Rafael Carneiro Reis de Souza (rcrs4@cin.ufpe.br)

Adotando modelagem de ameaças em projetos ágeis de desenvolvimento de software

Universidade Federal de Pernambuco
Centro de Informática
Graduação em Engenharia da Computação

Recife
2022

Rafael Carneiro Reis de Souza (rcrs4@cin.ufpe.br)

Adotando modelagem de ameaças em projetos ágeis de desenvolvimento de software

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Área: Engenharia de *Software* e Linguagens de Programação

Orientadora: Profa. Carla Taciana Lima Lourenco Silva

Recife

2022

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Souza, Rafael Carneiro Reis de.
Adotando modelagem de ameaças em projetos ágeis de desenvolvimento de software / Rafael Carneiro Reis de Souza. - Recife, 2023.
21 p. : il., tab.

Orientador(a): Carla Taciana Lima Lourenco Silva
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Engenharia da Computação - Bacharelado, 2023.

1. Modelagem de ameaças. 2. Desenvolvimento seguro. I. Silva, Carla Taciana Lima Lourenco. (Orientação). II. Título.

000 CDD (22.ed.)

Adotando modelagem de ameaças em projetos ágeis de desenvolvimento de software

Rafael Carneiro R. de Souza, Carla Taciana Lima Lourenco Silva

Centro de informática - Universidade Federal de Pernambuco (UFPE) - Recife - PE - Brasil

rcrs4@cin.ufpe.br,ctlls@cin.ufpe.br

Resumo. *Contexto: a modelagem de ameaças é uma atividade importante para a segurança, mas o seu uso no desenvolvimento ágil é difícil. Problema: para que a modelagem de ameaças seja aplicada no desenvolvimento ágil, é necessário entender seus desafios e boas práticas. Solução proposta: para compreender qual o cenário atual do uso de threat modeling em metodologias ágeis, foi feito um estudo na literatura sobre essa prática. Metodologia: foi feita uma RSL utilizando motores de busca para obter artigos que pudessem responder as perguntas feitas. Resultados: o estudo identificou os desafios, práticas e ferramentas utilizadas. Contribuições: o estudo trouxe as principais tendências da área estudada e uma proposta de ensino sobre o assunto.*

1. Introdução

A segurança da informação é um assunto que vem sendo abordado com cada vez mais relevância, isso é devido ao uso da tecnologia em todos os âmbitos das nossas vidas. Serviços na internet atualmente são extremamente atrativos para criminosos, que buscam dados pessoais de possíveis vítimas, realização de fraudes ou até participação em guerras cibernéticas. Sistemas críticos como bancos e gerenciamento de usinas hidrelétricas estão contidos na internet e podem ser alvo de criminosos.

Diante desse cenário, a modelagem de ameaças (*threat modeling*), se faz bastante importante por tentar mitigar as ameaças e vulnerabilidades ainda na fase de desenvolvimento do sistema [Shawn Hernan 2019], o que diminui as chances de incidentes de segurança e também os custos necessários para realizar as correções após o lançamento do sistema [G. McGraw 2004].

Ocorre que, nos dias de hoje, muitas empresas utilizam de metodologias ágeis para o desenvolvimento dos seus *softwares*, devido aos seus diversos benefícios de entrega contínua e interação constante com os clientes. Porém, o uso dessas metodologias podem ir de encontro com a segurança do sistema [de Vicente Mohino 2019], visto que seus princípios muitas vezes estão em discordância. Devido a isso, se fez necessário o estudo a respeito do uso de *threat modeling* em metodologias ágeis na forma de uma revisão sistemática de literatura. O foco em modelagem de ameaças se deu por ser uma atividade notavelmente importante para mitigar vulnerabilidades ainda na fase de desenvolvimento, sendo essas mitigações ainda na fase de desenvolvimento essenciais para para um *software* seguro [G. McGraw 2004].

Além disso, por não haver uma base de segurança da informação por parte dos desenvolvedores atualmente no mercado de trabalho [Hela Oueslati 2015], [Karin Bernsmed 2019], [Karin Bernsmed 2022], e pela pouca abordagem nas universidades

sobre este assunto, em particular no centro de informática ao qual os autores fazem parte, também foi realizado um estudo a respeito do ensino de *threat modeling*, principalmente em um ambiente que se utiliza metodologias ágeis.

As seções seguintes deste trabalho estão organizadas da seguinte forma: a seção 2 faz uma fundamentação teórica a respeito dos assuntos abordados, a seção 3 explica qual metodologia foi utilizada para a revisão de literatura realizada, a seção 4 contém os resultados obtidos e, por fim, a seção 5 contém as conclusões finais sobre o estudo realizado.

2. Fundamentação teórica

2.1. Desenvolvimento de Software Seguro

Nos últimos anos, o desenvolvimento de *software* foi uma atividade que cresceu rapidamente, grande parte da população possui acesso a internet ou *smartphones*. No entanto, proteger os dados dos usuários é uma questão mandatória para empresas e governos. Portanto, o desenvolvimento seguro de *software* tornou-se essencial para garantir a confidencialidade, integridade e disponibilidade dos sistemas e dos dados que eles possuem.

Um *software* pode ser considerado seguro uma vez que ele continua a funcionar corretamente mesmo sob ataque de um usuário malicioso [G. McGraw 2004]. Porém, tornar um *software* seguro é um processo difícil para os desenvolvedores, em muitos casos, devido à falta de conhecimento dos desenvolvedores em relação ao desenvolvimento de *software* seguro. Este foi um ponto identificado por McGraw em 2004 e que mesmo após mais de uma década, ainda é tido como um desafio para o desenvolvimento de *software* seguro [H. Oueslati 2015].

Outro desafio identificado é que os testes de segurança são difíceis de automatizar [H. Oueslati 2015], necessitando que os testes sejam feitos, muitas vezes, por terceiros, o que torna uma atividade demorada e custosa. Portanto, a atividade pode chegar a não acontecer devido estes impedimentos.

Para que seja possível obter um *software* seguro, é preciso abordar a segurança em um estágio inicial do desenvolvimento, projetando o *software* voltado para a segurança, realizando análises dos ativos e identificando quais são os possíveis riscos, aplicando testes de segurança e também educando os desenvolvedores, arquitetos e usuários sobre a segurança de um sistema [G. McGraw 2004].

2.2. Modelagem de Ameaças

A modelagem de ameaças é uma etapa muito importante dentro da segurança de *software*, dito pela Microsoft como fundamental para o seu ciclo de desenvolvimento seguro [Microsoft 2023, Security Development Lifecycle].

O *threat modeling* se resume em identificar todos os ativos importantes para o sistema, o que é comumente feito a partir da confecção de um diagrama de fluxo de dados. A partir dos ativos encontrados, é realizada uma análise de cada um dos itens identificados com o intuito de listar possíveis ameaças, que pode ser realizada como um levantamento de ideias e discussões a respeito dos ativos ou utilizando *frameworks* bem

estabelecidos na literatura e no mercado, como STRIDE [Microsoft 2022, STRIDE], Owasp Top 10 [Owasp 2023], árvores de ataques, entre outros. Uma vez identificadas as possíveis ameaças, os participantes das sessões de *threat modeling* podem discutir quais são as melhores mitigações para cada ameaça, ou até definir algumas delas como risco aceito, determinando que nenhuma ação necessita ser tomada.

A utilização do *threat modeling* tem como objetivo aumentar a resiliência do sistema contra possíveis ameaças, devendo estar presente em todas as etapas do desenvolvimento de *software* e ser revisitado principalmente em momentos que novas funcionalidades são introduzidas no sistema, incidentes de segurança ocorram ou mudanças na arquitetura e na infraestrutura aconteçam [Owasp 2021].

