hardware e software, por exemplo, sistemas incorporados)
- ◯ Consultoria / Suporte à Gestão de Projetos.
- ◯ Consultoria, treinamento e serviços de TI.
- ◯ Pesquisa e Desenvolvimento.
- ◯ Outro: _____

12. A base de seu projeto é localizada em sua região? *

Marcar apenas uma oval.

- ◯ Sim
- ◯ Não

13. Sua empresa atua de maneira (globalmente) distribuída? *

Marcar apenas uma oval.

- ◯ Não.
- ◯ Sim, nacionalmente. (mesmo país)
- ◯ Sim, regionalmente. (mesmo continente)
- ◯ Sim, globalmente.

14. Qual é a principal função que você desempenha nesta empresa? *

Marcar apenas uma oval.

- ◯ Arquiteto.
- ◯ CIO / CTO.
- ◯ Desenvolvedor.
- ◯ Gerente de Produto.
- ◯ Gerente de Projeto.
- ◯ Gerente de Qualidade.
- ◯ Scrum Master.
- ◯ Testador.
- ◯ Treinador.
- ◯ Outro: _____

15. (Opcional) Caso possível, nos informe o nome da empresa que você trabalha.

_____

16. Você gostaria de acrescentar algum comentário ou sugestão para essa pesquisa?

_____

_____

_____

_____

_____

C Apêndice C - Roteiro das entrevistas

Este roteiro foi utilizado para realização das entrevistas com os profissionais da área.

Roteiro das entrevistas sobre as práticas ágeis mais aceitas em ambientes AGSD.

Tópicos:

Planejamento

Pergunta – Qual sua experiência com a prática Planejamento no seu contexto atual ou em contextos passados?

Práticas de comunicação

Pergunta – Qual sua experiência com a prática Práticas de Comunicação no seu contexto atual ou em contextos passados?

Auto Gerenciamento

Pergunta – Qual sua experiência com a prática Auto Gerenciamento no seu contexto atual ou em contextos passados?

Backlog do produto

Pergunta – Qual sua experiência com a prática Backlog do produto no seu contexto atual ou em contextos passados?

Reuniões informais

Pergunta – Qual sua experiência com a prática Reuniões informais no seu contexto atual ou em contextos passados?

Documentação necessária

Pergunta – Qual sua experiência com a prática Documentação necessária no seu contexto atual ou em contextos passados?

Reunião diária

Pergunta – Qual sua experiência com a prática Reunião diária no seu contexto atual ou em contextos passados?

Padrões de codificação

Pergunta – Qual sua experiência com a prática Padrões de codificação no seu contexto atual ou em contextos passados?

Sprint

Pergunta – Qual sua experiência com a prática Sprint no seu contexto atual ou em contextos passados?

Colaboração entre equipes

Pergunta – Qual sua experiência com a prática Colaboração entre equipes no seu contexto atual ou em contextos passados?

Entrega contínua

Pergunta – Qual sua experiência com a prática Entrega contínua no seu contexto atual ou em contextos passados?

Revisão de código

Pergunta – Qual sua experiência com a prática Revisão de código no seu contexto atual ou em contextos passados?

Gerenciamento de tarefas

Pergunta – Qual sua experiência com a prática Gerenciamento de tarefas no seu contexto atual ou em contextos passados?