RMP
Revista dos Mestrados Profissionais
UFPE / CCSA – MGP
https://periodicos.ufpe.br/revistas/rmp
ISSN – 2317 - 0115
V. 14 - n. 1 (2025)
Recife – PE

Uma revisão sistemática da literatura sobre a modelagem de ameaças em projetos ágeis

Rafael Souza
Centro de Informática, Universidade Federal de Pernambuco
https://orcid.org/0009-0009-6373-1795
rafa.carneiror@gmail.com

Carla Silva
Centro de Informática, Universidade Federal de Pernambuco
https://orcid.org/0000-0002-0597-3851
ctlls@cin.ufpe.br

Jéssyka Vilela
Centro de Informática, Universidade Federal de Pernambuco
https://orcid.org/0000-0002-5541-5188
jffv@cin.ufpe.br

Mariana Peixoto
Universidade de Pernambuco – Campus Garanhuns
https://orcid.org/0000-0002-5399-4155
mariana.peixoto@upe.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. Método: para compreender qual o cenário atual do uso de *threat modeling* em metodologias ágeis, foi feita uma RSL utilizando bibliotecas digitais e *snowballing* para obter artigos que pudessem responder às perguntas de pesquisa. Resultados: o estudo identificou os desafios, práticas e ferramentas utilizadas. Contribuições: o estudo trouxe as principais tendências da área estudada.

Palavras-chave: segurança; métodos ágeis; ameaças

A systematic literature review on threat modeling in agile projects

Abstract – Context: Threat modeling is an important activity for security, however its use in agile development is difficult. Problem: For threat modeling to be applied in agile development, it is necessary to understand its challenges and good practices. Method: To understand the current scenario of using threat modeling in agile methodologies, an SLR was carried out using digital libraries and snowballing to obtain articles that could answer the research questions. Results: The study identified the challenges, practices and tools used. Contributions: The study brought out the main trends in the studied area.

Keywords: security; agile methods; threats

Data da Submissão: 19/08/2024 - **Data de aceitação:** 24/06/2025

RMP
Revista dos Mestrados Profissionais
UFPE / CCSA – MGP
https://periodicos.ufpe.br/revistas/rmp
ISSN – 2317 - 0115
V. 14 - n. 1 (2025)
Recife – PE

DOI: https://doi.org/10.51359/2317-0115.2025.265731

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. https://orcid.org/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. Um *software* pode ser considerado seguro uma vez que ele continua a funcionar corretamente mesmo sob ataque de um usuário malicioso (McGraw, 2004).

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 (Hernan et al., 2019), o que diminui as chances de incidentes de segurança e os custos necessários para realizar as correções após o lançamento do sistema (McGraw, 2004).

Ocorre que, nos dias de hoje, muitas empresas utilizam metodologias ágeis para o desenvolvimento de seu *software*, devido aos seus diversos benefícios de entrega contínua e interação constante com os clientes. Os princípios definidos no Manifesto Ágil (Manifesto ágil, 2001) visam possibilitar a entrega de *software* de forma mais rápida, interativa e que consiga responder às mudanças propostas pelos clientes. No entanto, o uso dessas metodologias pode ir de encontro com a segurança do sistema (Mohino et al., 2019), visto que seus princípios muitas vezes estão em discordância.

Devido a isso, se fez necessário o entendimento de como possibilitar o uso de modelagem de ameaças em metodologias tão difundidas na área de desenvolvimento de *software*, para que haja uma aplicação do desenvolvimento seguro em ambientes que prezam pela agilidade na entrega dos seus sistemas. Portanto, é imprescindível um 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 um *software* seguro (McGraw, 2004). Além disso, há pouca base de segurança da informação por parte dos desenvolvedores atualmente no mercado de trabalho (Oueslati et al., 2015), (Bernsmed and Jaatun, 2019), (Bernsmed et al., 2022), e pouco foco sobre esse assunto nas universidades.

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 do estudo.

2. Revisão Conceitual

A seguir são explicitados os conceitos fundamentais ao estudo.

