GERENCIAMENTO DE RISCOS ASSOCIADOS À SEGURANÇA DA INFORMAÇÃO EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARE ÁGEIS: UMA REVISÃO SISTEMÁTICA.

LUCIANO GONÇALVES DE CARVALHO[1]
MARIÂNGELA FERREIRA FUENTES MOLINA[2]

RESUMO

Embora os métodos ágeis de desenvolvimento de software tenham contribuído para aumentar a quantidade de projetos de software realizados com sucesso, eles não eliminaram todos os problemas enfrentados durante o processo de desenvolvimento de software, pois falhas relacionadas à segurança da informação continuam sendo encontradas. Uma forma de reduzir o número de falhas é aplicar conhecimentos, habilidades, ferramentas e técnicas relacionadas ao gerenciamento dos riscos do projeto. O presente artigo tem por objetivo apresentar as práticas de gerenciamento de riscos utilizadas atualmente em processos de desenvolvimento de software ágeis por meio de uma revisão sistemática da literatura. Os resultados indicam a existência de poucas soluções que podem ter sua eficácia medida e sugerem que novas pesquisas na área são necessárias.

Palavras-chave: Gerenciamento de Riscos; Segurança da Informação; Desenvolvimento de Software; Gerenciamento de Projetos; Ágil.

ABSTRACT

Although agile software development methods have contributed to increasing the number of successful software projects, they have not eliminated all the problems faced during the software development process, as failures related to information security continue to be found. One way to reduce the number of failures is to apply knowledge, skills, tools and techniques related to project risk management. This article aims to present the risk management practices currently used in agile software development processes through a systematic literature review. The results indicate that there are few solutions that can have their effectiveness measured and suggest that further research in the area is necessary.

Key Words: Risk Management; Information Security; Software Development; Project Management; Agile.

[1]Docente, Faculdade de Tecnologia de Mogi das Cruzes – FATEC-MC. Mogi das Cruzes-SP. E-mail: luciano.carvalho@fatec.sp.gov.br
[2]Docente, Faculdade de Tecnologia de Mogi das Cruzes – FATEC-MC. Mogi das Cruzes-SP. mariangela.molina@fatec.sp.gov.br

INTRODUÇÃO

Com o objetivo de melhorar a previsibilidade do desenvolvimento de software no final da década de 1960, foram desenvolvidos processos baseados em planos e documentos, os quais visavam construir software com qualidade, custo e tempo previsíveis, como feito em áreas da engenharia (FOX e PATTERSON, 2017). Os representantes mais conhecidos desse tipo de abordagem são o modelo Cascata e o Rational Unified Process (RUP).

Embora se tenha empregado processos de desenvolvimento de software rigorosos e controlados com sucesso em projetos de software de grande porte, como em sistemas de controle de aeronaves, que geralmente possuem grandes equipes e o desenvolvimento leva alguns anos, eles têm se mostrado ineficientes para o desenvolvimento de sistemas corporativos de pequeno e médio porte, onde os clientes podem mudar de ideia mais frequentemente e consequentemente alterar os requisitos de software.

Em virtude do insucesso de diversos projetos (CHARETTE, 2005) e da insatisfação com as abordagens existentes, foi proposto na década de 1990 novos métodos, cuja especificação, desenvolvimento e entrega do software são incrementais, permitindo que a equipe de desenvolvimento foque na construção do software propriamente dito e não na sua concepção e documentação (SOMMERVILLE, 2011). Tais métodos são conhecidos como ágeis.

Como os métodos ágeis são menos rigorosos, há o risco de que aspectos relacionados ao gerenciamento de projetos de software, que colaboram para o sucesso desse tipo de projeto, sejam ignorados, fazendo com que o software seja entregue com falhas. Segundo o Project Management Institute PMI (2017), “gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de cumprir os seus requisitos”.

Um exemplo típico de problema apresentado por softwares é a presença de falhas relacionadas à segurança da informação, cujos pilares são a

confidencialidade, integridade e disponibilidade, que podem ser mapeadas e evitadas por meio da aplicação de conhecimentos, habilidades, ferramentas e técnicas relacionadas à disciplina de gerenciamento dos riscos do projeto, uma dentre as dez disciplinas presentes no guia do Conhecimento em Gerenciamento de Projetos (Project Management Body of Knowledge - PMBoK) (PMI, 2017).

