Nurgalieva, Leysan; Frik, Alisa; Doherty, Gavin

# A Narrative Review of Factors Affecting the Implementation of Privacy and Security Practices in Software Development

*Published in:*
ACM Computing Surveys

*DOI:*
10.1145/3589951

Published: 17/07/2023

*Document Version*
Publisher's PDF, also known as Version of record

# A Narrative Review of Factors Affecting the Implementation of Privacy and Security Practices in Software Development

LEYSAN NURGALIEVA, Aalto University
ALISA FRIK, International Computer Science Institute (ICSI)
GAVIN DOHERTY, Trinity College Dublin

Privacy and security are complex topics, raising a variety of considerations and requirements that can be challenging to implement in software development. Determining the security and privacy related factors that have an influence on software systems development and deployment project outcomes has been the focus of extensive and ongoing research over the past two decades. To understand and categorize the factors that have an impact on developers' adoption and implementation of privacy and security considerations and practices in software development, we carried out a narrative review of the literature. The resulting mapping of factors provides a foundation for future interventions targeting organizational and individual behavior change, to increase the adoption of privacy and security practices in software development.

CCS Concepts: • **Security and privacy** → **Software security engineering**; **Human and societal aspects of security and privacy**; *Economics of security and privacy*; Privacy protections; • **Software and its engineering** → *Software development process management;*

Additional Key Words and Phrases: Privacy, security, design, software teams

**ACM Reference format:**
Leysan Nurgalieva, Alisa Frik, and Gavin Doherty. 2023. A Narrative Review of Factors Affecting the Implementation of Privacy and Security Practices in Software Development. *ACM Comput. Surv.* 55, 14s, Article 320 (July 2023), 27 pages.
https://doi.org/10.1145/3589951



## 1 INTRODUCTION AND RELATED WORK

A big part of today's digital economy relies on users' personal information. The collection of large amounts of user data introduces a variety of privacy and security risks. Although some threats (e.g., social engineering) target individual users [113], most threat models exploit system

This work received funding in part from the EU Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement 754489, in part from Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Progamme to Lero—the Irish Software Research Centre (www.lero.ie), and in part by a grant from the Center for Long-Term Cybersecurity (CLTC) at U.C. Berkeley, by National Science Foundation grants v and CNS-1528070, and by the National Security Agency's Science of Security program. Opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the funders.
Authors' addresses: L. Nurgalieva, Aalto University, Tietotekniikantalo, Konemiehentie 2, 02150 Espoo, Finland; email: leisan.nyr@gmail.com; A. Frik, International Computer Science Institute (ICSI), 1947 Center St, Berkeley, CA 94704, United States; email: afrik@icsi.berkeley.edu; G. Doherty, Trinity College Dublin, College Green, Dublin 2, Ireland; email: gavin.doherty@tcd.ie.


0360-0300/2023/07-ART320
https://doi.org/10.1145/3589951

vulnerabilities [18]. Therefore, it is important that privacy and security threats are recognized and addressed throughout the software development process, especially during the early software design and requirement stages. However, in practice, this is not always the case. For instance, a study by Spiekermann et al. [112] showed that 36% of the engineers surveyed rarely or never incorporate privacy mechanisms into the systems that they build, even though most of them believe that privacy and security engineering is useful, valuable, and important.

Prior research has identified a variety of reasons that implementing privacy and security in software development remains challenging. Some studies criticize inadequate enforcement of privacy regulations or blame the developers, their lack of knowledge, or lack of concern for privacy [37], and others believe that organizational structures and software development processes hinder the adoption of privacy and security practices [126]. However, the findings are fragmented: some reasons are repeatedly shown to have an impact, whereas other findings are contradictory or yield mixed results. Without a clear understanding of the barriers and challenges, efforts focused on designing and testing interventions to address the challenges remain scarce and unfocused.

Looking first at prior attempts at systematizing the knowledge on this topic, previous work has categorized the factors influencing the success of software development projects [46, 86], but not the success of implementing privacy and security practices specifically. Some studies have explored factors related to either only privacy [16, 126] or only security [70, 121]—or even more specifically on developer-centered security [120]—despite a large overlap between these factors. Moreover, individual studies typically consider only a subset of factors, without drawing a complete picture or acknowledging mixed and contradictory findings [83, 112]. Thus, we believe a wide-ranging review is needed to provide a comprehensive overview of the fragmented evidence from prior research and inform researchers, practitioners, and policy-makers about the drivers and barriers for implementing privacy and security practices in software development.

In this work, we present a narrative literature review of research that discusses the factors that affect the implementation of privacy and security practices in software development. Through a systematic search and synthesis of the literature, we identify patterns in the existing empirical evidence, categorize the relevant factors, and provide a critical assessment of the related work. Building on this analysis, we present a model of factors that provides a foundation for further exploration of the relative importance of the factors and relationships between them. Our model also provides a useful reference for systematically mapping the approaches for addressing the identified challenges, and driving organizational change in software companies, as well as individual behavior change among developers and engineers. Our findings categorize the factors into five main groups: environmental, organizational, product related, development process related, and individual. We discuss the implications and directions for future work, and map the potential interventions for leveraging the drivers and overcoming the barriers for implementing privacy and security practices in software development.

## 2 METHODS

In this work, we adopted a narrative review methodology. When compared to systematic literature reviews, narrative reviews produce a more selective survey of the literature [35, 50] and offer the flexibility to deal with "evolving knowledge and concepts" [21, p.2], such as the topic of this research. Following the literature review typology by Paré et al. [97], narrative reviews are considered as "a great starting point to bridge related areas of work, provoke thoughts, inspire new theoretical models, and direct future efforts in a research domain" (p. 185). This choice of methodology was driven by two main objectives: (1) to identify the factors and research hypotheses that affect the adoption of privacy and security practices in software development and design teams, and (2) to develop a conceptual model of the identified factors.

We developed the review protocol following a guiding framework by Walker et al. [127]. The framework itself relies on the "general framework for narrative synthesis" described by the Centre for Reviews and Dissemination [116]. Table 1 in Appendix A describes the main stages of the study protocol for selecting the most relevant documents related to the research topic, together with the timeline. The process included four stages: initial search strategy, building of the initial model, systematic search, and model refinement.

During the initial search (step 2 in Table 1), a group of three researchers worked together on the literature mapping using their expertise and knowledge on the subject to map prior studies and identify those relevant to the research objective. For this initial search, several strategies were used. Based on the author's knowledge, we started by identifying and then scanning relevant venues including ACM CHI, SOUPS, and PETS, as well as the output of research groups and individual researchers publishing on the topic. We then performed a snowball search based on this initial set of articles. This initial mapping was later complemented with a systematic search (step 4 in Table 1). All references identified as relevant were saved and organized in the shared repository, where we coded the publication year and venue, authors, abstract, and methodology, as well as major contributions in relation to security/privacy factors. We considered relevant the following contributions: (1) empirical evidence, for instance, resulting from user studies with individual developers or members of software development companies, or other experimental setups analyzing privacy and security practices in software development, and (2) theoretical contributions, for example, resulting from literature reviews and analytical processing, regarding the aspects that hinder or promote the implementation of privacy and security practices in software development. The subsequent analysis process summarizes the results by using descriptive parameters.

To build the initial model, two researchers read the selected papers and independently extracted the factors that were hypothesized (based on theoretical predictions) or observed (based on empirical evidence) to affect the implementation of privacy and security considerations in the software development process. Then, using affinity diagrams, the same two researchers independently categorized the factors into groups. The proposed factors and categorizations were discussed and merged. After resolving the disagreements, the researchers agreed on the initial model of factors.[1]

After the initial narrative review of the literature, the researchers conducted an additional systematic search (secondary search, step 4 in Table 1) to improve transparency and make sure all relevant studies are included. A title and abstract search was carried out on the following keywords: *security and privacy by design*, *software*, *developers and development*. The detailed search queries used are presented in Appendix A. Following the same procedure as described previously, we selected publications that took the form of literature reviews, experimental or quasi-experimental studies, or quantitative or qualitative analysis. The exclusion criteria comprised studies published in books and book chapters, which were not written in English, or were not related to software engineering.

Finally, during the model refinement stage, we inspected the additional documents selected during the systematic search to check whether any new factors emerged there and needed to be added to the model. This analysis did not reveal new factors that were not yet covered in our model, but did provide additional supporting evidence for the existing factors. This observation indicated that saturation was achieved and concluded our work on the model.

[1]Calculating the agreement rate was not relevant for this analysis. The goal of the analysis was to categorize the factors, which is equivalent to developing a codebook in qualitative analysis. Unlike the coding process, codebook development does not involve the calculation of agreement rates. Typical to codebook development, all differences in conceptualization were resolved after the initial codebooks were merged and discussed.

## 3 RESULTS

The initial search strategy (Table 1, step 2) resulted in 99 papers across the three databases. We excluded papers not relevant to our research questions based on their abstracts ($N = 45$) and on the full text of the article ($N = 11$) (Table 1, step 3). A systematic search in the Scopus database resulted in 185 additional documents. After removing duplicates, dissertations, and gray literature (commercial reports, policy statements, or editorial papers), the search yielded 99 unique articles. These papers were reviewed based on their abstracts and resulted in 47 articles, which were read in full, and the search results distilled to 26 publications (Table 1, step 4). The final set of papers considered for analysis included 69 relevant documents.

Factors affecting the implementation of privacy and factors affecting the implementation of security were typically studied separately. However, we observed a substantial overlap between them. In some cases, the factors did not overlap but appeared transferable—that is, although they were observed in one domain (e.g., security), we hypothesize a similar effect in another domain (e.g., privacy), although the original research may have focused on only one. Thus, in our model, we combine the factors affecting privacy and security.

Based on the analysis of the relevant literature, we identified the factors that affect the implementation of privacy and security considerations in the development process and developed a model that categorizes those factors into groups on five main levels (Table 2): environmental, organizational, product related, development process related, and individual. Next, we briefly describe the factors for each of these levels.

### 3.1 Environmental Factors

Environmental factors characterize the context that surrounds the company and affects the adoption of security and privacy practices in the development process, such as legal regulations, industry standards, perceived social norms, and economic and market trends.

*3.1.1 Laws, Regulations, and Industry Standards.* The reviewed studies discussed the regulatory bodies concerned with privacy violations in software development, among which three were the most prominent: government, platform authorities, and authorities enforcing industry standards.

*Government policies* represent the federal, state, local, and industry-specific laws and regulations (e.g., HIPAA, COPPA, FERPA) protecting consumer privacy and security. These regulations are often unclear and confusing, making it difficult to comply with them [12, 31, 100, 108], or lagging behind the rapid evolution of technology [26], and quickly becoming outdated [22, 26]. Some even raised concerns about governmental sovereignty over corporations in regulating privacy that can be entangled with national interests, such as when governments force companies to provide them with access to user data through so-called "backdoors" [16]. Moreover, data protection laws and regulations often prescribe vague directions that require substantial input from human judgment and expertise to interpret the implications in practice [13, 31]. For software companies operating in multiple countries, it is especially hard to comply with the variety of potentially contradictory regulatory requirements of different markets. Finally, the use of cloud services has major privacy implications and raises additional regional legal challenges, such as those dependent on where users' data is physically stored.