Tomando como exemplo um sistema de comércio eletrônico, que envolve transações financeiras, informações pessoais e de pagamento, uma modelagem de ameaças poderia começar com a identificação de atores e ativos no sistema. Alguns atores podem ser usuários, funcionários, fornecedores e atacantes. Já os ativos, podem incluir dados de pagamento, informações de identidade dos usuários, estoque dos produtos, carrinhos de compras, sessões dos usuários, entre outros. Após identificados, é possível criar um diagrama de fluxo de dados do sistema, que ilustra como os atores e ativos interagem com o sistema. A imagem 1 ilustra um possível, e simplificado, diagrama de fluxo de dados.

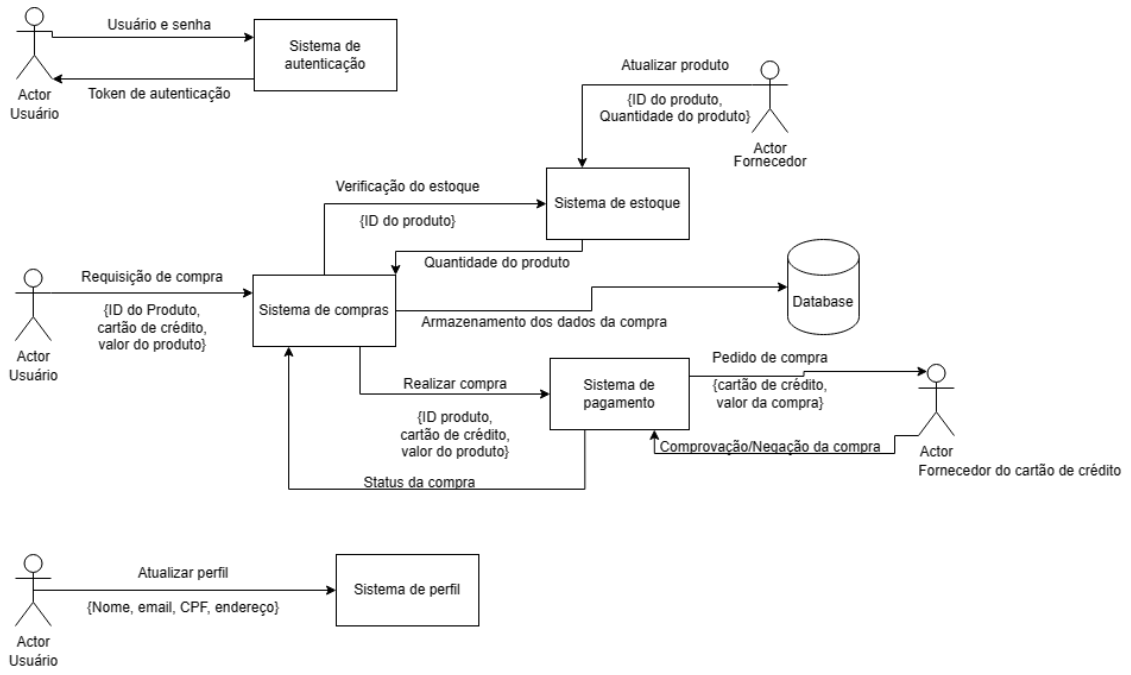

Imagem 1. Diagrama de fluxo de dados

Com base no diagrama de fluxo, podem ser identificadas as possíveis ameaças ao sistema, como: um ataque, realizado por um *hacker*, ao sistema de pagamentos para obter produtos gratuitos, uma captura de dados de cartão de crédito por um funcionário mal intencionado, uma tentativa de capturar os dados sensíveis e pessoais de outros usuários, entre outros. Esta etapa geralmente utiliza-se, assim como já mencionado, um *framework* auxiliar, por exemplo o STRIDE.

Após a identificação das ameaças, é possível determinar quais controles de segurança devem ser implementados para mitigar cada ameaça. Por exemplo, os dados de cartão de crédito não devem estar disponíveis para os funcionários que gerenciam a aplicação, deve ser utilizado um mecanismo de autorização para acesso aos dados pessoais de cada usuários, os dados relacionados aos preços dos produtos devem ser determinados exclusivamente pelo servidor da aplicação, entre outros.

Diante dessas informações obtidas, os desenvolvedores poderão realizar a codificação do sistema com base nas recomendações de segurança e mitigação de ameaças identificadas durante o processo de *threat modeling*.

2.3. Métodos Ágeis

Os métodos ágeis passaram a ser utilizados após a realização de um encontro de dezessete desenvolvedores de *software* que estabeleceram princípios sobre como melhorar a entrega de *software* para os clientes. Com o encontro, um manifesto [Manifesto ágil 2001] foi confeccionado e alguns princípios foram estabelecidos. O manifesto possui as seguintes afirmações sobre as prioridades da realização do desenvolvimento de *software*:

- Indivíduos e interações são mais importantes que processos e ferramentas.
- *Software* em funcionamento é mais importante que documentação abrangente.
- Colaboração com o cliente é mais importante que negociação de contratos.
- Responder a mudanças é mais importante que seguir um plano.

Essas sentenças são fundamentais para os princípios que abrangem o desenvolvimento ágil de *software* e com elas os desenvolvedores procuram entregar *softwares* de forma mais rápida, interativa e que consiga responder às mudanças propostas pelos clientes.

2.4. Trabalhos Relacionados

A utilização do *threat modeling* em metodologias ágeis não é uma tarefa simples, pois algumas atividades necessárias para o *threat modeling* vão de encontro com alguns princípios do desenvolvimento ágil de *software*, por exemplo possuir um *software* funcional em detrimento de uma documentação abrangente [Manifesto Ágil 2001], uma vez que a falta de uma documentação satisfatória do sistema torna a etapa de identificação de ameaças e vulnerabilidades inviável ou extremamente custosa [Karin Bernsmed 2022].

Devido a importância de ambos para o desenvolvimento de *software* na atualidade, alguns estudos foram realizados a respeito da melhor forma de incluir a modelagem de ameaças em metodologias ágeis. Karin Bernsmed et al 2019, fizeram um estudo sobre a abordagem de quatro empresas da Noruega em relação à modelagem de ameaças no desenvolvimento ágil, Daniela Cruzes et al 2018 trouxeram quais são os principais desafios dessa prática.

Também foi realizada uma Revisão Sistemática da Literatura (RSL) a respeito do estado da arte do *threat modeling* por Wenjun Xiong e Robert Lagerström (2019). As perguntas de pesquisa realizadas neste trabalho foram: "o que é *threat modeling*?" e "qual o estado da arte do *threat modeling*?". Para responder a essas perguntas, foram pesquisados artigos nos motores de busca IEEE Xplore[1], Scopus[2], Springer link[3], e Web of Science[4] utilizando as palavras chaves "*threat model*" e "*threat modeling*" sem nenhuma limitação de tempo para a busca, que resultou, após utilizar os critérios de exclusão e inclusão, em 54 artigos selecionados para uma maior análise, os quais foram divididos em três (3) categorias: artigos que citam as aplicações do *threat modeling*, os métodos na realização de *threat modeling* e o processo do *threat modeling*. As suas principais contribuições são as informações de que a modelagem de ameaças ainda é muito diversificada, sendo utilizada de diversas maneiras e que muito trabalho é feito de forma manual, o que traz um consumo grande de tempo e resulta em uma desmotivação na realização dessas atividades.

Outra RSL foi realizada por H. Oueslati et al 2015 a respeito dos desafios de desenvolver um *software* seguro em uma abordagem ágil. Nos seus estudos, foram realizadas as perguntas de pesquisa: "quais são os desafios de desenvolver *software* seguro usando a abordagem ágil?" e "os desafios encontrados são válidos?". Para a obtenção dos artigos da RSL, foram utilizadas as palavras chaves "*secure software agile*" nos motores de busca IEEE Xplore e ACM Digital Library[5] sem nenhum limite de tempo para a busca, que resultou em 28 artigos. Após utilizar os critérios de inclusão e exclusão, e utilizar a técnica de *snowballing*, foram selecionados 10 artigos que poderiam contribuir para a revisão sistemática da literatura. Como resultado, 20 desafios foram identificados, dos quais apenas 14 estão relacionados à práticas de segurança ou ao desenvolvimento ágil. O estudo traz a comprovação das dificuldades de introduzir segurança em uma metodologia de desenvolvimento ágil, como também uma necessidade de mais estudos que abordam os desafios encontrados. Seu estudo tem como foco os problemas de introduzir segurança no desenvolvimento ágil, discutindo tanto os problemas de algumas práticas de segurança, mas também problemas como os métodos ágeis não terem exigências em segurança, o que difere do estudo aqui proposto, que traz como foco os problemas e práticas da modelagem de ameaças no desenvolvimento ágil.

