# WiP: Factors Affecting the Implementation of Privacy and Security Practices in Software Development: a Narrative Review

Leysan Nurgalieva
leysan.nurgalieva@tcd.ie
Trinity College Dublin
Dublin, Ireland

Alisa Frik
afrik@berkeley.edu
University of California Berkeley
Berkeley, USA

Gavin Doherty
gavin.doherty@scss.tcd.ie
Trinity College Dublin
Dublin, Ireland

## Abstract

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 last two decades. In order 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, in order to increase the adoption of privacy and security practices in software development.

## Keywords

privacy, security, design, software teams

**ACM Reference Format:**
Leysan Nurgalieva, Alisa Frik, and Gavin Doherty. 2021. WiP: Factors Affecting the Implementation of Privacy and Security Practices in Software Development: a Narrative Review. In *HoTSoS'21: 8th Annual Hot Topics in the Science of Security (HoTSoS) Symposium, April 13–15, 2021.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/1122445.1122456

## 1 Introduction and related work

A big part of today's digital economy relies on users' personal information. Collection of large amounts of user data introduces a variety of privacy and security risks. While some of those attacks (such as social engineering) target individual users, most threat models exploit system vulnerabilities. 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 of Spiekermann et al. showed that 36% of the engineers surveyed rarely or never incorporate privacy mechanisms into the systems that they build, even though most of them think that privacy and security engineering is useful, valuable and important [99].

Prior research has identified a variety of reasons why 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, and others believe that organizational structures and software development processes hinder the adoption of privacy and security practices. However, the findings are fragmented. While some reasons are repeatedly shown to have an impact, other findings are contradictory or yield mixed results. Without a clear understanding of the barriers and challenges, the 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 [40; 76], but not the success of implementing privacy and security practices specifically. Some studies have explored factors related to either only privacy [15; 108] or only security [62; 105], 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 [73; 99]. 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 an in-progress narrative literature review of research that discusses the factors that affect the implementation of privacy and security practices in software development. Through a systematic 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 preliminary findings categorize the factors into five main groups: environmental, organizational, product-related, development process-related, and individual factors. 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.


*HoTSoS'21, April 13-15, 2021,*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

## 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 [32; 44] and offer the flexibility to deal with "evolving knowledge and concepts" [19, p.2], such as the topic of this research. Following the literature review typology by Paré et al., 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" [84, 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. [109]. The framework itself relies on the "general framework for narrative synthesis" described by the Centre for Reviews and Dissemination [102]. Table 1 in appendix A describes the main stages of the study protocol for selecting the most relevant documents related to the research topic. The process included four stages: initial search strategy, building of the initial model, systematic search, and model refinement.

During the initial search, 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. All references identified as relevant were saved and organized in the shared repository, where we coded the publication year and venue, authors, abstract, methodology, as well as major contributions in relation to security/privacy factors. Specifically, we collected the empirical evidence and theoretical predictions about what aspects hinder or promote the implementation of privacy and security considerations and practices in software development. The subsequent analysis process intends to summarize 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.

After the initial narrative review of 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 and are presented in the Appendix C. We selected publications that corresponded to literature reviews, experimental or quasi-experimental studies, or quantitative or qualitative analysis. The exclusion criteria comprised studies published in books and book chapters, papers whose proposed solution was not applied to software engineering, or which were not written in English.

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

## 3 Results

The initial search strategy (Table 1, Step 2) resulted in 99 papers across the 3 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, grey 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 includes 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 didn't overlap but appeared transferable, i.e. while they were observed in one domain (e.g. security), we hypothesize a similar effect in another domain (e.g. privacy) despite the lack of empirical evidence. Therefore, 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 will 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 (such as HIPAA, COPPA, FERPA) protecting consumer privacy and security. These regulations are often unclear and confusing, making it difficult to comply with them [11; 28; 87; 95], or lagging behind the rapid evolution of technology [24], and quickly becoming outdated [20; 24]. Some even raised concerns about governmental sovereignty over corporations in regulating privacy that can be entangled with national interests, for example, when governments force companies to provide them with access to user data through so-called "backdoors" [15]. 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 [12; 28].

**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 a more advisory role, suggesting the best but optional practices, while 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 [20], and provide a certain degree of data privacy by imposing privacy requirements [45]. However, these guidelines can be seen as inefficient, as different platforms may not be aligned and promote diverging or conflicting values [87; 92].

**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 [e.g., 60; 72] offer privacy and security guidance to software companies.

