



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

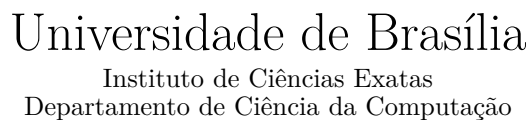
Características, desafios e práticas associados à integração de gestão de risco em projetos de software que empregam metodologias ágeis

André Larrosa Chimpliganond

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Daniel de Paula Porto

Brasília
2025



André Larrosa Chimpliganond

Brasília, 10 de julho de 2025

Dedicatória

Dedico esse trabalho à minha família, por todo os sacrifícios que fizeram por mim. Também aos meus colegas e professores, essenciais na minha formação profissional e pessoal.

Agradecimentos

Esse trabalho não poderia ter sido realizado sem o apoio e direcionamento do Professor Dr. Daniel de Paula Porto. Meus agradecimentos se estendem à toda Universidade de Brasília, pelo suporte e disposição em ajudar ao longo de toda minha formação.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Contexto: As metodologias ágeis são populares entre os desenvolvedores por permitirem flexibilidade e eficiência na produção de software. Gestão de risco segue, tradicionalmente, uma estrutura mais rigorosa e estratégias mais lentas de documentação e análise, características incompatíveis com desenvolvimento ágil. As diferenças entre metodologias ágeis e técnicas de gestão de risco criam situações particulares quando é preciso empregá-las juntas. Objetivo: Este trabalho apresenta uma revisão da literatura para entender como é realizada essa junção. Aborda-se sobre as características, os desafios e as práticas encontradas na gestão de risco de projetos que seguem as metodologias ágeis. Metodologia: Foi feita uma revisão da literatura segundo o modelo de Kitchenham. Foram analisados 34 estudos entre 2014 e 2024 extraídos das bases SciVerse Scopus, IEEE Xplore e ACM Digital Library. Resultados: Embora as metodologias ágeis tenham um sistema de gestão de risco inerente à sua estrutura, elas não cobrem todos os riscos. Assim, é necessário estabelecer uma rotina de gestão de risco explícita que siga a estrutura ágil. A adoção de uma estratégia de gestão de risco deve explicitar processos, funções e responsabilidades para a identificação e tratamento dos riscos, enquanto preserva o dinamismo de desenvolvimento ágil. Conclusões: O estudo contribui para uma visão abrangente do tema e propõe direções de pesquisas futuras.

Palavras-chave: Metodologia ágil, Scrum, XP, Extreme Programming, Kanban, Gestão de risco

Abstract

Context: Agile methodologies are popular among developers for enabling flexibility and efficiency in software production. Risk management, on the other hand, traditionally follows a more rigorous structure with slower strategies for documentation and analysis, characteristics often incompatible with agile development. The differences between agile methodologies and risk management techniques create unique challenges when both need to be applied together. **Objective:** This study presents a literature review aimed at understanding how the integration between agile development and risk management is carried out. It explores the characteristics, challenges, and practices involved in risk management within projects that follow agile methodologies. **Methodology:** A systematic literature review was conducted following Kitchenham's guidelines. A total of 34 studies published between 2014 and 2024 were analyzed, retrieved from the databases SciVerse Scopus, IEEE Xplore, and ACM Digital Library. **Results:** Although agile methodologies have an inherent risk management system within their structure, they do not address all types of risks. Therefore, it is necessary to establish an explicit risk management routine that aligns with the agile framework. Adopting a risk management strategy should clarify processes, roles, and responsibilities for risk identification and mitigation, while preserving the dynamism of agile development. **Conclusions:** This study contributes to a comprehensive understanding of the topic and proposes directions for future research.

Keywords: Agile methodology, Scrum, XP, Extreme Programming, Kanban, Risk management

Sumário

1	Introdução	1
1.1	Objetivos	2
1.2	Metodologia de pesquisa	3
1.3	Organização do trabalho	3
2	Fundamentação Teórica	4
2.1	Desenvolvimento de software	4
2.2	Gestão de projeto	5
2.2.1	Transição das metodologias tradicionais para as ágeis	5
2.3	Gestão de risco	6
2.3.1	Gestão de Risco em Desenvolvimento de Software	7
3	Protocolo da Revisão	9
4	Discussão	13
4.1	<i>Insights</i> gerais	13
4.2	Características, Desafios e Práticas	14
4.2.1	Características e Desafios	14
4.2.2	Práticas	15
4.3	Respondendo às perguntas de pesquisa	18
4.4	Ameaças à Validade	19
5	Conclusão	21
5.1	Contribuições do trabalho	21
5.2	Trabalhos futuros	22
	Referências	23
	Anexo	25
I	Resultados da revisão sistemática da literatura	26

Lista de Figuras

2.1	Exemplo de matriz de risco	7
-----	--------------------------------------	---

Lista de Tabelas

3.1	Artigos finais	11
I.1	Respostas dos artigos às questões de pesquisa	27

Capítulo 1

Introdução

O desenvolvimento de software deixou de ser um subproduto da matemática e da lógica, relevante apenas para um número restrito de profissionais e indústrias, para se tornar parte integrante das mais diversas áreas da atuação humana. Desde seu surgimento, o desenvolvimento de software tem experimentado um crescimento exponencial, tornando-se uma das principais fontes de inovação e crescimento econômico no mundo [1].

Inicialmente, a produção de software seguia os mesmos paradigmas e etapas dos projetos de engenharia [2]. No entanto, com o amadurecimento da disciplina, diversas metodologias para produção e manutenção de software foram desenvolvidas, incluindo as metodologias ágeis. Métodos como Scrum e XP surgiram como respostas às deficiências das abordagens tradicionais, priorizando a colaboração, a entrega de valor e a adaptabilidade, em vez de processos rígidos e documentação extensa. Desde o final do século XX, as metodologias ágeis têm ganhado cada vez mais popularidade, tornando-se as preferidas pelas equipes de desenvolvimento [3].

Assim como as tecnologias de desenvolvimento, as estratégias de gestão de projetos de software também evoluíram. Para garantir o sucesso de um projeto de desenvolvimento de software, é essencial uma gestão que inclua a definição de objetivos, a elaboração de um plano de ação, a alocação de recursos e o monitoramento contínuo do progresso [4]. Uma das etapas fundamentais da gestão de projetos é a gestão de risco, que envolve a identificação, avaliação e mitigação dos riscos que podem impactar o projeto [5]. Este processo é crucial, pois ameaças e oportunidades inevitavelmente surgirão ao longo do desenvolvimento. Com uma supervisão adequada, as ameaças que comprometem o trabalho e as oportunidades que impulsionam o projeto são tratadas de maneira eficaz. Caso contrário, oportunidades podem ser desperdiçadas e ameaças podem comprometer a execução do projeto dentro dos prazos e orçamentos planejados.

À medida que metodologias ágeis se tornam mais comuns, também surgem novas estratégias de gestão de risco adaptadas a essas abordagens. As técnicas tradicionais de

gestão de risco frequentemente envolvem atividades laboriosas, como a produção de documentação extensa, o que entra em conflito com os princípios de leveza das metodologias ágeis. Quais são, então, as particularidades da gestão de risco em metodologias ágeis? Quais desafios surgem ao integrar essas duas abordagens, com características tão distintas? Como os desenvolvedores ágeis conciliam a segurança proporcionada pela gestão de risco com a flexibilidade das metodologias? Equipes que adotam metodologias ágeis precisam entender como implementar a gestão de risco de forma eficaz, sem comprometer a agilidade ou expor o projeto a riscos excessivos. Este trabalho visa compreender, por meio de uma revisão sistemática da literatura, como a gestão de risco é implementada nas metodologias ágeis, oferecendo uma perspectiva abrangente sobre o tema para os engenheiros de software. Embora revisões de literatura possam incluir estudos quantitativos, este trabalho se concentrará em análises qualitativas.

1.1 Objetivos

Para termos uma visão geral da integração de gestão de risco em desenvolvimento de software ágil, esse trabalho realizou uma Revisão Sistemática da Literatura (RSL). Para isso foi utilizado o processo proposto por Kitchenham [6]. Com esse trabalho, busca-se responder as seguintes questões de pesquisa:

- RQ1: Quais são as características da integração de gestão de risco em projetos que empregam metodologias ágeis?
- RQ2: Quais são os desafios da integração de gestão de risco em projetos que empregam metodologias ágeis?
- RQ3: Quais são as práticas da integração de gestão de risco em projetos que empregam metodologias ágeis?

Para este trabalho, as definições de características, desafios e práticas são essenciais. Assim, é necessário especificá-las. Entende-se por *característica* todas as qualidades relevantes para a integração da gestão de risco nas metodologias ágeis. *Desafios* referem-se aos obstáculos encontrados ao tentar integrar as estratégias tradicionais de gestão de risco aos *frameworks* ágeis. Por fim, *práticas* são as abordagens propostas por profissionais ágeis e pela literatura para superar esses obstáculos e desenvolver métodos eficazes de gestão de riscos no contexto do desenvolvimento ágil, sejam elas ferramentas, técnicas, atividades ou outras soluções para o problema.

