An Evaluation of the Product Security Maturity Model Through Case Studies at 15 Software Producing Organizations

Elena Baninemeh[1](✉), Harold Toomey[2], Katsiaryna Labunets[1], Gerard Wagenaar[1], and Slinger Jansen[1,3]

[1] Utrecht University, Utrecht, The Netherlands
{e.baninemeh,k.labunets,g.wagenaar,slinger.jansen}@uu.nl
[2] Raytheon Technologies, Walthem, USA
Harold@Toomey.org
[3] LUT University, Lappeenranta, Finland

Abstract. Cybersecurity is becoming increasingly important from a software business perspective. The software that is produced and sold generally becomes part of a complex landscape of customer applications and enlarges the risk that customer organizations take. Increasingly, software producing organizations are realizing that they are on the front lines of the cybersecurity battles. Maintaining security in a software product and software production process directly influences the livelihood of a software business. There are many models for evaluating security of software products. The product security maturity model is commonly used in the industry but has not received academic recognition. In this paper we report on the evaluation of the product security maturity model on usefulness, applicability, and effectiveness. The evaluation has been performed through 15 case studies. We find that the model, though rudimentary, serves medium to large organizations well and that the model is not so applicable within smaller organizations.

Keywords: software product security · software engineering security · product security maturity model

1 Introduction

"Cybersecurity is the collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets." [42]. It strives to ensure the integrity, availability and confidentiality of software applications. There are plenty of tools, such as firewalls and antivirus software to prevent cyber-attacks and detect security breaches. A cyber-attack is action where a person tries to penetrate another person's computers or network for the purpose of causing damage or disruption [11]. Cybersecurity tries to prevent a cyber-attack from happening. We argue that

S. Hyrynsalmi et al. (Eds.): ICSOB 2023, LNBIP 500, pp. 327–343, 2024.
https://doi.org/10.1007/978-3-031-53227-6_23

cybersecurity is one of the recently introduced cost factors in SPOs and that this field deserves more attention from the software business research community. During the development phase of a software product, one of the key priorities for software engineers is ensuring the fulfillment of quality and security requirements [10]. Software business has benefited from maturity models [17,38]. Several maturity models 4 are being used by Software Producing Organizations (SPOs) to evaluate their software product and software production security. One of these models, called the Product Security Maturity Model (PSMM) that has not sufficiently been evaluated for its usefulness and applicability, so in this study, we improve this problem by evaluating the PSMM.

In the next Section, we introduce the PSMM. In Sect. 3 we reiterate the objective of this work and describe how we performed a model comparison and a holistic multiple case study at 15 organizations with a large number of small research teams.

1. In Sect. 4, we compared the PSMM with BSIMM and SAMM and discovered that the PSMM is unique in its agility and relative completeness for SPOs.
2. Secondly, we report on 15 case studies in Sect. 5, with the goal of identifying patterns in the data. We find that operational security is directly related to size of the company, but that technical product security is not dependent on a company's size.
3. With the participants in the case studies, we also evaluate the usefulness, applicability, and effectiveness of the PSMM and report on the findings from those evaluations in Sect. 6. We discovered that the model was proficient in suggesting new security practices to the participants in the case study. However, it does suffer from certain design flaws. Furthermore, in Sect. 6.1, we discussed various situational factors that were identified.

We conclude the work with a discussion about the role of maturity models as a scientific endeavor and their role in improving SPOs.

2 Introducing the PSMM

Evaluating the cybersecurity of any business is a difficult endeavor, comparing these evaluations is even more of a challenge, especially so if the evaluations were done according to different metrics. To solve this issue and evaluate whether partners were using proper cybersecurity protocols, an employee at semiconductor chip manufacturer Intel developed the "Product Security Maturity Model"[1].

The PSMM evaluates based on twenty criteria, which are split in two categories: Operational and Technical. Operational parameters in PSMM include measures of program support, staffing and resources, SDL implementation, protection from externally reported product vulnerabilities (PSIRT), adherence to product security policies and processes, security training, and efficiency of data tracking and security metrics. Technical parameters in PSMM include measures

[1] www.toomey.org/psmm/.

of software security requirements and verification, software architecture and design reviews, threat modeling, security testing, static and dynamic analysis, fuzz testing, vulnerability scans and penetration testing, manual code reviews, secure coding standards, security of open-source and third-party libraries, and protection of privacy and confidential data.

The model consists of five levels of maturity; none, initial, Basic, Acceptable, Mature. For each of the twenty parameters, five levels of maturity are defined, each with between 1–6 criteria that indicate whether a particular maturity level has been met for that practice. For instance, to achieve level 5 of the *Software Architecture and Design Reviews* parameter, you need to adhere to the following list of requirements:

1. Separation of privileges to address unknown attack vectors.
2. Reviews reveal multiple high and medium severity issues and the issues are effectively addressed early in the development cycle.
3. Architecture documents extensive enough to be used for Common Criteria (EAL-3) certification.
4. BSIMM-AA3.2: Drive analysis results into standard architecture patterns.

One of the more interesting parts of the PSMM is its inclusion of factors from other models (EAL-3, BSIMM-AA3.2) as adherence criteria. This leads to an explicit lists of requirements that the author would probably claim to be "the most suitable", but also to some complexity in the model.

To perform a PSMM assessment, an organization first defines the scope of the assessment, which includes determining the products or systems that will be evaluated and the level of detail of the assessment. Next, key stakeholders are identified and involved in the assessment, as they are able to provide valuable insights and perspectives on the organization's product security practices.