*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 [15; 50; 53]. Prior works agree that different perceptions of privacy or different needs based on individual preferences can lead to diverse types of concerns about privacy [12] and various expectations about usability, security, and privacy [20]. 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 [11].

*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 towards 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 [98]. On the other hand, 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 [53]. Such reputation risks encourage companies to pay more attention to security [6].

## 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 fund-raising and growth—the existential needs of a new business—and as their products mature, they may start giving more consideration to privacy matters [24; 87]. Expansion to international markets requires compliance with international privacy regulations [87], increasing the relevance of the environmental factors discussed earlier (§3.1.1).

*3.2.2 Financial and human resources.* Lack of resources is detrimental to the adoption of privacy and security practices [11; 95]. 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 [6]. Some companies prefer to have an expert specialized in security than to try to educate the whole team about it [114].

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 [6; 13; 41; 47; 62; 83; 119]. For instance, introducing security tools can take a substantial cut of a company budget [62; 114] but also result in indirect costs such as developers' time [10; 114]. Professional security training [13; 47; 62; 83; 119] 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 [53; 66]. Moreover, security risks are often underestimated in relation to the investment required to protect against them [99].

*3.2.3 Privacy and security culture.* Privacy and security culture represent shared perceptions, beliefs and social norms surrounding privacy and security [50; 114], the commitment to address concerns and promote a privacy and security mindset [53]. Privacy and security culture plays an informal role in affecting organizational privacy conduct [7], encouraging and supporting security practices [6; 53; 62; 114], development process [62] and developers' choices regarding security [53]. As engineers do not make independent decisions about system design and their work is situated in certain context [15], 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 [99]. Inefficient organizational norms and practices can put developers under the impression that privacy is not an important value in the organization, with negative consequences [50].

*3.2.4 Management support.* Top management privacy and security awareness and support have a strong influence on the implementation of security and privacy practices [41; 57; 62; 66]. Prior research recognizes management's responsibilities in supporting security and privacy culture both at the top level [43; 56; 61; 62] and through internal team supervision [50], by providing adequate resources for security implementation and communicating their expectations clearly [43; 57; 62], or mediating the communications between various interest groups to resolve related conflicts [56]. The lack of understanding of the security practices and their importance in the development process [62] may result in deferring security [6].

*3.2.5 Organizational incentives.* Providing developers with incentives (rewards and sanctions) can impact on their privacy and security practices [62]. Rewards can include monetary incentives [6; 52], feedback and empowerment [20; 50; 52; 103], and recognizing the value of employee work [52; 103]. The lack of incentives can encourage developers to prioritize functionality over security/privacy [20]. However, encouraging developers' intrinsic motivation has been recognized as a more efficient strategy than extrinsic, especially financial, rewards [6; 52; 96; 103].

As sanctions, developers can be penalized for failures to comply with security standards [20; 57; 96]. The certainty of detection has a stronger influence on security behaviours than the severity of penalty [57]. The combination of preventive methods for privacy protection and punitive mechanisms (such as reporting violations to authorities) can act as an efficient strategy to discourage developers from risky behaviours [12]. Instead of introducing sanctions, some researchers suggest companies should encourage developers to report errors and ensure fair investigation [29].

*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, etc. 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 [7; 86; 114]. Privacy/security training is considered valuable not only for the developers [20; 62; 86] and security advocates [52], but for all stakeholders involved in the software development process [62], as they ensure the support of security initiatives as an integral part of the organization [27]. Yet, security training rarely teaches developers to use security tools [53; 62; 85; 114], and ignores "soft skills," e.g., communication, collaboration, presentation and writing [51].

**Peer support.** User studies with developers identify peer support as a key resource in judging the ethics of their decisions about privacy [99], and encouraging them to discover new security tools [90], and adopt security best practices [14; 20; 114]. Developers seek peer advice from privacy/security specialists, current and former colleagues, friends, and other developers, e.g. from forums, meetups or work-related groups [10]. 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" [53], gradually shifting the privacy/security culture in a positive direction [16; 104], and even taking part in the development of effective organizational security policies for employees [14]. 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 and code sharing websites, e.g., Stack Overflow or GitHub [2; 10; 69; 71; 114] provide developers with technical support and privacy- and security-related knowledge, which might not be available within the company [10], and presented in a more comprehensible and less formal fashion than official documentation [71], such as code examples or examples of how an API works in a particular context (in contrast to a general API documentation) [71]. Despite their usefulness, relying on these resources, even when high-scoring answers are provided by the highly-ranked peers [114], can lead to less secure solutions [2], proliferating vulnerabilities [20], and ignoring the rationale behind the provided recommendations [69; 71].