1.2 Metodologia de pesquisa

A Revisão Sistemática da Literatura (RSL) realizada neste trabalho segue a metodologia proposta por Kitchenham [6]. Essa abordagem foi escolhida por permitir a análise e a síntese das informações de um conjunto significativo de estudos. A RSL adota um protocolo bem definido, o que garante a rigorosidade e a reprodutibilidade do trabalho. O objetivo dessa metodologia é eliminar vieses e fornecer uma visão abrangente do conhecimento existente. As etapas dessa metodologia são as seguintes:

1. Definição do problema: Identificar claramente o problema ou questão de pesquisa a ser abordada.
2. Criação das questões de pesquisa: Formular questões de pesquisa específicas e claras.
3. Definição dos critérios de inclusão: Estabelecer critérios para selecionar os estudos a serem incluídos na revisão.
4. Realização da busca: Executar uma busca sistemática em bases de dados e outras fontes relevantes.
5. Seleção dos estudos: Aplicar os critérios de inclusão para selecionar os estudos que atendem aos requisitos.
6. Extração de dados: Coletar e extrair dados relevantes dos estudos selecionados.
7. Análise dos dados: Realizar uma análise dos dados coletados.
8. Relato dos resultados: Apresentar os resultados da revisão de forma clara e concisa.

1.3 Organização do trabalho

O trabalho está organizado da seguinte forma: no Capítulo 2, será feita a contextualização. Os conceitos essenciais, que formam a base do conhecimento necessário para compreender a pesquisa, serão discutidos. No Capítulo 3, será abordado o protocolo de revisão da literatura, detalhando as etapas de execução da revisão. No Capítulo 4, serão apresentados e discutidos os resultados da revisão, juntamente com a análise das limitações do trabalho. Por fim, no Capítulo 5, será apresentada a conclusão, acompanhada de propostas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Desenvolvimento de software

Com a expansão da computação em diversos setores, surgiram estratégias de desenvolvimento fortemente influenciadas pelas metodologias utilizadas em projetos de engenharia. Essas estratégias seguiam estruturas sequenciais e rígidas, semelhantes à realidade dos projetos de engenharia, nos quais uma alteração nos requisitos no meio do processo de desenvolvimento é extremamente custosa. Com o amadurecimento da área, duas realidades começaram a se desenhar. Primeiro, as soluções de software passaram a ser ferramentas essenciais em praticamente todos os setores. Segundo, novos entendimentos sobre o processo de desenvolvimento começaram a surgir.

Em relação ao primeiro ponto, as tecnologias de software se popularizaram em áreas como saúde [7], finanças [8], comércio [9], entre outras. Com a expansão do uso dessas tecnologias, surgiu a necessidade de maior colaboração com o cliente e maior adaptabilidade. Os requisitos começaram a ser alterados ao longo do desenvolvimento para incorporar novos entendimentos sobre como o sistema deveria interagir com tecnologias existentes, como seria utilizado e quais funcionalidades deveria ter. A rigidez das metodologias tradicionais passou a ser questionada, e novas alternativas começaram a ser propostas.

Para superar as limitações das metodologias tradicionais, no final do século XX, surgiram novas abordagens para o desenvolvimento de software, com destaque para as metodologias ágeis. Essas metodologias são mais flexíveis e adaptáveis, baseando-se em princípios como a priorização de requisitos, a entrega contínua de valor e a colaboração entre as equipes. Caracterizam-se por uma abordagem iterativa e incremental, com foco na entrega de funcionalidades em ciclos curtos e na capacidade de responder rapidamente às mudanças e requisitos do projeto. As metodologias ágeis são amplamente adotadas em projetos de software que exigem alta flexibilidade e capacidade de adaptação, como projetos complexos ou que envolvem muitas mudanças [3].

2.2 Gestão de projeto

A gestão de projetos é um processo que envolve o planejamento, execução e controle de um projeto, visando alcançar seus objetivos [10]. Ela é fundamental em diversas áreas, incluindo o desenvolvimento de software. Essa disciplina ajuda a garantir que o projeto seja realizado de maneira eficiente, levando em consideração as restrições enfrentadas, sejam elas financeiras, técnicas ou relacionadas ao tempo disponível [10].

No contexto do desenvolvimento de software, a gestão de projetos é crucial para assegurar que o produto seja desenvolvido de forma eficiente. Ela envolve a definição de requisitos, o processo de programação, testes, entrega do software e sua manutenção ao longo de sua vida útil. A gestão de projetos garante que o software atenda às necessidades do cliente e entregue valor ao negócio [4]. Nesse contexto, as metodologias de gestão de projetos podem ser divididas [11] em duas categorias principais: metodologias tradicionais e metodologias ágeis.

2.2.1 Transição das metodologias tradicionais para as ágeis

A metodologia tradicional de desenvolvimento de software é uma abordagem baseada em uma série de etapas bem definidas, aplicadas de forma linear e sequencial [12]. Ela foi influenciada pela engenharia tradicional, onde os projetos são complexos, exigem uma abordagem estruturada e, geralmente, não acomodam bem mudanças nos requisitos durante o andamento do projeto.

Embora ofereça uma estrutura clara e definida, diversos problemas começaram a se tornar evidentes à medida que a indústria de software crescia e se tornava mais dinâmica. Sua estrutura rígida dificulta a adaptabilidade e a flexibilidade necessárias para lidar com alterações nos requisitos ao longo do projeto. Além disso, o alto custo e o tempo necessário para entregar valor ao cliente são grandes desafios. O software só é entregue ao final de todo o processo, o que aumenta a chance de insatisfação do cliente. À medida que o mercado se tornava mais dinâmico, a exigência de entregas rápidas de software se tornava cada vez mais comum.

É no contexto da dissonância entre as exigências do mercado e as estratégias tradicionais de produção de software que as metodologias ágeis surgem, momento marcado pela publicação do Manifesto Ágil em 2001 [13]. O Manifesto Ágil é composto por quatro valores fundamentais que orientam as diversas implementações das metodologias ágeis: Indivíduos e interações mais que processos e ferramentas, Software funcionando mais que documentação abrangente, Colaboração com o cliente mais que negociação de contratos e Responder a mudanças mais que seguir um plano. Esses valores buscam promover

flexibilidade, adaptação e colaboração nos projetos de desenvolvimento de software [3], atendendo, assim, às necessidades emergentes do mercado.

2.3 Gestão de risco

Parte da gestão de projetos é a gestão de riscos. Embora as práticas para entender e lidar com o risco sejam antigas, sua consolidação como disciplina científica é relativamente recente. A gestão de riscos, em sua perspectiva científica, pode ser definida como o processo de identificação, avaliação e resolução dos riscos de um empreendimento ou organização [5]. Para um melhor entendimento dessa disciplina, é importante esclarecer o que é risco, além de expor alguns princípios e etapas da gestão de riscos.

A definição de risco está intimamente relacionada ao contexto em que é aplicada. Por esse motivo, embora tentativas de unificar o entendimento sobre risco tenham sido feitas, ele permanece um elemento circunstancial. Uma tentativa de definição pode ser a de que risco são as consequências futuras indesejáveis de uma atividade [5]. Como o futuro não pode ser previsto, a incerteza se torna um componente essencial dessa definição. Uma outra interpretação possível de risco, na qual a incerteza está explicitada, é a relação entre a probabilidade de ocorrência do evento e seu impacto negativo [14]. Uma abordagem mais completa define o risco como puramente incerteza. Nesse caso, não há atribuição de valor, e o risco pode ser tanto negativo (ameaça) quanto positivo (oportunidade) [15]. Assim, parte da complexidade do fator risco, que se reflete na gestão de risco, está na dificuldade de identificar e tratar ameaças a um projeto, por exemplo, devido à incerteza quanto à natureza da ameaça ou às possíveis soluções.

O processo de gestão de risco, apesar de sua complexidade e variabilidade, pode ser dividido em algumas etapas padrão. Primeiramente, a identificação do risco. Em seguida, a priorização dos riscos. Depois, o estabelecimento de uma política de tratamento dos riscos. Por fim, o monitoramento e, quando necessário, o tratamento dos riscos.

A fase de identificação do risco consiste em analisar as diferentes áreas que compõem um empreendimento, como a financeira, tecnológica, organizacional, humana, entre outras. A partir dessa visão macro, realiza-se uma investigação mais detalhada para reconhecer, dentro de cada área, quais fatores representam um risco para o projeto.

A próxima etapa envolve, entre os riscos identificados, avaliá-los quanto ao seu potencial e à probabilidade de ocorrência. O potencial de um risco pode ser avaliado de acordo com o impacto negativo ou positivo que ele teria na conclusão bem-sucedida do projeto. A probabilidade de ocorrência, por sua vez, é avaliada intuitivamente, variando de *pouco provável* a *muito provável* [16]. A matriz de risco apresentada na Figura 2.1 ilustra esse conceito no caso de uma visão de risco como exclusivamente negativo.

		Potencial nocivo				
		Muito baixa	Baixa	Média	Alta	Muito alta
Probabilidade de ocorrência	Muito alta					
	Alta					
	Média					
	Baixa					
	Muito baixa					

Figura 2.1: Exemplo de matriz de risco

Após a classificação dos riscos, que permite sua priorização, é hora de estabelecer políticas de contingência e tratamento. Nessa etapa, as informações coletadas e processadas nas fases anteriores são compiladas e transformadas em ações e estratégias para lidar com os riscos identificados. Riscos considerados de menor prioridade (como aqueles com baixo potencial e probabilidade média de ocorrência, por exemplo) podem ser tratados apenas se ocorrerem. Já riscos de maior prioridade (como aqueles com alto potencial e alta probabilidade de ocorrência) exigem uma política mais rígida de monitoramento e práticas de prevenção ou aproveitamento [17].