3. Método de Pesquisa

Neste trabalho, uma Revisão Sistemática da Literatura foi realizada com o objetivo de compreender qual o cenário atual do uso de *threat modeling* em metodologias ágeis e, a partir disso, entender e decidir quais são conteúdos importantes para serem abordados em uma disciplina de segurança da informação.

---

[1] ieeexplore.ieee.org/
[2] www.scopus.com/
[3] www.link.springer.com/
[4] www.webofknowledge.com/
[5] dl.acm.org/

3.1. Revisão Sistemática da Literatura

A RSL é uma revisão da literatura que segue um protocolo pré definido e rigoroso. Uma RSL tem como objetivo analisar, interpretar, sintetizar e avaliar uma determinada área de pesquisa, quais são seus debates, suas lacunas e suas principais tendências [Kitchenham, B. A. & Charters 2007]. A Revisão Sistemática da Literatura é uma atividade importante para uma pesquisa acadêmica, pois ela permite uma compreensão mais profunda acerca do estado atual da pesquisa em sua área de estudo, possibilitando o desenvolvimento de questionamentos com base no que foi encontrado, a identificação de áreas que devem ser investigadas e também a justificativa de outros estudos.

Neste trabalho, a revisão de literatura foi realizada seguindo os passos descritos em Mariana Peixoto e Carla Silva 2017. Como estratégia de busca, foi utilizada uma busca automática, com os motores ACM Digital Library [ACM Digital Library 2023] e ScienceDirect [ScienceDirect 2023]. Em seguida, foi realizado um processo de criação dos critérios de seleção, para que os artigos que serão selecionados na revisão tenham relevância no estudo [Kitchenham, B. A. & Charters 2007]. Após a confecção dos critérios, o processo de seleção teve início, com a leitura dos resumos dos artigos. A partir dos artigos selecionados, foi utilizado o método de *snowballing*, uma técnica de busca manual, em que realizou-se novamente a etapa de seleção dos artigos, nas referências e citações dos artigos já selecionados. Uma vez obtidos todos os artigos selecionados, foi realizada uma extração de dados, com o objetivo de identificar os dados necessários para responder às questões da revisão [Kitchenham, B. A. & Charters 2007].

3.2. Contextualização

A princípio, a pergunta principal que abordaria esta revisão de literatura seria:

Qual o estado atual do ensino de *threat modeling* em metodologias ágeis?

Porém, nenhum dos artigos encontrados durante a revisão de literatura inicial abordaram o ensino atual de *threat modeling* em metodologias ágeis. Com isso, houve a necessidade de obter mais informações sobre o cenário do uso do *threat modeling* em metodologias ágeis para que fosse desenvolvida uma compreensão acerca dos assuntos mais importantes a serem abordados em uma disciplina de segurança da informação.

Portanto, a pergunta principal que foi utilizada para a revisão de literatura realizada foi:

Qual o estado atual do uso de *threat modeling* em metodologias ágeis?

A partir dessa pergunta, foram estabelecidas outras subperguntas para auxiliarem no entendimento das informações que viriam a ser encontradas. Elas são:

- Quais são os desafios e boas práticas do uso de *threat modeling* em *agile development*?
- Em qual momento é feito o *threat modeling* em metodologias ágeis?
- Quais ferramentas estão sendo utilizadas para facilitar a modelagem de ameaças?

- Quais conteúdos poderiam fazer parte do ensino de *threat modeling* numa disciplina de segurança da informação?

3.2. Escolha dos artigos

A seleção dos artigos foi realizada através de uma revisão sistemática de literatura [Mariana Peixoto e Carla Silva 2017]. Primeiro, para a obtenção dos artigos a serem analisados, foi construída uma *string* de busca, com base no PICO confeccionado (Imagem 2) e nas perguntas de pesquisa, que teve o seu valor como:

("agile methods" or "agile methodologies" or "agile development") and ("threat modeling" or "threat modelling")

Seu modelo mais genérico se deu pelo fato da escassa quantidade de artigos ao realizar a pesquisa anteriormente citada.

Utilizando a *string* de busca desenvolvida, foram pesquisados artigos nas ferramentas de busca ACM digital library [ACM Digital Library 2023] e ScienceDirect [ScienceDirect 2023], que nos retornaram 48 (quarenta e oito) artigos científicos[6].

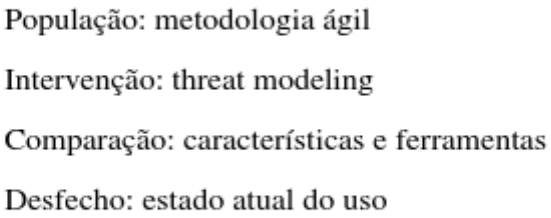

Imagem 2. PICO desenvolvido

Para compor a lista de artigos que seriam incluídos na revisão de literatura, foram criados alguns critérios de inclusão e exclusão, caso algum dos critérios de inclusão **não** fosse atingido, o artigo não seria incluído na lista de artigos que serão utilizados, e caso algum critério de exclusão fosse atingido o artigo também não seria incluído na lista. Os critérios utilizados foram:

Inclusão:

- Estudos que foram escritos em português ou em inglês.
- Estudos acessíveis.
- Estudos empíricos

[6] https://github.com/rcrs4/Planilha-de-artigos/

Exclusão:

- Estudos incompletos (short papers, meta-análises).
- Estudos que não abordam sobre metodologias ágeis.
- Estudos que não abordam sobre *threat modeling*.
- Estudos duplicados.

Após serem estabelecidos os critérios de inclusão e exclusão, foi realizada a leitura do resumo e conclusão de todos os artigos para verificar quais deles seriam incluídos com base nesses critérios.

Como parte do processo de seleção dos artigos, foi decidido que a técnica de *snowballing* seria utilizada para obtenção de uma maior quantidade de documentos. Portanto, foi lido o resumo e conclusão de todas as referências de cada artigo selecionado e também o resumo dos artigos que os citam, para que pudessem ser utilizadas as mesmas métricas de inclusão e exclusão. Além disso, devido à pequena quantidade de artigos selecionados, houve a inclusão de 9 (nove) artigos que abordam de forma mais genérica sobre segurança em metodologias ágeis, pois poderiam conter dados que auxiliam na resposta das perguntas realizadas. Um diagrama contendo as etapas da RSL se encontra na imagem 3.

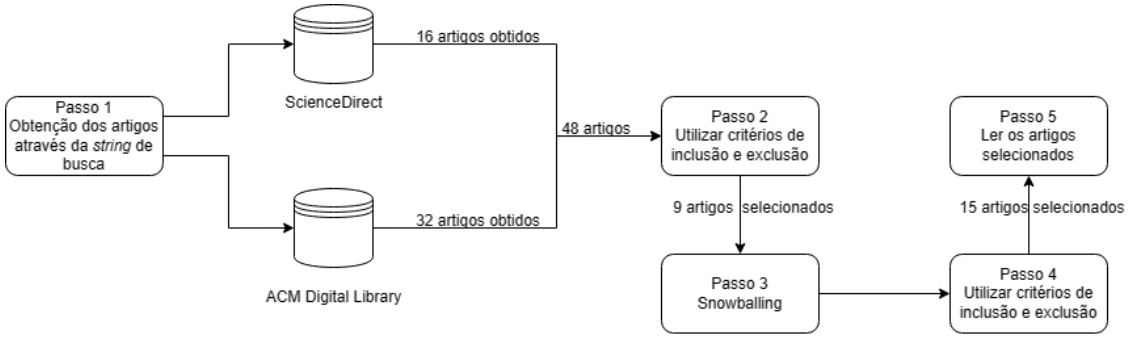

Imagem 3. Diagrama das etapas do RSL

O resultado da etapa de escolha de artigos se deu pela seleção de 15 (quinze) artigos que fazem referência à modelagem de ameaças em um ambiente que faz uso de metodologias de desenvolvimento ágil de *software*. Os artigos escolhidos estão listados na tabela 1. Os artigos que abordaram de forma mais genérica a segurança de informação foram marcados com um asterisco ao lado do identificador.

