

1 Introdução

Ao longo dos anos, o mundo foi passando por um processo de modernização que tornou a tecnologia indispensável ao dia a dia da sociedade (Castells, 2005, Lévy, 2010). Desde então, as empresas vêm aumentando seus investimentos na área da tecnologia da informação. Um levantamento feito pela ABES (Associação Brasileira de Empresas de Software) mostrou que o Brasil cresceu 10,5% no último ano e indicou uma projeção de crescimento de 10% do mercado de software em 2021 (ABES, 2020). Dessa forma, há um aumento na demanda por softwares, ao mesmo tempo que também aumentam a robustez e a complexidade dos sistemas (Agrawal *et al.*, 2020, Zielske & Held, 2021).

As metodologias de desenvolvimento de software são utilizadas para melhorar e agilizar o ciclo de produção de sistemas (Sommerville, 2011). Cada metodologia possui suas particularidades e devem ser aplicadas dependendo da necessidade existente em cada contexto. Com isso, é importante que as empresas escolham uma metodologia de desenvolvimento que melhor se adeque as necessidades do projeto, uma vez que a escolha de uma abordagem inadequada pode afetar diretamente no tempo de entrega, na qualidade e em outros aspectos importantes do produto (Pressman, 2011).

Em 1970 surgiu a primeira metodologia de desenvolvimento de software, que ficou conhecida como metodologia tradicional (Royce, 1970, Bell & Thayer, 1976). Com o passar do tempo, a abordagem tradicional começou a se mostrar ineficaz em alguns casos devido a sua rigidez, dificultando a adaptação as constantes mudanças que os projetos sofriam. A partir disso fez-se necessário a criação de uma nova forma de gerenciar projetos de *software* que recebeu o nome de metodologia ágil (Sungkur & Ramasawmy, 2014). Os métodos ágeis têm como características principais possuem uma abordagem iterativa de desenvolvimento de software, cujo foco é a realização de entregas constantes de um sistema funcional, que permita evoluções rápidas e que atendam os requisitos solicitados pelo cliente, mesmo quando estes sofrem constantes mudanças (Kavitha & Ahmed, 2015).

Algumas empresas que adotam metodologias tradicionais e desejam aplicar metodologias ágeis partem para a utilização das chamadas metodologias híbridas (Hoda & Noble, 2017). As metodologias híbridas são aquelas que combinam elementos e práticas das metodologias tradicionais e ágeis, podendo apresentar diferentes configurações e customizações a depender da necessidade da organização que a adote (Kuhmann *et al.*, 2017). Essa mistura de metodologias pode ser benéfica para organizações que, por algum motivo, precisam continuar utilizando o método tradicional, mas que desejam experimentar e se beneficiar de técnicas ágeis (Theocharis *et al.*, 2015, Cram & Marabelli, 2017).

Ao identificar e aplicar a metodologia de desenvolvimento mais adequada ao cenário do projeto, a empresa acabará diminuindo custos de produção, tornando o processo de desenvolvimento mais fluído e produtivo (Ramos *et al.*, 2020). Além disso, o produto final será entregue de forma mais rápida e com maior valor, consequentemente aumentando o nível de satisfação do cliente (Pressman, 2011). Por isso, além de entender a natureza da empresa e do projeto, é necessário ter uma maior compreensão de como a adoção destas metodologias pode impactar no sucesso do seu projeto. Considerando que estamos em um cenário onde os projetos cada vez mais adotam metodologias ágeis ou pelo menos algumas de suas técnicas, configurando assim a adoção de metodologias híbridas (Version One, 2020), este trabalho pretende responder a seguinte pergunta de pesquisa: ao comparar as metodologias ágil e híbrida, qual apresenta o melhor resultado percebido em relação aos indicadores de projeto?

2 Referencial Teórico

Sommerville (2011) afirma que a Engenharia de Software é uma área da computação voltada para a especificação, projeto e evolução dos sistemas. Essa área de estudo define metodologias de desenvolvimento de software que consistem em padronizar o processo de desenvolvimento, alinhar as fases de ciclo de vida do projeto e promover uma relação equilibrada entre os processos e dados (Dennis & Wixom, 2011).