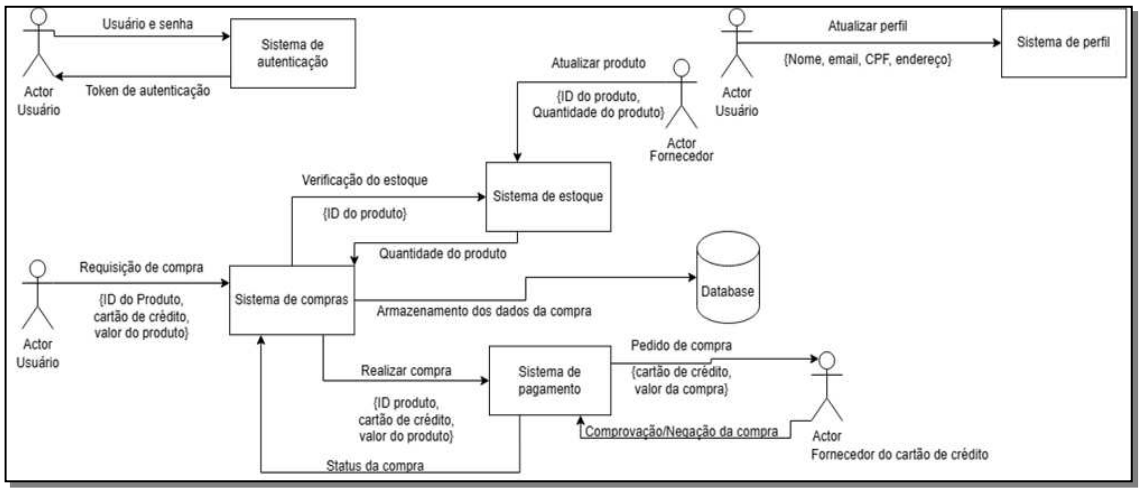
2.1 Modelagem de ameaças

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 Security Development Lifecycle, 2023).

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).

Figura 1 – Diagrama de fluxo de dados.



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 interagem com o sistema e quais ativos são utilizados pelo sistema. A figura 1 ilustra um possível, e simplificado, Diagrama de fluxo de dados (DFD).

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 o desenvolvimento do sistema com base nas recomendações de segurança e mitigação de ameaças identificadas durante o processo de *threat modeling*.

2.2 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, por recomendar um software funcional em vez 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 (Bernsmed et al., 2022).

Em uma revisão sistemática da literatura (RSL) a respeito do estado da arte do *threat modeling* (Xiong et al., 2019), existiram como perguntas de pesquisa: “o que é *threat modeling*?” “qual o estado da arte do *threat modeling*?”. Para responder a essas perguntas, os pesquisadores procuraram artigos nos motores de busca *IEEE Xplore, Scopus, Springer link,* e *Web of Science* 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 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, os pesquisadores utilizaram as palavras chaves "*secure*", "*software*" e "*agile*" nos motores de busca IEEE Xplore e ACM Digital Library 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 RSL foi realizada com o objetivo de compreender qual o cenário atual do uso de *threat modeling* em metodologias ágeis e, a partir disso, compilar quais são os desafios e as boas práticas deste uso.

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; 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 a justificativa de outros estudos.

Esta RSL foi realizada seguindo os passos descritos em Peixoto e Silva (2017). Como estratégia de busca, foi utilizada uma busca automática, com os motores *ACM Digital Library e ScienceDirect*. 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; 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 se realizou 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; Charters, 2007).

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?".

Foram também estabelecidas outras perguntas derivadas da pergunta de pesquisa principal. Elas foram:

- PP1: Quais são os desafios e boas práticas do uso de *threat modeling* no desenvolvimento ágil?
- PP2: Em qual momento é feito o *threat modeling* no desenvolvimento ágil?

- PP3: Quais ferramentas estão sendo utilizadas para facilitar a modelagem de ameaças?
- PP4: Quais conteúdos poderiam fazer parte do ensino de *threat modeling* numa disciplina de segurança da informação?

A seleção dos artigos foi realizada através de uma revisão sistemática de literatura (Peixoto; Silva, 2017).

Primeiro, para a obtenção dos artigos a serem analisados, foi construída uma *string* de busca, com base nas perguntas de pesquisa e na estratégia PICO:

- População: Métodos Ágeis;
- Intervenção: *Threat Modelling*;
- Comparação: Características e Ferramentas;
- Desfecho (*Outcomes*): estado atual do uso.

A aplicação da estratégia resultou na pergunta abaixo configurada:

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

Utilizando a *string* de busca desenvolvida, foram pesquisados artigos nas ferramentas de busca *ACM digital library* e *ScienceDirect,* que retornaram 48 (quarenta e oito) artigos científicos, sendo 16 artigos obtidos na *ScienceDirect* e 32 artigos obtidos na *ACM Digital Library*[1].

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. 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 aplicar esses 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 o resumo dos artigos que os citam, para que pudessem ser utilizadas as mesmas métricas de inclusão e exclusão.