Considerando a importância da gestão de riscos em projetos de desenvolvimento de software para a identificação e tratamento de questões relacionadas à segurança da informação, este artigo objetiva apresentar e discutir o resultado de uma revisão sistemática sobre o estado da arte das práticas de gerenciamento de riscos utilizadas com metodologias ágeis, pautada em um protocolo que define claramente as etapas de trabalho e que permite que o processo seja auditado e reproduzido por outros pesquisadores.

Para atingir o objetivo proposto, foram analisados artigos que tratam do gerenciamento de riscos em projetos de desenvolvimento software ágeis que visam tratar de questões relacionadas especificamente à segurança da informação.

REFERENCIAL TEÓRICO

Diferentemente dos métodos tradicionais de desenvolvimento de software, que privilegiam o planejamento e a documentação, os métodos ágeis estão baseados no desenvolvimento e entrega incrementais e compartilham um conjunto de valores e princípios descritos no Manifesto para Desenvolvimento Ágil de Software (AGILE ALLIANCE, 2001). Esta abordagem tem mostrado um melhor resultado, como apresentado em estudo realizado pelo The Standish Group entre 2011 e 2015 com mais de 10.000 projetos de software (Quadro 1).

QUADRO 1. Sucesso e fracasso de projetos segundo o método utilizado.

Tamanho	Método	Sucesso	Fracasso
Todos os tamanhos de projetos	Ágil	39%	9%
	Cascata	11%	29%
Projetos grandes	Ágil	18%	23%
	Cascata	3%	42%
Projetos médios	Ágil	27%	11%
	Cascata	7%	25%
Projetos pequenos	Ágil	58%	4%
	Cascata	44%	11%

Fonte: Adaptado de CHAOS Report 2015 do The Standish Group (2015).

Os métodos ágeis mais conhecidos e utilizados são (SOMMERVILLE, 2011):

- Extreme Programming (XP): tem como valores fundamentais a comunicação, simplicidade, feedback, coragem e respeito, e envolve as atividades de planejamento, projeto, codificação e testes (BECK, 2000);
- Scrum: *framework* para desenvolvimento ágil de produtos complexos e adaptativos, possui como fundamento principal o empirismo, especifica os papeis de Scrum master, dono de produto e equipe de desenvolvimento, que realizam sprints para a construção do produto (SCHWABER e SUTHERLAND, 2017);
- Crystal: concentra-se principalmente nas pessoas e suas interações ao trabalhar em um projeto, e não em processos e ferramentas, e é baseado na suposição de que as equipes podem otimizar seus processos conforme seu trabalho e se tornar uma equipe mais otimizada e que os projetos são únicos e dinâmicos e requerem métodos específicos (COCKBURN, 2004);
- Adaptative Software Development: concentra-se na colaboração e na auto-organização, cujo ciclo compreende especular, colaborar e aprender, proporcionando aprendizado e adaptação contínuos ao estado emergente do projeto (HIGHSMITH, 2000);

- Dynamic Systems Development Method (DSDM): método de desenvolvimento interativo e incremental, baseado em desenvolvimento rápido de aplicação, cujo objetivo é entregar software dentro do prazo e custo estimados por meio de ajustes de requisitos ao longo do desenvolvimento (STAPLETON, 1997);
- Feature-Driven Development (FDD): processo de iteração curta orientado por modelo, utiliza marcos para identificar o progresso feito em cada recurso e composto pelas atividades de desenvolvimento do modelo geral, criação da lista de recursos, planejamento por recurso, projeto por recurso e construção por recurso (PALMER e FELSING, 2002).

O desenvolvimento de software se enquadra na definição de projeto, que segundo o PMI (2017) “é um esforço temporário empreendido para criar um produto, serviço ou resultado único”. Portanto, pode ter a possibilidade de se beneficiar de boas práticas de gerenciamento de projetos que visam ajudar a cumprir os objetivos de negócio, satisfazer expectativas das partes interessadas, tornar o desenvolvimento mais previsível, aumentar as chances de sucesso, responder a riscos em tempo hábil etc.

Pode-se destacar duas abordagens amplamente difundidas para o gerenciamento de projetos:

- PRINCE2 (PRojects IN Controlled Environments): padrão de gerenciamento de projetos originalmente desenvolvido para projetos de tecnologia da informação e comunicação (TIC), que integra princípios, temas, processo e ambiente de projeto, e foca no controle de escopo, tempo, custo, qualidade, riscos e benefícios (AXELOS, 2017);
- PMBoK: conjunto de boas práticas de gerenciamento de projetos organizados nas áreas de conhecimento em gerenciamento da integração, escopo, cronograma, custos, qualidade, recursos, comunicações, riscos, aquisições e partes interessadas do projeto, e cujos respectivos processos são agrupados nos grupos de processo de iniciação, planejamento, execução, monitoramento e controle, e encerramento (PMI, 2017).