Como um projeto está sempre suscetível a ameaças e oportunidades durante seu desenvolvimento, a gestão de risco também envolve o estabelecimento de um sistema de monitoramento contínuo. Esse sistema permite a detecção de riscos e sua reavaliação conforme as circunstâncias mudam (riscos antes pouco prováveis podem se tornar mais prováveis à medida que o projeto avança, novas informações surgem e fatores que influenciam o projeto mudam) [17]. Além disso, à medida que esses riscos se concretizam, as estratégias de tratamento devem ser postas em prática.

2.3.1 Gestão de Risco em Desenvolvimento de Software

A gestão de riscos é especialmente relevante em projetos de software, dada a complexidade e os riscos associados ao seu desenvolvimento. Modelos de gestão de risco específicos para o desenvolvimento de software foram propostos, como o sugerido pelo Instituto de Engenharia de Software ou por Roy, G [18], e podem ser resumidos nas etapas: identificação, análise, planejamento, monitoramento e controle [19].

As etapas de identificação a monitoramento seguem uma lógica semelhante à descrita anteriormente, com adequações específicas para o contexto de software. Por exemplo, na etapa de análise, é crucial estudar o impacto de determinados riscos na performance de um sistema, já que os requisitos de desempenho são extremamente comuns em produtos de software. A etapa de controle, por sua vez, é uma extensão da etapa de monitoramento. Enquanto o monitoramento consiste no acompanhamento do status dos riscos identificados

e das ações para mitigá-los, o controle envolve a adaptabilidade do plano de gestão de risco às mudanças no projeto, respostas a gatilhos e melhorias nos processos de gestão.

Em metodologias tradicionais, a gestão de risco é realizada de maneira bem estruturada. Como os demais processos no desenvolvimento tradicional, a gestão de risco é formalmente integrada ao projeto. Dessa forma, existem diretrizes claras a serem seguidas, e os desenvolvedores têm plena consciência do processo de gestão de risco. No entanto, por ser parte integrante da estrutura tradicional, ela também sofre das mesmas limitações quanto à falta de flexibilidade e adaptabilidade.

Por outro lado, nas metodologias ágeis, o processo de gestão de risco é implícito. Como essas metodologias preveem mudanças ao longo do desenvolvimento, os riscos do projeto são tratados de maneira tácita. À medida que são percebidos, os riscos entram no pipeline de atividades como qualquer outra mudança de requisito. Assim, a gestão de risco é, em grande parte, inerente às atividades ágeis. No entanto, a falta de formalização do processo de gestão de risco pode gerar problemas. Seja pela ausência de uma distribuição clara das responsabilidades no tratamento de riscos, o que pode resultar na falta de iniciativa da equipe para lidar com eles, ou pela má apreciação dos riscos, tratá-los como se fossem simples mudanças de requisito.

Capítulo 3

Protocolo da Revisão

Seguindo o processo definido por Kitchenham, com o problema e perguntas de pesquisa já definidas na Seção 1.1, foram estabelecidos os critérios de inclusão (I) e exclusão (E):

- (I) Artigo aborda características da integração de gestão de risco em projetos que empregam metodologias ágeis.
- (I) Artigo aborda desafios da integração de gestão de risco em projetos que empregam metodologias ágeis.
- (I) Artigo aborda práticas da integração de gestão de risco em projetos que empregam metodologias ágeis.
- (E) Artigos não aborda características, desafios ou práticas da integração de gestão de risco em projetos que empregam metodologias ágeis.
- (E) Artigo não está em inglês.
- (E) Artigo não foi revisado por pares.

Em seguida, três bancos de dados: IEEE Xplore, ACM Digital Library e SciVerse Scopus. Foram selecionados devido à sua relevância e acessibilidade. As strings de busca foram desenvolvidas de forma iterativa, com o objetivo de identificar as melhores combinações que retornassem artigos relevantes. Por fim, adotou-se uma estrutura de string composta por quatro conjuntos de expressões: Expressões relacionadas à *gestão de risco*. Expressões que abordam *metodologias ágeis*. Expressões voltadas para *características, desafios* e *práticas*. Por último, expressões para a exclusão de artigos não relacionados, identificadas durante as iterações anteriores. Como o SciVerse Scopus indexa uma quantidade muito maior de artigos, a string de busca utilizada nesse banco de dados foi mais específica, incluindo um maior número de termos de exclusão, em comparação com as utilizadas no IEEE Xplore e na ACM Digital Library.

String de busca em IEEE Xplore e ACM Digital Library:

```
"risk management"OR "risk mitigation"OR "risk  
assessment"OR "risk analysis"  
AND  
agile OR Scrum OR kanban OR xp OR "extreme  
programming"  
AND  
characteristic* OR challenge* OR practice*  
AND NOT  
(school OR education OR health* OR medic* OR  
sustainab*)
```

String de busca SciVerse Scopus:

```
"risk management"OR "risk mitigation"OR "risk  
assessment OR "risk analysis"  
AND  
agile OR Scrum OR kanban OR xp OR "extreme  
programming"  
AND  
characteristic* OR challenge* OR practice*  
AND NOT  
( school OR education OR health* OR medic*  
OR sustainab* OR crypto* OR ml OR "machine  
learning"OR "supply chain"OR sales OR mining )
```

Além da string de busca, para garantir os critérios de seleção e a relevância dos trabalhos, foram escolhidos artigos em inglês, publicados entre 2014 e 2024, e provenientes de revistas científicas que adotam o protocolo de revisão por pares. A pesquisa nas bases de dados retornou: 76 artigos do IEEE Xplore, 179 artigos da ACM Digital Library e 489 artigos do SciVerse Scopus. A ferramenta StArt [20] foi utilizada para gerenciar a revisão.

Os artigos selecionados passaram por duas etapas de triagem. Na primeira, foram analisados os títulos, resumos (*abstract*) e palavras-chave (*keywords*), com base nos critérios de inclusão e exclusão. O estudo foi aceito caso pelo menos um critério de inclusão fosse atendido e nenhum critério de exclusão fosse identificado. O estudo foi rejeitado se ao menos um critério de exclusão fosse verificado. Essa seleção inicial resultou em 53 artigos, que seguiram para a próxima etapa de validação. Como a seleção de artigos revisados por

pares e em inglês foi realizada por meio dos filtros dos próprios bancos de dados, todos os artigos recusados nesta etapa não continham informação relevantes para o estudo.

A segunda fase consistiu na leitura integral dos artigos. As informações de cada artigo foram coletadas, desde que o conteúdo fosse diretamente relacionado ao objeto de pesquisa deste trabalho. Nessa etapa, também foi realizada uma análise qualitativa com base nas respostas fornecidas pelos estudos às questões de pesquisa. Ao final, essa seleção resultou nos 34 artigos finais.

A Tabela 3 apresenta os artigos finais da revisão. O resultado completo da revisão pode ser encontrado na Tabela I.1 em anexo.

Tabela 3.1: Artigos finais

ID	Título do Artigo
1	O A framework for risk management in Scrum development process
2	Agile risk management for multi-cloud software development
3	A Multi-agent Reinforcement Learning Risk Management Model for Distributed Agile Software Projects
4	An alternative approach for risk assessment in Scrum
5	An Empirical Evaluation of a Model for dealing with Epistemic Uncertainty in Agile Software Project Management
6	A novel risk management model in the Scrum and extreme programming hybrid methodology
7	Architectural Risk Analysis in Agile Development of Cloud Software
8	A Review of Risk Management in Agile Development
9	A Review on Risk Management Framework for large scale Scrum
10	A risk management framework for distributed Scrum using prince2 methodology
11	A Risk Management Framework for Scrum Projects
12	A risk poker based testing model for Scrum
13	Artificial Intelligence based Risk Management Framework for Distributed Agile Software Development
14	Assessing the Risk of Software Development in Agile Methodologies Using Simulation
15	A Survey of Agile vs. Traditional Methods on Project Risk Management
16	A survey on product backlog change management and requirement traceability in agile (Scrum)
17	Best practices for managing risk in adaptive agile process
18	Categorization of risk factors for distributed agile projects

ID	Título do Artigo
19	Combining agile approaches and risk management for mechatronic product development - A case study
20	Continuous Requirements Risk Profiling in Information Systems Development
21	Development of Scrum-Tree-KNN Algorithm for Distributed Agile Development
22	Explaining software project risks in agile teams: an action case using causal mapping
23	Improving the Organization Technology competitiveness through Effective Management of Conflict-Risk within Agile Teams
24	Integrating risk management in Scrum framework
25	Lightweight risk management: The development of agile risk tool agents
26	Mapping of Agile Processes into Project Management Knowledge Areas and Processes
27	Quantitative planning and risk management of Agile Software Development
28	Risk assessment forum: A proposal for agile software development teams ruled by Scrum
29	Risk management for Agile projects in offshore Vietnam
30	Risk management framework in Agile software development methodology
31	Suitability of PMBOK 6th edition for agile-developed IT Projects
32	Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities
33	User Story Risk Prioritization Model for Agile Software Development
34	Using velocity, acceleration, and jerk to manage agile schedule risk

Capítulo 4

Discussão

4.1 *Insights* gerais

Inicialmente, os trabalhos podem ser analisados em duas categorias: tipo de abordagem e definição de risco. Em relação à abordagem, temos aqueles que elaboram um modelo, *framework* ou estratégia para lidar com a gestão de risco em metodologias ágeis e aqueles que abordam o tema de uma perspectiva teórica, como revisões da literatura, entrevistas com praticantes e compreensões gerais sobre as metodologias e desafios da área. Quanto à definição de risco, é possível dividir os trabalhos em dois grupos: aqueles que consideram o risco como algo negativo e aqueles que o veem como incerteza, tanto positiva (oportunidade) quanto negativa (ameaça). Um ponto importante a ser destacado é que, independentemente da definição de risco, as estratégias de gestão são bastante similares. Algumas poucas diferenças podem surgir, mas os processos se desenvolvem de maneira bastante semelhante.