**Media and other resources.** Mass and social media, and blogs are increasingly used as the channels of privacy and security knowledge dissemination[51; 114]. The content of such channels can be more engaging than formal documentation due to the use of images, metaphors, or pop culture references [51]. 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 [92].

*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 [15] or business departments [11; 38; 43; 48]. However, such collaborations can face challenges, as lawyers and developers often "don't speak each other's language," i.e. don't share the vocabulary and conceptual frameworks of privacy [15]. 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 [11]. In contrast, privacy professionals in "engineer-only design teams" might not have an opportunity to raise and address privacy issues during the design process [108]. Bureaucratic barriers can further hinder the institutionalization and spread of privacy norms within the organization [87; 108]. 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 [11; 99].

**Team diversity.** The lack of demographic and background diversity limits engineers' perspectives and increases the likelihood of discriminatory implicit biases around privacy and security [87; 108]. Increasing team diversity [51; 62; 108], involving security experts and employees responsible for facilitating cross-departmental connections could narrow the proficiency gaps [51], and help establish trust and shared knowledge among team members [108], leading to more successful self-management of teams [38] and a greater sense of belonging and collaboration [52].

## 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 (such as internal organizational documentation), or at a specific stage of the software development life cycle—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 [6; 62; 90; 96] is necessary for the implementation of secure software development [8; 29; 62] and privacy engineering [88], especially in the companies with privacy as a core value or companies operating in privacy-sensitive domains, such as healthcare and finance [50] (see §3.4.1). While 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 [53; 72; 107]. As a result, developers might lack suitable resources [20; 114]. Unlike security tools, privacy tools to assist software development are more scarce [10; 103], or address privacy through security mechanisms, such as secure data sharing [28].

**Awareness.** While 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 [66]. While some studies report high awareness about internal procedures and policies among developers [6; 50; 99], other studies suggest that developers are often unaware of privacy recommendations [24], privacy threat models, mitigation strategies, less privacy invasive coding alternatives [71], and privacy specific tools and checklists [10; 108]. Developers also have little knowledge regarding privacy and security regulations [10], security tools [6] or secure development lifecycles [41].

**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 [14] and that the guidelines and documentation are comprehensive and easy to understand [71]. Developers are less likely to implement recommendations when they doubt their effectiveness and usefulness [41; 66]. For instance, some developers find privacy recommendations provided by the Federal Trade Commission (FTC) outdated, irrelevant, too generic, and, therefore, not useful [24]. 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 [94]. Developers also think that certain privacy mechanisms can be easily broken, overridden, overruled, or de-anonymized [15], while the guidelines for implementing such mechanisms are complex and too theoretical to be used in practice [90]. Some third-party privacy tools even raise concerns, as they might collect information that developers are unaware of [10]. Security code analysis tools might be considered not very useful due to their complexity [6] or due to time resources they require to implement [114]. Academic resources on security might become outdated as well, and are often perceived as distant from real-world challenges [53]. The value of security certification may be doubted due to its costs exceeding the perceived value [66]. 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's voided following every software update [53]. Some developers don't trust existing standards, for instance, due to evidence of government intentions to purposefully weaken cryptographic protections [46].

#### 3.3.2 Requirements stage.

Software entrepreneurs tend to underestimate the role of privacy at the initial stages of business development [24], despite prior research agreeing that it is important [28]. Developers generally more easily agree that they should consider *security* from the earliest planning phases, than privacy [6; 26; 75]. Auditing the security of the code only before the code integration or its release can pose significant security risks [20; 114]. 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.** While development teams are usually familiar with the concept of software security requirements, the concept of privacy in software development is considered to be rather abstract and vague [12; 15; 20; 68; 90; 95; 97; 99] and context-dependent [15] thus difficult to implement. Since, conceptually and methodologically, privacy is often confounded with security [97], developers sometimes use the data security vocabulary to approach privacy, which limits their consideration of privacy [50], or even sacrifice privacy for security [6; 28]. 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 [12; 28; 79; 80; 97] and security requirements [62].