After the scope and stakeholders have been defined, the organization then collects and analyzes data on its product security practices. This involves reviewing documentation, conducting interviews, and gathering data from systems and tools. The data is then used to determine the organization's current level of product security maturity, as well as any areas for improvement.

3 Research Approach

Object of Study. The study focuses on PSMM. The model was developed by Intel and is being used by a number of large IT companies including McAfee, Intel, and Deloitte. PSMM aims to be a simple, quantitative tool with low overhead that allows organizations to determine how well each Security Development Lifecycle activity is being performed. The PSMM is unique in that it provides relatively low-touch assessments, compared to more extensive models.

To perform this task, the model has operational parameters, such as Resources, Processes and Training, and technical parameters such as threat modelling and dynamic analysis. For each parameter five maturity levels are defined. Each of the maturity levels is associated with several questions per parameter.

If the answer to each of those questions is positive, the maturity level can be seen as obtained for that maturity level. As the model is simple and these levels are quantified and fully defined, minimal training and effort is needed to apply the model and create insightful metrics.

Evaluating Design Science Artifacts. Design science is the science of designing new information systems artifacts, that have a positive effect on science or society [12]. An essential step in the scientific process of design science, is the evaluation of design science artifacts. We frame our evaluation of the PSMM using Venable et al.'s framework [40]. The framework takes input from contextual factors such as goals, conditions, and constraints and supports the researcher in selecting the appropriate evaluatory techniques. These techniques are sorted into four categories that consist of two properties, being ex post (after creation of the artefact) or ex ante (before creation of the artefact) and a naturalistic (for example, in a field setting) or artificial (for example, in a laboratory) evaluation. After selecting one or more categories the framework proposes methods that can best be used with the selected evaluatory techniques.

Following the Design Science Research Evaluation Framework results in a focus on utility and efficacy. Essentially, posing that the evaluation should focus on the questions, 'Does the model do what it needs to do?' and 'Can PSMM be effective?'. The framework subsequently suggests, based on contextual factors, that a naturalistic ex post approach is the best fit for this study. For this approach a number of methods are recommended including focus groups, surveys, and case studies. In this work, we use the case study method [32] for the evaluation, by performing a holistic multiple case study in Sect. 5.

4 Related Models

In this study, Snowballing was applied as the primary method to investigate the existing literature regarding the security maturity models. During the initial hypothesis search phase, we explored literature based on the following search keywords: "(security or SDL) maturity model", and "Secure Development Lifecycle". Accordingly, We collected a set of papers based on the snowballing method during this phase. Hence, we found 97 papers for security maturity models with different activities and features. Inclusion and exclusion criteria ensure that relevant manuscripts are included and irrelevant manuscripts are excluded. We extracted the required information, including the title, abstract, the Maturity Models considered in the paper, the venue where the paper was presented, the number of citations, and the year as inclusion and exclusion criteria.

The first and second authors conducted a quality assessment of the resulting studies. We collaboratively analyzed and discussed the studies for inclusion in the final list. We used quality criteria such as whether the paper contains (1) a problem statement, (2) research questions, (3) research challenges, (4) explicit research results, and (5) real-world use cases. Based on these qualities, we indicated each paper's relevance to our study's research question. Based on this information, we have ranked the studies using four qualitative values: No relevance, low, medium, and high. The high-ranked results are listed in Table 1.

We ended up selecting 29 studies from various domains through a literature review based on snowballing that was presented in Table 1. We discovered that the studies we examined incorporated various security maturity models, such as BSIMM, SAMM, SSE-CMM, C2M2, MSSDL, CLASP, SAFECode, and OpenSAMM. However, upon analyzing the frequency of each framework's appearance in these studies, it became evident that BSIMM and SAMM were the popular choices. These two models demonstrated a consistent presence across the studies we considered in our research and they are open community projects and widely utilized within the IT industry.

OWASP Software Assurance Maturity Model (SAMM) - SAMM [35] is an open framework developed by OWASP, designed to assist organizations in assessing their current software security practices across the entire organization. This flexible model is intended for use by companies of all sizes, including small, medium, and large enterprises. SAMM is structured around key business functions within the software development life cycle, with each business function associated with three specific security practices. These business functions include Governance, Construction, Verification, and Operations [43].

Building Security In Maturity Model - BSIMM is founded on real-world practices observed in a large number of companies, making it a reflection of the prevailing state of software security. This framework is instrumental in evaluating the effectiveness of the Secure Software Development Lifecycle (SSDL). BSIMM covers 12 practices, which are further categorized into four primary domains: Governance, Intelligence, SSDL Touchpoints, and Deployment [16,19].

The practices and activities outlined in these models differ slightly in their approaches to what each model takes to achieve a higher maturity level. For instance, SAMM provides a comprehensive view by detailing activities, performance metrics, associated assurance benefits, personnel roles, and cost considerations. Conversely, BSIMM primarily focuses on security activities, the individuals engaged in them, and performance measurement [26].

We conducted a comparative analysis between PSMM and BSIMM, and SAMM. The results of this analysis are presented in the Table 2. The mappings were established based on comprehensive documentation and the respective activities defined in each model. In this mapping, we used a binary notation, with'1' denoting the presence of each activity from either the BSIMM or SAMM within specific parameters of the PSMM. For example, by considering the activity [SM1.1] from the "Strategy and Metrics" category, which involves 'publishing processes (roles, responsibilities, plan) and evolving them as necessary', we can realize that this particular activity can be effectively mapped to the "Process" parameter within the operational parameters of PSMM.