Os diversos modelos e *frameworks* seguem, em sua maioria, o mesmo paradigma: identificação, planejamento e monitoramento/tratamento. Todos propõem alguma forma de identificar o risco (podendo também classificá-lo e priorizá-lo), planejar uma estratégia para lidar com ele e monitorá-lo, implementando a estratégia de tratamento quando necessário. Alguns artigos abordam apenas uma parte do processo de gestão de risco, mas, mesmo assim, ainda seguem, em essência, o paradigma tradicional.

Uma questão recorrente é o entendimento de que as metodologias ágeis lidam bem com riscos devido à sua estrutura flexível. Contudo, a gestão de risco nessas metodologias não é formalizada. Dessa forma, os riscos são tratados de maneira informal, mas a falta de uma estratégia formal deixa lacunas nos projetos ágeis. Outro ponto comum é a escassez de ferramentas específicas para gestão de risco em metodologias ágeis. O que existe, na maioria das vezes, são adaptações de ferramentas e metodologias tradicionais de gestão de projetos.

Além disso, as diferenças fundamentais entre a gestão de risco tradicional e as metodologias ágeis fazem com que muitos praticantes ágeis não se interessem por essa atividade, já que ela envolve um processo excessivamente documental, o que não se alinha com a cultura ágil. Essa desconexão é, inclusive, abordada em alguns artigos.

4.2 Características, Desafios e Práticas

O processo de integração da gestão de risco em projetos de desenvolvimento de software que utilizam metodologias ágeis enfrenta a difícil tarefa de conciliar dois fatores contraditórios. De um lado, há atividades lentas, com alto nível de documentação, que, em sua maioria, foram planejadas com base em outro *framework* de trabalho. Do outro, existe uma estratégia de produção voltada para velocidade, flexibilidade e entrega constante de valor.

4.2.1 Características e Desafios

Dentro do processo ágil, a gestão de risco precisa considerar algumas peculiaridades, que podem ser resumidas em um princípio fundamental das metodologias ágeis: a flexibilidade. Flexibilidade, no sentido de que o processo é estruturado com a expectativa de que mudanças ocorrerão. É dessa consideração essencial que surgem características marcantes, como o formato iterativo das metodologias ágeis. Como as mudanças nos requisitos do sistema já são previstas, a sequência de identificação de requisitos e desenvolvimento do sistema se repete. Novos requisitos são incorporados ao trabalho em andamento, e modificações no software são feitas para acomodar essas alterações.

Dessa forma, o processo de gestão de risco tem que se encaixar nessa estrutura sequencial (assim como em metodologias tradicionais), mas que se repete (muito diferente de metodologias tradicionais) de maneira leve e adaptada aos processos de trabalhos dinâmicos. O adjetivo leve faz referência à necessidade de que as estratégias de gestão de risco não interfiram nos processos ágeis. Assim, rituais ágeis, como as reuniões diárias, não podem ser excessivamente prolongados para incluir atividades de gestão de risco.

Como a gestão de risco foi originalmente baseada nas metodologias tradicionais, a maioria dos modelos segue uma estrutura tradicional. No processo clássico de gestão de risco, a identificação dos fatores de risco ocorre primeiro, seguida pelo seu tratamento. Em um *framework* ágil, esse processo deve ser repetido para se alinhar com a forma iterativa da metodologia. Se o processo de identificação for muito custoso (ou seja, demorado), o problema é agravado, pois terá de ser realizado várias vezes. Por isso, a gestão de risco em ambientes ágeis deve, assim como as demais atividades, ser leve, de forma a não comprometer o andamento das *sprints*. Um dos maiores desafios de integrar uma

gestão explícita de riscos em projetos ágeis é gerenciá-los sem prejudicar as características fundamentais dos processos ágeis.

Outra complicação surge pelo fato de que as metodologias ágeis já antecipam mudanças. Isso traz benefícios, pois cria uma mentalidade de gestão de risco implícita nas equipes. Times ágeis têm plena consciência da alta probabilidade de mudanças ao longo do ciclo de vida do desenvolvimento do software. No entanto, essa gestão implícita de risco pode gerar pontos cegos, uma vez que não é formalmente estruturada. Como as alterações são naturalmente previstas, pode-se ter a falsa sensação de que todos os riscos estão sendo gerenciados junto aos requisitos. Na realidade, enquanto os riscos relacionados aos requisitos estão sendo tratados, outros não recebem a devida atenção. Portanto, a gestão de risco em metodologias ágeis precisa abordar explicitamente todas as facetas dos riscos, garantindo que todos sejam devidamente gerenciados.

Outra característica das metodologias ágeis é o foco nas pessoas e nas suas interações, o que resulta em equipes multidisciplinares, com ênfase na entrega de valor e na distribuição compartilhada de responsabilidades. No entanto, isso pode gerar um problema de falta de clareza quanto às responsabilidades pela gestão dos riscos, o que pode resultar em apatia na hora de resolvê-los. Para lidar com essa questão, a gestão de risco em metodologias ágeis deve esclarecer não apenas as atividades relacionadas ao gerenciamento de risco, mas também quem é responsável por cada uma delas.

4.2.2 Práticas

Para lidar com esses desafios, diversas estratégias são propostas, podendo ser agrupadas em algumas categorias. Primeiramente, aquelas que adaptam e implementam métodos e práticas de gestão de risco tradicionais nos *frameworks* ágeis, como o mapeamento das atividades de gestão de risco propostas no PRINCE2 nas etapas do Scrum, por exemplo. Em segundo lugar, aquelas que desenvolvem uma metodologia própria de gestão de risco para o desenvolvimento ágil, seja combinando Scrum e XP para tratar riscos ou utilizando agentes de IA para interpretar riscos.

Em *A Review of Risk Management in Agile Development*, os autores propõem a integração das atividades de gestão de risco do PRINCE2 no processo de desenvolvimento de software que segue o Scrum. O artigo observa que a gestão de risco existente no Scrum, e em muitos outros modelos ágeis, é implícita. Isso significa que não há atividades claras de gestão de risco, deixando esse trabalho para ser resolvido ao longo das *sprints* como parte dos demais requisitos. O problema gerado por essa gestão tácita é a tendência de subestimar riscos e deixá-los desassistidos. Portanto, os autores propõem a integração das atividades de gestão de risco do PRINCE2 como parte das *sprints*.

Outro trabalho que incorpora estratégias da gestão de risco tradicional em desenvolvimento ágil, especificamente no Scrum, é *A Risk Management Framework for Scrum Projects*. Os autores reconhecem as mesmas deficiências apontadas pelo artigo anterior: o Scrum não possui uma maneira estruturada de gerir riscos. Para superar esse problema, propõem uma estratégia em que o *Product Owner* (PO) assume as responsabilidades de um gestor de risco, gerenciando os requisitos e garantindo que o produto final entregue valor ao cliente. O estudo identifica tipos de riscos que o PO pode encontrar e propõe adaptações aos processos de gestão de risco do PMBOK para alinhá-los à metodologia Scrum. A execução dos processos de gestão de risco é iterativa. O planejamento de gestão de risco, a identificação de riscos, a análise de riscos, o planejamento das respostas a riscos, a implementação das respostas e o monitoramento de riscos são realizados em todas as *sprints*.

Outra abordagem encontrada nesta revisão da literatura é a criação de *frameworks* de gestão de risco que, desde sua concepção, são ágeis. *A novel risk management model in the Scrum and extreme programming hybrid methodology* reconhece que existem diferenças fundamentais entre gestão de risco tradicional e desenvolvimento ágil. Além disso, identifica características do Scrum e do XP que podem ser benéficas para a gestão de risco. O Scrum assume que alterações ocorrerão ao longo do projeto, enquanto o XP observa o projeto em termos de comunicação, simplicidade e *feedback*. Com isso, os autores propõem a combinação dessas metodologias para criar uma nova abordagem de gestão de risco. O processo macro dessa nova abordagem segue o *framework* do Scrum. As *sprints*, por sua vez, seguem os processos do XP. Os riscos são identificados e priorizados, e os riscos com maior prioridade entram no ciclo de iteração do Scrum, que se baseia no método XP. A estratégia de tratamento dos riscos dessa iteração é planejada por meio de discussões entre os membros da equipe.

Agile risk management for multi-cloud software development, assim como o trabalho anterior, reconhece que metodologias tradicionais de gestão de risco não são facilmente traduzidas para *frameworks* ágeis. O estudo enumera problemas importantes: ferramentas de gestão de risco em ágil não incentivam a colaboração, e há uma resistência cultural significativa nas equipes ágeis em adotar estratégias formais de manejo de risco. As equipes ágeis se apoiam fortemente na comunicação e colaboração, enquanto as atividades de controle de risco, por outro lado, têm características mais estruturadas e hierárquicas (o gestor de risco é o tomador de decisões). Essas divergências tornam a combinação pouco eficiente. Assim, os autores propõem uma nova abordagem, projetada especificamente para a cultura ágil. O sistema visa registrar os riscos em formato Kanban e automatizar os processos de risco. Os riscos são identificados e dispostos seguindo a metodologia Kanban. À medida que os riscos passam pelas etapas de gestão, automações são ativadas

para oferecer, por exemplo, sugestões aos *stakeholders* sobre os passos seguintes. Essas automações visam facilitar a gestão de risco, promovendo a cooperação de uma equipe multidisciplinar.