A palavra método deriva do latim *methodus* e tem como significado caminho definido para atingir algo (Rangel, 2005). A partir do método, surgiram as metodologias. Uma metodologia pode ser definida como um estudo sobre o método mais eficaz a ser utilizado para alcançar determinado objetivo (Cajueiro, 2015). Os primeiros sistemas desenvolvidos em sua maioria não utilizavam uma metodologia de desenvolvimento de software, fazendo com que o processo de implementação ficasse restrito apenas a codificação e correção dos erros que eram identificados (Turban *et al.*, 2008). A falta de um processo formal acabou trazendo vários problemas como alto custo de desenvolvimento e produção de softwares com um número alto de erros e pouco manuteníveis (Boehm, 1988). Assim, já na década de 1950, houve tentativas de estruturar o desenvolvimento de software com base em etapas (Benington, 1983).

As primeiras metodologias de gerenciamento de projetos surgiram nos anos 60 com a realização dos projetos de desenvolvimento de software da NASA (Larman & Basili, 2003). Em 1965, foi criado o International Project Management Association (IPMA), primeira associação mundial de gerenciamento de projetos. Em 1969, com a publicação do PMBOK, surgiu o Project Management Institute (PMI), instituição internacional que reúne profissionais de gestão de projetos. O PMBOK é um livro que apresenta as melhores práticas de gerenciamento de projetos e que se tornou um referencial utilizado por muitas organizações (Seymour & Hussein, 2014).

No fim da década de 70, as empresas precisaram modificar a forma de desenvolver seus produtos de forma padronizada. Desde então, foram surgindo várias metodologias de desenvolvimento de software (Pressman, 2011). As metodologias tradicionais são baseadas em requisitos estáticos que trabalhados por etapas; desta forma, são tratados com o cliente no início do projeto, sendo desenvolvidos e testados em seguida (Yang *et al.*, 2009). O artigo de Royce (1970) foi o primeiro a descrever um ciclo de desenvolvimento de software que funcionava em sequência. Os projetos que utilizam esse modelo devem ter desde o início seu escopo, prazo e custos bem definidos, já que mudanças não são bem-vindas no cenário dessa metodologia, pois qualquer alteração no escopo acaba gerando um grande impacto nos prazos e custos definidos inicialmente (Nerur *et al.*, 2005).

Em resposta às limitações da metodologia Waterfall, nos anos 2000 surgiram as metodologias ágeis de desenvolvimento de software, um paradigma “leve” de desenvolvimento, baseado em desenvolvimento iterativo, que se opõe a limitações de metodologias de desenvolvimento tradicionais (Sommerville, 2011; Sungkur & Ramasawmy, 2014). A abordagem ágil é uma forma de trabalho que possibilita entregas rápidas e constantes, realizadas por equipes auto-gerenciadas, com trocas frequentes de funções e que mantém um relacionamento próximo com o cliente (Beck *et al.*, 2001). O objetivo das metodologias ágeis é entregar maior produtividade, qualidade e projetos de sucesso, ao promover planejamento proativo e respostas rápidas e flexíveis (Kavitha & Ahmed, 2015). As metodologias ágeis têm alcançado notável sucesso, sendo adotado por empresas de todas as indústrias e tamanhos, em times locais ou remotos (Version One, 2020).

Tanto a metodologia tradicional como a ágil apresentam pontos fortes e fracos, fazendo com que muitos projetos apliquem práticas ágeis e tradicionais de forma conjunta, utilizando assim uma abordagem denominada híbrida (Barlow et al., 2012). O desenvolvimento híbrido pode ser considerado como uma evolução do desenvolvimento tradicional até chegar ao ágil (Hoda & Noble, 2017). Em alguns casos, os gestores relutam para utilizar a abordagem ágil, visto que os processos gerenciais não são tão ágeis (Kuhrmann *et al.*, 2017). Portanto, a abordagem híbrida acaba sendo uma boa opção pelo fato de combinar os processos tradicionais da gestão que já estão bem estabelecidos na organização e funcionam bem com as técnicas ágeis que podem melhorar o desempenho e os resultados das equipes (Theocharis *et al.*, 2015).

3 Metodologia

Este trabalho emprega um *survey* para a coleta de dados, um método que utiliza o questionário e é adequado para a coleta de dados de uma grande amostra de indivíduos (Hair *et al.*, 2005). A amostra deste estudo abrange pessoas de 18 a 60 anos, que trabalhem em projetos de desenvolvimento de software no Brasil e utilizam metodologia ágil ou híbrida.