Table 1. An overview of the results of the literature study

Ref	Research type	Maturity Models
[21]	Research paper	BSIMM, SAMM, SSE-CMM, C2M2
[26]	Research paper	BSIMM, SAMM
[9]	Research paper	BSIMM, SAMM
[31]	Research paper	BSIMM, SAMM
[27]	Book	BSIMM, SAMM, MSSDL
[1]	Research paper	BSIMM, SAMM, MSSDL, CLASP, SAFECode
[22]	Research paper	BSIMM, SAMM, MSSDL, CLASP, SAFECode, OpenSAMM
[13]	Research paper	BSIMM, SAMM, MSSDL
[23]	Research paper	BSIMM, SAMM
[29]	Research paper	BSIMM, SAMM, MSSDL, CLASP
[3]	Research paper	BSIMM, SAMM, MSSDL
[30]	Research paper	BSIMM, SAMM, MSSDL, SSE-CMM
[34]	Research paper	BSIMM, SAMM, MSSDL, MSSDL, SAFECode
[44]	Research paper	BSIMM, SAMM, SAFECode
[20]	Research paper	BSIMM, SAMM
[18]	Research paper	BSIMM, SAMM
[37]	Thesis	BSIMM, SAMM, SAFECode
[41]	Research paper	BSIMM, MSSDL, CLASP, SAFECode
[45]	White paper	BSIMM, SAMM
[6]	Research paper	BSIMM, SAMM
[8]	Thesis	BSIMM, SAMM
[15]	Chapter	BSIMM, SAMM
[5]	Research paper	BSIMM, SAMM
[33]	Research paper	BSIMM, SAMM, MSSDL
[36]	Thesis	BSIMM, SAMM
[28]	Research paper	BSIMM, SAMM, MSSDL
[25]	Research paper	BSIMM, SAMM, MSSDL
[4]	Thesis	BSIMM, SAMM
[2]	Research paper	BSIMM, SAMM, MSSDL, CLASP, SAFECode

Through this mapping process, as shown in Table 2, we are able to quantify the number of activities from both BSIMM and SAMM that can be mapped to the PSMM framework. For activities where at least a'1' is assigned, it can be inferred that PSMM incorporates those activities within its scope. Thus, this analysis demonstrates of the extent to which PSMM aligns with and covers activities outlined in BSIMM and SAMM. Moreover, in the coverage column, we indicated the activities and practices by'0' that they do not map to PSMM. For instance, the environment hardening practice in SAMM and part of the software environment practices in BSIMM. After analyzing this mapping, we realized that PSMM mapped to approximately 95% of the activities and practices outlined within BSIMM and it mapped to approximately 90% of the activities defined within SAMM (full table of mapping). On the other hand, PSMM assists organizations in advancing through the four stages of maturity management, establishing a clear path from their current product security status to the desired state. Within each stage of the maturity model, the team can showcase tangible achievements by evaluating specific requirements. This proactive approach outlined in the model enables the organization to set and reach milestones to minimize product-related risks and detect potential risks earlier in SDL. The implementation of this maturity model will establish multiple layers of defense within the product, significantly raising the difficulty for malicious actors to breach it. The model's efficacy is evident at each security level as it enables the team to address security concerns in the early stages of development proactively.

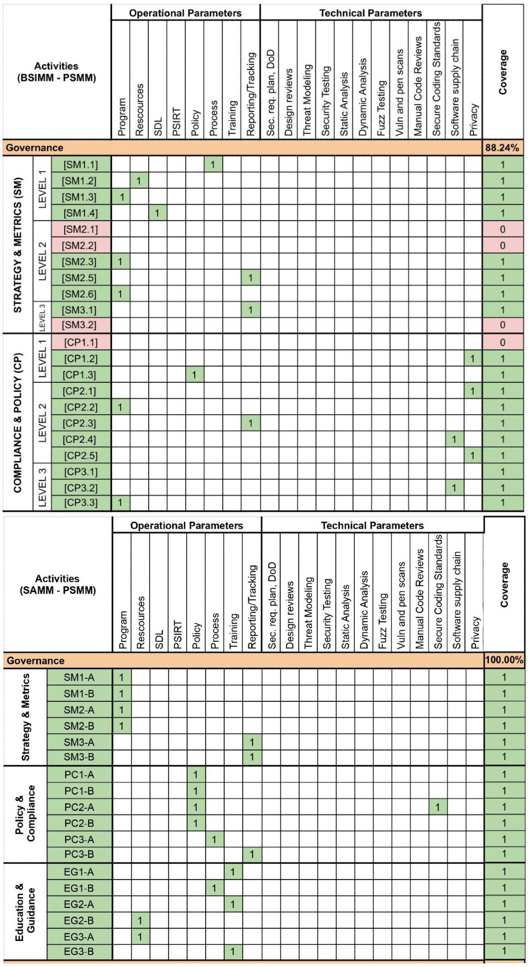
5 Case Studies: 15 Software Producing Organizations

The case studies were performed at fifteen SPOs from 2021–2023. The organizations were companies ranging from one to 67.000 employees. In Table 3 the company sizes are indicated (Small: 1–49, Medium: 50–999, Large: 1000+). We do not provide exact numbers to protect the identity of some of the larger organizations, which are easily identifiable through their employee numbers. The PSMM was applied on one product per SPO. The organizations range from SPOs providing administration products for small businesses to SPOs producing products for maintaining public transportation vehicles. All SPOs are business to business companies. The SPOs are located in the Netherlands (12x), the USA (2x), and Canada (1x), although they all had a presence in the Netherlands. All interviews were conducted in Dutch and transcribed. The transcriptions are available upon request from the authors and were translated into English by the last author.