Alguns artigos trataram de aspectos específicos da gestão de risco. *An Empirical Evaluation of a Model for dealing with Epistemic Uncertainty in Agile Software Project Management* estuda como equipes ágeis lidam com incerteza epistêmica e sugere uma abordagem aprimorada. A incerteza epistêmica é um tipo de incerteza que surge da falta de conhecimento, dados ou informações sobre um fenômeno, sistema ou modelo [21]. Em equipes ágeis, essa incerteza é tratada por meio de julgamentos de especialistas, o que pode ser difícil de avaliar e, portanto, gera um risco associado. O artigo propõe uma maneira de gerenciar a incerteza, apresentando-a como uma etapa anterior (e mais abrangente) à gestão de risco. Os autores desenvolvem um modelo para identificar essas incertezas e aplicam a Teoria de Evidência Dempster-Shafer [22] e redes bayesianas [23] para quantificar as opiniões dos especialistas sobre as incertezas e as relações entre elas. Dessa forma, eles oferecem uma metodologia para lidar com as ambiguidades e imprecisões que surgem ao longo do projeto.

The Risk Poker Based Testing Model for Scrum também se preocupa com o fato de que metodologias ágeis dependem de especialistas, mas neste caso, para priorizar tarefas. O modelo propõe uma análise de custo das tarefas, com o objetivo de identificar os defeitos mais custosos e mais importantes o mais cedo possível. O modelo *Risk Poker* utiliza discussões entre os membros da equipe para medir a probabilidade do risco (fator de probabilidade). Em seguida, o *Product Owner* e os *stakeholders* discutem o custo do risco. Com os riscos priorizados, eles são decompostos em atividades. As atividades mais importantes devem ser testadas com maior rigor.

Práticas alternativas

Existem também alguns trabalhos que abordam a gestão de risco de maneira não convencional. Por isso, vale a pena dedicar um tempo para descrever com mais detalhes suas sugestões. Todos eles utilizam o conceito de automação na tomada de decisão. O princípio comum entre esses trabalhos é acelerar e facilitar o processo de decisão no âmbito da gestão de risco. Seja pelo uso de inteligência artificial ou por simulação de cenários, o objetivo é poupar o gestor das atividades mais trabalhosas do manejo de risco.

Artificial Intelligence based Risk Management Framework for Distributed Agile Software Development propõe o desenvolvimento de uma IA que, alimentada com dados relevantes, seja capaz de avaliar a situação de exposição ao risco de um projeto. Por meio das histórias de usuário e de um banco de dados de risco, o sistema consegue entender as dependências e brechas de risco. A IA libera o gerente do processo de gestão, permitindo

que ele se concentre na identificação de novos riscos, ou seja, na atualização do banco de dados de risco da IA.

Assessing the Risk of Software Development in Agile Methodologies Using Simulation, por sua vez, propõe a integração de um simulador Monte Carlo [24] ao software de gerenciamento de atividades de um projeto, como o JIRA [25]. O objetivo é auxiliar na tomada de decisões, utilizando dados probabilísticos. O simulador Monte Carlo gera vários cenários randômicos com base nas histórias de usuários, habilidades da equipe e atividades do projeto. Realizando múltiplas simulações, ele prevê custo, duração e outros fatores relevantes para o projeto. Ao conectar o simulador com o JIRA, vários cenários podem ser administrados, apoiando assim a tomada de decisões.

Development of Scrum-Tree-KNN Algorithm for Distributed Agile Development aborda uma maneira de estimar o custo do projeto e, por consequência, prevenir riscos relacionados a estimativas incorretas. Para isso, sugere a estruturação do projeto em forma de árvore. O nó raiz é o projeto, os ramos são os módulos ou funcionalidades do sistema, e as folhas representam as tarefas necessárias para desenvolver esses módulos. O sistema busca alocar de maneira mais eficiente os membros da equipe aos itens de trabalho. Para isso, utiliza o algoritmo de aprendizado de máquina KNN [26]. A partir das similaridades entre tarefas atuais e anteriores, o algoritmo determina o desenvolvedor mais apto para resolver aquela demanda. Por meio dessa alocação eficiente, o sistema consegue calcular o custo do projeto, reduzindo sua exposição a riscos relacionados ao custo.

Lightweight risk management: The development of agile risk tool agents se baseia no desenvolvimento de agentes para automatizar o processo de identificação, avaliação e gestão de riscos. Na fase inicial, coletam-se informações sobre os requisitos do sistema, as tarefas a serem realizadas, a equipe de desenvolvimento e os riscos. Esses riscos são definidos por gatilhos e consequências. Na fase de processo, o sistema funciona de maneira iterativa, com os agentes monitorando os riscos. A fase final gera um Registro de Riscos.

4.3 Respondendo às perguntas de pesquisa

Considerando as observações anteriores, respondemos agora as perguntas de pesquisa:

RQ1: Quais são as características da integração de gestão de risco em projetos que empregam metodologias ágeis?

- Metodologias ágeis, por se anteciparem a mudanças ao longo do projeto, já lidam, parcialmente, com os riscos. Tem, portanto, uma gestão de risco implícita e inerente.
- O processo de gestão de risco é tradicionalmente pesado, não se encaixando na estrutura ágil.
- Times ágeis são multidisciplinares. Logo, muitos não tem nem experiência nem conhecimento sobre gestão de risco.
- Desenvolvimento ágil se apoia fortemente em opinião de especialistas para a tomada de decisão, o que pode gerar riscos específicos.

RQ2: Quais são os desafios da integração de gestão de risco em projetos que empregam metodologias ágeis?

- Como o processo de gestão de risco em metodologias ágeis é iterativo, ele deve ser leve a fim de não comprometer a flexibilidade de desenvolvimento ágil. Contudo, gestão de risco é tradicionalmente um processo demorado e custoso.
- Há falta de ferramentas e métodos de gestão de risco voltadas para desenvolvimento ágil.
- A falta de formalização e explicitação do processo de gestão pode expor o projeto a riscos.

RQ3: Quais são as práticas da integração de gestão de risco em projetos que empregam metodologias ágeis?

- Gestão de risco em desenvolvimento ágil deve ser um processo explícito, tanto em atividades quanto em quem é responsável por implementá-las.
- *Frameworks* de gestão de risco ágeis, sejam eles originais ou adaptados das metodologias tradicionais, devem ser leves, iterativos e intuitivos. Dessa forma, não comprometem as qualidades das metodologias ágeis e são facilmente integrados pela equipe.
- Gestão de risco deve, idealmente, ter métodos de lidar com os riscos provenientes da tomada de decisão por meio de opinião de especialista.

4.4 Ameaças à Validade

O trabalho apresenta algumas características que podem representar riscos à sua validade, os quais dividimos e explicamos em categorias específicas. Validade interna: Apesar de o estudo ter sido realizado por um revisor independente, não se pode descartar completamente a possibilidade de viés ou erros decorrentes da coleta manual das informações. Validade externa: Foram considerados apenas artigos revisados por pares, publicados em revistas científicas. A exclusão da literatura cinzenta pode ter limitado a abrangência e

a generalização dos achados. Validade de construção: Embora as estratégias de busca tenham sido validadas por meio de testes iterativos, estudos importantes podem ter sido excluídos devido a variações na nomenclatura utilizados pelos autores ou a limitações das bases de dados consultadas. Validade de conclusão: As conclusões foram baseadas em análises qualitativas. Análises quantitativas formais não foram realizadas, o que pode ter restringido a robustez estatística dos achados e limitado a precisão das inferências realizadas.

Capítulo 5

Conclusão

5.1 Contribuições do trabalho

Conforme o desenvolvimento de software se populariza, as estratégias de gestão desses projetos também ganham destaque. Dentre as diversas etapas da gestão de projetos de software, a gestão de risco exige atenção especial. A capacidade de se adaptar às mudanças e aproveitar as oportunidades, sem comprometer as condições do projeto, é fundamental para o sucesso no mercado de desenvolvimento de software. Este trabalho busca oferecer aos desenvolvedores uma análise abrangente das características, desafios e práticas da integração da gestão de risco em projetos de software que adotam metodologias ágeis. Dessa forma, possibilita a identificação dos principais fatores a serem observados ao adotar uma estratégia de gestão de risco em um projeto ágil.

Estratégias de gestão de risco tradicionais não podem ser facilmente adaptadas para os *frameworks* ágeis. Para que os desenvolvedores ágeis possam produzir software levando em consideração tanto as ameaças quanto as oportunidades, é necessário que as práticas de gestão de risco sejam adaptadas. Os processos de gestão de risco precisam ser transformados de um modelo linear para um modelo iterativo. Documentação e processos pesados devem ser otimizados para não comprometer o maior benefício das metodologias ágeis: a velocidade. Discussões sobre riscos devem ser incorporadas à reunião diária Scrum, por exemplo, permitindo melhorar o processo ágil enquanto mantêm sua familiaridade, o que facilita a adesão dos desenvolvedores. As responsabilidades e funções, mesmo que temporárias, devem ser explicitamente definidas para evitar apatia e garantir a resolução de problemas ou o aproveitamento de oportunidades.

As metodologias ágeis são as mais populares no universo do desenvolvimento de software. Para que esse processo seja realizado de maneira eficiente e segura, atividades de gestão de risco precisam ser incorporadas explicitamente às tarefas ágeis. Assim, as van-

tagens do desenvolvimento ágil são preservadas, enquanto sua exposição ao risco é gerida de forma mais eficaz, garantindo um melhor desenvolvimento de software.