Práticas associadas ao gerenciamento de risco em projetos de desenvolvimento de software podem contribuir para a mitigação de riscos associados à segurança da informação, que envolve principalmente a preservação dos seguintes conceitos:

- Confidencialidade: a informação poderá ser acessada apenas por quem tem o direito de o fazer, isto é, tem a devida autorização;
- Integridade: a informação deverá permanecer exata e completa, isto é, da mesma forma em que foi concebida, bem como seus métodos de processamento, não podendo ser alterada indevidamente;
- Disponibilidade: a informação e ativos correspondentes deverão estar disponíveis para acesso sempre que requerido, desde que se tenha a devida autorização.

A ocorrência de um incidente de segurança, em muitos casos, pode determinar o fim dos negócios de uma empresa ou gerar muitos problemas às pessoas que dele dependem.

MATERIAL E MÉTODOS

A execução da revisão sistemática abrangeu as seguintes fases:

1. Planejamento: estabelecimento de um protocolo para guiar a pesquisa de artigos, que teve como ponto de partida uma pesquisa exploratória para determinar a principal questão de pesquisa (“Quais são as práticas de gerenciamento de risco empregadas para tratar questões relacionadas à segurança da informação em projetos de desenvolvimento de software ágeis?”) e as palavras chaves (“agile”, “project management”, “risk management”, “security” e “software development”) empregadas nos mecanismos de busca das bases de dados pesquisadas (IEEE, ACM, Science Direct, Scopus, Web of Science e Portal CAPES), e onde foram determinados todos os critérios de inclusão (“Apresentar prática de gerenciamento de risco relacionado à segurança da informação” e “Estar relacionado a processo de desenvolvimento de software ágil”) e exclusão (“Impossibilidade de acesso à

íntegra do trabalho", "Não apresenta detalhes do processo de desenvolvimento do software", "Não aborda riscos relacionados à segurança da informação", "Apresentação de produto ou ferramenta de suporte") de artigos;

2. Condução: construção das *strings* de busca (IEEE: "(((((("All Metadata":security) AND "All Metadata":risk management) AND "All Metadata":project management) AND "All Metadata":software development) AND "All Metadata":agile)"; ACM: "[All: security] AND [All: "software development"] AND [All: "project management"] AND [All: "risk management"] AND [All: "agile"] AND [Publication Date: (01/01/2015 TO 05/31/2020)]"; Science Direct: "security risk management agile"; Scopus: "security AND "software development" AND "project management" AND "risk management" AND agile"; Web of Science: "security software development project management risk management agile"; Portal CAPES: "agile security risk"), execução da busca e consequente seleção de artigos baseada nos critérios de inclusão e exclusão;

3. Extração de dados: análise aprofundada dos artigos selecionados, buscando os elementos que respondem à pergunta de pesquisa, identificando os aspectos de segurança da informação tratados e a apresentação de cenários de aplicação dos conceitos abordados (Nenhum, Hipotético, Real ou ambos).

As buscas pelos artigos aconteceram no dia vinte e cinco de abril de dois mil e vinte, retornando um total de 44 trabalhos, dois quais apenas 14 foram selecionados em virtude dos critérios de inclusão e o restante descartado em virtude dos critérios de exclusão.

Após a análise integral de cada artigo, 2 trabalhos foram rejeitados por também se enquadrarem em critérios de exclusão. Portanto, 12 trabalhos foram considerados para a confecção do presente artigo.

RESULTADOS

Jaatun (2019) propõe em seu trabalho a utilização de uma análise de risco arquitetural que contempla as etapas de visualização da arquitetura em nível geral, determinação da resistência do sistema a ataque utilizando listas de verificação como o modelo STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service e Elevation of Privilege), realização de análise de ambiguidade e identificação de vulnerabilidades em estruturas subjacentes. Este trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada e não propõe um cenário de aplicação.

Albadarneh et al. (2015) defende que o método ágil já é uma estratégia de mitigação de riscos, pois a identificação de risco ágil é feita nas reuniões diárias. A análise de risco ágil é feita pelo gerente de projeto ou especialista por meio da atribuição de nota à probabilidade e impacto.