Case Study Protocol. The evaluation of the PSMM with experts was conducted by different student teams in the context of either a bachelor course at Utrecht University (Cases A-L) or in the context of a graduation project (M, N, O). A case study protocol (Link to the case protocol) was provided that included a case report format, a set of interview questions, and a guide to the PSMM. All teams were briefed in a two-hour session about the PSMM and about the case study approach in another lecture. Furthermore, they were provided with accompanying literature and prepared the case study interviews by discussing the protocol. All teams recorded their interviews and transcribed them. The case study data and PSMM assessment, collected by the researchers, consisted of: a filled in PSMM spreadsheet as provided by Toomey, spider graphs presenting the scores, a descriptive case study report (15–35 pages LNCS, available by request from the last author), and a transcription of the interviews performed (usually one or two per case study). The teams also reported on which document resources (website, provided documents, etc.) were used for the data gathering.

To analyze the effect of a company's size on the Operational, Technical, and combined scores, we use the Kruskal-Wallis (KW) test as our data are ordinal in nature and have more than two levels (small, medium, and large sizes). To explore any statistically significant results identified by the KW test, we use a post-hoc Mann-Whitney (MW) test (corrected for multiple tests with Bonferroni method). We adopt 5% as a threshold of α (i.e., the probability of committing Type-I error). We also provide the Cliff's δ, a non-parametric effect size measure, when reporting any statistically significant result identified with the MW test.

Table 2. The first table provides an overview of how PSMM maps to BSIMM, and the second table presents an overview of the mapping between SAMM and BSIMM. In this mapping process, we utilized a binary notation, where '1' signifies the existence of each activity from either the BSIMM or SAMM within the defined parameters of the PSMM. For instance, examining the activity [CP1.3] in the "Compliance & Policy (CP)" category of "BSIMM" reveals that this specific activity can be effectively mapped to the "Policy" parameter within the operational framework of PSMM. The full table for mapping PSMM - BISIMM and PSMM- SAMM is available as a spreadsheet at this Google Drive Spreadsheet.



Activities (BSIMM - PSMM)			Operational Parameters								Technical Parameters												Coverage
			Program	Rescources	SDL	PSIRT	Policy	Process	Training	Reporting/Tracking	Sec. req. plan, DoD	Design reviews	Threat Modeling	Security Testing	Static Analysis	Dynamic Analysis	Fuzz Testing	Vuln and pen scans	Manual Code Reviews	Secure Coding Standards	Software supply chain	Privacy	
Governance																							88.24%
STRATEGY & METRICS (SM)	LEVEL 1	[SM1.1]						1															1
		[SM1.2]		1																			1
		[SM1.3]	1																				1
		[SM1.4]			1																		1
	LEVEL 2	[SM2.1]																					0
		[SM2.2]																					0
		[SM2.3]	1																				1
		[SM2.5]								1													1
		[SM2.6]	1																				1
	LEVEL 3	[SM3.1]								1													1
		[SM3.2]																					0
COMPLIANCE & POLICY (CP)	LEVEL 1	[CP1.1]																					0
		[CP1.2]																				1	1
		[CP1.3]					1																1
	LEVEL 2	[CP2.1]																				1	1
		[CP2.2]	1																				1
		[CP2.3]								1													1
		[CP2.4]																			1		1
		[CP2.5]																				1	1
	LEVEL 3	[CP3.1]																					1
		[CP3.2]																			1		1
		[CP3.3]	1																				1

Activities (SAMM - PSMM)		Operational Parameters								Technical Parameters												Coverage
		Program	Rescources	SDL	PSIRT	Policy	Process	Training	Reporting/Tracking	Sec. req. plan, DoD	Design reviews	Threat Modeling	Security Testing	Static Analysis	Dynamic Analysis	Fuzz Testing	Vuln and pen scans	Manual Code Reviews	Secure Coding Standards	Software supply chain	Privacy	
Governance																						100.00%
Strategy & Metrics	SM1-A	1																				1
	SM1-B	1																				1
	SM2-A	1																				1
	SM2-B	1																				1
	SM3-A								1													1
	SM3-B								1													1
Policy & Compliance	PC1-A					1																1
	PC1-B					1																1
	PC2-A					1													1			1
	PC2-B					1																1
	PC3-A						1															1
	PC3-B								1													1
Education & Guidance	EG1-A							1														1
	EG1-B						1															1
	EG2-A							1														1
	EG2-B		1																			1
	EG3-A		1																			1
	EG3-B							1														1

The KW test identified statistically significant effect of the company's size on the Operational and combined PSMM score ($p = 0.009$ and $p = 0.03$, correspondingly). For the Technical score the KW test returned $p = 0.15$ indicating no significant effect. The MW test requires the homogeneity of variance of samples.

Table 3. The 15 companies are listed here with their evaluation scores. The PSMM discriminates well across different companies, as many different values are given for different cases. The patterns in this table are discussed in Sect. 6.