**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 [65]. Security [6; 28; 72] and privacy can interfere with other requirements, such as functional requirements, integrity requirements, performance, and usability [12; 15; 20; 53; 71; 80; 90; 95; 108]. For instance, the collection and use of end user data for the optimization of services and design might compromise data protection agreements [48], a nuanced user authentication process (such as two-factor authentication) requires additional user effort [20; 45], and developers often find it hard to obtain a meaningful informed user consent using existing mechanisms [10; 15; 48; 65; 100]. On the other hand, ignoring the privacy needs of users can negatively impact their trust and loyalty to a product or a service [28]. Thus, developers face the challenge of balancing security and usability [14; 17; 55]. While 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 [28], the benefits of such user-centered approaches are undeniable [28; 53; 87].

Complex requirements engineering 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 [71] violating data minimization principles [58]. To prevent privacy risks, previous studies recommend incorporation of privacy considerations into the definition of software requirements and specifications [15], which might require improving general software requirements that are often not sufficiently maintained and managed [50].

#### 3.3.3 *Implementation stage.*

Even with privacy and security requirements in place, developers might not satisfy them during the implementation stage, because it is not easy to operationalize them, 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 [15; 28; 34; 65; 87; 90; 117] and security [20; 53; 116] requirements into specific software development processes. Partially it is related to the underlying difficulties with defining privacy concepts (see §3.3.2) and lack of knowledge (see §3.5.2), as well as with technical challenges, such as identification of sensitive information [117], technical complexity of the system (e.g. in a cloud environment) [21], and technical implementation of data anonymization [28; 65], data minimization [90], and encryption [65; 117], especially if organizations fail to provide developers with the methods and resources necessary for supporting the implementation of privacy and security requirements [97]. Developers might also direct their attention to the formal procedures and fail to implement a distributed privacy architecture [11] or implement privacy methods in isolation, via different stakeholders that have different levels of knowledge of the system [94]. Certain development approaches, such as Agile and DevOps, might require ad-hoc solutions to address security requirements [21; 28], due to the privacy risks posed by the modularity of these approaches and concentration of user data "in the hands of specialized service providers" [48]. To address it, similarly to waterfall development methodologies [89], organizations might include privacy and security practices in every phase of the development process [62; 65].

**Tension with time priorities.** Limited time, especially in the conditions of time-to-market competition pressure [20; 53], can become an impediment for data protection implementation [10; 99; 107]. Privacy is usually not a priority task developers ready to allocate time and resources to [9; 97; 108]. Some engineers believe that implementing privacy features can slow down the development process [15; 66]. Similarly, security is pushed down the priority list in the conditions of tight deadlines in which most software companies have to operate [41; 62; 64; 83; 99; 114; 116]. While keeping the overall development time within adequate limits is important [62], 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 [48].

**Usability issues of privacy/security tools**. Usability of privacy and security tools is important [62; 113]. Issues with usability reduce the adoption of such tools by developers [6; 71; 110]. Poor default configurations in tools and libraries, confusing security APIs, and insufficient documentation lead to errors in their usage [20; 35; 37; 42; 65]. The lack of interoperability of cryptographic libraries on multiple platforms also impedes collaboration between teams and oversight of a security architect [53].

#### 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 [65]. 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 [11; 15]. While app marketplaces provide certain assurance seals based on the app review process [92] and automated compliance checks are used to verify the compliance with privacy regulations [15], clear and objective criteria and metrics for assessing success in addressing privacy issues are still largely lacking [28; 90; 103]. As systems often change, introduce and remove features [48], it is challenging to keep the security standards (e.g. for cryptographic products) [53] and privacy assessments up-to-date, creating a need for continuous privacy management and monitoring [97]. The lack of automated tools for privacy and security assessment makes developers rely mainly on their own expertise [21], increasing the amount of time required for assessment, and the probability of human error. On the other hand, manual code reviews may improve developers' understanding of underlying data processes [103]; in conjunction with security tools, manual code reviews lead to the best results [114].

## 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 [6]. For instance, developers working on B2B products [24; 53] or internal applications [114] 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 [87; 114]. 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 [10; 28; 82]), also leads to a higher degree of privacy and security concerns among developers [12; 20; 24; 28; 50]. Conversely, products not collecting personally identifiable information lead developers to demote the importance of privacy and security in product design [10].

#### 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 [50; 66; 88], especially when data-driven business models rely on the monetization of personal information as a source of revenue [48; 99] and in early-stage startups [24]. Time and budget spent on privacy engineering is believed to be better invested in innovation, development and growth [99].