Tabela 1. Lista dos artigos selecionados

Identificador	Artigo
acm1*	Maria Teresa Baldassarre, Vita Santa Barletta, Danilo Caivano, and Antonio Piccinno. 2021. Integrating Security and Privacy in HCD-Scrum. In CHItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter (CHItaly '21). Association for Computing Machinery, New York, NY, USA, Article 37, 1–5. https://doi.org/10.1145/3464385.3464746

Tabela 1. Lista dos artigos selecionados (continuação)

Identificador	Artigo
acm2*	Daniele Granata, Massimiliano Rak, and Giovanni Salzillo. 2022. MetaSEnD: A Security Enabled Development Life Cycle Meta-Model. In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22). Association for Computing Machinery, New York, NY, USA, Article 152, 1–10. https://doi.org/10.1145/3538969.3544463
acm3*	Kalle Rindell, Sami Hyrynsalmi, and Ville Leppänen. 2018. Aligning security objectives with agile software development. In Proceedings of the 19th International Conference on Agile Software Development: Companion (XP '18). Association for Computing Machinery, New York, NY, USA, Article 3, 1–9. https://doi.org/10.1145/3234152.3234187
acm14*	Jessica Nguyen and Marc Dupuis. 2019. Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations. In Proceedings of the 20th Annual SIG Conference on Information Technology Education (SIGITE '19). Association for Computing Machinery, New York, NY, USA, 93–98. https://doi.org/10.1145/3349266.3351420
acm27	Sigrid Marita Kvamme, Espen Gudmundsen, Tosin Daniel Oyetoyan, and Daniela Soares Cruzes. 2023. Data Protection Fortification: An Agile Approach for Threat Analysis of IoT Data. In Proceedings of the 12th International Conference on the Internet of Things (IoT '22). Association for Computing Machinery, New York, NY, USA, 151–154. https://doi.org/10.1145/3567445.3569164
science3	Valentina Casola, Alessandra De Benedictis, Massimiliano Rak, Umberto Villano,A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach, Journal of Systems and Software, Volume 163, 2020, 110537, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2020.110537.
science4*	Inger Anne Tøndel, Daniela Soares Cruzes, Continuous software security through security prioritisation meetings, Journal of Systems and Software, Volume 194, 2022, 111477, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2022.111477.
science15	Karin Bernsmed, Daniela Soares Cruzes, Martin Gilje Jaatun, Monica Iovan, Adopting threat modelling in agile software development projects, Journal of Systems and Software, Volume 183, 2022, 111090, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2021.111090.

Tabela 1. Lista dos artigos selecionados (continuação)

Identificador	Artigo
science16*	Kalle Rindell, Jukka Ruohonen, Johannes Holvitie, Sami Hyrynsalmi, Ville Leppänen, Security in agile software development: A practitioner survey, Information and Software Technology, Volume 131, 2021, 106488, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2020.106488.
ieee1	K. Bernsmed and M. G. Jaatun, "Threat modelling and agile software development: Identified practice in four Norwegian organisations," 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 2019, pp. 1-8, doi: 10.1109/CyberSecPODS.2019.8885144.
acm33*	Kalle Rindell, Sami Hyrynsalmi, and Ville Leppänen. 2017. Busting a Myth: Review of Agile Security Engineering Methods. In Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES '17). Association for Computing Machinery, New York, NY, USA, Article 74, 1–10. https://doi.org/10.1145/3098954.3103170
ieee2	D. Soares Cruzes, M. Gilje Jaatun, K. Bernsmed and I. A. Tøndel, "Challenges and Experiences with Applying Microsoft Threat Modeling in Agile Development Projects," 2018 25th Australasian Software Engineering Conference (ASWEC), Adelaide, SA, Australia, 2018, pp. 111-120, doi: 10.1109/ASWEC.2018.00023.
ieee3*	H. Oueslati, M. M. Rahman and L. b. Othmane, "Literature Review of the Challenges of Developing Secure Software Using the Agile Approach," 2015 10th International Conference on Availability, Reliability and Security, Toulouse, France, 2015, pp. 540-547, doi: 10.1109/ARES.2015.69.
science17*	Inger Anne Tøndel, Daniela Soares Cruzes, Martin Gilje Jaatun, Guttorm Sindre, Influencing the security prioritisation of an agile software development project, Computers & Security, Volume 118, 2022, 102744, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2022.102744.
mdpi1	de Vicente Mohino J, Bermejo Higuera J, Bermejo Higuera JR, Sicilia Montalvo JA. The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies. *Electronics*. 2019; 8(11):1218. https://doi.org/10.3390/electronics8111218

3.3. Leitura dos artigos

A etapa final da metodologia se compreendeu em ler todos os documentos selecionados e realizar anotações das informações mais relevantes, tanto para as perguntas aqui realizadas, quanto para o entendimento geral dos assuntos propostos na revisão de literatura.

3.4. Ameaças à Validade do Estudo

Algumas das ameaças descritas por Wohlin et al. 2012 tentaram ser enfrentadas, sendo elas:

- A ameaça interna, em que ao surgir uma dúvida sobre a inclusão de um artigo, o mesmo seria incluído, devido a essa abordagem, alguns artigos que discorrem sobre segurança da informação de forma mais genérica foram incluídos
- A ameaça construtiva, em que foram utilizados sinônimos das palavras chaves para compor a *string* de busca. Porém, essa ameaça talvez não tenha sido enfrentada de uma forma satisfatória, devido à falta de mais sinônimos que podiam compor a *string* de busca descrita neste trabalho, como “modeling security”, “modeling vulnerabilities”, “security analysis", “ threat analysis", entre outros.
- A ameaça de confiabilidade pôde ser enfrentada ao utilizar um protocolo de RSL construído com base em outros estudos científicos.

Em contrapartida, a ameaça externa não foi abordada de forma significativa e não possui garantias a seu respeito.

4. Resultados

Neste tópico serão apresentadas as respostas obtidas para cada pergunta da pesquisa, após a realização da leitura dos artigos selecionados, sendo cada pergunta dividida em um subtópico, também será introduzida uma proposta de um roteiro de ensino de modelagem de ameaças.

4.1. Quais são os desafios e boas práticas do uso de *threat modeling* em *agile development*?

Diversos desafios e boas práticas foram identificados durante a revisão da literatura. Os desafios que foram encontrados em uma quantidade maior de artigos foram: a falta de motivação dos desenvolvedores em realizar as atividades, principalmente na confecção de Data Flow Diagrams (DFDs), que são diagramas contendo os ativos do sistema, suas interações com outros sistemas, como os dados são distribuídos, entre outros; e o tempo levado para que as atividades de *threat modeling* fossem concluídas [Karin Bernsmed 2022], [Karin Bernsmed and M. G. Jaatun 2019], [D. Soares Cruzes 2018]. A imagem 4 ilustra um gráfico da quantidade de artigos que identificam cada desafio.

Além desses desafios principais, outros desafios trazem dificuldades no uso de *threat modeling* em desenvolvimentos ágeis, como o gerenciamento e atualização dos DFDs, já que os desenvolvedores não estão acostumados a documentar nada, visto que é um dos princípios descritos no manifesto ágil [Manifesto ágil 2001]. A tabela 2 demonstra os desafios encontrados e também quais artigos os citaram. Os desafios que foram julgados como particulares para o cenário proposto pelo artigo ou que não tiveram relação com as atividades de *threat modeling* não foram incluídos.