A priorização de risco ágil se dá pela aplicação do cálculo de risco, o qual é calculado pela multiplicação da probabilidade pelo impacto e cujo resultado é avaliado segundo uma escala de valor que varia entre 1 e 9, sendo que os riscos avaliados entre 6 e 9 são considerados significativos e precisam ser gerenciados, e os riscos avaliados entre 1 e 5 são riscos que não necessitam de gerenciamento.

O planejamento de gerenciamento de risco ágil é feito pelo gerente de projeto que define a abordagem (Reter (aceitar), Evitar, Reduzir e Transferir) e técnica (responsabilidade compartilhada, jogar para frente o item de risco para se ter mais tempo para avaliar e encontrar a solução, e investigação de risco), a resolução de risco ágil que consiste na execução do plano de gerenciamento de risco, e o monitoramento de risco ágil que consiste no monitoramento continuo do plano de gerenciamento de risco. Este trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada e não propõe um cenário de aplicação.

Frijns et al. (2018) concluem que somente o desenvolvimento ágil não trata a segurança da informação, mas a sua associação com o conceito de DevOps

(Development and Operations), que é uma prática que unifica o desenvolvimento e operação de software e defende a automação e monitoramento das fases de sua construção. Este trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada e não propõe um cenário de aplicação.

Ramadani e Utama (2015) identificaram na literatura adaptações da engenharia de segurança aos métodos ágeis, que contemplam a modificação do próprio processo, a modificação do gerenciamento de projeto e outras soluções complementares (matriz de segurança, treinamento de segurança e avaliação de risco por especialista). Este trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada e não propõe um cenário de aplicação.

A modificação do próprio processo ágil consiste no uso de estórias ou casos de mau uso, adoção de planos de teste nas fases de projeto, implementação e verificação, e utilização de técnicas de gerenciamento de riscos tais como identificação de riscos, modelagem de ameaças e desenvolvimento de documento de arquitetura de segurança, e emprego de fase de teste e melhora de segurança. Já a modificação do gerenciamento de projeto envolve o comprometimento por escrito em favor da segurança, identificação de risco, cálculo de oportunidade perdida, gerenciamento de performance/risco e ferramenta de controle de segurança.

Singh (2018) propôs a integração do modelo XP com processo seguro para seleção de requisitos, tal processo utiliza o conceito de autenticação e autorização para se fazer uma análise de risco baseado em estórias de usuário. Para garantir a confidencialidade, integridade e disponibilidade, é solicitado a autenticação do usuário cuja estória será desenvolvida, e sua validação é efetuada (aceitação da solução, execução de análise de risco e inspeção da entrega). Este trabalho não propõe um cenário de aplicação.

Chadli e Idri (2017) apresentam uma série de estratégias de mitigação divididas entre tarefas relacionadas a atores (como realizam as tarefas) e tarefas relacionadas à estrutura (organização do projeto para execução de tarefas de

desenvolvimento). Com isso, dentre os vários riscos possíveis, seria possível tratar aqueles relacionados com a confidencialidade, integridade, disponibilidade e privacidade. Este trabalho não propõe um cenário de aplicação.

Hassani et al. (2018) propõem um método híbrido que considera a utilização do processo cascata em conjunto com o processo ágil, ficando a cargo do primeiro o gerenciamento de riscos. O processo resultante possui uma fase de planejamento híbrido que requer um plano de projeto completo, o que favorece a análise de risco. Este trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada e não propõe um cenário de aplicação.

Hayat et al. (2019) consideram uma abordagem de engenharia de requisitos baseada na SysML (System Modeling Language) na qual os riscos de um requisito são representados de forma gráfica. É definido então como mitigar cada um deles, a partir dos seguintes passos:

1. classificação dos requisitos em funcional e não funcional;
2. análise do risco de cada requisito;
3. criação de um impacto positivo e negativo no sistema durante a implementação do requisito;
4. modelagem gráfica do risco do requisito com o diagrama de requisito SysML;
5. ordenação e modelagem dos requisitos de forma a serem compreensíveis às partes interessadas; e
6. controle dos riscos por meio de ações preventivas e de detecção.

O trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada, mas propõe um cenário de aplicação hipotético.

Franqueira et al. (2011) assumem que práticas isoladas de garantia de segurança devem ser totalmente integradas e embutidas em iterações curtas de avaliação de riscos, tratamento e aceitação, provendo insumos para a atualização de requisitos de segurança e para o gerenciamento de riscos de segurança.