*Development platform policies*, such as Google Play Store, Apple's App Store, Amazon Web Services, Microsoft Store Policies, and Code of Conduct offer developer policies and guidelines outlining requirements for systems. Some recommendations play an advisory role, suggesting the best but optional practices, whereas others are mandatory—their implementation is reviewed and is necessary for approval by the platforms. Development platform policies help to inform developers about security standards and encourage them to adopt secure coding practices [22], and provide a

certain degree of data privacy by imposing privacy requirements [51]. However, these guidelines can be seen as inefficient, because different platforms may not be aligned in terms of definitions of privacy-related terms [81] and promote diverging or even conflicting values [100, 105]. The general challenges of cross-platform development can be further exacerbated by different platforms requiring developers to comply with different responsibilities for data protection and privacy disclosures. Better alignment across platform policies, potentially enforced by regulations, will reduce the overhead of developing software for a variety of different platforms, and eventually improve the adoption of best practices for privacy and security.

*Industry standards* attempt to self-regulate the privacy and security practices in a specific industry and represent the set of privacy and security requirements that are generally accepted and followed by most members of a particular industry. Industry privacy and security standards such as the Payment Card Industry Data Security Standards (PCI DSS) and ISO/IEC [68, 82] offer privacy and security guidance to software companies. These standards often focus on requirements around *what* privacy and security outcomes software companies need to achieve, without adequate guidance on *how* to best achieve these goals. Having a clear set of widely agreed upon best practices, standards, and processes for implementing them would positively affect developers' adoption of best practices for privacy and security.

*3.1.2 Perceived Social Norms and User Expectations About Privacy and Security.* Developers' perceptions about norms prevalent in society or certain cultures affect their propensity to deploy security and privacy principles in the product design and development process [16, 57, 60]. Prior works agree that different perceptions of privacy or different needs based on individual preferences can lead to diverse types of concerns about privacy [13] and various expectations about usability, security, and privacy [22]. To account for variability in social norms, researchers highlight the importance of involving broader societal groups into the discourse and enforcement of privacy norms and regulations [12].

*3.1.3 Competition and Reputation.* Market competition and company reputation could either motivate organizations to implement privacy and security engineering or diminish its priority. For instance, strong competition might push organizations toward aggressive business models (e.g., focusing on personal data monetization or invasive data-driven targeting approaches) and diminish ethical practices in the race for the market share [111]. However, companies with a large market share may be less concerned about the loss of some customers due to a data breach incident than companies operating in a highly competitive environment, where a publicized data breach scandal can create a wave of customer switching, significantly affecting the business [60]. Such reputation risks encourage companies to pay more attention to security [7].

### 3.2 Organizational Factors

Organizational factors represent the aspects pertinent to the company, such as its maturity, available financial and human resources, privacy and security culture, management support, organizational incentives, the proliferation of privacy/security knowledge within the organization, and organizational and team structure.

*3.2.1 Organizational Maturity.* The maturity of an organisation (not solely determined by age) plays a role in prioritizing security and privacy practices within it, and can be correlated with other factors in our model. For instance, leaders of startups may initially be very focused on fundraising and growth—the existential needs of a new business—and as their products mature, they may start giving more consideration to privacy matters [26, 100]. Expansion to international markets

requires compliance with international privacy regulations [100], increasing the relevance of the environmental factors discussed earlier (Section 3.1.1).

*3.2.2 Financial and Human Resources.* A lack of resources is detrimental to the adoption of privacy and security practices [12, 108]. Conversely, the availability of sufficient human resources who could take on the job of ensuring security and privacy practices is an important factor for their adoption [7]. Some companies prefer to have an expert specialized in security rather than to try to educate the whole team about it [132].

The portion of company's financial resources allocated to the privacy and security budget, specifically, plays an important role in the adoption of privacy and security practices in the development process [7, 14, 47, 53, 70, 94, 137]. For instance, introducing security tools can take a substantial cut of a company budget [70, 132] but also result in indirect costs such as developers' time [11, 132]. Professional security training [14, 53, 70, 94, 137] and security certification (e.g., ISO) are often seen as too costly to implement in terms of time and resources while their value is questioned by many companies [60, 74]. Moreover, security risks are often underestimated in relation to the investment required to protect against them [112].

*3.2.3 Privacy and Security Culture.* Privacy and security culture represent shared perceptions, beliefs, and social norms surrounding privacy and security [57, 132], the commitment to address concerns, and promote a privacy and security mindset [60]. Privacy and security culture plays an informal role in affecting organizational privacy conduct [6, 8, 122], encouraging and supporting security practices [7, 60, 70, 132], the development process [70], and developers' choices regarding security [60]. As engineers do not make independent decisions about system design and their work is situated in a certain context [16], previous research recognizes the strong effect of an organization's privacy norms on developers' privacy design behavior, conditional on engineers' motivation to comply with them [112]. Inefficient organizational norms and practices can put developers under the impression that privacy is not an important value in the organization, with negative consequences [57].

*3.2.4 Management Support.* Senior management privacy and security awareness and support have a strong influence on the implementation of security and privacy practices [47, 64, 70, 74]. Prior research recognizes management's responsibilities in supporting security and privacy culture both at the top level [49, 63, 69, 70] and through internal team supervision [57], by providing adequate resources for security implementation and communicating their expectations clearly [49, 64, 70], or mediating the communications between various interest groups to resolve related conflicts [63]. The lack of understanding of the security practices and their importance in the development process [70] may result in deferring security [7].

*3.2.5 Organizational Incentives.* Providing developers with incentives (rewards and sanctions) can impact their privacy and security practices [70]. For example, in a recent qualitative analysis of developers' comments on Reddit, it was found that they perceive little additional benefit from the substantial amount of additional effort required for compliance with Google Play privacy and security requirements [80]. Rewards can include monetary incentives [7, 59], feedback and empowerment [22, 57, 59, 117], and recognizing the value of employee work [59, 117]. The lack of incentives can encourage developers to prioritize functionality over security/privacy [22]. However, encouraging developers' intrinsic motivation has been recognized as a more efficient strategy than extrinsic, especially financial, rewards [7, 59, 109, 117].

As sanctions, developers can be penalized for failures to comply with security standards [22, 64, 109]. The certainty of detection has a stronger influence on security behaviors than the severity of penalty [64]. The combination of preventive methods for privacy protection and punitive

mechanisms (e.g., reporting violations to authorities) can act as an efficient strategy to discourage developers from risky behaviors [13]. Instead of introducing sanctions, some researchers suggest that companies should encourage developers to report errors and ensure fair investigation [32].

*3.2.6 The Proliferation of Privacy/Security Knowledge Within the Organization.* The knowledge that organizations circulate in the form of training, educational courses and materials, peer and privacy/security champion support, and so on has also been shown to influence the security and privacy practices of the developers.

*Privacy/security education and training* presumes the exposure of company employees to privacy and security knowledge resources that help developers understand the potential impact of privacy and security problems on the organization at large [8, 99, 132]. Application of a situated learning framework, where knowledge transfer happens directly within a development team, has been found to be more effective in incorporating such knowledge into practice than theory-oriented learning curricula [122]. Privacy/security training is considered valuable not only for the developers [22, 70, 99] and security advocates [59], but for all stakeholders involved in the software development process [70], as they ensure the support of security initiatives as an integral part of the organization [30]. Yet, security training rarely teaches developers to use security tools [60, 70, 98, 132] and ignores "soft skills," such as communication, collaboration, presentation, and writing [58].

***Peer Support.*** User studies with developers identify peer support as a key resource in judging the ethics of their decisions about privacy [112], and encouraging them to discover new security tools [103], and adopt security best practices [15, 22, 132]. Developers seek peer advice from privacy/security specialists, current and former colleagues, friends, and other developers, such as from forums, meetups, or work-related groups [11]. Yet, guidance by others' examples can also be counterproductive, as it may not address important topics and may include outdated advice [3].

***The Role of Privacy/Security Champions.*** Instead of trying to educate each employee about privacy and security, some research recognizes the value of privacy and security champions who act as experts or enthusiasts "leading by example" [60], gradually shifting the privacy/security culture in a positive direction [17, 117, 122], and even taking part in the development of effective organizational security policies for employees [15]. In contrast to regular peer support, champions take on a proactive role in promoting privacy and security values in the organizations.

***Q&A and Code Sharing Websites.*** Q&A (questions and answers) and code sharing websites, such as Stack Overflow or GitHub [2, 11, 77, 79, 118, 132], provide developers with technical support and privacy and security related knowledge, which might not be available within the company [11], and presented in a more comprehensible and less formal fashion than official documentation [79], such as code examples or examples of how an API works in a particular context (in contrast to a general API documentation) [79]. Despite their usefulness, relying on these resources, even when high-scoring answers are provided by the highly ranked peers [132], can lead to less secure solutions [2, 42, 90], proliferating vulnerabilities [22], and ignoring the rationale behind the provided recommendations [77, 79].

***Media and Other Resources.*** Mass and social media, as well as blogs, are increasingly used as the channels of privacy and security knowledge dissemination [58, 132]. The content of such channels can be more engaging than formal documentation due to the use of images, metaphors, or pop culture references [58]. The exchange of moral and cautionary tales, news, or stories about legal repercussions and other consequences for developers help developers justify privacy values, and rationalize their technical and instrumental realizations of privacy [105].

*3.2.7 Organizational and Team Structure.* The structure of development teams varies in the degree of specialization from narrow-focused specialized units (that might not communicate with other departments) to teams with wide domain diversity.

***Siloed Teams.*** Privacy and security practices could be improved through the collaboration of software development and design teams with legal [16] or business departments [12, 43, 49, 54]. However, such collaborations can face challenges, as lawyers and developers often "don't speak each other's language"—that is, don't share the vocabulary and conceptual frameworks of privacy [16]. Such communication issues often result in siloed teams, leading to a variety of problems at different levels. For instance, privacy professionals might be locked in legal compliance departments, hindering the access of the development teams to their expertise [12]. In contrast, privacy professionals in "engineer-only design teams" might not have an opportunity to raise and address privacy issues during the design process [126]. Bureaucratic barriers can further hinder the institutionalization and spread of privacy norms within the organization [100, 126]. Even when developers have access to relevant experts, they may choose not to consult them due to the shortage of time and communications overhead it would require [55]. To resolve the communication issues between different departments, privacy experts and teams may be called on to mediate the interests of different stakeholders, balancing the external and internal privacy requirements and practices [12, 112].

***Team Diversity.*** The lack of demographic and background diversity limits engineers' perspectives and increases the likelihood of discriminatory implicit biases around privacy and security [100, 126]. Increasing team diversity [58, 70, 126] by involving security experts and employees responsible for facilitating cross-departmental connections could narrow the proficiency gaps [58] and help establish trust and shared knowledge among team members [126], leading to more successful self-management of teams [43] and a greater sense of belonging and collaboration [59].

### 3.3 Process-Related Factors

This section describes the factors related to the software development process that affect the implementation of privacy and security considerations in software products. These factors can have an impact throughout the process (e.g., internal organizational documentation) or at a specific stage of the software development lifecycle—requirements, implementation, or review and evaluation.

*3.3.1 Internal Organizational Documentation and Procedures.* These include organizational policies and guidelines that recommend certain practices and tools, aiming at ensuring privacy and security in software products. For such documentation to be effective, it not only needs to be available in the organization, but developers also need to be aware of it and perceive it as useful.

***Availability.*** The existence or even enforcement of certain policies and procedures to address software security and data privacy [7, 70, 103, 109] is necessary for the implementation of secure software development [9, 32, 70] and privacy engineering [101], especially in the companies with privacy as a core value or companies operating in privacy-sensitive domains, such as healthcare and finance [57] (see Section 3.4.1). Although there is a wide variety of available security resources, companies might lack a formal plan or process for choosing, adapting, and integrating them in practice [55, 60, 82, 96, 124]. As a result, developers might lack suitable resources [22, 132]. Unlike security tools, privacy tools to assist software development are more scarce [11, 117], or address privacy through security mechanisms, such as secure data sharing [31].