5.2 Trabalhos futuros

Trabalhos futuros podem realizar um estudo mais aprofundado sobre as diferentes estratégias de gestão de risco em metodologias ágeis, buscando identificar quais são mais eficazes. É possível, por exemplo, investigar a integração de *frameworks* consagrados, como a gestão de risco no PMBOK, em metodologias ágeis, ou ainda o desenvolvimento de novas estratégias, como a combinação de Scrum e XP.

Outro ponto relevante para futuras pesquisas é a metrificação das abordagens de gestão de risco. Será importante verificar, por meio de evidências quantitativas, quais estratégias de gestão de risco são mais eficazes e em que medida elas impactam as atividades ágeis.

Referências

- [1] Dedrick, Jason, Vijay Gurbaxani e Kenneth L. Kraemer: *Information technology and economic performance: A critical review of the empirical evidence*. ACM Comput. Surv., 35(1):1–28, março 2003, ISSN 0360-0300. <https://doi.org/10.1145/641865.641866>. 1
- [2] Royce, Winston W: *Managing the development of large software systems: concepts and techniques*. Em *Proceedings of the 9th international conference on Software Engineering*, páginas 328–338, 1987. 1
- [3] Abrahamsson, Pekka, Outi Salo, Jussi Ronkainen e Juhani Warsta: *Agile software development methods: Review and analysis*, 2017. <https://arxiv.org/abs/1709.08439>. 1, 4, 6
- [4] Varajão, João, Rui Pedro Marques e António Trigo: *Project management processes – impact on the success of information systems projects*. Informat-ica, 33(2):421–436, 2022. <https://journals.sagepub.com/doi/abs/10.15388/22-INFOR488>. 1, 5
- [5] Aven, Terje: *Risk assessment and risk management: Review of recent advances on their foundation*. European Journal of Operational Research, 253(1), 2016. 1, 6
- [6] Kitchenham, Barbara: *Procedures for performing systematic reviews*. Keele, UK, Keele University, 33(2004):1–26, 2004. 2, 3
- [7] Sager, Naomi, Margaret Lyman, Christine Bucknall, Ngo Nhan e Leo J. Tick: *Natural language processing and the representation of clinical data*. Journal of the American Medical Informatics Association, 1(2):142–160, março 1994, ISSN 1067-5027. <https://doi.org/10.1136/jamia.1994.95236145>. 4
- [8] Bygari, Ramya, Aayush Gupta, Shashwat Raghuvanshi, Aakanksha Bapna e Birendra Sahu: *An ai-powered smart routing solution for payment systems*, 2021. <https://arxiv.org/abs/2111.00783>. 4
- [9] B V, Santhosh Krishna, B Rajalakshmi, Immanuel Dsouza, Jeremie Dsouza, Jeferson X e Ashok K: *E-commerce trend prediction software*. Em *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, páginas 1–4, 2024. 4
- [10] Project Management Institute: *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Project Management Institute, Newtown Square, PA, 6th edição, 2017, ISBN 978-1-62825-184-5. 5

- [11] Pressman, Roger S e Bruce R Maxim: *Engenharia de software-9*. McGraw Hill Brasil, 2021. 5
- [12] Islam, A.K.M Zahidul e Dr. Alex Ferworn: *A comparison between agile and traditional software development methodologies*. Global Journal of Computer Science and Technology, 20(C2):7–42, Jul. 2020. <https://computerresearch.org/index.php/computer/article/view/1987>. 5
- [13] Beck, Kent, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas: *Manifesto for agile software development*. <https://agilemanifesto.org/>, 2001. 5
- [14] Verma, Bhawna e Mamta Dhanda: *A review on risk management in software projects*. International Journal for Innovative Research in Science & Technology, 2(11):499–503, 2016. 6
- [15] International Organization for Standardization: *ISO 31000:2018 - Risk management – Guidelines*. <https://www.iso.org/standard/65694.html>, 2018. Accessed: 2025-05-23. 6
- [16] George, Crispin: *The essence of risk identification in project risk management: An overview*. International Journal of Science and Research (IJSR), 9(2):1553–1557, 2020. 6
- [17] Boehm, B.W.: *Software risk management: principles and practices*. IEEE Software, 8(1):32–41, 1991. 7
- [18] Roy, Geoffrey G: *A risk management framework for software engineering practice*. Em *2004 Australian Software Engineering Conference. Proceedings.*, páginas 60–67. IEEE, 2004. 7
- [19] Chowdhury, Abdullah Al Murad e Shamsul Arefeen: *Software risk management: importance and practices*. IJCIT, ISSN, páginas 2078–5828, 2011. 7
- [20] Fabbri, Sandra, Cleiton Silva, Elis Hernandes, Fábio Octaviano, André Di Thommazo e Anderson Belgamo: *Improvements in the start tool to better support the systematic review process*. Em *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, EASE '16, New York, NY, USA, 2016. Association for Computing Machinery, ISBN 9781450336918. <https://doi.org/10.1145/2915970.2916013>. 10
- [21] Hester, Patrick: *Epistemic uncertainty analysis: an approach using expert judgment and evidential credibility*. Journal of Quality and Reliability Engineering, 2012(1):617481, 2012. 17
- [22] Gordon, Jean e Edward H Shortliffe: *The dempster-shafer theory of evidence*. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, 3(832-838):3–4, 1984. 17

- [23] Stephenson, Todd Andrew: *An introduction to bayesian network theory and usage*. 2000. 17
- [24] Harrison, Robert L: *Introduction to monte carlo simulation*. Em *AIP conference proceedings*, volume 1204, página 17, 2010. 18
- [25] Li, Patrick: *Jira Software Essentials: Plan, track, and release great applications with Jira Software*. Packt Publishing Ltd, 2018. 18
- [26] Guo, Gongde, Hui Wang, David Bell, Yaxin Bi e Kieran Greer: *Knn model-based approach in classification*. Em *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, páginas 986–996. Springer, 2003. 18

Anexo I

Resultados da revisão sistemática da literatura

Tabela I.1: Respostas dos artigos às questões de pesquisa

ID	Título	Características	Desafios	Práticas
1	A framework for risk management in Scrum development process	Não apresentam atividades específicas e explícitas de gestão de risco. Ela é feita de maneira implícita	-	Integração do modelo tradicional de gestão de risco no framework Scrum
2	Agile risk management for multi-cloud software development	-	Metodologias tradicionais de gestão de risco não são facilmente traduzidas para ágil. Análise de risco deve ser contínua, Equipes não em expertise em análise de risco, Ferramentas para gestão de risco em ágil não incentivam colaboração e Dificuldade/Resistência cultural	Registro e análise de riscos em Kanban. Automatização de processos como recomendações aos stakeholders e verificação de progresso. Processo de gestão de risco é contínuo e visual para facilitar a colaboração em um time multi-disciplinar
3	A Multi-agent Reinforcement Learning Risk Management Model for Distributed Agile Software Projects	-	-	Propõem um modelo multi-agentes para tomada de decisão. Os agentes monitoram fatores de risco e, por meio de iterações, realização ações para mitigar os riscos.
Continua na próxima página				

ID	Título	Características	Desafios	Práticas
4	An alternative approach for risk assessment in Scrum	Metodologias de gestão de risco tradicionais precisam ser adaptadas para ágil	Alguns riscos permanecem desconhecidos. Como múltiplas fases de desenvolvimento se sobrepõem, riscos ficam sem o devido tratamento	Modelo de rede de risco é construído a partir da relação entre causa e efeito dos riscos
5	An Empirical Evaluation of a Model for dealing with Epistemic Uncertainty in Agile Software Project Management	Análise de incerteza epistêmica usa julgamento de especialistas	Julgamentos de especialistas são difíceis de avaliar	Propõem modelo com as seguintes etapas: Nivelamento e conscientização, Identificação de fontes de incerteza, Análise de incerteza por especialistas, Análise de interdependência entre incertezas, Análise visual da rede: interpretação visual da rede bayesiana, Integração com gestão de risco e Alimentando o corpo de conhecimento
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
6	A novel risk management model in the Scrum and extreme programming hybrid methodology	Scrum prevê alterações ao longo do projeto. XP observa um projeto em termos de comunicação, simplicidade feedback. Enquanto o Scrum se concentra na gestão de projetos, o XP fornece práticas úteis no desenvolvimento de software	Diferenças fundamentais entre gestão de risco tradicionais e metodologias ágeis	Combinação entre Scrum e XP. Todos os riscos têm de ser identificados numa primeira fase. Prioriza-se os riscos. Os riscos com maior prioridade entram no ciclo de iteração do Scrum. Neste ciclo, que se baseia no método XP, a forma de enfrentar os riscos selecionados na iteração atual é planejada através de diferentes estratégias discutidas. Aplica-se a estratégia associada a esse risco. Este ciclo é seguido para todos os riscos selecionados para serem considerados nesta <i>sprints</i> .
7	Architectural Risk Analysis in Agile Development of Cloud Software	Os desenvolvedores ágeis não implementam nenhuma análise de risco de arquitetura	Difícil integração de análise de arquitetura com DevOps, CI e CD. Já que modelagem de ameaças e atividades relacionadas não são facilmente automatizadas	Análise de risco de arquitetura: Visão macro (forest-view), Determinar a resistência a ataques do sistema, Análise de ambiguidades, Identificar vulnerabilidades no framework
Continua na próxima página				