	Company	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**		
	Size	M	S	S	L	L	M	L	S	S	S	M	M	S	S	S		
Operational Parameters																	**Avg**	**StDv**
O1	Program	5	4	1	5	5	5	4	1	1	2	2	4	3	1	1	4.17	1.71
O2	Resources	4	4	1	4	2	5	4	2	1	1	2	2	3	2	3	3.33	1.29
O3	SDL	1	3	3	3	5	4	3	0	1	1	5	5	3	2	1	3.17	1.63
O4	PSIRT	3	3	2	4	2	2	5	2	1	4	1	4	4	2	2	2.67	1.22
O5	Policy	3	4	4	4	3	5	3	1	1	5	4	4	3	2	1	3.83	1.36
O6	Process	1	5	2	5	5	4	5	4	3	4	2	4	4	1	3	3.67	1.41
O7	Training	2	2	1	5	4	5	5	1	1	3	4	3	2	0	3	3.17	1.62
O8	Reporting & Tracking	4	3	4	5	4	5	2	2	4	2	3	3	4	2	1	4.17	1.21
Technical Parameters																	**Avg**	**StDv**
T1	Sec. req. plan, DoD	1	5	2	5	4	5	4	2	3	2	0	4	4	2	2	3.67	1.56
T2	Design reviews	4	5	2	5	4	3	0	2	3	2	1	4	3	2	3	3.83	1.41
T3	Threat Modeling	2	1	2	4	2	4	3	3	3	1	0	3	1	1	1	2.50	1.22
T4	Security Testing	3	4	2	5	5	4	5	2	2	4	1	5	3	1	3	3.83	1.44
T5	Static Analysis	5	5	3	4	5	5	0	2	1	4	2	3	3	3	3	4.50	1.52
T6	Dynamic Analysis	4	4	0	4	5	5	0	1	0	4	2	2	3	2	2	3.67	1.77
T7	Fuzz Testing	1	4	0	5	5	4	3	0	1	1	1	3	1	1	1	3.17	1.75
T8	Vuln and pen scans	3	4	2	5	5	4	3	1	1	4	1	4	3	1	1	3.83	1.52
T9	Manual Code Reviews	5	3	4	5	3	4	4	3	4	5	3	5	1	2	3	4.00	1.18
T10	Secure Coding	2	3	3	5	3	4	3	3	3	3	1	3	4	5	1	3.33	1.16
T11	Software supply chain	2	4	3	4	3	5	4	1	1	2	1	4	4	4	0	3.50	1.52
T12	Privacy	3	4	2	5	4	0	3	4	4	4	3	3	4	2	1	3.00	1.33
	Operational score	2.9	3.5	2.3	4.4	3.8	4.4	3.9	1.6	1.6	2.8	2.9	3.6	3.3	1.5	1.9		
	Technical score	2.9	3.8	2.1	4.7	4.0	3.9	2.7	2.0	2.2	3.0	1.3	3.6	2.8	2.2	1.8		
	PSMM Score	2.9	3.7	2.2	4.5	3.9	4.1	3.3	1.8	1.9	2.9	2.1	3.6	3.0	1.8	1.8		

We checked this parameter with the Levene's test confirmed that the samples for the three scores met this requirements (Levene's $p > 0.61$). The post-hoc MW test with Bonferroni correction ($\alpha = 0.05/3 = 0.0167$) revealed several statistically significant results. For the Operational score we observed a statistically significant difference between Medium over Small (mean $Op_{s}mall = 2.16$ and $Op_{m}ed = 3.4$, MW $p = 0.014$ and Cliff's $\delta = 0.83$, considered a large effect size) and Large over Small organizations (mean $Op_{s}mall = 2.16$ and $Op_{l}arge = 4.0$, MW $p = 0.0167$ and Cliff's $\delta = 1$, large effect size). For the combined PSSM score the post-hoc test revealed similar trend between Small and Medium (mean $Op_{s}mall = 2.3$ and $Op_{m}ed = 3.16$, MW $p = 0.07$) and Small and Large organizations (mean $Op_{s}mall = 2.3$ and $Op_{l}arge = 3.89$, MW $p = 0.03$), but these results are not statistically significant.

We can draw several conclusions from the relationship between company size and PSMM score. First, the operational security within an SPO is directly related to its size. Second, technical security is not observably related to its size, which can be explained by technical prowess: each company will have its own security requirements for a product and its skill levels, independent of size [14].

6 Analysis: Evaluating the PSMM

We evaluated the model in a free format; throughout interviews, the case study participants were allowed and encouraged to criticize parts of the PSMM during the assessment. At the end of the interviews, we also asked them what their general feelings about the model was. We report on these using quotes from the interviews and mark the finding with the companies where it was observed (e.g., A, B, *C*). If one of the companies' code names is in italics, that means the transcript shows this quote literally (company C in the example).

There were many positive remarks about the model. All organizations indicated that *"it is a great standardized test to benchmark one's operational security"*. While we never shared the data from other organizations with them, the benchmarking capabilities were still recognized. Another positive remark we heard from the participants concerned that it was timely to take a look through this lens. Each organization found low hanging fruits for improvement, and this generally helped the organization. A final positive remark we heard was about how to prioritize security in the software development process: *"The model proved useful to us, because we typically prioritize features over security, we should start writing security "features" down as user stories"* (*H*, I, K).

We collected 24 unique criticisms from the interviews, after grouping them for occurrence. The following texts report on the ones that are common (three or more companies) or stand out for other reasons.