***Awareness.*** Although security and privacy procedures might be in place, developers may not be aware of them, or whether they are mandated to read, use, and comply with them [74]. Some studies report high awareness about internal procedures and policies among developers [7, 57, 112], whereas other studies suggest that developers are often unaware of privacy recommendations [26], privacy threat models, mitigation strategies, less privacy invasive coding alternatives [79], privacy-specific tools and checklists [11, 126], and code analysis and testing tools [6]. Developers also

have little knowledge regarding privacy and security regulations [11], security tools [7] or secure development lifecycles [47], and concepts related to usable security [80].

***Perceived Usefulness.*** In addition to developer awareness of existing documentation and policies, companies should ensure that developers perceive the security and privacy practices described in these policies as useful and feasible [15] and that the guidelines and documentation are comprehensive and easy to understand [79]. Developers are less likely to implement recommendations when they doubt their effectiveness and usefulness [47, 74]. For instance, some developers find privacy recommendations provided by the Federal Trade Commission (FTC) outdated, irrelevant, too generic, and, therefore, not useful [26]. Another widely used tool, Data Protection Impact Assessment (DPIA), may not match the system architecture and thus be perceived as obsolete, outdated, or even incorrect [107]. Developers also believe that certain privacy mechanisms can be easily broken, overridden, overruled, or de-anonymized [16], whereas the guidelines for implementing such mechanisms are complex and too theoretical to be used in practice [103]. Some third-party privacy tools even raise concerns, as they might collect information that developers are unaware of [11]. Security code analysis tools might be considered not very useful due to their complexity [7] or due to time resources they require to implement [132]. Academic resources on security might become outdated as well and are often perceived as distant from real-world challenges [60]. The value of security certification may be doubted due to its costs exceeding the perceived value [74]. Security certification might even be perceived as counterproductive: it can discourage product updates, as the company needs to apply and pay for the certification after it is voided following every software update [60]. Some developers do not trust existing standards, for instance, due to evidence of government intentions to purposefully weaken cryptographic protections [52].

*3.3.2 Requirements Stage.* Software entrepreneurs tend to underestimate the role of privacy at the initial stages of business development [6, 26], despite prior research agreeing that it is important [31]. Developers generally more easily agree that they should consider *security* from the earliest planning phases rather than privacy [7, 29, 85]. Auditing the security of the code only before the code integration or its release can pose significant security risks [22, 132]. The failure to consider privacy and security from the early stages of software development is associated with the difficulties with defining privacy and security as concepts and requirements, and with the tensions between privacy/security and other technical/system requirements.

***Difficulties with Defining Privacy and Security Concepts and Requirements.*** Development teams are usually familiar with the concept of software security requirements; however, the concept of privacy in software development is considered to be rather abstract and vague [13, 16, 22, 76, 103, 108, 110, 112] and context dependent [16], thus difficult to implement. Since, conceptually and methodologically, privacy is often confounded with security [110], developers sometimes use the data security vocabulary to approach privacy, which limits their consideration of privacy [57], or even sacrifice privacy for security [7, 31]. This is particularly concerning, because improving security does not always imply improving privacy (as in case of confidentiality), and can even have the opposite effect. Difficulties with conceptualizations, complexity, constant evolution, and context dependency also translate into difficulties with defining privacy [13, 31, 89, 91, 110] and security requirements [70].

***Tension Between Privacy/Security and Other Technical and System Requirements.*** Generally, data protection requirements are considered as non-functional and might not fit into standard software development practices [73]. Security [7, 31, 82] and privacy can interfere with other requirements, such as functional requirements, integrity requirements, performance, and usability [6, 13, 16, 22, 55, 60, 79, 91, 103, 108, 126]. For instance, the collection and use of end user data

for the optimization of services and design might compromise data protection agreements [54], a nuanced user authentication process (e.g., two-factor authentication) requires additional user effort [22, 51], and developers often find it hard to obtain a meaningful informed user consent using existing mechanisms [11, 16, 54, 73, 114]. However, ignoring the privacy needs of users can negatively impact their trust and loyalty to a product or a service [31]. Thus, developers face the challenge of balancing security and usability [15, 19, 62]. Although the involvement of users in the design process via user studies and reconciling their sometimes conflicting preferences for privacy, security, and usability is not easy [31, 55], the benefits of such user-centered approaches are undeniable [31, 60, 100].

Complex and more nuanced requirements might be traded for simplification of the development process. For instance, developers might request just one permission on the multiple data items from users, which will lead to excessive data collection [79] violating data minimization principles [66]. To prevent privacy risks, previous studies recommend incorporation of privacy considerations into the definition of software requirements and specifications [16], which might require improving general software requirements that are often not sufficiently maintained and managed [57].

*3.3.3 Implementation Stage.* Privacy and security requirements might be in place, but developers still might not consider them during the implementation stage. The reasons vary: it is not easy to operationalize them or prioritize privacy and security over time pressure, or due to usability issues with the privacy/ security engineering tools.

***Difficulties with Translating Requirements into Practice.*** A large number of studies recognize that it is difficult for developers to translate privacy [16, 31, 38, 73, 100, 103, 135] and security [22, 60, 134] requirements into specific software development processes. Partially, it is related to the underlying difficulties with defining privacy concepts (see Section 3.3.2) and lack of knowledge (see Section 3.5.2), as well as with technical challenges, such as identification of sensitive information [135], technical complexity of the system (e.g., in a cloud environment) [23], and technical implementation of data anonymization [31, 73], data minimization [103], and encryption [73, 135], especially if organizations fail to provide developers with the methods and resources necessary for supporting the implementation of privacy and security requirements [110]. Developers might also direct their attention to the formal procedures and fail to implement a distributed privacy architecture [12] or implement privacy methods in isolation, via different stakeholders that have different levels of knowledge of the system [107]. Certain development approaches, such as Agile and DevOps, might require ad hoc solutions to address security requirements [23, 31], due to the privacy risks posed by the modularity of these approaches and concentration of user data "in the hands of specialized service providers" [54]. To address it, similarly to waterfall development methodologies [102], organizations might include privacy and security practices in every phase of the development process [70, 73]. The need to comply with the variety of potentially contradictory requirements, such as if an app is networked to communicate with multiple fragmented services, introduces further complexity in searching, reading, and reconciling differences in the documentation and requirements of each of those services [119].

***Tension with Time Priorities.*** Limited time, especially in the conditions of time-to-market competition pressure [22, 60], can become an impediment for data protection implementation [11, 112, 124] and for improvements in the usability of privacy and security features [55]. Privacy is usually not a priority task that developers are ready to allocate time and resources to [10, 96, 110, 126]. Some engineers believe that implementing privacy features can slow down the development process [16, 74]. Similarly, security is pushed down the priority list in the conditions of tight deadlines in which most software companies have to operate [47, 70, 72, 94, 112, 132, 134]. Although keeping the overall development time within adequate limits is important [70], the

shortage of time dedicated to security can ultimately result in a technical debt with later security issues leading to increased costs, system fragility, and reduced rates of innovation [54].

***Usability Issues of Privacy/Security Tools.*** Usability of privacy and security tools is important [70, 95, 131]. Issues with usability, steep learning curves, and limited library support reduce the adoption of such tools by developers [7, 27, 44, 79, 128]. Poor default configurations in tools and libraries, confusing security APIs, and insufficient documentation lead to errors in their usage [22, 39, 41, 48, 73] and in developers' correct disclosure of libraries' data practices [81]. The lack of interoperability of cryptographic libraries on multiple platforms also impedes collaboration between teams and oversight of a security architect [60].

*3.3.4 Review and Evaluation Stage.* The review and evaluation stage involves an assessment of whether and how the initial requirements are implemented in the system [73]. In this stage, issues with the evaluation process and metrics may arise.

***Evaluation Process and Metrics.*** Privacy assessment mechanisms recommended and used by the legal enforcement authorities are limited to a narrow set of privacy mechanisms, and lack guidance on how to technically implement them [12, 16]. Although app marketplaces provide certain assurance seals based on the app review process [105] and automated compliance checks are used to verify the compliance with privacy regulations [16], clear and objective criteria and metrics for assessing success in addressing privacy issues are still largely lacking [31, 103, 117]. As Assal and Chiasson [6], Votipka et al. [125] indicate in their research, developers admit limited testing for privacy and security risks. As systems often change, introduce, and remove features [54], it is challenging to keep the security standards (e.g., for cryptographic products) [60] and privacy assessments up-to-date, creating a need for continuous privacy management and monitoring [110]. The lack of automated tools for privacy and security assessment makes developers rely mainly on their own expertise [23], increasing the amount of time required for assessment, and the probability of human error. However, manual code reviews may improve developers' understanding of underlying data processes [117]; in conjunction with security tools, manual code reviews lead to the best results [132].

### 3.4 Product-Related Factors

This dimension includes product-oriented factors, namely to what extent the product requires access to the user data and what is the relevance and importance of privacy and security for the product; how much the product relies on user data to generate revenue, exacerbating the tensions between privacy and business priorities; and whether privacy implications of a product are recognized as its potential competitive advantage on the market.

*3.4.1 Relevance/Importance of Privacy/Security for the Product.* Companies' beliefs about whether their products are an interesting target for security attacks influences their eagerness to address privacy and security concerns [7]. For instance, developers working on B2B products [26, 60] or internal applications [132] do not feel the need to deploy strong security safeguards. In contrast, a large or growing user base is believed to make a product an attractive target for attack and invokes developers' concerns about data security [100, 132]. High perceived sensitivity of the data or context in which it is collected (e.g., finance, health, child- or education-related contexts, especially if they are subject to special regulation [11, 31, 93]) also leads to a higher degree of privacy and security concerns among developers [13, 22, 26, 31, 57]. Conversely, products not collecting personally identifiable information lead developers to demote the importance of privacy and security in product design [11].

*3.4.2 Tensions with Business Priorities.* The tension between privacy and primary business priorities focused on revenue maximization often hinders the implementation of privacy-preserving

features in software products [57, 74, 101], especially when data-driven business models rely on the monetization of personal information as a source of revenue [54, 112] and in early-stage startups [26]. Time and budget spent on privacy engineering is believed to be better invested in innovation, development, and growth [55, 112].

*3.4.3 Competitive Advantage.* Security and privacy centric features can help organizations to distinguish their products in the marketplace [13, 31, 51, 54, 60, 61, 67, 70, 105, 112]. For instance, in a "crowded" software market such as Android apps, privacy features can be used to differentiate products from competitors [105]. Greater competitiveness can also be achieved by efficient privacy management [13, 26] and providing users with support and transparency regarding their data [51, 100]. Engaging in privacy research further helps companies understand and better address user needs and preferences, consequently improving the product overall [31]. However, some developers fail to recognize the competitive advantages of embedding privacy in products or obtaining privacy and security certification [74], due to the lack of awareness regarding the benefits and risks associated with user data privacy and security practices [110].

### 3.5 Personal Characteristics

Developers' personal characteristics and backgrounds include developers' position and role in an organization, their expertise and knowledge, privacy and security attitudes, previous experience with privacy and security violations in the software development context, and personality traits.

*3.5.1 Position and Role.* Hierarchical position and role, perceived personal responsibility, and autonomy and control over privacy and security decisions have been found to affect their implementation.

With respect to *hierarchical position*, engineers in senior or managerial positions tend to have more responsibility and control over privacy and security engineering compared to employees in more junior roles [112]. Previous studies identify that such division can lead to negative consequences. For instance, limited involvement of non-senior level employees in high-level firm decision making can lead to poor adoption of privacy principles in the development process [12]. Besides, those in managing roles can lack substantive expertise to make privacy decisions [108].