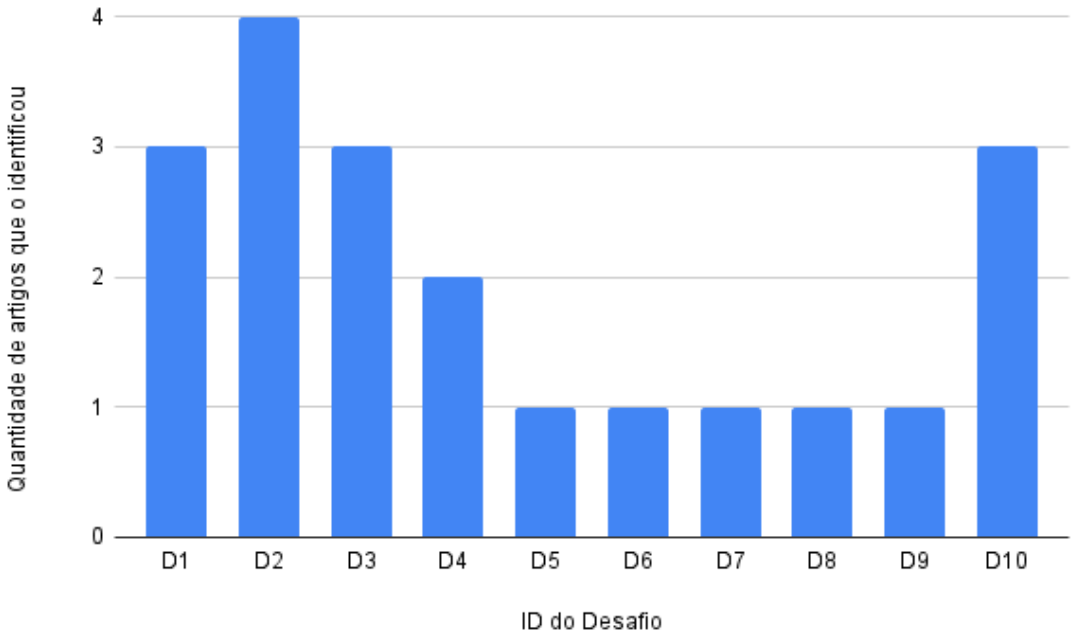

Imagem 4. Número de artigos por desafio

Em sua maioria, os desafios mostraram-se relacionados à construção dos DFDs. Dentre eles, notou-se que alguns poderiam ser evitados a partir de um maior conhecimento dos desenvolvedores sobre o processo de confecção destes DFDs. Problemas como falta de motivação estavam, em algumas empresas estudadas, ligados possivelmente à falta de entendimento do porquê estão realizando essas atividades, visto que as mesmas empresas que descreveram a falta de motivação, também informaram que um dos desafios era entender a relevância e qual o uso dos DFDs. Portanto, algumas das dificuldades poderiam ser resolvidas através do ensino de *threat modeling* para os desenvolvedores.

Tabela 2. Desafios identificados e quais artigos os identificaram

Identificador	Desafio identificado	Quais artigos o identificou
D1	Falta de motivação dos desenvolvedores para realizarem as atividades	science15, ieee1, ieee2
D2	Tempo necessário para realizar as atividades de *threat modeling* são altos	science15, ieee1, ieee2, ieee3
D3	Difícil compreensão sobre quais elementos devem ser incluídos nos DFDs e quão detalhada devem ser as informações	science15, ieee1, ieee2
D4	O gerenciamento de DFDs se torna difícil pela falta de costume em documentação em metodologias ágeis	science15, ieee3
D5	Os desenvolvedores não sabem qual a usabilidade das atividades, nem dos resultados	science15

Tabela 2. Desafios identificados e quais artigos os identificaram (continuação)

Identificador	Desafio identificado	Quais artigos o identificou
D6	Difícil identificação das ameaças e compreender quais delas são relevantes	ieee1
D7	Ter a perspectiva de um atacante	ieee1
D8	Entender quando deve ser realizado novamente o *threat modeling*	science15
D9	O STRIDE se mostrou muito focado na identificação de ameaças em canais de comunicação	ieee2
D10	Falta de conhecimento dos desenvolvedores acerca de segurança da informação	science15, ieee3, ieee1

Apesar de um dos maiores desafios ser o custo temporal para realizar as atividades de *threat modeling*, principalmente quando falamos da etapa de confecção das DFDs, a única boa prática que se repetiu dentre os artigos foi a realização das atividades de *threat modeling* de forma regular. A imagem 5 indica a quantidade de artigos que identificaram cada uma das boas práticas.

Dentre as boas práticas encontradas, algumas se mostraram, assim como dito anteriormente, contraditórias com os desafios. Isso demonstra que apesar de ser uma prática difícil em um ambiente de desenvolvimento ágil, os times de desenvolvimento conseguem ter a percepção de que algumas práticas são benéficas para a empresa e para o time. As boas práticas identificadas estão expostas na tabela 3.

Tabela 3. Boas práticas identificadas e quantidade de artigos que as identificaram

Identificador	Boa prática identificada	Quais artigos a identificou
B1	Realizar atividades de modelagem de ameaças de forma regular	ieee1, science15
B2	Realizar a confecção dos DFDs	science15
B3	Utilizar os DFDs construídos para o *onboarding* de novos desenvolvedores	science15
B4	Analisar todas as interações do sistema	science15
B5	Envolver um especialista em segurança	science15
B6	Envolver os desenvolvedores nas atividades	ieee1
B7	Utilizar *checklists* dos tópicos a serem discutidos	ieee1
B8	Ter processos e rotinas claras	ieee1

4.2. Em qual momento é feito *threat modeling* em metodologias ágeis?

Esta pergunta tem como objetivo identificar quando os times de desenvolvimento ágil decidem realizar atividades relacionadas a modelagem de ameaças.

À medida que novos *softwares* são desenvolvidos, e que novos requisitos são incorporados no sistema, novos problemas de segurança podem surgir. Portanto, a prática de *threat modeling* não deve ser realizada apenas uma vez, ela deve ter uma constância, a fim de sempre entregar *softwares* mais seguros.

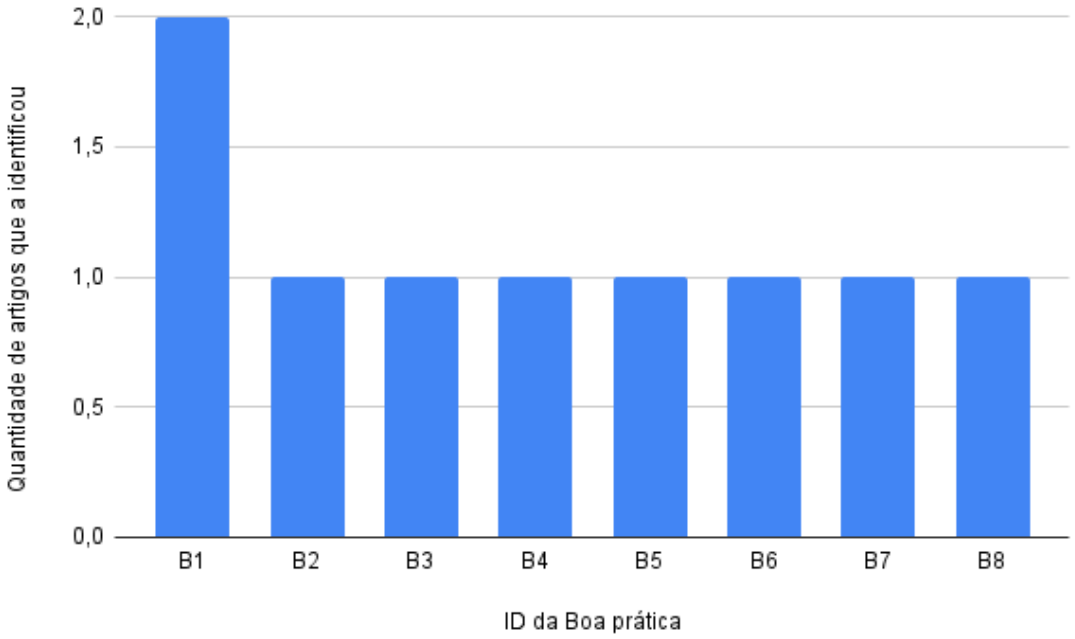

Imagem 5. Número de artigos por boa prática

A realização de *threat modeling* em intervalos regulares de tempo é uma prática tida como benéfica para empresas que utilizam de metodologias ágeis [Karin Bernsmed 2022]. Porém, apesar de ser uma atividade benéfica, não foi encontrado um estudo que informasse um padrão, ou que propusesse um padrão, para a realização de sessões de *threat modeling* dentro das empresas de desenvolvimento de *software*.