O questionário auto-administrável foi criado com a ferramenta Google Forms®. O questionário é uma “técnica estruturada para coleta de dados que consiste em uma série de perguntas (...) que um entrevistado deve responder” (Malhotra, 2012, p. 290). O questionário desta pesquisa foi elaborado com base no relatório State of Agile da Version One (2020). Ele foi distribuído via redes sociais e aplicativo de troca de mensagens. Ao todo, foram respondidos 111 questionários.

Foram coletadas as percepções dos profissionais de TI em relação ao projeto em que está trabalhando atualmente nos seguintes pontos: produtividade da equipe, qualidade do código, tempo de entrega do software, troca de conhecimento entre a equipe, adaptação do time ao uso da metodologia, gerenciamento de prioridades, alinhamento da TI com os negócios, previsibilidade da entrega, gerenciamento do time e manutenibilidade do software. Os itens foram medidos com escalas Likert de 5 pontos. Os dados foram coletados no período de 14 de setembro a 01 de novembro de 2021.

O método empregado nesta pesquisa é o de comparação de médias. Ao término da coleta, os questionários foram avaliados quanto à limpeza dos dados, onde os questionários que estão sem um preenchimento adequado ou com itens faltantes são eliminados, seguindo as recomendações de Hair *et al.* (2009). Em seguida, são realizados os testes de Komolgorov-Smirnov e Shapiro-Wilk, a fim de determinar se os testes adotados serão paramétricos ou não-paramétricos (Hair *et al.*, 2009). Finalmente, é aplicado o teste de Mann-Whitney de diferenças de médias (Field, 2009), a fim de comparar as respostas daqueles que fazem partes de projetos que utilizam abordagens ágil ou híbrida. Os procedimentos foram realizados com o auxílio do software SPSS® (*Statistical Package for Social Sciences*).

4 Análise dos Resultados

Inicialmente, foi feita uma limpeza nos dados para eliminar respostas que não tiveram preenchimento adequado. Foram respondidos 102 questionários válidos. A maioria dos respondentes é do estado de Pernambuco, sendo que a amostra abrange também Rio Grande do Sul, São Paulo, Paraná, Distrito Federal, Paraíba, Santa Catarina, Ceará, Goiás, Rio de Janeiro e Espírito Santo. A maioria parte da amostra possui cargo técnico, como desenvolvedor ou analista de sistemas, contudo a pesquisa também contou com participantes de cargos gerenciais. A maioria dos participantes trabalham em empresas do segmento tecnológico, mas houve

respostas diversificadas, com respondentes que trabalham na área da saúde, agricultura, beleza, serviços, entre outras.

As outras características sociodemográficas estão sumarizadas na Tabela 1.

Tabela 1 – Características sociodemográficas da amostra

Gênero	Masculino – 87,3%
	Feminino – 11,8%
	Não-binário – 1%
Idade	De 18 a 24 anos - 36,3%
	De 25 a 35 anos - 45,1%
	De 36 a 50 anos - 14,7%
	De 51 a 60 anos - 3,9%
Escolaridade	Ensino Médio – 5,9%
	Técnico – 5,9%
	Graduação – 60,8%
	Pós-graduação <i>latu sensu</i> – 14,7%
	Mestrado – 9,8%
	Doutorado – 2,9%
Experiência	Júnior - 30.9%
	Pleno - 30.9%
	Sênior - 38.2%
Porte da empresa (nro. funcionários)	Até 9 funcionários – 3,9%
	10 a 49 funcionários – 10,8%
	50 a 99 funcionários – 8,8%
	Acima de 99 funcionários – 76,5%
Metodologia	Ágil – 72,5%
	Híbrida – 27,5%

A análise inicia-se com os procedimentos de estatística descritiva dos itens de produtividade da equipe, qualidade do código, tempo de entrega do software, troca de conhecimento entre a equipe, adaptação do time ao uso da metodologia, gerenciamento de prioridades, alinhamento da TI com os negócios, previsibilidade da entrega, gerenciamento do time e manutenibilidade do software. As estatísticas descritivas são mostradas na Tabela 2.

Tabela 2 – Estatística descritiva dos dados

	Média	Desvio padrão	Quartis		
			25%	50% (Mediana)	75%
Produtividade	3,88	,871	3,00	4,00	5,00
Qualidade código	3,65	,740	3,00	4,00	4,00
Tempo entrega	3,37	,867	3,00	3,00	4,00
Compartilhamento	3,73	,956	3,00	4,00	5,00
Adaptação metod.	3,69	,890	3,00	4,00	4,00
Gerenc. Prioridades	3,44	1,030	3,00	3,00	4,00
Alinhamento	3,50	,993	3,00	4,00	4,00
Previsibilidade	3,19	1,012	2,00	3,00	4,00
Gerenc. Time	3,71	,950	3,00	4,00	4,00
Manutenabilidade	3,58	,949	3,00	3,00	4,00

Em seguida, foram realizados os testes de Kolmogorov-Smirnov e Shapiro-Wilk para determinar a normalidade das variáveis. Em todos os testes, $p < 0.001$, de forma que as variáveis não são normais. Assim, os próximos testes aplicados serão não paramétricos. A Tabela 3 mostra os resultados dos testes de normalidade dos dados.