*Perceived personal responsibility* of developers and engineers does not always depend on professional position or role [74, 112]. Prior research often reports the lack of perceived responsibility of the developers and engineers in implementing and enforcing security and privacy, or limiting of perceived responsibilities to, for instance, only minimizing data usage [79, 90]. Specifically, "not my problem" mentality [98], lack of "moral responsibility," and absence of privacy and security requirements among deliverables [16, 76] in a job description [7] or formal responsibilities [6, 57] often lead developers to neglect security and privacy engineering [112]. Absence of personal responsibility can be also caused by the developers' misconception that their mistakes are unlikely to cause security vulnerabilities in the system [70]. Some developers believe that users themselves are supposed to protect their personal data [16, 51, 96, 112, 115]. Even when the primary responsibilities directly related to privacy and security engineering and coding are clearly defined, it may be hard to define, especially in small organizations, responsibilities for secondary tasks, such as user interface design of privacy features, audits, privacy policy updates, or privacy nutrition label assignment [119]. In the absence of specialized experts, these tasks may be assigned to the developers [55], who may not have sufficient expertise to perform these tasks, leading to subpar quality of implementation.

A general lack of a clear distribution of privacy and security roles and responsibilities in many organizations or teams presents another challenge [20, 74, 96, 110]. In some cases, such roles are substituted with collective responsibility [16, 26, 76], reducing a more systematic privacy engineering

approach to ad hoc decisions that lack enforcement [31]. On the one hand, in the presence of specialized privacy or security experts, the developers shift the responsibility over to them [11, 79, 100, 112, 132]. On the other hand, the lack of specialized experts could result in non-expert staff taking on part-time responsibilities for implementing security and privacy [74].

When developers perceive a lack of *autonomy and control* over decisions and implementation of privacy and security features [30, 109], they are less likely to take action to influence or execute such decisions in software design and more likely to rely on external advice about it [79]. The autonomy to act according to personal beliefs may also be overridden by the tendency to comply and conform with organizational decisions [104, 112].

*3.5.2 Privacy Expertise/Knowledge.* The factors related to privacy and security expertise, knowledge, and skills of developers are commonly mentioned as predictors of implementation of privacy and security practices [6, 16, 55, 90, 112]. Important skills for implementing security practices include a wide range of expertise including technical security skills [58, 59], competence in assessing security risks [17, 125], efficiency in applying security tools [22], and the ability to deal with a great degree of technical and organizational complexity [56]. Even though developers might have the skills necessary to implement some security mechanisms, they do not always have security expertise [2, 22], as it often requires knowledge from different fields and interdisciplinary collaborations [55, 82, 96, 122]. Compared to security, developers might be less prepared to deal with privacy challenges [100] and may not have appropriate privacy expertise, characterized as the ability to incorporate information privacy mechanisms in practice [16]. Developers find it difficult to make decisions about appropriate levels of privacy and when in the software development process they should incorporate it [103], especially when there are no guidelines about what it means to implement privacy and how to balance it against business priorities [101]. The lack of formal training is particularly evident when it comes to privacy [106], and such discipline is much needed to train privacy experts [13, 33, 34, 75, 123]. The lack of sufficient knowledge and awareness to implement security and/or privacy often results in adoption of unreliable third-party services [11, 79, 112], introduction of security vulnerabilities during the development process [7, 14, 40, 60, 70], misunderstanding of potential privacy threats and corresponding coping strategies [79], and frustration over decision making about embedding privacy in the development process and making developers rely on their personal opinions rather than objective knowledge [103]. Even when the developers have appropriate privacy and security knowledge, keeping that knowledge up-to-date may not be always easy [6, 90].

*3.5.3 Instrumental Privacy and Security Attitudes.* Based on the Theory of Planned Behavior [4], instrumental privacy attitudes reflect developers' opinions about the importance of information privacy [16, 112], which we extend to security attitudes as well. Such attitudes may be shaped by the developers' background, including their multicultural environments [100], and their personal opinions and beliefs in relation to privacy [8, 16, 103]. Some developers perceive privacy practices as relevant and important [16, 112], and others as unimportant [103], for example, due to the lack of awareness [74] or motivation to protect privacy unless it is required [79]. Positive instrumental security attitudes are associated with the higher uptake of security tools [130] and act as a strong motivation for implementing security practices [7, 59]. Some developers do not recognize the value of the effort invested in software security [25]. Developers who doubt the feasibility of building secure systems may also doubt the importance of following the security practices and be less motivated to incorporate them [112], which illustrates the necessity to consistently motivate and support their confidence in the importance of protecting data security [70].

*3.5.4 Experiential Privacy and Security Attitudes.* Experiential attitudes indicate developers' spontaneous feelings and emotions about security and privacy practices that affect its adoption and implementation [16, 17]. Motivation to advocate for software security among developers is also characterized by their interest in the field and self-challenging with security tasks [7, 59]. Engineers working in industry find security engineering less unpleasant than privacy engineering [112].

*3.5.5 Prior Experiences with Privacy/Security Violations in Software Development.* The prior experience of developers in collecting and storing personal information affects their privacy and security practices [6, 8]. Experiencing a security issue can increase developers' awareness of, concern about, and attention to security for a long time [7], or motivate them to use security tools [132].

*3.5.6 Personality Traits.* Some personality traits are associated with the adoption of privacy and security practices as well. For instance, the locus of control, which captures "the beliefs of individuals about whether the outcomes of their actions are contingent on what they do or on the machinations of outside forces" [71, p. 4], is a positive predictor of the adoption of ethical engineering in general, and privacy and security engineering specifically [112]. Inquisitiveness may affect the adoption of security tools [129]. People with pronounced imagination and emotionality, low immoderation [45], high proactiveness and reactiveness [56], and good soft skills [59], such as communication and people skills, context awareness, and service orientation [58], are more likely to become successful security advocates. One study mentioned the religiousness can positively affect a developer's ethical decision-making process and, consequently, privacy-related decisions [112].

## 4 DISCUSSION AND CONCLUSION

Prior work provides a wealth of insights about the barriers and enablers to adoption and implementation of privacy and security practices in software development. It is important to note that most of the factors are interconnected and there is nearly an infinite number of ways to present their relations and prioritization in a given context. Different factors may be more or less useful or even implementable in different circumstances. However, such efforts are segmented (e.g., between security and privacy domains, theoretical and empirical studies), use diverse terminology, and sometimes find contradicting results without acknowledging how their new findings contrast with earlier ones. Thus, our work addresses the need for systematization of knowledge on this topic.

In this section, we highlight the most prominent potential implications of our research and directions for future work. Based on a well-known behavior change model, we map the strategies that are aiming to remove the barriers and stimulate the adoption of security and privacy practices in software development.

### 4.1 Multi-Level View of Security and Privacy Practices

Our review highlights the importance of evaluating and promoting the adoption of security and privacy practices on various levels: from the environments such organizations are operating in to the organization-related factors and factors related to the development process and produced software products or services to the factors related to personal characteristics of developers.

From previous research, it is evident that organizations or individual developers can have different levels of control of various factors related to security and privacy practices, which is important to identify and consider. For instance, on an environmental level, laws, regulations, and industry standards are often given conditions for specific regions and industries, but organizations can influence their privacy and security reputation. At the organizational level, depending

on the organizational maturity, individual developers within development teams might not have control over the allocation of financial resources to security and privacy practices but can act as privacy/security champions promoting privacy and security values and eventually improving the privacy and security culture within the organization.

Another important point to consider is possible value tensions and conflicts that can arise when it comes to the allocation of organizational resources to security and privacy practices. For instance, such tensions can arise between privacy/security practices and system requirements at the process level or business priorities at the product level. Although economic motivation is not the only driver for companies' decisions to adopt strong privacy/security practices, the main goal of a business is to generate profit, and economic arguments are likely to play a dominant role in organizational decision making. Hence, as long as privacy violations have no explicit costs, and privacy engineering has high implementation costs, companies are less likely to adopt good privacy practices. Companies' "costs"may be direct and expressed as penalties/fines for regulatory violations, other legal expenses associated with litigation and policy compliance (e.g., lawyers, audit, third-party compliance tools), and re-issuing of compromised credit cards, as well as privacy software licensing, training, and so on. These "costs"can also be indirect and associated with decreased user engagement, reputational damage, and increased marketing and PR budget to mitigate negative impact, loss of competitive advantage, and user trust (e.g., after a recent ToS update, WhatsApp lost millions of users to competitors [65]). A strong value proposition may indeed offset some of the indirect costs; however, their mere existence increases the chances that companies will take privacy and security into consideration when defining business strategies. Hence, the choice of factors to manipulate in order to improve the adoption and implementation of security and privacy practices in software development teams should consider both the opportunities to leverage such practices but also have a realistic view of possible value tensions within the organization.

### 4.2 Privacy and Security Concepts

The concepts of privacy and security are closely interconnected in the development process but still different, which is not always recognized. Although security appears to be conceptually clearer to developers and engineers, and often recognized as important both at an organizational and personal level, awareness about the impact of privacy is often lacking. Moreover, conceptually and methodologically, privacy is often confounded with security [110], and developers sometimes use the vocabulary of data security to approach privacy, which limits their consideration of privacy [57]. This is particularly concerning because improving security does not always imply improving privacy (as in the case of confidentiality), and can have the opposite effect. For example, attempts at improving security can increase surveillance and the extent of data collection [31].

There is also a contradiction between privacy and security for the developers from the point of view of users: although they sometimes believe that privacy is not important for users, which may discourage them, security is the opposite, and user expectations may encourage more attention to security.

Privacy engineering is less mature than security engineering. However, although some barriers are more pronounced, or less addressed, in the privacy domain than in the security domain (automatic tools for vulnerability discovery, clear taxonomies of risks/attack surfaces, mitigation approaches, etc.), we observed that *factors* affecting their implementation are similar. In other words, although relative magnitude, prevalence, or importance of a particular factor may differ between privacy and security engineering fields, the factors themselves are overlapping between the two fields.

### 4.3 Behavior Change Implications

We propose to map the potential strategies for leveraging the factors described in our model using the **Capability-Opportunity-Motivation Behavior (COM-B)** model from the widely recognized behavior change theory [87, 88]. The COM-B model describes a framework, in which capability, opportunity, and motivation are three essential conditions for generating behavior. Thus, we discuss our findings of barriers and enablers for implementing privacy and security practices in software development from the perspectives of stakeholders' motivations, capabilities, and opportunities to engage in behaviors aiming at protecting end user privacy and security. Stakeholders include any actors involved in the model (engineers, companies' management, policy-makers, etc.).

*4.3.1 Leveraging Motivation.* To increase companies' motivation to take privacy and security seriously, policy-makers, industry associations, and software platforms need to create and enforce regulations and requirements that oblige companies to protect users' data, make those regulations and guidelines easy to understand and interpret, and include quantified metrics for measuring success in compliance with them (Section 3.1.1). Penalties for violations of users' privacy and security would further impact product-related factors, such as tension between privacy/security and business priorities (Section 3.4.2). This will make it more costly for organizations to ignore privacy and security aspects of the products and services they create, and encourage the inclusion of the potential costs of violations into the profit calculus, thereby moving privacy and security objectives up in the list of business priorities. Providing rewards for better privacy practices is another way to improve incentive structures (Section 3.2.5). For example, app marketplaces can offer better search rankings, special app featuring options, and privacy/security badges to those apps with better privacy and security practices. User-facing certificates and badges can be offered outside of the app ecosystem as well, to signal positive privacy/security practices to users, as a way to differentiate from competitors.