O momento mais comum para a realização de novas sessões de modelagem de ameaças é ao serem incluídas grandes mudanças no *software* desenvolvido [Karin Bernsmed 2019] [Owasp 2023], nos estudos conduzidos por Karin Bernsmed et al 2019 uma das empresas entrevistadas também realizava atividades de *threat modeling* em determinados cenários de segurança que poderiam afetar a empresa, por exemplo ao acontecerem incidentes em uma empresa concorrente. Por outro lado, também foram identificadas organizações que exigem que a atividade seja feita apenas uma vez a cada ano [Karin Bernsmed 2022].

Apesar das tentativas citadas de estabelecer um momento para a realização do *threat modeling*, elas podem ser consideradas falhas pelo excesso de abstração. O entendimento de grandes mudanças no *software* varia de acordo com a empresa e o desenvolvedor, podendo ser após a implementação de uma funcionalidade nova ou

apenas após uma mudança de infraestrutura do sistema, por exemplo ao mudar o ambiente em que o sistema será hospedado, o que torna esse método difícil de ser seguido, principalmente para empresas que estão iniciando com o uso de *threat modeling*. A abordagem anual também apresenta possíveis falhas, pois a quantidade de *software* produzido dentro de um ano provavelmente é alto, o que introduz possíveis vulnerabilidades, que serão abordadas apenas após um ano. Além disso, com a alta quantidade de mudanças que ocorrem dentro de um ano, as sessões de *threat modeling* custarão muito tempo para serem concluídas, o que pode tornar as sessões desmotivadoras, que é um dos desafios encontrados durante a revisão de literatura.

Contudo, não foi encontrado na literatura qual o melhor momento para serem realizadas atividades relacionadas a *threat modeling*, sendo ainda um assunto que necessita de maiores estudos.

4.3. Quais ferramentas estão sendo utilizadas para facilitar a modelagem de ameaças?

Este tópico abordará quais são as principais ferramentas utilizadas por desenvolvedores, identificadas durante a RSL.

O uso de ferramentas se mostrou muito positivo ao realizar atividades relacionadas à *threat modeling*. Em um estudo feito com estudantes de duas universidades [Karin Bernsmed 2022], foi possível comparar a importância de ferramentas para auxiliarem na modelagem de ameaças. Ao separar os estudantes em dois grupos, um que realizou atividades de *threat modeling* utilizando apenas caneta e papel e outro que utilizou a ferramenta Microsoft Threat Modeling Tool (MS-TMT) [Microsoft 2022, Threat Modeling Tool], notou-se que os alunos que utilizaram a ferramenta foram mais otimistas do que os alunos que realizaram as atividades utilizando apenas caneta e papel, ao afirmarem que realizar o *threat modeling* é uma tarefa fácil e que exige pouco esforço mental.

Quatro ferramentas utilizadas por desenvolvedores ágeis foram identificadas durante a revisão de literatura, sendo elas:

Microsoft Threat Modeling Tool: esta ferramenta possui vários módulos, dentre os quais é possível realizar a confecção de Data Flow Diagrams (DFD), os DFDs são amplamente utilizados nas atividades de *threat modeling* por auxiliar na identificação do fluxo dos dados do sistema, e assim, identificar possíveis ameaças. Além dos diagramas, é possível gerar uma lista com as ameaças identificadas através do STRIDE, priorizar as ameaças identificadas e gerar relatórios [Microsoft 2022, Threat Modeling Tool Features].

Apesar da MS-TMT ser a ferramenta mais utilizada em *threat modeling* [D. Soares Cruzes 2018], não foram mencionadas, nos estudos, muitas vantagens em relação a outras ferramentas, as únicas vantagens identificadas foram: a possibilidade de adicionar quais são as propriedades de segurança de cada elemento descrito no DFD e a sua facilidade do entendimento dos resultados gerados e consequentemente na apresentação dos resultados para terceiros [Karin Bernsmed 2022]. Por outro lado, diversas desvantagens foram apontadas pelos desenvolvedores de *software* que utilizam a ferramenta, como a dificuldade de utilizá-la em colaboração com outros integrantes do

time; a limitação da lista de elementos pré-definidos, que não contém componentes muito utilizados em sistemas atuais, principalmente em ambientes *cloud*; a falta de um fluxo de dados bidirecional; a possibilidade de utilizar a ferramenta apenas no sistema operacional Windows; e, por fim, a alta complexidade e dificuldade na leitura ao gerenciar diagramas muito grandes [Karin Bernsmed 2022].

Draw.io: o draw.io [Draw.io 2023] é uma ferramenta de confecção de diagramas, com ela é possível começar a criação de diagramas sem nenhum elemento ou utilizando *templates* que podem facilitar na criação dos diagramas. É uma ferramenta versátil, com diversos elementos pré-definidos.

As vantagens mencionadas ao utilizar o draw.io é a sua maior facilidade de desenhar DFDs em colaboração com outros integrantes, sendo um dos pontos mais fortes em comparação com o MS-TMT, e também a possibilidade de utilizar a ferramenta em qualquer sistema operacional, visto que ela é acessada através de uma aplicação *web* [Karin Bernsmed 2022].

Não foi mencionado quais seriam as desvantagens em relação ao draw.io, porém é notável que algumas dificuldades podem ser encontradas ao utilizá-la, como a falta de foco em segurança, a falta de realizar outras atividades de *threat modeling* além da confecção de DFDs e, para algumas empresas e desenvolvedores, pode ser difícil de começar a desenhar DFDs sem um conhecimento prévio, assim como descrito em **desafios e boas práticas**, o que tornar a ferramenta mais difícil nesses cenários, já que ela não tem foco a construção de DFDs e sim em diagramas genéricos.

The Elevation of Privilege (EoP) card game: Esta ferramenta, também desenvolvida pela Microsoft [Adam Shostac 2014], tem como objetivo ajudar na identificação de ameaças dentro do sistema, de uma forma mais descontraída e gamificada.

Não foram encontradas vantagens ou desvantagens no uso da ferramenta nos artigos analisados, porém essa ferramenta é bastante focada na identificação de ameaças e não ajudará na confecção de diagramas de fluxo de dados. Dito isso, por ser uma atividade importante, será preciso utilizar outras ferramentas para auxiliar na construção de DFDs.

VIS-PRISE (Visually Inspection to Support Privacy and Security): O VIS-PRISE é uma ferramenta proposta por Maria Teresa Baldassarre et al 2022, que visa adicionar aspectos de segurança em todas as iterações de um ciclo de desenvolvimento ágil. Em seu estudo, um grupo de estudantes sem conhecimento de segurança e com pouco conhecimento de metodologias ágeis, foi capaz de realizar algumas tarefas de identificação de ameaças.

A colaboração entre os usuários da ferramenta se mostrou como uma vantagem da ferramenta [Maria Teresa Baldassarre 2021], outra vantagem mencionada seria a possibilidade de adicionar novas vulnerabilidades [Maria Teresa Baldassarre 2022], uma funcionalidade não disponível, por exemplo, no **MS-TMT** e no **The Elevation of Privilege (EoP) card game**, que ambos utilizam o STRIDE, para identificação de vulnerabilidades.

Apesar da ferramenta não ser específica para atividades de *threat modeling*, o que pode ser considerado uma desvantagem, ela pode auxiliar na identificação de ameaças, que é uma das etapas do *threat modeling*. Caso o uso de ferramentas como o draw.io já seja uma prática comum pelos desenvolvedores, o VIS-PRISE pode ser uma opção na identificação de vulnerabilidades a partir dos diagramas construídos.