Tabela 3 – Normalidade dos dados

	Kolmogorov-Smirnov			Shapiro-Wilk		
	Est.	gl	Sig.	Est.	gl	Sig.
Produtividade	,211	102	,000	,858	102	,000
Qualidade código	,280	102	,000	,817	102	,000
Tempo entrega	,245	102	,000	,884	102	,000
Compartilhamento	,217	102	,000	,873	102	,000
Adaptação metod.	,241	102	,000	,862	102	,000
Gerenc. Prioridades	,244	102	,000	,881	102	,000
Alinhamento	,212	102	,000	,898	102	,000
Previsibilidade	,191	102	,000	,910	102	,000
Gerenc. Time	,203	102	,000	,881	102	,000
Manutenabilidade	,249	102	,000	,875	102	,000

O próximo passo é a aplicação de um teste de comparação de médias independente, portanto prossegue-se com a aplicação do teste de Mann-Whitney, seguindo os procedimentos indicados por Field (2009). A Tabela 4 e a Tabela 5 mostram os resultados dos testes de Mann-Whitney para as variáveis analisadas.

Tabela 4 – Mann-Whitney: Estatísticas de Teste

	Mann-Whitney U	Wilcoxon W	Z	Sig. Assint.
produtividade	939	1345	-0,76758	0,442736
qualidadeCodigo	787	1193	-2,03063	0,042293
tempoEntrega	964,5	1370,5	-0,5724	0,567051
compConhec	912,5	1318,5	-0,97218	0,330962
adaptacaoUso	754	1160	-2,2345	0,02545
gerencPrioridades	860	1266	-1,38214	0,166929
alinhamentoNegTI	894,5	1300,5	-1,1124	0,265968
previsibilidade	934,5	1340,5	-0,79422	0,427066
gerenciamentoTime	871,5	1277,5	-1,29368	0,195775
manutenabilidade	837,5	1243,5	-1,5733	0,11565

Tabela 5 – Mann-Whitney: Ordenação dos Postos

	metodologia	N	Média dos postos	Soma dos postos
Produtividade	Ágil	74	52,81	3908,00
	Híbrida	28	48,04	1345,00
Qualidade Código	Ágil	74	54,86	4060,00
	Híbrida	28	42,61	1193,00
Tempo Entrega	Ágil	74	52,47	3882,50
	Híbrida	28	48,95	1370,50
Compartilhamento	Ágil	74	53,17	3934,50
	Híbrida	28	47,09	1318,50
Adaptação metod.	Ágil	74	55,31	4093,00
	Híbrida	28	41,43	1160,00
Gerenc. Prioridades	Ágil	74	53,88	3987,00
	Híbrida	28	45,21	1266,00
Alinhamento	Ágil	74	53,41	3952,50
	Híbrida	28	46,45	1300,50
Previsibilidade	Ágil	74	52,87	3912,50
	Híbrida	28	47,88	1340,50
Gerenc. Time	Ágil	74	53,72	3975,50
	Híbrida	28	45,63	1277,50
Manutenabilidade	Ágil	74	54,18	4009,50
	Híbrida	28	44,41	1243,50

Ao analisar a Tabela 4, notamos que quase todas as variáveis não apresentam diferenças estatísticas, com exceção da qualidade do código ($U = 727,000$; $p = 0,042293$) e da adaptação do uso da metodologia ($U = 754,000$; $p = 0,02545$). Na Tabela 5, pode-se verificar que para a qualidade do código, a metodologia ágil (média dos postos = 54,86) apresenta médias dos postos maior que a da metodologia tradicional (média dos postos = 42,61). O mesmo ocorre com a adaptação do time ao uso da metodologia, onde a metodologia ágil (média dos postos = 55,31) também apresenta médias dos postos maior que a da metodologia tradicional (média dos postos = 41,43). Ou seja, pode-se inferir que a qualidade percebida do código e a adaptação do time ao uso da metodologia é maior no uso de metodologias ágeis do que no uso de metodologias híbridas.