Após o processo de seleção, 9 artigos foram selecionados e 6 foram posteriormente acrescentados através da técnica de *snowballing*. Dentre os artigos

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

selecionados, devido à pequena quantidade de artigos escolhidos após utilizar as métricas de inclusão e exclusão, 9 (nove) abordam de forma mais genérica sobre segurança em metodologias ágeis, pois poderiam conter dados que auxiliam na resposta das perguntas realizadas. 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.* O quadro 1 lista os artigos selecionados. Os artigos que abordaram de forma mais genérica a segurança de informação foram marcados com um asterisco ao lado do identificador.

Quadro 1 – Lista dos artigos selecionados

Identificador	Artigo
acm1*	Baldassarre et al. (2021)
acm2*	Granata et al. (2022)
acm3*	Rindell et al. (2018)
acm14*	Nguyen and Dupuis (2019)
acm27	Kvamme et al. (2023)
science3	Casola et al. (2020)
science4*	Tøndel and Cruzes (2022)
science15	Bernsmed et al. (2022)
science16*	Rindell et al. (2021)
ieee1	Bernsmed and Jaatun (2019)
acm33*	Rindell et al. (2017)
ieee2	Cruzes et al. (2018)
ieee3*	Oueslati et al. (2015)
science17*	Tøndel et al. (2022)
mdpi1	Mohino et al. (2019)

Algumas das ameaças descritas por Wohlin *et al.* (2012) foram consideradas. A ameaça interna, em que ao surgir uma dúvida sobre a inclusão de um artigo, ele seria incluído. Devido a essa abordagem, alguns artigos que discorrem sobre segurança da informação de forma mais genérica foram incluídos. A ameaça de construto, em que foram utilizados sinônimos das palavras chaves para compor a string de busca. Porém, essa ameaça talvez não tenha sido mitigada satisfatoriamente, 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 mitigada ao utilizar um protocolo de RSL construído com base em outros estudos científicos. Com relação à ameaça externa, não é possível generalizar os resultados, pois trabalhos relevantes armazenados em bibliotecas digitais não consideradas neste estudo podem não ter sido capturados por meio do *snowballing*.

4. Análise e Resultados

São descritos a seguir os achados da pesquisa.

4.1 PP1: Quais são os desafios e boas práticas do uso de *threat modeling* no desenvolvimento ágil?

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 (Bernsmed et al. 2022), (Bernsmed; Jaatun, 2019), (Cruzes et al.,2018).

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).

O quadro 2 demonstra os desafios encontrados e 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.

Quadro 2 – Desafios identificados e quais artigos os identificaram.

ID	Desafio identificado	Artigos Identificadoes
	Falta de motivação dos desenvolvedores para realizarem as atividades	science15, ieee1, ieee2
2	Tempo necessário para realizar as atividades de *threat modeling* são altos	science15, ieee1, ieee2, ieee3
3	Difícil compreensão sobre quais elementos devem ser incluídos nos DFDs e quão detalhada devem ser as informações	science15, ieee1, ieee2
4	O gerenciamento de DFDs se torna difícil pela falta de costume em documentação em metodologias ágeis	science15, ieee3
5	Os desenvolvedores não sabem qual a usabilidade das atividades, nem dos resultados	science15
6	Difícil identificação das ameaças e compreender quais delas são relevantes	ieee1
7	Ter a perspectiva de um atacante	ieee1
8	Entender quando deve ser realizado novamente o *threat modeling*	science15
9	O STRIDE se mostrou muito focado na identificação de ameaças em canais de comunicação	ieee2
10	Falta de conhecimento dos desenvolvedores acerca de segurança da informação	science15, ieee3, ieee1

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 ensinando *threat modeling* aos desenvolvedores.

Apesar de um dos maiores desafios ser o custo temporal para realizar as atividades de *threat modeling,* principalmente quando se fala da etapa de criação das DFDs, a única boa prática que se repetiu nos artigos foi a realização das atividades de *threat modeling* de forma regular. 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 e quais artigos as citaram estão demonstradas no quadro 3.

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

ID	Boa prática identificada	Quais artigos a identificou
1	Realizar atividades de modelagem de ameaças de forma regular	ieee1, science15
2	Realizar a confecção dos DFDs	science15
3	Utilizar os DFDs construídos para o *onboarding* de novos desenvolvedores	science15
4	Analisar todas as interações do sistema	science15
5	Envolver um especialista em segurança	science15
6	Envolver os desenvolvedores nas atividades	ieee1
7	Utilizar checklists dos tópicos a serem discutidos	ieee1
8	Ter processos e rotinas claras	ieee1

4.2 PP2: Em qual momento é feito *threat modeling* no desenvolvimento ágil?

Esta pergunta visava identificar quando os times de desenvolvimento ágil decidem realizar atividades relacionadas à modelagem de ameaças. À medida que um novo *software* é 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 ser contínua, a fim de sempre entregar *software* mais seguros.