ID	Título	Características	Desafios	Práticas
8	A Review of Risk Management in Agile Development	Gestão de risco é feito de maneira implícita em ágil	Possibilidade de subestimar muitos riscos	Integração de frameworks como PRINCE2 nas <i>sprints</i> para realizar uma gestão explícita de risco
9	A Review on Risk Management Framework for large scale Scrum	As atividades de gestão de risco devem ser aplicadas de forma iterativa	-	Framework deve facilitar a comunicação e troca de conhecimento entre os times. Risco deve ser considerado no Planejamento de <i>sprint</i> , Scrum Diário, Revisão de <i>sprint</i> , Retrospectiva de <i>sprints</i> .
10	A risk management framework for distributed Scrum using prince2 methodology	-	Forte dependência dos membros da equipe e a área envolvida no projeto ainda não é mensurável e não pode ser estimada em Scrum	Integração entre Scrum e PRINCE2. Identificar as responsabilidades dos indivíduos. Reunir os requisitos e os custos. Abrangência do projeto envolvido pode ser identificada. A partir de onde se pode identificar os riscos do projeto. Gestão das <i>sprints</i> leva em consideração a verificação dos riscos

Continua na próxima página

ID	Título	Características	Desafios	Práticas
11	A Risk Management Framework for Scrum Projects	-	Scrum não tem uma maneira estruturada de gerenciar riscos	Risk Management <i>Product Owner</i> (RIMPO). <i>Product Owner</i> tem a responsabilidade de gerenciar os requisitos e garantir que o produto final trata valor para o cliente. Três tipos de risco de <i>Product Owner</i> : Engenharia de requisitos, Qualidade de software e Migração de times tradicionais para ágil. RIMPO propõem mudanças aos processos do PMBOK para abarcar os riscos que o <i>Product Owner</i> corre. Os processos devem ser executados de maneira iterativa. Planejamento de gestão de risco, Identificação de risco, Análise de risco, Planejamento de resposta de risco, Implementação de resposta de risco e Monitoramento de risco

Continua na próxima página

ID	Título	Características	Desafios	Práticas
12	A risk poker based testing model for Scrum	Metodologias ágeis se baseiam em análise de especialistas quando precisam priorizar tarefas. Embora ágil trata de riscos em seus desenvolvimento, não possui fases de gestão de risco explícitas	Incompletude de uma análise de risco feita por um único especialista	Teste baseado em risco tem como objetivo descobrir os mais custosos e mais importantes defeitos o mais cedo possível. Poker de Risco é estratégia em que são discutidos riscos associados aos requisitos da <i>sprints</i> e suas prioridades. Ao invés de realizar uma análise de exposição ao risco feito por um especialista, Poker de Risco pede ao membros do time que julguem a probabilidade do risco (fator de probabilidade). Em seguida, <i>Product Owner</i> e stakeholders discutem o custo do risco (fator de impacto).
Continua na próxima página				

ID	Título	Características	Desafios	Práticas
13	Artificial Intelligence based Risk Management Framework for Distributed Agile Software Development	-	-	Framework de gestão de risco baseado em IA. Coletar dados para machine learning (estórias de usuários, objetivos de projeto, ambiente e banco de dados de risco). O framework vai automatizar o processo de gestão de riscos e liberar os gerentes de risco para a identificação e estudo de novos riscos

Continua na próxima página

ID	Título	Características	Desafios	Práticas
14	Assessing the Risk of Software Development in Agile Methodologies Using Simulation	-	-	<p>Avaliação de risco utilizando simulação de processos e o método de Monte Carlo. Baseado em histórias de usuário, habilidades da equipe e atividades, o simulador modela o desenvolvimento ágil por meio de eventos e agentes autônomos. Ele analisa seis dimensões de risco, focando na etapa de análise da gestão de risco. O sistema realiza milhares de simulações para prever duração, custo e outros indicadores. Conectado ao JIRA, permite avaliar cenários variáveis e apoiar decisões de planejamento com base em dados probabilísticos</p>
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
15	A Survey of Agile vs. Traditional Methods on Project Risk Management	-	Metodologias ágeis são mais incertos e necessitam de maior comunicação.	A combinação de ágil e tradicional cria um balanço entre flexibilidade e controle. Ajustar as estratégias ao ambiente de desenvolvimento. Ágil também pode focar em qualidade e satisfação do consumidor para combater o risco. Integração de Failure Mode and Effect Analysis (FMEA), identificar e priorizar falhas de acordo com sua severidade e ocorrência, e Critical Path Method (CPM), identifica o caminho crítico e estima a duração de um projeto. Matriz de suporte de decisão que auxilia na escolha do melhor modelo de gestão de risco para cada necessidade.
16	A survey on product backlog change management and requirement traceability in agile (Scrum)	Volatilidade de requisitos é vista como um importante fator de risco para projeto ágeis	-	Rastreamento de cada requisito e seus efeitos ao longo do ciclo de vida do projeto.

Continua na próxima página

ID	Título	Características	Desafios	Práticas
17	Best practices for managing risk in adaptive agile process	Resposta a mudanças ao invés de seguir o plano. Scrum não tem práticas de gestão de risco	Ágil não define nem avalia os riscos dentro dos atributos impacto e probabilidade. O que pode causar má priorização de risco. Metodologias tradicionais definem stakeholders, tanto externos quanto internos. Em ágil, todos são tratados como um coletivo, isso pode dificultar a atribuição de responsabilidades na resolução dos riscos	Atividades de gestão de risco devem ser realizadas nos eventos ágeis: Definir a tecnologia a ser utilizada, Estimar quantidade de iterações a serem realizadas, Encontrar desafios (que podem ser riscos), Identificação de má interpretação de requisitos. Deve ser documentado em uma matriz de risco (impacto x probabilidade) e por diferentes stakeholders em diferentes níveis de abstração
18	Categorization of risk factors for distributed agile projects	Metodologias ágeis não tem maneira formal própria de gerir riscos.	-	Identificação de cinco categorias de risco: Ciclo de Vida de Desenvolvimento de Software, Gestão de Projeto, Consciência de Grupo, Colaboração com stakeholder externo, Setup de tecnologia
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
19	Combining agile approaches and risk management for machatronic product development - A case study	-	-	No planejamento da <i>sprints</i> , uma Mapa de Risco de Estórias do Usuário é criado para avaliar o risco de cada estória ainda não implementada. Risco é calculado com probabilidade de ocorrência e impacto. De acordo com a classificação do risco, medidas devem ser tomadas. Depois que todas as medidas para lidar com os riscos foram tomadas, as estórias são reavaliadas
20	Continuous Requirements Risk Profiling in Information Systems Development	Gestão de risco interativa	Como o processo é iterativo, os requisitos estão sempre mudando e novos requisitos vão surgindo. Assim, a gestão dos riscos associados a esses requisitos é contínua	Lista de riscos e suas gravidades. Identificar o risco e entender sua gravidade, assim como a chance de afetar todo o desenvolvimento.

Continua na próxima página

ID	Título	Características	Desafios	Práticas
21	Development of Scrum-Tree-KNN Algorithm for Distributed Agile Development	-	-	Desenvolve um sistema que modela eficazmente custos do projeto. Itens de trabalho são estruturados em árvore. Nó raiz é o projeto, ramos são os módulos ou funcionalidades e as folhas são os itens de trabalho. Algoritmo KNN é utilizado para encontrar a melhor combinação entre desenvolvedores e itens de trabalho. Mede a similaridade entre tarefas antigas e atuais. Os vizinhos-K mais perto (tarefas similares) determinam o desenvolvedor apropriado para a tarefa atual

Continua na próxima página

ID	Título	Características	Desafios	Práticas
22	Explaining software project risks in agile temas: an action case using casual mapping	Metodologias ágeis não propõem uma técnica formal de avaliação de risco	-	O mapeamento causal é uma forma de mapeamento cognitivo em que se identificam e representam visualmente relações causais entre conceitos ligados a uma situação real. O objetivo é esclarecer como um conceito leva a outro, por meio de afirmações como "A causa B" ou "B resulta de A". Essas relações são representadas em modelos compostos por nós e setas, onde os nós representam conceitos percebidos por atores, como gestores de projetos, em seu contexto, e as setas indicam as crenças desses atores sobre as ligações causais entre os conceitos
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
23	Improving the Organization Technology competitiveness through Effective Management of Conflict-Risk within Agile Teams	Como metodologias ágeis focam em pessoas, gestão de risco voltada para conflitos entre pessoas é importante	Como metodologias ágeis encorajam colaboração, existe maior probabilidade de conflito entre a equipe	Quando o conflito entre os membros não é bem gerido, o efeito é negativo para as futuras colaborações entre a equipe. No entanto, quando é bem gerido, implica no compartilhamento de ideias de forma construtiva entre a equipe
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
24	Integrating risk management in Scrum framework	Metodologias ágeis seguem um processo implícito de gestão de risco.	A falta de um processo explícito de gestão de risco pode fazer com que alguns riscos fiquem sem tratamento	Integração das atividades de risco (identificação, análise, mitigação e controle de risco) em Scrum de maneira iterativa. Identificação é realizada na <i>sprints</i> de planejamento. Os riscos são passados para o Registro de Risco. Nas <i>sprints</i> de retrospectiva são realizados a análise, controle e mitigação. <i>Product Owner</i> é o gestor de risco. Ele é responsável por monitorar, tomar providências quanto aos riscos. Registro de riscos de projetos anteriores podem auxiliar na resolução de riscos. Matriz de prioridade em que verifica-se a probabilidade e o impacto de cada risco