O processo de gerenciamento de risco de segurança ágil proposto considera que os insumos de várias práticas de avaliação de segurança existentes e dados

empíricos de catálogos e bases de dados públicos são recebidos nas iterações e a saída alimenta atualizações dos requisitos de segurança e funcionalidades e provê feedback ao processo de gerenciamento de segurança, que coleta de forma incremental riscos residuais (avaliados na próxima iteração).

O trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada e não propõe um cenário de aplicação.

Casola et al. (2020) apresentam a metodologia SSDE (Security SLA-based Security-by-Design) baseada em acordo de nível de serviço (Service Level Agreement - SLA) de segurança que pode ser integrada ao processo de desenvolvimento ágil e que é capaz de suportar o ciclo de vida do gerenciamento de risco.

A referida metodologia apresenta as fases de modelagem (especificação da arquitetura funcional do software), avaliação de segurança por componente (processo de projeto e avaliação de cada componente que contempla a análise de risco para identificação de requisitos de segurança e a avaliação de segurança para a verificação da correta aplicação de contramedidas) e avaliação de segurança por aplicativo (avaliação geral de segurança).

É importante salientar que também é proposto a integração com o Scrum, onde a cada reunião de planejamento é executado a fase de modelagem e a avaliação de segurança é feita na reunião de revisão (requisitos de segurança são identificados e alocados para a próxima reunião de planejamento).

O trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada, mas propõe cenários de aplicação hipotético e real.

Organ e Stapleton (2015) discutem a abordagem SSM (Soft Systems Methodology) que consiste nas seguintes fases: Fase 1 – A Situação (representar a situação do problema de forma pictórica, a fim de obter uma compreensão da situação complexa e muitas vezes arriscada); Fase 2 - Gerar modelos de atividades intencionais (debater representações da atividade humana intencional, que são construídas em torno da visão global da pessoa que constrói o diagrama, a fim de questionar o envolvimento do gerenciamento de riscos dos sistemas e

determinar quais mudanças são necessárias para melhorar esse envolvimento); e Fase 3 - Usando os modelos para aprender sobre a situação a fim de trazer melhorias (desenvolver os modelos intencionais e discutir de forma estruturada a situação modelada juntamente com uma conversa sobre quais mudanças são necessárias para determinada situação). Este trabalho não identifica os conceitos de segurança da informação tratados pela abordagem apresentada e não propõe um cenário de aplicação.

Por fim, Sadler (2019) considera que atualmente o desenvolvimento de software é uma entrega contínua de versões e aplica o mesmo modelo para desenvolvimento e manutenção, propondo um novo método baseado em normas relacionadas ao gerenciamento de riscos denominado ER2C SDMLC (Enterprise Release Risk-Centric Systems Development and Maintenance Life Cycle), modelo de ciclo de vida para desenvolvimento e manutenção de sistemas que fornece um pipeline unificado para cenários de desenvolvimento e manutenção.

Para a ER2C SDMLC, o sistema de interesse é um conjunto de versões e cada uma delas pode ser categorizada em dimensões de escopo, perfil de risco e ambiente de hospedagem.

DISCUSSÃO

Após a análise dos artigos selecionados, pode-se perceber que a grande maioria das soluções propostas não são aplicadas em qualquer tipo de cenário, dificultando uma compreensão mais ampla e a realização de uma análise de eficácia.

Um ponto preocupante é a não identificação clara dos princípios de segurança da informação que cada abordagem visaria proteger, pois isso auxiliaria na escolha pela abordagem mais adequada a cada situação encontrada e possibilitaria uma eventual adequação ao processo de desenvolvimento ágil adotado, evitando que perca características importantes e deixe de ser interessante a quem o emprega.

É possível afirmar que muito do conhecimento em gerenciamento de riscos aplicados aos métodos tradicionais de desenvolvimento de software, que apresentam bons resultados nessa questão, foram praticamente ignorados pela maioria dos trabalhos. Embora uma estratégia ou técnica não possa ser utilizada exatamente como havia sido concebida, isso não significa que ela não possa ser adaptada a um novo contexto, e assim produzir bons resultados. É importante salientar que, no trabalho de Hassani et al. (2018), é utilizado um método híbrido, com a utilização de aspectos do processo cascata.

CONCLUSÃO

Uma estratégia adequada para reduzir o número de falhas de segurança em software é fazer o gerenciamento de riscos durante o processo de desenvolvimento de software.