A realização de *threat modeling* em intervalos regulares de tempo é uma prática tida como benéfica para empresas que utilizam de metodologias ágeis (Bernsmed et al., 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 (Bernsmed; Jaatun, 2019; Owasp, 2023), nos estudos conduzidos por Bernsmed e Jaatun (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 (Bernsmed et al., 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 na infraestrutura do sistema, 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 alta, 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 foi um dos desafios encontrados durante a revisão de literatura. Contudo, não foi encontrado na literatura qual o melhor momento para serem realizadas atividades relacionadas à *threat modeling,* sendo um assunto que necessita de mais estudos.

4.3 PP3: 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 modelagem de ameaças. Em um estudo feito com estudantes de duas universidades (Bernsmed et al., 2022), foi possível inferir a importância de ferramentas 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). 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* (Microsoft 2022), *Draw.io* (Draw.io, 2023), *Elevation of Privilege* (EoP) *card game* (Shostac, 2014), *Visually Inspection to Support Privacy and Security* - VIS-PRISE (Baldassarre et al, 2022). Uma análise sobre essas ferramentas pode ser encontrada no material suplementar (Souza et al., 2024).

4.4 PP4: 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 (Bernsmed et al. ,2022) e demonstrar as vantagens da ferramenta desenvolvida (VIS-PRISE) (Baldassarre et al., 2022). Apesar disso, é possível abordar o assunto a partir das respostas obtidas nas perguntas anteriores e de algumas afirmações obtidas 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 (Bernsmed et al., 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 (Bernsmed et al., 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 f*rameworks* 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.

5. Conclusão

Neste trabalho, foi realizada uma revisão sistemática da literatura para entender o cenário atual da modelagem de ameaças em desenvolvimento ágil. 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*. Trazendo a conclusão que é necessário realizar mais estudos sobre o ensino de *threat modeling* para os desenvolvedores de *software*.

A principal contribuição 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.

Uma possibilidade de trabalhos futuros é a síntese dos resultados para uma proposta de ensino e a sua aplicação, 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 e mitigar os desafios encontrados nos artigos analisados, visto que existem artigos que abordam quais os desafios enfrentados por empresas e desenvolvedores, mas ainda não existem artigos que discorrem sobre quais são as possíveis soluções para tais desafios.

Referências

BALDASSARRE, M. T., BARLETTA, V. S., DIMAURO, G., GIGANTE, D., PAGANO, A., & PICCINNO, A. Supporting Secure Agile Development: the VIS-PRISE Tool. Proceedings of the 2022 International Conference on Advanced Visual Interfaces (AVI 2022). Association for Computing Machinery, New York, NY, USA, Article 69, 1–3, jun, 2022.

BALDASSARRE, M., BARLETTA, V., CAIVANO, D., & PICCINNO, A. Integrating Security and Privacy in HCD-Scrum. CHItaly 2021: 14th Biannual Conference of the

Italian SIGCHI Chapter (CHItaly '21). Association for Computing Machinery, New York, NY, USA, Article 37, 1–5, jul, 2021.

BECK, K. et al. Manifesto para Desenvolvimento Ágil de Software. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 7 maio. 2022.

BERNSMED, K., CRUZES, D. S., JAATUN, M. G., & IOVAN, M Adopting threat modelling in agile software development projects. Journal of Systems and Software, Volume 183, 111090, ISSN 0164-1212, jan, 2022.

BERNSMED, K; JAATUN, M. 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, p. 1-8, jun, 2019.

CASOLA, V., DE BENEDICTIS, A., RAK, M., & VILLANO, U.A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach. [s.l.] Journal of Systems and Software, 2020. v. 163.

CRUZES, D., JAATUN, M. G., BERNSMED, K., & TØNDEL, I. A. Challenges and Experiences with Applying Microsoft Threat Modeling in Agile Development Projects. (IEEE, Ed.)Adelaide, SA, Australia: 2018 25th Australasian Software Engineering Conference (ASWEC), 2018.

DE VICENTE MOHINO, J. et al. The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies. Electronics, v. 8, n. 11, p. 1218, 2019.