Pode-se dizer que o resultado da variável adaptação do time ao uso da metodologia era esperado, visto que o grupo ágil usa apenas uma abordagem enquanto a híbrida combina práticas das abordagens ágil e tradicional, fazendo com que possa surgir conflitos ao combinar as regras dessas duas abordagens (Barlow *et al.*, 2012). Algumas das novas técnicas ágeis podem ser difíceis de combinar com aquelas tradicionais que já eram utilizadas, fazendo com que a combinação fique mais complexa do que se houvesse somente uma metodologia sendo aplicada (Hoda & Noble, 2017).

O resultado da variável qualidade do código está de acordo com a literatura, visto que muitos autores elencam esse aspecto como uma das principais vantagens da metodologia ágil. Inclusive essa característica é mencionada em um dos princípios ágeis que diz que a satisfação do cliente deve ser obtida com a entrega de um software com qualidade (Beck *et al.*, 2001).

No entanto, é interessante notar que os projetos apresentam resultados similares para a grande maioria dos resultados percebidos pelos membros dos projetos. Esses resultados são especialmente importantes para os projetos híbridos. Se os líderes destes projetos forem capazes de se adaptar a ponto de absorver as dificuldades das diferenças das metodologias e prestando máxima atenção na qualidade do código, é possível que os projetos híbridos cheguem em um equilíbrio. Os líderes devem tomar atitudes para mitigar potenciais efeitos negativos nos projetos híbridos: aplicando, por exemplo, treinamentos naquelas técnicas onde ainda existem dificuldades e incentivando ainda mais o compartilhamento do conhecimento dentro da equipe e fora dela; e utilizando técnicas do ágil que incentivem especialmente a qualidade do código, como a programação em pares (Sungkur & Ramasawmy, 2014). Assim, se possa realmente obter o melhor dos dois mundos, com projetos de software adaptados à realidade da empresa sem perder em qualidade (Theocharis *et al.*, 2015).

5 Considerações Finais

Esta pesquisa pretendia comparar as metodologias ágil e híbrida de desenvolvimento de *software*, investigando qual apresenta o melhor resultado percebido em relação aos indicadores de projeto. Para isso, foi realizado um *survey* com 102 integrantes de projetos de desenvolvimento de *software* no Brasil. Os dados foram analisados frente à metodologia utilizada no projeto do respondente, utilizando comparação de média.

Os resultados mostram que existem diferenças nas variáveis qualidade de código e adaptação do time ao uso da metodologia, sendo que em ambas o maior valor é atrelado à metodologia ágil. É interessante notar que nas variáveis produtividade da equipe, tempo de entrega do software, troca de conhecimento entre a equipe, gerenciamento de prioridades, alinhamento da TI com os negócios, previsibilidade da entrega, gerenciamento do time e manutenibilidade do software o resultado foi igual para metodologia ágil e metodologia híbrida.

Esse trabalho traz contribuições acadêmicas, com os resultados já citados e por mostrar que, em indicadores, as metodologias ágeis e híbridas estão bastante próximas. São também confirmados os resultados já indicados desde por princípios das metodologias ágeis por Beck *et al.* (2001) quanto à qualidade de código. Traz também contribuições gerenciais, ao mostrar que as equipes podem adotar a metodologia ágil ou híbrida, escolhendo a melhor dentro da realidade do projeto, pois os resultados são similares. Gerentes de projetos que utilizam metodologia híbrida devem estar atentos às dificuldades à adaptação do time à metodologia, que pode atrapalhar o andamento do projeto, e à qualidade do código.

Como limitação, aponta-se que a amostra não é representativa de todo o território nacional. Como sugestões de trabalhos futuros, sugere-se a expansão da coleta para mais estados do Brasil e em maior volume. Além disso, sugere-se uma coleta de dados longitudinal para comparação dos resultados ao longo do tempo. Finalmente, sugere-se uma pesquisa comparando os resultados de equipes tradicionais com ágeis e híbridas.

Referências

- Agrawal, A., Seh, A. H., Baz, A., Alhakami, H., Alhakami, W., Baz, M., ... & Khan, R. A. (2020). Software security estimation using the hybrid fuzzy ANP-TOPSIS approach: design tactics perspective. *Symmetry*, 12(4), 598.