4.4. Quais conteúdos poderiam fazer parte do ensino de *threat modeling* numa disciplina de segurança da informação?

Os artigos vistos não abordam sobre o ensino de *threat modeling* em desenvolvimento ágil, apenas dois deles trouxeram um estudo com alunos de universidades, porém com foco em entender algumas características da ferramenta MS-TMT [Karin Bernsmed 2022] e demonstrar as vantagens da ferramenta desenvolvida (VIS-PRISE) [Maria Teresa Baldassarre 2022]. Apesar disso, é possível abordar o assunto a partir das respostas obtidas nas perguntas anteriores e de algumas afirmações encontradas na leitura dos artigos.

Uma das afirmações mencionadas foi que era necessário esclarecer quais são os benefícios de *threat modeling* para os times de desenvolvimento, não só isso como também qual a continuação do processo de *threat modeling* para tornar o *software* produzido mais seguro [Karin Bernsmed 2022], assim como descrito nos desafios apresentados anteriormente. Portanto, o primeiro tópico que deve ser abordado em sala de aula é qual a importância do *threat modeling* para a segurança do sistema e qual a importância de abordar segurança ainda na fase de *design* do sistema.

Outro assunto muito abordado foi a falta de entendimento acerca dos DFDs, como eles poderiam ser utilizados em discussões de segurança, quais os níveis de detalhamento que deveria ser empregado ao construir os diagramas, o tempo que se gastava para a confecção desses DFDs e até para o que eles serviriam. Apesar de não ser uma atividade mandatória no *threat modeling*, a construção de DFDs é altamente recomendada, sendo parte do Microsoft Threat Modeling Framework [Microsoft 2023, Threat Modeling Framework]. Com isso, um tópico que se identifica muito necessário é o que são DFDs, como devem ser realizadas as construções dos DFDs e como devem ser realizadas as suas atualizações, pois é um dos tópicos identificados como boas práticas.

A falta de entendimento dos resultados obtidos após sessões de *threat modeling* também se mostrou um problema relevante, muitas vezes desmotivando os desenvolvedores por não saberem qual o sentido de realizar algo que não seria utilizado posteriormente. Portanto, o aprendizado de como transpor os resultados em código seguro pode ser valioso para os desenvolvedores.

Um dos desafios abordados foi entender a perspectiva de um atacante [Karin Bernsmed 2022], no que diz respeito a quais ferramentas e recursos um atacante poderia ter. Neste caso, a relevância de entender as ferramentas de um atacante pode não ser muito alta, pois as ferramentas serão inutilizadas uma vez que as vulnerabilidades são mitigadas. Porém, ainda é importante ter a visão de como um atacante pode utilizar os ativos em seu benefício e qual seria o objetivo principal de um atacante ao tentar explorar vulnerabilidades naquele ativo. Entretanto, ter uma visão aprofundada de um

possível atacante pode levar muito tempo, e até não ser atingido por um desenvolvedor, mas apenas por um especialista em segurança. Neste caso, um tópico que pode ser abordado em salas de aula é a respeito de *frameworks* que auxiliam os desenvolvedores a identificar vulnerabilidades, como STRIDE e OWASP Top 10.

O uso de ferramentas mostrou ser uma questão de afinidade do time de desenvolvimento, então talvez não fosse um tópico interessante para ser abordado. Por outro lado, as ferramentas diminuem o tempo gasto nas atividades de *threat modeling* e podem ser úteis para os alunos. O uso da ferramenta **The Elevation of Privilege (EoP) card game** também pode ser muito útil no entendimento do STRIDE, que é o *framework* tomado como base do jogo. Além disso, por ser um ensino gamificado, é possível que tenha um engajamento alto por parte dos alunos.

4.5. Proposta de um roteiro de ensino de modelagem de ameaças

Com base nos resultados encontrados, um possível roteiro de ensino para modelagem de ameaças é:

- Introdução: deve ser introduzida a modelagem de ameaças e o que ela é, também deve ser mostrado, e frizado, quais são os principais benefícios e importâncias do uso contínuo de *threat modeling*, pois um dos principais desafios dos desenvolvedores era entender qual a importância e usabilidade da modelagem de ameaças [Karin Bernsmed 2022]. Além disso, um dos desafios mais abordados foi o de falta de motivação dos desenvolvedores ao realizar tais atividades [D. Soares Cruzes 2018], [Karin Bernsmed 2019], [Karin Bernsmed 2022], que, apesar de não ser informado de forma explícita pelos estudos, notou-se que parte dessa desmotivação é devido à falta de entendimento das atividades que estão sendo realizadas.
- Fundamentos de segurança da informação: para realizar algumas das atividades de *threat modeling*, por exemplo na etapa de descoberta de ameaças, é necessário ter um conhecimento, ao menos básico, em relação à segurança da informação. Esta falta de conhecimento foi vista como um desafio entre as empresas de *software* [Hela Oueslati 2015], [Karin Bernsmed 2019], [Karin Bernsmed 2022]. Com isso, é importante que seja abordado sobre fundamentos de segurança da informação.
- Quais os processos de modelagem de ameaças e da confecção de DFDs: ao obter um conhecimento fundamental sobre segurança da informação, é possível abordar sobre as etapas da modelagem de ameaças, como essas etapas são distribuídas e realizadas. Também deve ser abordado como realizar a confecção de um diagrama de fluxo de dados, por ser um tópico importante para a realização da modelagem de ameaças e por apresentar diversos desafios, como a difícil compreensão dos elementos que um DFD deve possuir e como atualizar estes DFDs, pela falta de costume em documentação [D. Soares Cruzes 2018], [Karin Bernsmed 2019], [Karin Bernsmed 2022].
- *Frameworks* de identificação de ameaças: um benefício indicado por Karin Bernsmed et al 2019 é o uso de *checklists* e de processos bem estruturados ao realizar a modelagem de ameaças. Devido a isso, é importante abordar sobre a análise de ameaças baseada em STRIDE, Owasp Top 10, PASTA, entre outros.

- Aplicação prática da modelagem de ameaças: por fim, é importante que seja realizada uma aplicação prática da modelagem de ameaças, para que um conhecimento mais profundo acerca dos passos que devem ser realizados e de como gerar uma lista de ameaças ocorra. Além disso, nessa etapa podem ser introduzidas algumas ferramentas de modelagem de ameaças, como MS-TMT, pois elas trarão uma maior agilidade no processo do *threat modeling*, sendo esta maior rapidez um ponto muito positivo, devido ao desafio da alta quantidade de tempo tomado para realizar atividades de modelagem de ameaças [Hela Oueslati 2015], [D. Soares Cruzes 2018], [Karin Bernsmed 2019], [Karin Bernsmed 2022].

5. Conclusão

Neste trabalho, foi realizada uma Revisão Sistemática da Literatura para entender o cenário atual da modelagem de ameaças em desenvolvimento ágil e para, a partir dos resultados obtidos, realizar uma proposta de ensino em modelagem de ameaças. As principais conclusões foram:

- O cenário do uso de *threat modeling* em metodologias ágeis ainda foi pouco explorado e necessita de mais estudos, principalmente no que diz respeito ao momento ou qual a frequência que devem ser realizadas as atividades de *threat modeling* dentro do desenvolvimento ágil;
- Ainda existem muitos desafios na junção das atividades de modelagem de ameaças e de desenvolvimento ágil e mais estudos devem ser realizados para obter informações sobre como enfrentar os desafios encontrados na literatura;
- Um dos desafios encontrados mais abordado foi a falta de conhecimento dos desenvolvedores a respeito da segurança da informação e das atividades de *threat modeling*. Ainda assim, não foi encontrado na literatura artigos que abordassem o ensino como seu foco. Trazendo a conclusão que é necessário realizar mais estudos sobre o ensino de *threat modeling* para os desenvolvedores de *software*.