Draw.io. Disponível em: <https://www.draw.io/>. Acesso em: 2023

GEIB, J, SANTOS, B., BERRY, D., BALDWIN, M., & BARBARA, K. Microsoft Threat Modeling Tool threats. Disponível em: <https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats>. Acesso em: 10 out. 2022.

GEIB, J. et al. Threat Modeling Tool feature overview. Disponível em: <https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-feature-overview>. Acesso em: 12 out. 2022.

GEIB, J. et al. Microsoft Threat Modeling Tool. Disponível em: <https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool>. Acesso em: 15 out. 2022.

GRANATA, D.; RAK M.; SALZILLO G. MetaSEnD: A Security Enabled Development Life Cycle Meta-Model. 17th International Conference on Availability, Reliability and Security (ARES '22). Association for Computing Machinery, New York, NY, USA, Article 152, pp. 1–10, August, 2022.

HERNAN, S., LAMBERT, S., & OSTWALD, T. Uncover Security Design Flaws Using The STRIDE Approach. Disponível em: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach>. Acesso em: 12 jul. 2022.

KITCHENHAM, B.; CHARTERS, S.. Guidelines for performing Systematic Literature Reviews in Software Engineering (EBSE 2007-001). Keele University and Durham University Joint Report. jun, 2007.

KVAMME, S, GUDMUNDSEN, E., OYETOYAN, T. D., & CRUZES, D. S Data Protection Fortification: An Agile Approach for Threat Analysis of IoT Data. 12th International Conference on the Internet of Things (IoT '22). Association for Computing Machinery, New York, NY, USA. pp. 151–154. January, 2023

MCGRAW, G. Software security. IEEE Security & Privacy. Volume: 2, Issue: 2, March-April 2004), p. 80–83, 2 ago. 2004.

NGUYEN, J.; DUPUIS, M. Closing the Feedback Loop Between UX Design, Software Development, Security Engineering, and Operations. New York, NY, United States: Association for Computing Machinery, p. 93-98, 2019.

OUESLATI, H.; RAHMAN, M.; OTHMANE, L. Literature Review of the Challenges of Developing Secure Software Using the Agile Approach. Toulouse, France: 10th International Conference on Availability, Reliability and Security, p. 540-547, 2015.

OWASP Top Ten. Disponível em: <https://owasp.org/www-project-top-ten/>.

OWASP Threat Modeling Project. Disponível em: <https://owasp.org/www-project-threat-model/>.

PEIXOTO, M.; SILVA, C. A gamification requirements catalog for educational software: results from a systematic literature review and a survey with experts. Marrakech, Morocco: SAC 2017: Symposium on Applied Computing, abr. 2017.

RINDELL, K.; HYRYNSALMI, S.; LEPPÄNEN, V. Aligning security objectives with agile software development. Porto, Portugal: XP ’18 Companion: 19th International Conference on Agile Software Development, maio 2018.

RINDELL, K., RUOHONEN, J., HOLVITIE, J., HYRYNSALMI, S., & LEPPÄNEN, V. Security in agile software development: A practitioner survey. Information and Software Technology, v. 131, n. 106488, mar. 2021.

RINDELL, K.; HYRYNSALMI, S.; LEPPÄNEN, V. Busting a Myth: Review of Agile Security Engineering Methods. Reggio Calabria Italy: ARES ’17: International Conference on Availability, Reliability and Security, ago. 2017.

Security Development Lifecycle (SDL) Practices. Disponível em: <https://www.microsoft.com/en-us/securityengineering/sdl/practices>. Acesso em: 26 jun. 2023.

SHOSTACK, A. Elevation of Privilege: Drawing Developers into Threat Modeling. 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14). Anais...San Diego, CA: USENIX Association, ago. 2014.

SOUZA, R. et al. Material Suplementar. Disponível em: <https://docs.google.com/document/d/19Qxm0YVn_JL3egC68m_Y-18qkII0VbNdtyTjrfTb894/edit>.

TØNDEL, I.; CRUZES, D. S. Continuous software security through security prioritisation meetings. Journal of Systems and Software, v. 194, n. 111477, 2022.

TØNDEL, I., CRUZES, D. S., JAATUN, M. G., & SINDRE, G. Influencing the security prioritisation of an agile software development project. Computers & Security, v. 118, n. 102744, 2022.

Threat Modeling. Disponível em: <https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling>. Acesso em: 12 set. 2023.

WOHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M. C., REGNELL, B., & WESSLÉN, A. Experimentation in software engineering. Berlim, Germany: Springer, 2024.

XIONG, W.; LAGERSTRÖM, R. Threat modeling: A systematic literature review. Computers & Security, v. 84, p. 53-69, 2019.