Continua na próxima página

ID	Título	Características	Desafios	Práticas
25	Lightweight risk management: The development of agile risk tool agents	-	Como metodologias ágeis tem baixo nível hierárquico, motivação individual para a identificação, tratamento e discussão de risco é fundamental. Como também focam em velocidade, discussões de problemas podem ser deixada de lado quando o projeto está atrasado	A Ferramenta de Risco Ágil (FRA) é composta por três fases: input, processo e output. Na fase de input, coleta-se dados sobre requisitos, tarefas, equipe e riscos. Os riscos são definidos por indicadores e regras IF/THEN. No processo, o projeto ocorre em iterações e agentes de risco monitoram e gerenciam riscos com base nos dados coletados. A fase final gera um registro consolidado de riscos. O foco do estudo é o desenvolvimento de quatro agentes — gestor, identificador, avaliador e monitorador — que automatizam as etapas de identificação, avaliação e monitoramento de riscos, promovendo uma gestão mais prática e eficiente
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
26	Mapping of Agile Processes into Project Management Knowledge Areas and Processes	Metodologias ágeis carecem de gestão de risco apropriada. Subintende-se que riscos podem ser tratados ao longo das <i>sprints</i> junto dos demais requisitos	A falta de foco em gestão de risco pode causar problemas	Metodologias ágeis requerem uma abordagem proativa para gestão de risco. Registros de risco devem ser mantidos e essa atividade deve ser parte da <i>sprints</i> diária. Quando os Processos de Gestão de Risco de Projeto do PMBOK mapeados para as atividades das metodologias ágeis vemos que, no geral, a implementação de planejamento, identificação, análise, resposta e controle são executados de maneira implícita ao lado das demais atividades. A proatividade da equipe é elemento fundamental para o tratamento de riscos
Continua na próxima página				

ID	Título	Características	Desafios	Práticas
27	Quantitative planning and risk management of Agile Software Development	-	-	Modelo quantitativo e qualitativo de gestão de risco em projetos ágeis de software, inspirado em abordagens do setor financeiro. O modelo coleta dados durante o projeto e utiliza simulações de Monte Carlo para calcular métricas de risco baseadas em tempo, custo, abrangência e qualidade. O risco é definido como o desvio entre valores atuais e desejados, e o sistema usa o conceito de Value at Risk para prever e mitigar desvios. Quando os limites são ultrapassados, o sistema sugere ajustes no projeto. A abordagem busca evitar riscos relacionados tanto à entrega quanto à qualidade do software final.

Continua na próxima página

ID	Título	Características	Desafios	Práticas
28	Risk assessment forum: A proposal for agile software development teams ruled by Scrum	Não há método específico para gestão de risco em Scrum . O mais perto de um método são as reuniões diárias que visam tratar de problemas	-	Envolvimento dos desenvolvedores na identificação de risco. Criação do Fórum de Análise de Risco (FAR). Visa identificar os riscos e expor o time a maior compreensão de risco. Integrado às ideias de Scrum. FAR inicia após a reunião diária. Membros debatem sobre todos as facetas do projeto e os desafios encontrados ou esperados. Esse processo de debate se repete até que tenha se chagado a um consenso ou não haja mais comentários. Ao final, tem-se uma lista com riscos que devem ser analisados. Então, o Scrum Master passa a gerenciar esses riscos de acordo com práticas tradicionais de gestão de risco. Uma ferramenta de auxílio é a Matriz de Marcação de Risco.
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
29	Risk management for Agile projects - in offshore Vietnam	-	<p>A gestão de risco tradicional entra em conflito com metodologias ágeis, levando à super ou sub-implementação. A super-implementação, comum em gestores experientes na abordagem tradicional, reduz a agilidade e impacta a velocidade do projeto. Já a sub-implementação, frequente entre gestores mais jovens, resulta na negligência de riscos devido ao desinteresse por processos pesados. Além disso, metodologias ágeis não cobrem todos os riscos, e a responsabilidade compartilhada faz com que membros da equipe ignorem a gestão de risco, esperando que outros a realizem. Isso compromete o tratamento adequado de ameaças e oportunidades durante o projeto.</p>	<p>Processo de Gestão de Risco Leve para ser integrado em metodologias ágeis. Leve e consistente com atividades de gestão de risco já existente. Consiste de cinco atividades: Identificação, Filtragem, Avaliação, Planejamento e Monitoramento e controle de risco. Em Scrum, o Scrum master é responsável pelo processo de gestão. Os etapas são realizadas nas <i>sprints</i>.</p>
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
30	Risk management framework in Agile software development methodology	Equipes ágeis são colaborativas, a responsabilidade por certos riscos pode ser compartilhada. Metodologias ágeis promovem a análise de risco de requisitos	Risco de má colaboração entre stakeholders. Diferentes perspectivas de risco podem ser um risco em si. Curta iteração de desenvolvimento pode ser um risco na medida em que mudanças são frequentes	O framework propõe um processo iterativo de gestão de risco em projetos ágeis, com seis etapas. Inicia-se com a identificação, comunicação e consulta dos riscos durante as reuniões diárias. Em seguida, analisa-se a probabilidade e o impacto do risco, classificando-os como alto, médio ou baixo. A priorização é feita multiplicando esses fatores; apenas riscos com pontuação elevada são tratados. O tratamento envolve aceitar, evitar, reduzir ou transferir o risco. Após a ação, ocorre a resolução e mitigação. O processo inclui monitoramento contínuo e pode ser aplicado em dois níveis: projeto e <i>sprints</i> , adaptando-se ao ciclo ágil.
Continua na próxima página				

ID	Título	Características	Desafios	Práticas
31	Suitability of PMBOK 6th edition for agile-developed IT Projects	-	<p>Lacunas entre PMBOK e Scrum: Na gestão de projetos clássica, o gestor é responsável pelas entregas e suas integrações. Em Scrum, a integração é gerida pelo time de desenvolvimento. Isso torna difícil para o gestor ser responsável pelo projeto. Gestão tradicional tem em vista o longo prazo, o projeto completo. Em Scrum, o que mais interessa é a próxima iteração. Como não há, em Scrum, uma visão ao longo prazo, dificuldades de estimar custos e cronogramas podem ocorrer. Como times ágeis são auto-organizados, gestores de risco podem perder responsabilidade, o que pode causar conflitos em alguns ambientes e culturas de trabalho.</p>	<p>A abordagem iterativa de Scrum, permite que o gestor faça a integração de cada <i>sprints</i> nas reuniões de revisão. Práticas como produto mínimo viável podem ajudar quando é preciso conciliar visão de longo prazo e desenvolvimento iterativo. Práticas como definir a maturidade das histórias de usuários podem ajudar quando é necessário estimar custos e cronograma. Muitas das áreas de gestão de risco do PMBOK não tem equivalente em Scrum e as atividades devem ser integradas, principalmente, nas reuniões diárias.</p>
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
32	Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities	Metodologias ágeis foca em redução de incerteza (ameaças), mas não muito em gestão de incerteza (ser capaz de aproveitar as oportunidades). Metodologias ágeis tem atividades que minimizam a incerteza de requisitos e reduzem volatilidade.	-	A gestão de incerteza envolve planejar antecipando surpresas e adotar estratégias para lidar com situações imprevistas. Isso inclui vigilância constante, uso de feedback entre iterações para ajustes contínuos, análise coletiva de tarefas com dúvidas, e compartilhamento de conhecimento. Outras práticas incluem prototipação para validar ideias, criação de alternativas para aumentar a flexibilidade, e definição de novas funções conforme surgem necessidades. A integração com stakeholders ajuda a reduzir ambiguidades, e mudar de atividade evita paralisações. As estratégias usadas para lidar com riscos também podem ser aplicadas para explorar oportunidades ao longo do desenvolvimento do projeto.
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
33	User Story Risk Prioritization Model for Agile Software Development	-	-	<p>O modelo consiste em duas partes. A primeira consiste de identificação de risco, que são associados a baixa qualidade de estórias de usuários. A segunda, analisar riscos com base em impacto e probabilidade. O modelo é aplicado com Processo Hierarquico Analítico (PHA). PHA é uma técnica de decisão multi-critério. PHA inicia com um objetivo que deve ser decomposto em uma hierarquia de critérios e alternativas. Então, estabeleça prioridade entre os critérios e as alternativas ao comparar cada para de alternativa com respeito a cada critério. Depois, verifique a consistência entre as comparações, calcule os pesos relativos das comparações e calcule as prioridades definitivas das alternativas.</p>
<i>Continua na próxima página</i>				

ID	Título	Características	Desafios	Práticas
34	Using velocity, acceleration, and jerk to manage agile schedule risk	Metodologias ágeis tem estratégias de controle de risco. Contudo, não especificaram processos para tratar e resolver riscos. Passos específicos são negligenciados. Esses passos são: Definição de processo, Estratégias de mitigação, Definição de reposição de risco e Definição de gatilhos para indicar a necessidade de mudança de estratégia de mitigação. Assim, de acordo com a definição da ISO, essas metodologias não apresentam gestão de risco por não possuírem atividades bem especificadas e explícitas	-	A velocidade, medida em pontos de estória por <i>sprints</i> , é uma métrica de produtividade que indica quanto trabalho é realizado em um intervalo de tempo. Reduções na velocidade podem sinalizar risco de atraso, enquanto aumentos indicam possível adiantamento, ambos configurando riscos de agenda. A velocidade é definida no planejamento e calculada ao final de cada <i>sprints</i> . Aceleração, a variação da velocidade entre <i>sprints</i> , serve como gatilho para mudanças no processo. Já o Jerk Ágil, que representa a variação da aceleração, é usado como indicador de estabilidade nas entregas. Quando a aceleração tende a zero, a entrega se estabiliza, reduzindo o risco de atrasos em tarefas não concluídas.