The demonstration of evidence that privacy and security of software are enforced can be a valid competitive advantage (Section 3.4.3) that may attract and retain users. In contrast, violations of privacy and security may repel users and harm a companies' reputation (Section 3.1.3), thus motivating companies to include it in their strategic planning. To demonstrate such effects, more academic and independent market research needs to be conducted about the economic impact of negative and positive privacy and security reputation. The results need to be disseminated not only in academic outlets but also in mass and social media, business magazines and blogs, and other resources consulted by business executives. Similarly, academic and market research needs to regularly survey user expectations and perceived social norms around privacy, security, and data collection and sharing (Section 3.1.2), and disseminate the results, to raise awareness, dispel potential misconceptions about user beliefs, and engage broader societal groups in discussions regarding privacy and security. Companies' engagement in user research and direct involvement of users in testing software prototypes could further align the views of software companies' employees with user beliefs, expectations, and preferences.

Software engineers often believe that privacy and security concerns are not relevant or important to certain products, such as B2B or internal software services (Section 3.4.1). However, practically no software is safe from privacy/security risks. Threat modeling exercises (e.g., using Security Cards[2]), regular vulnerability discovery activities (e.g., bug bounty, penetration testing, threat analysis), and less formal activities (e.g., hackathons) can help engineers to correctly assess the relevance of privacy/security issues and vulnerability of the systems.

[2]https://securitycards.cs.washington.edu.

To leverage the personal motivation of software developers, job descriptions need to include the protection of privacy and security as part of the official personal responsibilities of software developers, regardless of whether they are part of the privacy or security team, or not (Section 3.5.1). Direct engagement of developers in code review may also promote their perceived responsibility for privacy and security aspects of the developed system, instead of entirely relying on a security team to do the reviews and making them fully responsible for privacy and security aspects of the system [134]. To leverage instrumental attitudes (Section 3.5.3), companies should emphasize the importance of addressing privacy and security concerns and explain the reasons. In the absence of personal experiences with violations (Section 3.5.5), case studies can be deployed to further raise developers' empathy and motivation to protect users' privacy and security. However, they could also try to change the experiential attitudes (Section 3.5.4) by making solving privacy and security issues more engaging, for example, by introducing gamified incentives, organizing competitions and hackathons, and using humor and positive framing in communications about this topic.

Organizational privacy and security culture (Section 3.2.3), including companies' vision, values, and code of conduct, can emphasize the importance of privacy and security at every stage of software development. Security and privacy champions and advocates act as experts or enthusiasts leading by example [60]. By motivating and encouraging such champions, companies can gradually shift and proliferate positive privacy and security culture to the rest of the organization [17, 117]. (See more strategies for promoting organizational privacy culture in the work of Tahaei et al. [117].) Evidence about the effectiveness of monetary incentives to encourage developers to protect users' privacy and security is mixed, often suggesting that intrinsic motivation is a stronger predictor than extrinsic rewards [7, 59, 109, 117], thus more research on this topic is encouraged.

*4.3.2 Leveraging Capability.* The COM-B model defines *capability* as the physical and psychological capacity to perform the behavior (e.g., engage in the necessary thought processes, like comprehension or reasoning) [87, 88]. To leverage developers' capability to engage in user privacy and security protection, it is important to ensure an adequate level of privacy expertise and knowledge (Section 3.5.2), for example, through training, peer support, other experts in the company exchanging their knowledge, and other resources (Section 3.2.6). Organizations can encourage and support informal procedures in relation to security and privacy, such as the sharing of empirical problem-solving knowledge among employees and development of personally devised security checklists [7, 132] or validation of security code libraries by peers before implementation [32]. Although peer support, Q&A websites (e.g., Stack Overflow), and other media resources can be more engaging than formal documentation, these sources of information may be less reliable [2].

Requirements engineering tools [136] may enable the identification and prioritization of privacy and security requirements and strategies (Section 3.3.2). Furthermore, better tools for implementing privacy and security best practices would reduce barriers to adoption and improve regulatory compliance. Examples of such tools include privacy-preserving libraries, data mapping tools, databases for implementing differential privacy, or tools for building GDPR-compliant user interfaces for obtaining user consent. It is also important to provide engineers an appropriate level of authority, autonomy, and control (Section 3.5.1) over their decisions about privacy and security features of the systems.

Finally, to support the ability of developers to detect and address privacy and security vulnerabilities, it is important to not only provide appropriate tools and libraries but also make them easy to use (Section 3.3.3). Similarly, internal organizational documentation and procedures (Section 3.3.1) and external privacy and security guidelines and documentation should be readily available to the developers, comprehensive, useful, and easy to understand [79]. It can be achieved by providing security reference guidelines that are adapted to non-experts [70], and by providing reputable

interactive third-party security implementations and tools, which could free developers from writing complex security code from scratch [60] and help to reduce programming errors [133].

*4.3.3 Leveraging Opportunity.* It is important to create opportunities for implementing privacy and security in software development, for example, by providing management support (Section 3.2.4), including security and privacy in the board's agenda [12], clearly communicating to employees their support of security advocates [59], and dedicating to privacy and security sufficient financial and human resources (Section 3.2.2), and time (Section 3.3.3), such as by budgeting time for privacy and security requirements and evaluation stages, and set more adequate deadlines and goals. Given that privacy and security are complex issues, companies should improve organizational and team structures (Section 3.2.7) to facilitate communication between teams about privacy and software, increase the diversity of opinions to obtain a variety of perspectives on controversial topics, and integrate privacy and security experts into all software development teams instead of creating a separate siloed privacy/security team.

To create opportunities for evaluation of a system's privacy and security, it is necessary to incorporate privacy and security reviews, and the principles of privacy-by-design [24], into formal software development practices, provide UI and UX guidelines and templates for obtaining informed consent, and develop metrics for evaluating privacy and security aspects of the systems (Section 3.3.4).

### 4.4 Limitations and Future Work

The goal of this study was to systematize existing knowledge about the factors that affect the implementation of privacy and security considerations and practices in software development. Future work is needed to validate the model, and further discuss the quantitative insights regarding the prevalence, relative importance, and relationship between the factors.

There are several limitations to the research described in this article. First, it is possible that the narrative review missed some studies and associated security or privacy factors. However, our secondary systematic search has expanded the set of studies included in the analysis, and revealed that our model reached saturation: all factors identified in those papers corresponded to factors discovered in the initial search. Thus, although there could be additional supporting evidence for our factors, we are confident that the model of factors is comprehensive and provides coverage of the major factors currently discussed in the literature. Our model is just the first step, and future research may modify it as needed. Second, as with any qualitative work, judgment is inevitably involved in categorizing the factors derived from the reviewed studies. There could be other approaches to categorization of factors into groups, different from ours (e.g., different logic in grouping the factors, or different choices about up-coding certain groups and breaking them into smaller sub-groups). However, we believe that differences in approaches to categorization do not have a significant impact on the core goal of this work—systematization of knowledge about what factors affect the implementation of privacy and security considerations and practices in software development.

Further research could focus on the strategies for overcoming the identified barriers and their effectiveness or operationalizing the model. For example, the model could be used to build assessment tools for evaluating the barriers to privacy/security practices adoption in the organizations, improving the mechanisms of translating requirements into practice or the metrics reliably measuring privacy-related performance. Future studies could assess the relative importance of the identified factors on the eventual implementation of privacy and security practices in software development.

## A NARRATIVE REVIEW PROCESS

Table 1. A Guiding Framework for Narrative Review

| Review Step | Description |
|---|---|
| 1. Developing a concept (March 2021) | (a) A research objective was defined to search for the factors or research hypotheses important in security and privacy practices of software developers and designers. |
| 2. Developing a preliminary synthesis (April 2021) | (a) The initial search strategy was developed and included the following: defining and scanning relevant venues and research groups publishing on the topic, snowballing search based on the initial set of relevant articles.<br>(b) The initial search resulted in a set of 99 documents.<br>(c) Repetition of ranked papers indicated saturation.<br>(d) 54 papers were identified as relevant. |
| 3. Categorization process and development of the initial model (April 2021) | (a) Preliminary reading and review of articles and checking/reading for relevant references. At this stage, 11 articles were excluded, as they did not focus on security/privacy factors within the development process.<br>(b) 43 papers were retained.<br>(c) Data extraction and categorization of the factors.<br>(d) Developing separate models, reaching saturation of factors, merging of models, and developing the initial model. |
| 4. Secondary systematic search (May 18, 2021) | (a) Following the initial research objective, search terms were defined and included *security and privacy by design*, *software*, *developers and development*.<br>(b) Search strategy resulted in 99 documents.<br>(c) Duplicates, dissertation, and gray literature (commercial reports, policy statements, or editorial papers) were excluded.<br>(d) 47 papers were identified as relevant based on their abstract and were read in full.<br>(e) 26 papers were included in the validation and refinement of the model. |
| 5. Refining the model (May 2021) | The factors extracted from the systematic search at stage 4 and not identified in step 3 were added to the model. The final set of papers included 69 publications selected in steps 3 and 4. |

## CONCEPTUAL MODEL

Table 2. Model of Security and Privacy Factors

| Categories | Sub-Categories | References |
|---|---|---|
| *Environmental factors* describe the context that surrounds the company and affects the adoption of security and privacy practices in the development process. | Laws, regulations, and industry standards:<br>• Government policies;<br>• Development platforms' policies;<br>• Industry standards. | [12, 13, 16, 22, 26, 31, 51, 68, 82, 100, 105, 108] |
| | Perceived social norms and user expectations about privacy and security | [12, 13, 16, 22, 57, 60] |
| | Competition and reputation | [7, 60, 111] |

(Continued)

Table 2. Continued

| Categories | Sub-Categories | References |
|---|---|---|
| *Organizational factors* are aspects pertinent to the company. | The proliferation of privacy/security knowledge within the organization:<br>• Privacy/security education and training;<br>• Peer support;<br>• Role of privacy/security champions;<br>• Q&A and code sharing websites;<br>• Media and other resources. | [2, 3, 8, 11, 15, 17, 22, 30, 58–60, 70, 77, 79, 98, 99, 103, 105, 112, 117, 132] |
| | Privacy and security culture:<br>• Organizational security culture;<br>• Organizational privacy culture. | [7, 8, 12, 16, 57, 60, 70, 112, 126, 132] |
| | Organizational maturity | [26, 60, 89, 100, 100] |
| | Financial and human resources | [7, 11, 12, 14, 47, 53, 60, 70, 74, 94, 108, 112, 132, 137] |
| | Management support | [7, 12, 47, 49, 57, 59, 63, 64, 69, 70, 74] |
| | Organizational incentives | [7, 13, 22, 32, 57, 59, 64, 70, 109] |
| | Organizational team structure:<br>• Siloed teams;<br>• Team diversity. | [1, 5, 11, 12, 16, 20, 31, 43, 47, 49, 54, 58, 59, 70, 74, 82, 100, 101, 110, 112, 126, 132, 136] |
| *Process-related factors* are the ones that can have an impact throughout the process (e.g., internal organizational documentation), or at a specific stage of the software development lifecycle. | Internal organizational documentation and procedures:<br>• Availability;<br>• Awareness;<br>• Perceived usefulness. | [7–9, 11, 15–17, 26, 28, 31, 32, 36, 47, 52, 54, 57, 60, 70, 74, 79, 101, 103, 107, 109, 112, 112, 126, 132, 133] |
| | Requirements stage factors:<br>• Difficulties with defining privacy and security concepts and requirements;<br>• Tension between privacy/security and other technical and system requirements. | [7, 13, 15, 16, 19, 22, 25, 31, 51, 54, 57, 60, 62, 66, 70, 73, 76, 79, 82, 84, 89, 91, 100, 101, 103, 108, 110, 126, 131, 136] |
| | Implementation stage factors:<br>• Difficulties with translating requirements into practice;<br>• Tension between privacy/security and time priorities;<br>• Usability issues of privacy and security tools. | [7, 8, 10–12, 16, 22, 23, 31, 38, 39, 41, 47, 48, 54, 57, 60, 70, 72–74, 79, 91, 94, 100–103, 107, 110, 112, 126, 128, 131, 132, 134, 135] |
| | Review and evaluation stage factors:<br>• Evaluation process and metrics. | [11, 12, 16, 23, 26, 31, 54, 60, 66, 73, 79, 92, 103, 105, 110, 114, 126, 132, 134] |