- Barlow, J. B., Giboney, J., Keith, M. J., Wilson, D., Schuetzler, R. M., Lowry, P. B., & Vance, A. (2011). Overview and guidance on agile development in large organizations. *Communications of the Association for Information Systems*, 29(2), 25-44.
- Beck, K *et al.* (2001) Manifesto for Agile Software development. <http://agilemanifest.org>
- Bell, T. E., & Thayer, T. A. (1976, October). Software requirements: Are they really a problem?. In *Proceedings of the 2nd international conference on Software engineering* (pp. 61-68).
- Benington, H. D. (1983). Production of large computer programs. *Annals of the History of Computing*, 5(4), 350-361.
- Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61-72.
- Castells, M. (2005). *A sociedade em rede: do conhecimento à política*. Editora Record.
- Cajueiro, R. L. P. (2013). Manual para elaboração de trabalhos acadêmicos: guia prático do estudante. *Rio de Janeiro: Saraiva*.
- Cram, W. A., & Marabelli, M. (2018). Have your cake and eat it too? Simultaneously pursuing the knowledge-sharing benefits of agile and traditional development approaches. *Information & Management*, 55(3), 322-339.
- Dennis, A. & Wixon, H. B. (2011) *Análise e projeto de sistemas*. 2. ed. Rio de Janeiro: LTC.
- Field, A. (2009). *Descobrimo a estatística usando o SPSS*. Penso Editora.
- Hair, J., Babin, B., Money, A., & Samouel, P. (2005). *Fundamentos de métodos de pesquisa em administração*. Bookman Companhia Ed.
- Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2009). *Análise multivariada de dados*. Bookman editora.
- Hoda, R., & Noble, J. (2017, May). Becoming agile: a grounded theory of agile transitions in practice. In 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE) (pp. 141-151). IEEE.
- Kavitha, R. K., & Ahmed, M. I. (2015). Knowledge sharing through pair programming in learning environments: An empirical study. *Education and Information Technologies*, 20(2), 319-333.
- Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., ... & Prause, C. R. (2017, July). Hybrid software and system development in practice: waterfall, scrum, and beyond. In *Proceedings of the 2017 International Conference on Software and System Process* (pp. 30-39).

- Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, 36(6), 47-56.
- Lévy, P. (2010). *Cibercultura*. Editora 34.
- Malhotra, N. K. (2001). *Pesquisa de Marketing: uma orientação aplicada*. Bookman Editora.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.
- Pressman, Roger S. (2011) *Engenharia de Software, Sétima Edição*. Porto Alegre: Editora McGrawHill.
- Ramos, G. R., de Assis Silva, W., Seleme, R., Marques, M. A. M., & Gressler, F. (2020). Comparação entre metodologias de gerenciamento de projetos ágil e tradicional: Um estudo de caso em uma Startup de pequena escala. *Brazilian Journal of Development*, 6(3), 15025-15047.
- Rangel, M. (2014). *Métodos de ensino para a aprendizagem e a dinamização das aulas*. Papirus Editora.
- Royce, W. W. (1970, August). Managing the development of large systems: Concepts and techniques. In *9th International Conference on Software Engineering*. ACM (pp. 328-38).
- Seymour, T., & Hussein, S. (2014). The history of project management. *International Journal of Management & Information Systems (IJMIS)*, 18(4), 233-240.
- Sommerville, Ian. (2011). *Engenharia de software*. 9. ed. São Paulo: Pearson Addison – Wesley.
- Sungkur, R. K., & Ramasawmy, M. (2014). Knowledge4Scrum, a novel knowledge management tool for agile distributed teams. *Vine*. 44(3), 394-419.
- Theocharis, G., Kuhrmann, M., Münch, J., & Diebold, P. (2015, December). Is water-scrum-fall reality? on the use of agile and traditional development practices. In *International Conference on Product-Focused Software Process Improvement* (pp. 149-166). Springer, Cham.
- Turban, E., Leidner, D., Mclean, E., & Wetherbe, J. (2008). *INFORMATION TECHNOLOGY FOR MANAGEMENT, (With CD)*. John Wiley & Sons.
- Version One. (2020). *14th Annual State of Agile Survey*. <https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494>.
- Yang, H., Huff, S., & Strode, D. (2009). Leadership in software development: Comparing perceptions of agile and traditional project managers. AMCIS 2009 Proceedings, p. 184, 2009.



Zielske, M., & Held, T. (2021). Application of agile methods in traditional logistics companies and logistics startups: Results from a German Delphi Study. *Journal of Systems and Software*, 177, 110950.