Completeness - The participants were particularly critical of the model completeness. Most of them found it *"overcomplete"* (*F*, G, L, K, M, N, O) and *"practically impossible to be fully compliant"* (*K*, M, N, O) *"without huge budgets"* (*all*). For example, one participant mentioned that if you follow the model strictly *"being available 24/7 is a requirement, so maximum maturity cannot be reached, because we don't need 24/7 availability"* (*F*). On the other hand, it was judged to be *"more or less sufficient for what it's trying to do"* (*A, F, D*).

Flexibility - *"Maturity Models are generally too static"* (*A*, B, L, K), and the participants want the *"Model [to] be more 'need-based', and take the company goals into account."* (*F*, K). Furthermore, the PSMM is judged to be *"too strict on particular guidelines, e.g. ISO"* (*A*, B, D, G, J, K, M, N, O)

Score Representation and Correctness - One important critique was also that the comprised score that is assigned at the end of the process does not fairly represent the status of a company and can be *"misleading"* (A, D, *K*, M, N, O). A relevant detail is that the way in which the score is calculated in the provided spreadsheets, is different from how it is described in the description text of the model. Some organizations also wondered whether the model might give *"a false sense of security"* (*A*, F, D).

Security Culture - Some of the case participants that found the model too inflexible, also mentioned that the model insufficiently allows for situationality in security culture. This was observed on different levels, such as culture on the work floor: *"The model assumes zero trust within the company itself, which may be an American thing."* (*A*, E, L), but also the situation that customers of a product may be more demanding regarding security and may be more vigilant and in a more trusting relationship with the SPO.

Assessment Complexities - One interesting complexity was that in some of the cases, we could not find all details on security processes, as they had *"some processes ... outsourced, such as pen testing"* (*C*, G, L). Furthermore, we heard from some organizations that by "following modern certifications for security, we scored high by default" (*E*, F). In larger organizations, we also encountered case participants who did not precisely know how particular functions were filled in within the organization (E).

6.1 PSMM Usability and Situational Factors

The PSMM instructions are somewhat unclear on its use; should the PSMM be applied regularly or is it a one-time instrument? Should the scores be trusted and have an impact on the improvement policies within the organization? And for whom is the model suitable? In this Section, we answer those questions using the evaluations and general knowledge about maturity models.

The models are generally tailored towards larger organizations, and the PSMM, with its origins at Intel, seems to suffer from this more than others. This has some funny side effects, such as interpretations leading to smaller (single product) organizations being able to much more rapidly adhere to some of the requirements. For example, to achieve level 5, an organization needs to have a Product Security Champion for a product, which is relatively easy for a one-product company.

For some of the other requirements, the inverse is true. A small-scale organization would not be able to meet some of the other requirements or only with immense and unnecessary difficulty. An example of this can be found in the resources parameter; To achieve level three the organisation needs to have a budget for the growth of the number of product security champions and have one product security champion per product. However, if a small organization has only a single product with a product security champion, then budgeting for multiple new product security champions seems unnecessary.

Situational Factors. A situational factor is any factor relevant to product development and product services. Examples are company size, branch and the number of submitted requirements per month, whether or not currently a waterfall-based method is used for product software development, etc. [7]. The organization's context is considered by evaluating different situational factors that define its surroundings and structure, subsequently helping the choice of relevant capabilities [7]. We suggest incorporating two situational factors that could improve the PSMM. Such factors can serve multiple purposes: they can either automatically disregard or introduce specific practices, or they can facilitate branching within the model to another variation. After identifying four potential situational factors through the interviews, we have chosen to introduce only two of them as real options.

The first situational factor we identify is company size. There are two sides of the spectrum that the interviewees addressed: small one-product companies should be given exemptions from practices in the model. On the other hand, large organizations require flexibility for the implementation of processes, as they may have more or less centralized security services within the organization, and at times the PSMM is too prescriptive in this respect. The second situational factor we identify is *"the development method (agile or waterfall)"* (*A*, H, I), especially because agile takes a different approach to security [30].

There were also proposed situational factors that we mention here, but question the validity of, and we currently do not propose implementing them in the PSMM. The third situational factor concerns the product characteristics, with two variation points. First, one of the companies operates from an open source perspective and provides a large part of its code base to the open source community (D), inherently leading to more secure products. One of the participants stated that product maturity has a strong influence on security; "it's easier to score better with a mature product." (*F*, H, I, K).

General Usage and Frequency. From the case studies we find that the model is best usable for medium to large product organizations with multiple products. As future work, we propose that a lighter version of the model is developed for smaller one-product companies. Assessments can be done in a relatively short time, ranging from around four to eight hours to get a first score, but obviously the lessons are found in the next steps: where is the organization now, where does it want to go, and how does the PSMM help in deciding what to do next? With regards to maturity models [17,24,39], from experience we can say that a yearly assessment is frequent enough and many organizations only use the same maturity model for one to four iterations, after which they abandon the maturity model or move on to another more extensive model.

6.2 Threats to Validity

Conclusion Validity. Possible threats to conclusion validity are related to the inaccurate data and data analysis process. Each of the case study reports was checked by one of the authors using the associated transcript, which are available upon request from the last author. Furthermore, two lower quality case study reports were excluded from the study, because they were incomplete and did not appear to represent the data. As for data analysis, we used the non-parametric tests as they do not require a normal distribution of the sample. To mitigate low statistical power, we adopted $\alpha = 0.05$ for the difference test, with reported Cliff's δ effect sizes for significant results.