5.1. Principais contribuições

Uma das principais contribuições deste trabalho foi a obtenção de uma visão geral do cenário da modelagem de ameaças em desenvolvimento ágil, trazendo seus principais desafios encontrados na literatura e quais as boas práticas que podem ser utilizadas ao realizar essas atividades. Além disso, o trabalho abordou sobre as ferramentas que podem ser utilizadas, ou que já estão sendo utilizadas por empresas, para auxiliar nas atividades de modelagem de ameaças.

Outra contribuição deste artigo foi a proposta de ensino desenvolvida com base nos resultados obtidos da RSL.

5.2. Limitações e dificuldades

Algumas dificuldades e limitações puderam ser identificadas durante a realização do trabalho, sendo elas:

- A pouca quantidade de artigos relacionados exclusivamente com a modelagem de ameaças, resultando em perguntas sem uma resposta clara da abordagem da literatura sobre as questões desenvolvidas;
- O uso de poucos sinônimos na *string* de busca, para realizar a busca de artigos. Essa limitação pode ter resultado na obtenção de menos artigos que se esperava e ter afetado nas respostas de algumas das perguntas realizadas durante o trabalho.
- A falta de validação da proposta de ensino desenvolvida e apresentada;
- A falta da ilustração de um modelo que utilize as ferramentas apresentadas no trabalho, resultando na falta de um pensamento crítico das vantagens e desvantagens acerca das ferramentas mencionadas.

5.3. Trabalhos futuros

Uma possibilidade de trabalhos futuros é a aplicação da proposta de ensino, para que seja validada, em conjunto com os alunos, realizando pesquisas sobre quais dos assuntos abordados foram sucedidos e quais devem ter uma reorganização e melhorias, como também novos assuntos que deveriam ser ensinados. Como também, uma validação em conjunto com professores da área de segurança da informação e uma consulta sobre a distribuição do conteúdo no plano de curso de uma disciplina de graduação.

Outro possível trabalho futuro é a realização de pesquisas para abordar os desafios encontrados nos resultados do trabalho, visto que existem artigos que abordam quais os desafios enfrentados por empresas e desenvolvedores, mas ainda não existem artigos que discorrem quais são as possíveis soluções para os problemas.

Devido às limitações do trabalho, também se vê como trabalho futuro uma nova RSL utilizando uma outra *string* de busca, que possa ter uma abrangência maior dos artigos obtidos.

Agradecimentos

Primeiramente, gostaria de agradecer a professora Carla pela disponibilidade e dedicação ao aceitar me orientar neste trabalho.

Aos meus pais, pelo incentivo contínuo no estudo e pela oportunidade que me deram em possuir um ensino de qualidade. Também agradeço pelo exemplo que foram de motivação e perseverança e pelo carinho durante todos os anos da minha vida.

A Lara, que foi muito mais do que uma parceira em praticamente todas as etapas da minha graduação, estava presente em todas as conquistas, dores, complicações e estresses. Agradeço por sempre me alegrar nas derrotas e comemorar as vitórias, por sempre me apoiar independentemente das escolhas tomadas e pelo amor que sempre me foi dado.

A Rodrigo, meu irmão, que durante toda a minha vida foi a maior inspiração que eu poderia ter, por ter me ensinado a tentar ser um exemplo em todas as atividades feitas e pelo amor e companheirismo que me foi dado durante todos esses anos.

Aos integrantes do Nomu, que estiveram presentes nos melhores momentos de toda a graduação, em particular a Thalisson e Gabriel, que estiveram presentes em toda a graduação.

A professora Edna Barros, pela crença nos projetos de diversos alunos, incluindo o Nomu, que sem dúvidas só foi possível pelo seu amor pelo ensino e pela dedicação no seu trabalho.

E não menos importante, a minha família, pelo amor e carinho que me proporcionam.

Referências

Microsoft 2023, Security Development Lifecycle https://www.microsoft.com/en-us/securityengineering/sdl/practices

Microsoft 2023, Threat Modeling Framework https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling

Microsoft 2022, Threat Modeling Tool, STRIDE https://learn.microsoft.com/pt-br/azure/security/develop/threat-modeling-tool-threats

Owasp 2023 https://owasp.org/www-project-top-ten/

Owasp 2021 https://owasp.org/www-community/Threat_Modeling

Manifesto ágil 2001 https://agilemanifesto.org/iso/ptbr/manifesto.html

Hannah Snyder (2019), Literature review as a research methodology: An overview and guidelines, Journal of Business Research, Volume 104.

ACM Digital Library 2023 https://dl.acm.org/

ScienceDirect 2023 https://www.sciencedirect.com/

Microsoft 2022, Threat Modeling Tool Features https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-feature-overview

Microsoft 2022, Threat Modeling Tool https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool

Draw.io 2023, https://www.draw.io/

D. Soares Cruzes, M. Gilje Jaatun, K. Bernsmed and I. A. Tøndel, "Challenges and Experiences with Applying Microsoft Threat Modeling in Agile Development Projects," *2018 25th Australasian Software Engineering Conference (ASWEC)*, Adelaide, SA, Australia, 2018, pp. 111-120, doi: 10.1109/ASWEC.2018.00023.

Adam Shostac, “Elevation of privilege: Drawing developers into threat modeling,” in 2014 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14), 2014.

Maria Teresa Baldassarre, Vita Santa Barletta, Giovanni Dimauro, Domenico Gigante, Alessandro Pagano, and Antonio Piccinno. 2022. Supporting Secure Agile Development: the VIS-PRISE Tool. In Proceedings of the 2022 International

Conference on Advanced Visual Interfaces (AVI 2022). Association for Computing Machinery, New York, NY, USA, Article 69, 1–3.

Maria Teresa Baldassarre, Vita Santa Barletta, Danilo Caivano, and Antonio Piccinno. 2021. Integrating Security and Privacy in HCD-Scrum. In CHItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter (CHItaly '21). Association for Computing Machinery, New York, NY, USA, Article 37, 1–5. https://doi.org/10.1145/3464385.3464746

de Vicente Mohino, Juan, Javier Bermejo Higuera, Juan Ramón Bermejo Higuera, and Juan Antonio Sicilia Montalvo. 2019. "The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies" *Electronics* 8, no. 11: 1218. https://doi.org/10.3390/electronics8111218

G. McGraw, "Software security," in *IEEE Security & Privacy*, vol. 2, no. 2, pp. 80-83, March-April 2004, doi: 10.1109/MSECP.2004.1281254.

Shawn Hernan, Scott Lambert, Tomasz Ostwald, and Adam Shostack. 2019. Uncover Security Design Flaws Using The STRIDE Approach. https://docs.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach

Karin Bernsmed, Daniela Soares Cruzes, Martin Gilje Jaatun, Monica Iovan, Adopting threat modelling in agile software development projects, Journal of Systems and Software, Volume 183, 2022, 111090, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2021.111090.

K. Bernsmed and M. G. Jaatun, "Threat modelling and agile software development: Identified practice in four Norwegian organisations," *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Oxford, UK, 2019, pp. 1-8, doi: 10.1109/CyberSecPODS.2019.8885144.

H. Oueslati, M. M. Rahman and L. b. Othmane, "Literature Review of the Challenges of Developing Secure Software Using the Agile Approach," *2015 10th International Conference on Availability, Reliability and Security*, Toulouse, France, 2015, pp. 540-547, doi: 10.1109/ARES.2015.69.

Kitchenham, B. A. & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering* (EBSE 2007-001). Keele University and Durham University Joint Report .

Wenjun Xiong, Robert Lagerström, Threat modeling – A systematic literature review, Computers & Security, Volume 84, 2019, Pages 53-69, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2019.03.010.

Mariana Peixoto and Carla Silva. 2017. A gamification requirements catalog for educational software: results from a systematic literature review and a survey with experts. In Proceedings of the Symposium on Applied Computing (SAC '17). Association for Computing Machinery, New York, NY, USA, 1108–1113. https://doi.org/10.1145/3019612.3019752.

Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. Experimentation in software engineering. Springer Science & Business Media.