(Continued)

Table 2. Continued

| Categories | Sub-Categories | References |
|---|---|---|
| *Product-related factors* are pertinent to the type of software product, its target audience, and its economic potential. | Relevance/importance of privacy/security for the product | [7, 11, 13, 22, 26, 31, 57, 60, 93, 100, 132] |
| | Tensions between privacy/security and business priorities | [11, 16, 26, 31, 54, 57, 74, 78, 101, 112, 124] |
| | Competitive advantage | [13, 26, 31, 51, 54, 60, 61, 67, 70, 74, 100, 105, 110, 112] |
| *Personal factors* include developers' personal characteristics and backgrounds. | Position and role:<br>• Hierarchical position;<br>• Perceived personal responsibility;<br>• Autonomy and control. | [7, 11, 12, 16, 20, 26, 30, 31, 51, 57, 70, 74, 76, 79, 98, 100, 104, 108–110, 112, 115, 132] |
| | Privacy expertise and knowledge | [2, 7, 11, 13, 14, 16, 17, 22, 33, 34, 40, 56, 58–60, 70, 75, 79, 82, 100, 101, 103, 106, 112, 123] |
| | Instrumental privacy/security attitudes | [4, 7, 8, 16, 25, 59, 70, 74, 79, 100, 103, 112, 130] |
| | Experimental privacy/security attitudes | [7, 16, 17, 59, 112] |
| | Prior experiences with privacy/security violations in the software development context | [7, 8, 132] |
| | Personality traits | [45, 56, 58, 59, 71, 112, 129] |

## SEARCH QUERIES

*Scopus.* ABS ("privacy by design") AND ABS (software) AND ABS (developer OR development) AND (LIMIT-TO (PUBSTAGE, "final")) AND (LIMIT-TO (LANGUAGE, "English"))—18 May, resulted in 42 documents

ABS ("security by design") AND ABS (software) AND ABS (developer OR development) AND (LIMIT-TO (PUBSTAGE, "final")) AND (LIMIT-TO (LANGUAGE, "English"))—18 May, resulted in 16 documents

ABS(security OR privacy) AND ABS("by design") AND ABS(software) AND ABS(developer OR development) AND (LIMIT-TO (PUBSTAGE, "final")) AND ( LIMIT-TO ( LANGUAGE, "English"))—18 May, resulted in 84 documents

*IEEE Xplore.* ((((("Abstract": Privacy OR security) AND "Abstract": Development OR developer) AND "Abstract": Software) AND "Abstract": "by design"))—18 May, resulted in 43 documents.

## REFERENCES

[1] Jenny Abramov, Omer Anson, Michal Dahan, Peretz Shoval, and Arnon Sturm. 2012. A methodology for integrating access control policies within database development. *Computers & Security* 31, 3 (2012), 299–314.

[2] Yasemin Acar, Michael Backes, Sascha Fahl, Doowon Kim, Michelle L. Mazurek, and Christian Stransky. 2016. You get where you're looking for: The impact of information sources on code security. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP'16)*. IEEE, Los Alamitos, CA, 289–305.

[3] Yasemin Acar, Christian Stransky, Dominik Wermke, Charles Weir, Michelle L. Mazurek, and Sascha Fahl. 2017. Developers need support, too: A survey of security advice for software developers. In *Proceedings of the 2017 IEEE Cybersecurity Development Conference (SecDev 17)*. IEEE, Los Alamitos, CA, 22–26.

[4] Icek Ajzen. 1991. The theory of planned behavior. *Organizational Behavior and Human Decision Processes* 50, 2 (1991), 179–211.

[5] Abdulaziz Alkussayer and William H. Allen. 2009. The ISDF framework: Integrating security patterns and best practices. In *Advances in Information Security and Its Application*, Jong Hyuk Park, Justin Zhan, Changhoon Lee, Guilin Wang, Tai-Hoon Kim, and Sang-Soo Yeo (Eds.). Springer, Berlin, Germany, 17–28.

[6] Hala Assal and Sonia Chiasson. 2018. Security in the software development lifecycle. In *Proceedings of the 14th Symposium on Usable Privacy and Security (SOUPS'18)*. 281–296.

[7] Hala Assal and Sonia Chiasson. 2019. "Think secure from the beginning": A survey with software developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[8] Oshrat Ayalon, Eran Toch, Irit Hadar, and Michael Birnhack. 2017. How developers make design decisions about users' privacy: The place of professional communities and organizational climate. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, New York, NY, 135–138.

[9] Maria Teresa Baldassarre, Vita Santa Barletta, Danilo Caivano, and Michele Scalera. 2019. Privacy oriented software development. In *Proceedings of the International Conference on the Quality of Information and Communications Technology*. 18–32.

[10] Rebecca Balebako and Lorrie Cranor. 2014. Improving app privacy: Nudging app developers to protect user privacy. *IEEE Security & Privacy* 12, 4 (2014), 55–58.

[11] Rebecca Balebako, Abigail Marsh, Jialiu Lin, Jason I Hong, and Lorrie Faith Cranor. 2014. The privacy and security behaviors of smartphone app developers. In *Proceedings of the Workshop on Usable Security (USEC'14)*.

[12] Kenneth A. Bamberger and Deirdre K. Mulligan. 2015. Privacy on the ground: Driving corporate behavior in the United States and Europe (chapter 1). In *Privacy on the Ground: Driving Corporate Behavior in the United States and Europe*. MIT Press, Cambridge, MA, 1–20.

[13] Pedro Barbosa, Andrey Brito, and Hyggo Almeida. 2020. Privacy by evidence: A methodology to develop privacy-friendly software applications. *Information Sciences* 527 (2020), 294–310.

[14] Steffen Bartsch. 2011. Practitioners' perspectives on security in agile development. In *Proceedings of the 2011 6th International Conference on Availability, Reliability, and Security*. IEEE, Los Alamitos, CA, 479–484.

[15] Ingolf Becker, Simon Parkin, and M. Angela Sasse. 2017. Finding security champions in blends of organisational culture. In *Proceedings of the 2nd European Workshop on Usable Security (EuroUSEC'17)*.

[16] Kathrin Bednar, Sarah Spiekermann, and Marc Langheinrich. 2019. Engineering privacy by design: Are engineers ready to live up to the challenge? *Information Society* 35, 3 (2019), 122–142.

[17] Odette Beris, Adam Beautement, and M. Angela Sasse. 2015. Employee rule breakers, excuse makers and security champions: Mapping the risk perceptions and emotions that drive security behaviors. In *Proceedings of the 2015 New Security Paradigms Workshop*. ACM, New York, NY, 73–84. https://doi.org/10.1145/2841113.2841119

[18] Parnika Bhat and Kamlesh Dutta. 2019. A survey on various threats and current state of security in Android platform. *ACM Computing Surveys* 52, 1 (2019), 1–35.

[19] David Botta, Rodrigo Werlinger, André Gagné, Konstantin Beznosov, Lee Iverson, Sidney Fels, and Brian Fisher. 2007. Towards understanding IT security professionals and their tools. In *Proceedings of the 3rd Symposium on Usable Privacy and Security*. ACM, New York, NY, 100–111.

[20] D. Byers and N. Shahmehri. 2007. Design of a process for software security. In *Proceedings of the 2nd International Conference on Availability, Reliability, and Security (ARES'07)*. IEEE, Los Alamitos, CA, 301–309. https://doi.org/10.1109/ARES.2007.67

[21] Jennifer A. Byrne. 2016. Improving the peer review of narrative literature reviews. *Research Integrity and Peer Review* 1, 1 (2016), 12.

[22] Jean Camp, Ryan Henry, Tadayoshi Kohno, Shrirang Mare, Steve Myers, Shwetak N. Patel, and Joshua Streiff. 2020. Toward a secure internet of things: Directions for research. *IEEE Security & Privacy* 18, 4 (2020), 28–37.

[23] Valentina Casola, Alessandra De Benedictis, Massimiliano Rak, and Umberto Villano. 2020. A novel security-by-design methodology: Modeling and assessing security by SLAs with a quantitative approach. *Journal of Systems and Software* 163 (2020), 110537.

[24] Ann Cavoukian. 2009. Privacy by Design: The 7 Foundational Principles. Retrieved April 13, 2023 from https://iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf.

[25] Golriz Chehrazi, Irina Heimbach, and Oliver Hinz. 2016. The impact of security by design on the success of open source software. In *Proceedings of the 2016 European Conference on Information Systems (ECIS'16)*.

[26] Wenhong Chen, Gejun Huang, Joshua Miller, Kye-Hyoung Lee, Daniel Mauro, Bryan Stephens, and Xiaoqian Li. 2018. "As we grow, it will become a priority": American mobile start-ups' privacy practices. *American Behavioral Scientist* 62, 10 (2018), 1338–1355.

[27] Partha Das Chowdhury, Joseph Hallett, Nikhil Patnaik, Mohammad Tahaei, and Awais Rashid. 2021. Developers are neither enemies nor users: They are collaborators. In *Proceedings of the 2021 IEEE Secure Development Conference (SecDev'21)*. IEEE, Los Alamitos, CA, 47–55.

[28] Michael Colesky and Julio C. Caiza. 2018. A system of privacy patterns for informing users: Creating a pattern system. In *Proceedings of the 23rd European Conference on Pattern Languages of Programs (EuroPLoP'18)*. ACM, New York, NY, Article 16, 11 pages. https://doi.org/10.1145/3282308.3282325

[29] John Colley. 2010. Why secure coding is not enough: Professionals' perspective. In *ISSE 2009 Securing Electronic Business Processes*. Springer, 302–311.

[30] Lena Connolly, Michael Lang, and J. Doug Tygar. 2015. Investigation of employee security behaviour: A grounded theory approach. In *Proceedings of the IFIP International Information Security and Privacy Conference*. 283–296.

[31] Computing Community Consortium. 2015. *Privacy by Design–Engineering Privacy*. Workshop 3 Report. Computing Community Consortium. https://cra.org/ccc/wp-content/uploads/sites/2/2015/12/PbD3-Workshop-Report-v2.pdf.

[32] Barnaby Craggs. 2019. A just culture is fundamental: Extending security ergonomics by design. In *Proceedings of the 2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS'19)*. IEEE, Los Alamitos, CA, 46–49.

[33] Lorrie Faith Cranor. 2015. *Wanted: Privacy Engineers*. Technical Report. IAPP. https://iapp.org/news/a/wanted-privacy-engineers/.

[34] Lorrie Faith Cranor and Norman Sadeh. 2013. A shortage of privacy engineers. *IEEE Security & Privacy* 11, 2 (2013), 77–79.

[35] Philip Davies. 2000. The relevance of systematic reviews to educational policy and practice. *Oxford Review of Education* 26, 3-4 (2000), 365–378.