Internal Validity. To perform the maturity assessments, we used the instructions as provided with the PSMM. We strongly depended on the information provided by the interviewees, and when vague answers were given, we were critical to ensure that we did not assess a practice or capability as present when it was not. The interviews had a dual nature: we performed the assessment and simultaneously asked the interviewee to provide feedback on the PSMM itself. This may have influenced the correctness of our findings, but we often found that asking deeper questions about each practice, led to better more detailed assessments and better shared understanding of each of the practices.

External Validity. To ensure the generalisability of our findings, we conducted a series of case studies with real product companies of different sizes, backgrounds, and from different regions. Therefore, we collected a diverse set of cases of applying the PSMM to evaluate the security maturity of real product development cases. However, it should be noted that we refrain from making any claims to generalization, but that we suspect that the PSMM is suitable for use by medium SPOs. We find that our model observations in this Section are rather generic and could be made about other maturity models or security assessment models as well. We hope that in the future, model designers will take these challenges into account, especially regarding applicability and situationality.

7 Conclusion

In this work, we provide an academic evaluation of a model rooted in practice entitled the Product Security Maturity Model, by evaluating it with 15 case studies and comparing it to existing models. We provide an extensive criticism of the model itself and how it may be improved, but we also praise it for its usefulness and effectiveness in providing organizations with improvement advice. We identify several situational factors that could lead to variations in the model that better fit an organization's size or development method.

We observe that maturity models are a well accepted standard for the diffusion of knowledge in organizations and are frequently used within organizations with highly skilled workers, such as in information technology. The 15 case participants all agree that even though the model is not perfect, it immediately gave the interviewees new ideas and concepts to implement and check within the organization. As such, we dare state that our work has already made an impact at the time of writing this work.

As part of our future work, we consider exploring other models and their applicability to software businesses, also to circumvent the challenges that have been identified in Sect. 6. In December 2023 we will start a new set of case studies with the OWASP SAMM 2.0 model. We experience that maturity models are seen as a relevant instrument for disseminating (scientific) knowledge among organizations, but are not necessarily seen as scientific. After all, aren't they just collections of ideas without much scientific merit? We consider it a challenge to give maturity models more solid footing in the scientific community, for instance by performing more empirical studies on the longevity of maturity models and their usage. We have already created a platform for the dissemination of maturity models and ensure their visibility: MaturityModels.org.

Acknowledgments. We want to thank the student teams that so diligently performed the case studies according to our protocol.

References

1. Al-Matouq, H., Mahmood, S., Alshayeb, M., Niazi, M.: A maturity model for secure software design: a multivocal study. IEEE Access **8**, 215758–215776 (2020)
2. M. Alenezi, H. A. Basit, M. A. Beg, and M. S. Shaukat. Synthesizing secure software development activities for linear and agile lifecycle models. Softw.: Pract. Exp. **52**(6), 1426–1453 (2022)
3. Ardo, A.A., Bass, J.M., Gaber, T.: An empirical investigation of agile information systems development for cybersecurity. In: Themistocleous, M., Papadaki, M. (eds.) Information Systems: 18th European, Mediterranean, and Middle Eastern Conference, EMCIS 2021, Virtual Event, December 8–9, 2021, Proceedings, pp. 567–581. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-030-95947-0_40
4. Assal, H.: The human dimension of software security and factors affecting security processes. PhD thesis, Carleton University (2018)
5. Assal, H., Chiasson, S.: Security in the software development lifecycle. In: 14th Symposium on Usable Privacy and Security (SOUPS 2018), pp. 281–296 (2018)
6. Attwood, S., Onumah, N., Paxton-Fear, K., Kharel, R.: Security-focused prototyping: A natural precursor to secure development. In: 2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), pp. 356–361. IEEE (2022)
7. Bekkers, W., Spruit, M.R., van de Weerd, I., van Vliet, R. and Mahieu, A., et al.: A situational assessment method for software product management. In: Proceedings of the 18th European Conference on Information Systems (ECIS2010) (2010)
8. Bideh, P.N.: Contributions to Securing Software Updates in IoT. Department of Electrical and Information Technology, Faculty of Engineering (2022)
9. Bugeja, J., Vogel, B., Jacobsson, A., Varshney, R.: IoTSM: an end-to-end security model for IoT ecosystems. In: 2019 International Conference on Pervasive Computing and Communications Workshops, pp. 267–272. IEEE (2019)
10. Farshidi, S.: Multi-criteria decision-making in software production. PhD thesis, Utrecht University (2020)
11. Hathaway, O.A., et al.: The law of cyber-attack. California law review, pp. 817–885 (2012)
12. Hevner, A., Chatterjee, S., Hevner, A., Chatterjee, S.: Design science research in information systems. Design research in information systems, pp. 9–22 (2010)