Atualmente muitas empresas utilizam os métodos ágeis para desenvolver seus produtos de software, que podem não contemplar especificamente o gerenciamento de riscos relacionados à segurança da informação, por isso, existe a necessidade de se adaptar esses métodos.

A partir da revisão sistemática realizada neste trabalho, pode-se concluir que atualmente existem poucas soluções para gerenciar os riscos relacionados à segurança da informação e a maioria delas não faz um mapeamento de problemas de segurança a serem tratados e não foram testadas em projetos reais, prejudicando uma análise adequada sobre a sua eficácia.

Fica claro também que existem diversas lacunas com relação à adaptação dos métodos ágeis para tratamento mais amplo dos riscos, significando que mais pesquisas devem ser feitas sobre o assunto.

REFERÊNCIAS BIBLIOGRAFIAS

AGILE ALLIANCE. **The agile manifesto**. 2001. Recuperado em 13 de junho, 2020, de https://agilemanifesto.org/

ALBADARNEH, A., ALBADARNEH, I. e QUSEF, A. **Risk management in agile software development: a comparative study**. In IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2015.

AXELOS. **Managing successful projects with PRINCE2**. 6th Edition, TSO, 2017.

BECK, K. **Extreme Programming Explained**. Reading, Mass.: Addison-Wesley, 2000.

CASOLA, V., De BENEDICTIS, A., RAK, M. & VILLANO, U. **A novel security-by-design methodology: modeling and assessing s security-by-design SLAs with a quantitative approach.** The Journal of Systems and Software. Elsevier, 2020.

CHADLI, S. e IDRI, A. **Identifying and mitigating risks of software project management in global software development**. In Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, 2017.

CHARETTE, R. **Why software fails**. IEEE Spectrum, 42(9), 42-49, 2005.

COCKBURN, A. **Crystal clear: a human-powered methodology for small teams**. Boston: Addison-Wesley, 2004.

FOX, A. & PATTERSON, D. **Construindo software como serviço: uma abordagem ágil usando computação em nuvem**. 1ª edição, Strawberry Canyon LLC, 2017.

FRANQUEIRA, V., BAKALOVA, Z. TUN, T. e DANEVA, M. **Towards agile security risk management in RE and beyond**. In Proceedings - 1st International Workshop on Empirical Requirements Engineering, EmpiRE, 2011.

FRIJNS, P., BIERWOLF, R. E ZIJDERHAND, T. **Reframing security in contemporary software development life cycle**. In IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), 2018.

HASSANI, R., El BOUZEKRI, El IDRISSI, Y. e ABOUABDELLAH, A. **Digital project management in the era of digital transformation: hybrid method**. In Proceedings of the 2018 International Conference on Software Engineering and Information Management, 2018.

HAYAT, F., ANWAR, M., AZAM, F. e KIRAN, A. **A SYSML-Based approach for requirements risk management and change control**. In Proceedings of the 2019 11th International Conference on Information Management and Engineering, 2019.

HIGHSMITH, J. **Adaptative software development: a collaborative approach to managing complex systems**. Nova York: Dorset House, 2000.

JAATUN, M. **Architectural risk analysis in agile development of cloud software**. In IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2019.

ORGAN, J. e STAPLETON, L. **Technologist engagement with risk management practices during systems development? Approaches, effectiveness and challenges**. Springer-Verlag, 2015.

PALMER, S. e FELSING, J. **A practical guide to Feature-Driven Development**. Englewood Cliffs, NJ: Prentice Hall, 2002.

Project Management Institute. **Guia do conhecimento em gerenciamento de projetos**. 6ª edição, PMI, 2017.

RAMADANI, L. e UTAMA, N. **Secure software engineering for agile methodology preliminary investigation**. In Second International Conference on Computing Technology and Information Management (ICCTIM), 2015.

SADLER, H. **ER2C SDMLC: enterprise release risk-centric systems development and maintenance life cycle**. Springer Science+Business Media, 2019.

SCHWABER, K. & SUTHERLAND, J. **The Scrum guide**. 2017. Recuperado em 13 de junho, 2020, de https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

SINGH, A. **Integrating the extreme programing model with secure process for requirement selection**. In Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018.

SOMMERVILLE, I. **Engenharia de sofware**. 9ª edição, Pearson, 2011.

STAPLETON, J. **DSDM Dynamic Systems Development Method**. Harlow, Reino Unido: Addison-Wesley, 1997.