[36] Vasiliki Diamantopoulou, Nikolaos Argyropoulos, Christos Kalloniatis, and Stefanos Gritzalis. 2017. Supporting the design of privacy-aware business processes via privacy process patterns. In *Proceedings of the 2017 11th International Conference on Research Challenges in Information Science (RCIS'17)*. IEEE, Los Alamitos, CA, 187–198.

[37] Edna Dias Canedo, Angelica Toffano Seidel Calazans, Eloisa Toffano Seidel Masson, Pedro Henrique Teixeira Costa, and Fernanda Lima. 2020. Perceptions of ICT practitioners regarding software privacy. *Entropy* 22, 4 (2020), 429.

[38] Laurence Diver and Burkhard Schafer. 2017. Opening the black box: Petri nets and privacy by design. *International Review of Law, Computers & Technology* 31, 1 (2017), 68–90.

[39] Manuel Egele, David Brumley, Yanick Fratantonio, and Christopher Kruegel. 2013. An empirical study of cryptographic misuse in Android applications. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*. 73–84.

[40] Sascha Fahl, Yasemin Acar, Henning Perl, and Matthew Smith. 2014. Why eve and mallory (also) love webmasters: A study on the root causes of SSL misconfigurations. In *Proceedings of the 9th ACM Symposium on Information, Computer, and Communications Security*. 507–512.

[41] Sascha Fahl, Marian Harbach, Henning Perl, Markus Koetter, and Matthew Smith. 2013. Rethinking SSL development in an appified world. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*. 49–60.

[42] Felix Fischer, Konstantin Böttinger, Huang Xiao, Christian Stransky, Yasemin Acar, Michael Backes, and Sascha Fahl. 2017. Stack overflow considered harmful? The impact of copy&paste on Android application security. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP'17)*. IEEE, Los Alamitos, CA, 121–136.

[43] Pieter Frijns, Robert Bierwolf, and Tom Zijderhand. 2018. Reframing security in contemporary software development life cycle. In *Proceedings of the 2018 IEEE International Conference on Technology Management, Operations, and Decisions (ICTMOD'18)*. IEEE, Los Alamitos, CA, 230–236.

[44] Kelsey R. Fulton, Anna Chan, Daniel Votipka, Michael Hicks, and Michelle L. Mazurek. 2021. Benefits and drawbacks of adopting a secure programming language: Rust as a case study. In *Proceedings of the 17th Symposium on Usable Privacy and Security (SOUPS'21)*. 597–616.

[45] Trevor Gabriel and Steven Furnell. 2011. Selecting security champions. *Computer Fraud & Security* 2011, 8 (2011), 8–12.

[46] Vahid Garousi, Ayça Tarhan, Dietmar Pfahl, Ahmet Coşkunçay, and Onur Demirörs. 2019. Correlation of critical success factors with success of software projects: An empirical investigation. *Software Quality Journal* 27, 1 (2019), 429–493.

[47] David Geer. 2010. Are companies actually using secure development life cycles? *Computer* 43, 6 (2010), 12–16.

[48] Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov. 2012. The most dangerous code in the world: Validating SSL certificates in non-browser software. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. 38–49.

[49] William Bradley Glisson and Ray Welland. 2005. Web development evolution: The assimilation of web engineering security. In *Proceedings of the 3rd Latin American Web Congress (LA-WEB'05)*. IEEE, Los Alamitos, CA, 5.

[50] Bart N. Green, Claire D. Johnson, and Alan Adams. 2006. Writing narrative literature reviews for peer-reviewed journals: Secrets of the trade. *Journal of Chiropractic Medicine* 5, 3 (2006), 101–117.
[51] Daniel Greene and Katie Shilton. 2018. Platform privacies: Governance, collaboration, and the different meanings of "privacy" in iOS and Android development. *New Media & Society* 20, 4 (2018), 1640–1657.
[52] Larry Greenemeier. 2013. NSA efforts to evade encryption technology damaged US cryptography standard. *ACM News*. Retrieved April 13, 2023 from https://cacm.acm.org/news/168046-nsa-efforts-to-evade-encryption-technology-damaged-s-cryptography-standard/fulltext?mobile=false.
[53] Hui Guan, Weiru Chen, Lin Liu, and Hongji Yang. 2011. Environment-driven threats elicitation for web applications. In *Proceedings of the KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. 291–300.
[54] Seda Gurses and Joris Van Hoboken. 2017. Privacy after the agile turn. In *The Cambridge Handbook of Consumer Privacy*, Evan Selinger, Jules Polonetsky, and Omer Tene (Eds.). Cambridge University Press, 579–691.
[55] Marco Gutfleisch, Jan H. Klemmer, Niklas Busch, Yasemin Acar, M. Angela Sasse, Sascha Fahl, Dominik Wermke, Nicolas Huaman, Christian Stransky, and Alexander Krause. 2022. How does usable security (not) end up in software products? Results from a qualitative interview study. In *Proceedings of the 43rd IEEE Symposium on Security and Privacy (S&P'22)*. 22–26.
[56] Eben Haber and Eser Kandogan. 2007. Security administrators: A breed apart. In *Proceedings of the Symposium on Usable Privacy and Security (SOUPS'07)*. 3–6.
[57] Irit Hadar, Tomer Hasson, Oshrat Ayalon, Eran Toch, Michael Birnhack, Sofia Sherman, and Arod Balissa. 2018. Privacy by designers: Software developers' privacy mindset. *Empirical Software Engineering* 23, 1 (2018), 259–289.
[58] Julie M. Haney and Wayne G. Lutters. 2017. Skills and characteristics of successful cybersecurity advocates. In *Proceedings of the 13th Symposium on Usable Privacy and Security (SOUPS'17)*. https://www.usenix.org/conference/soups2017/workshop-program/wsiw2017/haney.
[59] Julie M. Haney and Wayne G. Lutters. 2019. Motivating cybersecurity advocates: Implications for recruitment and retention. In *Proceedings of the 2019 on Computers and People Research Conference (SIGMIS-CPR'19)*. ACM, New York, NY, 109–117. https://doi.org/10.1145/3322385.3322388
[60] Julie M. Haney, Mary Theofanos, Yasemin Acar, and Sandra Spickard Prettyman. 2018. "We make it a big deal in the company": Security mindsets in organizations that develop cryptographic products. In *Proceedings of the 14th Symposium on Usable Privacy and Security (SOUPS'18)*. 357–373. https://www.usenix.org/conference/soups2018/presentation/haney-mindsets.
[61] Woodrow Hartzog. 2018. *Privacy's Blueprint: The Battle to Control the Design of New Technologies*. Harvard University Press.
[62] Kirstie Hawkey, David Botta, Rodrigo Werlinger, Kasia Muldner, Andre Gagne, and Konstantin Beznosov. 2008. Human, organizational, and technological factors of IT security. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, 3639–3644. https://doi.org/10.1145/1358628.1358905
[63] Daniel Hein and Hossein Saiedian. 2009. Secure software engineering: Learning from the past to address future challenges. *Information Security Journal: A Global Perspective* 18, 1 (2009), 8–25.
[64] Tejaswini Herath and H. Raghav Rao. 2009. Encouraging information security behaviors in organizations: Role of penalties, pressures and perceived effectiveness. *Decision Support Systems* 47, 2 (2009), 154–165.
[65] Alex Hern. 2021. WhatsApp loses millions of users after terms update. *The Guardian* 24 (2021).
[66] Jaap-Henk Hoepman. 2014. Privacy design strategies. In *ICT Systems Security and Privacy Protection*, Nora Cuppens-Boulahia, Frédéric Cuppens, Sushil Jajodia, Anas Abou El Kalam, and Thierry Sans (Eds.). Springer, Berlin, Germany, 446–459.
[67] David Hoffman. 2014. Privacy is a business opportunity. *Harvard Business Review* 18 (2014), 2–7.
[68] ISO. 2019. Security Techniques – Extension to ISO/IEC 27001 and ISO/IEC 27002 for Privacy Information Management – Requirements and Guidelines. Retrieved April 13, 2023 from https://www.iso.org/obp/ui/#iso:std:iso-iec:27701:ed-1:v1:en.
[69] Russell L. Jones and Abhinav Rastogi. 2004. Secure coding: Building security into the software development life cycle. *Information Systems Security* 13, 5 (2004), 29–39.
[70] Sri Lakshmi Kanniah and Mohd Naz'ri Mahrin. 2016. A review on factors influencing implementation of secure software development practices. *International Journal of Computer and Systems Engineering* 10, 8 (2016), 3032–3039.
[71] Jennifer J. Kish-Gephart, David A. Harrison, and Linda Klebe Treviño. 2010. Bad apples, bad cases, and bad barrels: Meta-analytic evidence about sources of unethical decisions at work. *Journal of Applied Psychology* 95, 1 (2010), 1.
[72] David Kleidermacher and Mike Wolf. 2008. Using static analysis to improve communications infrastructure. In *Proceedings of the 2008 IEEE/AIAA 27th Digital Avionics Systems Conference*. IEEE, Los Alamitos, CA, 1.
[73] Ralf Kneuper. 2019. Integrating data protection into the software life cycle. In *Proceedings of the International Conference on Product-Focused Software Process Improvement*. 417–432.

[74] Barbara Krumay and Marie Caroline Oetzel. 2011. Security and privacy in companies: State-of-the-art and qualitative analysis. In *Proceedings of the 2011 6th International Conference on Availability, Reliability, and Security*. IEEE, Los Alamitos, CA, 313–320.

[75] Susan Landau. 2014. Educating engineers: Teaching privacy in a world of open doors. *IEEE Security & Privacy* 12, 3 (2014), 66–70.

[76] Marc Langheinrich and Saadi Lahlou. 2003. Troubadour approach to privacy. *Ambient Agoras Report* 15, 1 (2003), 2–29.

[77] Pedro Giovanni Leon, Lorrie Faith Cranor, Aleecia M. McDonald, and Robert McGuire. 2010. Token attempt: The misrepresentation of website privacy policies through the misuse of p3p compact policy tokens. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*. ACM, New York, NY, 93–104.

[78] Ilias Leontiadis, Christos Efstratiou, Marco Picone, and Cecilia Mascolo. 2012. Don't kill my ads! Balancing privacy in an ad-supported mobile application market. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*. 1–6.

[79] Tianshi Li, Yuvraj Agarwal, and Jason I. Hong. 2018. Coconut: An IDE plugin for developing privacy-friendly apps. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–35.

[80] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I. Hong. 2021. How developers talk about personal data and what it means for user privacy: A case study of a developer forum on Reddit. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–28.

[81] Tianshi Li, Kayla Reiman, Yuvraj Agarwal, Lorrie Faith Cranor, and Jason I. Hong. 2022. Understanding challenges for developers to create accurate privacy nutrition labels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–24.

[82] Thomas Loruenser, Henrich C. Pöhls, Leon Sell, and Thomas Laenger. 2018. CryptSDLC: Embedding cryptographic engineering into secure software development lifecycle. In *Proceedings of the 13th International Conference on Availability, Reliability, and Security*. 1–9.

[83] Zulfikar Ahmed Maher, Humaiz Shaikh, Mohammad Shadab Khan, Ammar Arbaaeen, and Asadullah Shah. 2018. Factors affecting secure software development practices among developers—An investigation. In *Proceedings of the 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS'18)*. IEEE, Los Alamitos, CA, 1–6.

[84] Yod-Samuel Martín, Jose M. Del Alamo, and Juan C. Yelmo. 2014. Engineering privacy requirements valuable lessons from another realm. In *Proceedings of the 2014 IEEE 1st International Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE'14)*. IEEE, Los Alamitos, CA, 19–24.