13. Höst, M., Hell, M.: Evaluation of the havoss software process maturity model. In: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 137–140. IEEE (2020)
14. Hou, F., Jansen, S.: A systematic literature review on trust in the software ecosystem. Empir. Softw. Eng. **28**(1), 8 (2023)
15. Iovan, M., Cruzes, D.S., Johansen, E.A.: A framework for a sustainable software security program. Evolving Software Processes, pp. 47–69 (2022)
16. Jaatun, M.G.: The building security in maturity model as a research tool. In: Empirical Research for Software Security, pp. 201–208. CRC Press (2017)
17. Jansen, S.: A focus area maturity model for software ecosystem governance. Inform. Softw. Technol. **1**, 118 (2020)
18. Kudriavtseva, A., Gadyatskaya, O.: Secure software development methodologies: a multivocal literature review. arXiv preprint arXiv:2211.16987 (2022)
19. McGraw, G.: Software security and the building security in maturity model (bsimm). J. Comput. Sci. Coll. **30**(3), 7–8 (2015)
20. Moyón, F., Bayr, C., Mendez, D., Dännart, S., Beckers, K.: A light-weight tool for the self-assessment of security compliance in software development – an industry case. In: Chatzigeorgiou, A., Dondi, R., Herodotou, H., Kapoutsis, C., Manolopoulos, Y., Papadopoulos, G.A., Sikora, F. (eds.) SOFSEM 2020: Theory and Practice of Computer Science: 46th International Conference on Current Trends in Theory and Practice of Informatics, SOFSEM 2020, Limassol, Cyprus, January 20–24, 2020, Proceedings, pp. 403–416. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-38919-2_33
21. Nikbakht Bideh, P., Höst, M., Hell, M.: HAVOSS: a maturity model for handling vulnerabilities in third party OSS components. In: Kuhrmann, M., Schneider, K., Pfahl, D., Amasaki, S., Ciolkowski, M., Hebig, R., Tell, P., Klünder, J., Küpper, S. (eds.) PROFES 2018. LNCS, vol. 11271, pp. 81–97. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03673-7_6
22. Núñez, J.C.S., Lindo, A.C., Rodríguez, P.G.: A preventive secure software development model for a software factory: a case study. IEEE Access, **8**, 77653–77655 (2020)
23. Onumah, N., Attwood, S., Kharel, R.: Towards secure application development: A cyber security centred holistic approach. In: 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), pp. 1–6. IEEE (2020)
24. Overeem, M., Mathijssen, M., Jansen, S.: Api-m-famm: a focus area maturity model for API management. Inform. Software Tech. **147**, 106890 (2022)
25. Palma, F., Realista, N., Serrão, C., Nunes, L., Oliveira, J., Almeida, A.: Automated security testing of android applications for secure mobile development. In: 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp. 222–231. IEEE (2020)
26. Ramirez, A., Aiello, A., Lincke, S.J.: A survey and comparison of secure software development standards. In: 2020 13th CMI Conference on Cybersecurity and Privacy, pp. 1–6. IEEE (2020)
27. Ransome, J., Misra, A.: Core software security. CRC Press (2018)
28. Rindell, K., Holvitie, J.: Security risk assessment and management as technical debt. In: 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1–8. IEEE (2019)
29. Rindell, K., Hyrynsalmi, S., Leppänen, V.: Aligning security objectives with agile software development. In: Proceedings of the 19th International Conference on Agile Software Development: Companion, pp. 1–9 (2018)

30. Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., Leppänen, V.: Security in agile software development: a practitioner survey. Inf. Softw. Technol. **131**, 106488 (2021)
31. Rindell, K., Ruohonen, J., Hyrynsalmi, S.: Surveying secure software development practices in finland. In: Proceedings of the 13th International Conference on Availability, Reliability and Security, pp. 1–7 (2018)
32. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**, 131–164 (2009)
33. Ryan, I., Roedig, U., Stol, K.-J.: Insecure software on a fragmenting internet. In: 2022 Cyber Research Conference-Ireland (Cyber-RCI), pp. 1–9. IEEE (2022)
34. Ryan, I., Roedig, U., Stol, K.-J.: Measuring secure coding practice and culture: A finger pointing at the moon is not the moon. In 2023 IEEE/ACM 45th Int'l Conference on Software Engineering (ICSE), pp. 1622–1634. IEEE (2023)
35. Teodoro, N., Serrão, C.: Web application security: improving critical web-based applications quality through in-depth security analysis. In: International Conference on Information Society (i-Society 2011), pp. 457–462 (2011)
36. Tøndel, I.A.: Prioritisation of security in agile soft. dev. projects (2022)
37. van de Werfhorst, M., Poll, E., Schoemaker, H.: and C. Kop, Security recommendations for agile and devops development at ridder data systems (2020)
38. van Steenbergen, M., Bos, R., Brinkkemper, S., van de Weerd, I., Bekkers, W.: The design of focus area maturity models. In: Winter, R., Zhao, J.L., Aier, S. (eds.) Global Perspectives on Design Science Research, pp. 317–332. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13335-0_22
39. van Steenbergen, M., Bos, R., Brinkkemper, S., van de Weerd, I., Bekkers, W.: Improving is functions step by step: the use of focus area maturity models. Scand. J. Inf. Syst. **25**(2), 35–56 (2013)
40. Venable, J., Pries-Heje, J., Baskerville, R.: Feds: a framework for evaluation in design science research. Eur. J. Inf. Syst. **25**, 77–89 (2016)
41. Venson, E., Alfayez, R., Gomes, M.M., Figueiredo, R.M., Boehm, B.: The impact of software security practices on development effort: An initial survey. In: 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pages 1–12. IEEE (2019)
42. Von Solms, R., Van Niekerk, J.: From information security to cyber security. Comput. Secur. **38**, 97–102 (2013)
43. Wen, S.-F.: Software security in open source development: a systematic literature review. In: 2017 21st Conference of Open Innovations, pp. 364–373. IEEE (2017)
44. White, C.A.: Root causes of insecure internet of things and holistically addressing them. In: 2020 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 1066–1074. IEEE (2020)
45. Williams, L.: Secure software lifecycle knowledge area issue. The National Cyber Security Center (2019)