[85] G. McGraw. 2004. Software security. *IEEE Security & Privacy* 2, 2 (March 2004), 80–83.

[86] Laurie McLeod and Stephen G. MacDonell. 2011. Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys* 43, 4 (2011), 1–56.

[87] Susan Michie, Lou Atkins, and Robert West. 2014. *The Behaviour Change Wheel: A Guide to Designing Interventions*. Silverback Publishing. https://behaviourchangewheel.com.

[88] Susan Michie, Maartje M. van Stralen, and Robert West. 2011. The behaviour change wheel: A new method for characterising and designing behaviour change interventions. *Implementation Science* 6, 1 (April 2011), 42. https://doi.org/10.1186/1748-5908-6-42

[89] Miguel Ehécatl Morales-Trujillo and Gabriel Alberto Garcia-Mireles. 2018. Extending ISO/IEC 29110 basic profile with privacy-by-design approach: A case study in the health care sector. In *Proceedings of the 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC'18)*. IEEE, Los Alamitos, CA, 56–64.

[90] Alena Naiakshina, Anastasia Danilova, Eva Gerlitz, Emanuel Von Zezschwitz, and Matthew Smith. 2019. "If you want, I can store the encrypted password": A password-storage field study with freelance developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[91] Nicolás Notario, Alberto Crespo, Yod-Samuel Martín, Jose M. Del Alamo, Daniel Le Métayer, Thibaud Antignac, Antonio Kung, Inga Kroener, and David Wright. 2015. PRIPARE: Integrating privacy best practices into a privacy engineering methodology. In *Proceedings of the 2015 IEEE Security and Privacy Workshops*. IEEE, Los Alamitos, CA, 151–158.

[92] Leysan Nurgalieva, David O'Callaghan, and Gavin Doherty. 2020. Security and privacy of mhealth applications: A scoping review. *IEEE Access* 8 (2020), 104247–104268.

[93] Elin Merethe Oftedal, Lene Foss, and Tatiana Iakovleva. 2019. Responsible for responsibility? A study of digital e-health startups. *Sustainability* 11, 19 (2019), 5433.

[94] T. Okubo, H. Kaiya, and N. Yoshioka. 2012. Mutual refinement of security requirements and architecture using twin peaks model. In *Proceedings of the 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*. IEEE, Los Alamitos, CA, 367–372.

[95] Daniela Seabra Oliveira, Tian Lin, Muhammad Sajidur Rahman, Rad Akefirad, Donovan Ellis, Eliany Perez, Rahul Bobhate, Lois A. DeLong, Justin Cappos, and Yuriy Brun. 2018. API blindspots: Why experienced developers write vulnerable code. In *Proceedings of the 14th Symposium on Usable Privacy and Security (SOUPS'18)*. 315–328.

[96] Hernan Palombo, Armin Ziaie Tabari, Daniel Lende, Jay Ligatti, and Xinming Ou. 2020. An ethnographic understanding of software (in) security and a co-creation model to improve secure software development. In *Proceedings of the 16th Symposium on Usable Privacy and Security (SOUPS'20)*. 205–220.

[97] Guy Paré, Marie-Claude Trudel, Mirou Jaana, and Spyros Kitsiou. 2015. Synthesizing information systems knowledge: A typology of literature reviews. *Information & Management* 52, 2 (2015), 183–199.

[98] Jeffery Payne. 2010. Integrating application security into software development. *IT Professional* 12, 2 (2010), 6–9.

[99] Vijay Raghavan and Xiaoni Zhang. 2009. Building security in during information systems development. *AMCIS 2009 Proceedings* 2009 (2009), 687.

[100] Rivka Ribak. 2019. Translating privacy: Developer cultures in the global world of practice. *Information, Communication & Society* 22, 6 (2019), 838–853.

[101] Ira S. Rubinstein and Nathaniel Good. 2013. Privacy by design: A counterfactual analysis of Google and Facebook privacy incidents. *Berkeley Technology Law Journal* 28 (2013), 1333.

[102] Hanne Rygge and Audun Jøsang. 2018. Threat poker: Solving security and privacy threats in agile software development. In *Proceedings of the Nordic Conference on Secure IT Systems*. 468–483.

[103] Awanthika Senarath and Nalin A. G. Arachchilage. 2018. Why developers cannot embed privacy into software systems? An empirical investigation. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering (EASE'18)*. ACM, New York, NY, 211–216. https://doi.org/10.1145/3210459.3210484

[104] Thomas R. Shaw. 2003. The moral intensity of privacy: An empirical study of webmaster' attitudes. *Journal of Business Ethics* 46, 4 (2003), 301–318.

[105] Katie Shilton and Daniel Greene. 2019. Linking platforms, practices, and developer ethics: Levers for privacy discourse in mobile application development. *Journal of Business Ethics* 155, 1 (2019), 131–146.

[106] Johanneke Siljee. 2015. Privacy transparency patterns. In *Proceedings of the 20th European Conference on Pattern Languages of Programs (EuroPLoP'15)*. ACM, New York, NY, Article 52, 11 pages. https://doi.org/10.1145/2855321.2855374

[107] Laurens Sion, Pierre Dewitte, Dimitri Van Landuyt, Kim Wuyts, Ivo Emanuilov, Peggy Valcke, and Wouter Joosen. 2019. An architectural view for data protection by design. In *Proceedings of the 2019 IEEE International Conference on Software Architecture (ICSA'19)*. IEEE, Los Alamitos, CA, 11–20.

[108] H. Jeff Smith. 1994. *Managing Privacy: Information Technology and Corporate America*. UNC Press Books.

[109] Teodor Sommestad, Jonas Hallberg, Kristoffer Lundholm, and Johan Bengtsson. 2014. Variables influencing information security policy compliance. *Information Management & Computer Security* 22, 1 (2014), 42–75.

[110] Sarah Spiekermann. 2012. The challenges of privacy by design. *Communications of the ACM* 55, 7 (2012), 38–40.

[111] Sarah Spiekermann. 2015. *Ethical IT Innovation: A Value-Based System Design Approach*. CRC Press, Boca Raton, FL.

[112] Sarah Spiekermann, Jana Korunovska, and Marc Langheinrich. 2018. Inside the organization: Why privacy and security engineering is a challenge for engineers. *Proceedings of the IEEE* 107, 3 (2018), 600–615.

[113] Alex Sumner and Xiaohong Yuan. 2019. Mitigating phishing attacks: An overview. In *Proceedings of the 2019 ACM Southeast Conference*. 72–77.

[114] Theeraporn Suphakul and Twittie Senivongse. 2017. Development of privacy design patterns based on privacy principles and UML. In *Proceedings of the 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'17)*. IEEE, Los Alamitos, CA, 369–375.

[115] Ivan Szekely. 2013. What do IT professionals think about surveillance? *Internet and Surveillance: The Challenges of Web 2.0 and Social Media* 16 (2013), 198.

[116] Evelina Tacconelli. 2010. Systematic reviews: CRD's guidance for undertaking reviews in health care. *Lancet Infectious Diseases* 10, 4 (2010), 226.

[117] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. 2021. Privacy champions in software teams: Understanding their motivations,strategies,and challenges. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[118] Mohammad Tahaei, Tianshi Li, and Kami Vaniea. 2022. Understanding privacy-related advice on Stack Overflow. *Proceedings on Privacy Enhancing Technologies* 1 (2022), 18.

[119] Mohammad Tahaei, Kopo M. Ramokapane, Tianshi Li, Jason I. Hong, and Awais Rashid. 2022. Charting app developers' journey through privacy regulation features in ad networks. *Proceedings on Privacy Enhancing Technologies* 1 (2022), 24.

[120] Mohammad Tahaei and Kami Vaniea. 2019. A survey on developer-centred security. In *Proceedings of the 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW'19)*. IEEE, Los Alamitos, CA, 129–138.

[121] Inger Anne Tøndel and Martin Gilje Jaatun. 2020. Towards a conceptual framework for security requirements work in agile software development. *International Journal of Systems and Software Security and Protection* 11, 1 (2020), 33–62.

[122] Anwesh Tuladhar, Daniel Lende, Jay Ligatti, and Xinming Ou. 2021. An analysis of the role of situated learning in starting a security culture in a software company. In *Proceedings of the 17th Symposium on Usable Privacy and Security (SOUPS'21)*. 617–632.

[123] Jaideep Vaidya, Basit Shafiq, David Lorenzi, and Nazia Badar. 2013. Incorporating privacy into the undergraduate curriculum. In *Proceedings of the 2013 Information Security Curriculum Development Conference*. ACM, New York, NY.

[124] Veracode. 2016. *State of Software Security 2016*. Veracode. https://www.veracode.com/sites/default/files/Resources/Reports/state-of-software-security-volume-7-veracode-report.pdf.

[125] Daniel Votipka, Kelsey R. Fulton, James Parker, Matthew Hou, Michelle L. Mazurek, and Michael Hicks. 2020. Understanding security mistakes developers make: Qualitative analysis from build it, break it, fix it. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security'20)*. 109–126.

[126] Ari Ezra Waldman. 2017. Designing without privacy. *Houston Law Review* 55 (2017), 659.

[127] Rachel Walker, Marie Cooke, Amanda Henderson, and Debra K. Creedy. 2011. Characteristics of leadership that influence clinical learning: A narrative review. *Nurse Education Today* 31, 8 (2011), 743–756.

[128] Roman Wirtz and Maritta Heisel. 2019. Managing security risks: Template-based specification of controls. In *Proceedings of the 24th European Conference on Pattern Languages of Programs*. 1–13.

[129] Jim Witschey, Shundan Xiao, and Emerson Murphy-Hill. 2014. Technical and personal factors influencing developers' adoption of security tools. In *Proceedings of the 2014 ACM Workshop on Security Information Workers*. ACM, New York, NY, 23–26.

[130] Jim Witschey, Olga Zielinska, Allaire Welk, Emerson Murphy-Hill, Chris Mayhorn, and Thomas Zimmermann. 2015. Quantifying developers' adoption of security tools. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. 260–271.

[131] Sven Wohlgemuth. 2014. Adaptive user-centered security. In *Proceedings of the International Conference on Availability, Reliability, and Security*. 94–109.

[132] Shundan Xiao, Jim Witschey, and Emerson Murphy-Hill. 2014. Social influences on secure development tool adoption: Why security tools spread. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, New York, NY, 1095–1106.

[133] Jing Xie, Heather Lipford, and Bei-Tseng Chu. 2012. Evaluating interactive support for secure programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2707–2716.

[134] Jing Xie, Heather Richter Lipford, and Bill Chu. 2011. Why do programmers make security errors? In *Proceedings of the 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'11)*. IEEE, Los Alamitos, CA, 161–164.

[135] Zeineb Zhioua, Yves Roudier, and Rabea Ameur-Boulifa. 2017. Formal specification of security guidelines for program certification. In *Proceedings of the 2017 International Symposium on Theoretical Aspects of Software Engineering (TASE'17)*. IEEE, Los Alamitos, CA, 1–8.

[136] Tanveer A. Zia and Aftab Rizvi. 2011. Source code embedded (SCEM) security framework. In *Proceedings of the 9th Australian Information Security Management Conference*. 262–269.

[137] Albin Zuccato, Nils Daniels, and Cheeverat Jampathom. 2011. Service security requirement profiles for telecom: How software engineers may tackle security. In *Proceedings of the 2011 6th International Conference on Availability, Reliability, and Security*. IEEE, Los Alamitos, CA, 521–526.

Received 18 October 2021; revised 21 January 2023; accepted 23 March 2023