#### 3.4.3 *Competitive advantage.*

Security and privacy-centric features can help organizations to distinguish their products in the marketplace [12; 28; 45; 48; 53; 54; 59; 62; 92; 99]. For instance, in a "crowded" software market such as Android apps, privacy features can be used to differentiate products from competitors [92]. Greater competitiveness can also be achieved by efficient privacy management [12; 24] and providing users with support and transparency regarding their data [45; 87]. Engaging in privacy research further helps companies understand and better address user needs and preferences, consequently improving the product overall [28]. However, some developers fail to recognize the competitive advantages of embedding privacy in products or obtaining privacy and security certification [66], due to the lack of awareness regarding the benefits and risks associated with user data privacy and security practices [97].

## 3.5 Personal characteristics

Developers' personal characteristics and backgrounds include developers' position and role in an organisation, 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 [99]. 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 [11]. Besides, those in managing roles can lack substantive expertise to make privacy decisions [95].

**Perceived personal responsibility** of developers and engineers doesn't always depend on professional position or role [66; 99]. 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 [71]. Specifically, "not my problem" mentality [85], lack of "moral responsibility" and absence of privacy and security requirements among deliverables [15; 68], in a job description [6] or formal responsibilities [50] often lead developers to neglect security and privacy engineering [99]. Absence of personal responsibility can be also caused by the developers' misconception that their mistakes are unlikely to cause security vulnerabilities in the system [62]. Some developers believe that users themselves are supposed to protect their personal data [15; 45; 99; 101].

General lack of a clear distribution of privacy and security roles and responsibilities in organization or team presents another challenge [18; 66; 97]. In some cases such roles are substituted with collective responsibility [15; 24; 68], reducing a more systematic privacy engineering approach to ad-hoc decisions that lack enforcement [28]. On the one hand, in the presence of specialized privacy or security experts, the developers shift the responsibility over to them [10; 71; 87; 99; 114]. 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 [66].

When developers perceive a lack of **autonomy and control** over decisions and implementation of privacy and security [27; 96] features, 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 [71]. The autonomy to act according to personal beliefs may also be overridden by the tendency to comply and conform with organizational decisions [91; 99].

*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 [15; 99]. Important skills for implementing security practices include a wide range of expertise including technical security skills [51; 52], competence in assessing security risks [16], efficiency in applying security tools [20], and the ability to deal with a great degree of technical and organizational complexity [49]. Even though developers might have the skills necessary to implement some security mechanisms, they do not always have security expertise [2; 20], as it often requires knowledge from different fields and requires interdisciplinary collaborations [72]. Compared to security, developers might be less prepared to deal with privacy challenges [87] and may not have appropriate privacy expertise, characterized as the ability to incorporate information privacy mechanisms in practice [15]. Developers find it difficult to make decisions about appropriate levels of privacy and when in the software development process they should incorporate it [90], especially when there are no guidelines about what it means to implement privacy and how to balance it against business priorities [88]. The lack of formal training is particularly evident when it comes to privacy [93] and such discipline is much needed to train privacy experts [12; 30; 31; 67; 106]. The lack of sufficient knowledge and awareness to implement security and/or privacy often results in adoption of unreliable third-party services [10; 71; 99], introduction of security vulnerabilities during the development process [6; 13; 36; 53; 62], misunderstanding of potential privacy threats and corresponding coping strategies [71], and frustration over decision-making about embedding privacy in the development process and making developers rely on their personal opinions rather than objective knowledge [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 [15; 99], which we extend to security attitudes as well. Such attitudes may be shaped by the developers' background, including their multicultural environments [87], and their personal opinions and beliefs in relation to privacy [7; 15; 90]. Some developers perceive privacy practices as relevant and important [15; 99], others as unimportant [90], for example, due to the lack of awareness [66] or motivation to protect privacy unless it's required [71]. Positive instrumental security attitudes are associated with the higher uptake of security tools [112] and act as a strong motivation for implementing security practices [6; 52]. Some developers don't recognize the value of the effort invested in the software security [23]. 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 [99], which illustrates the necessity to consistently motivate and support their confidence in the importance of protecting data security [62].

*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 [15; 16]. Motivation to advocate for software security among developers is also characterized by their interest in the field and self-challenging with security tasks [6; 52]. Engineers working in industry find security engineering less unpleasant than privacy engineering [99].

*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 [7]. Experiencing a security issue can increase developers' awareness of, concern about, and attention to security for a long time [6], or motivate them to use security tools [114].

*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"[63, p.4], is a positive predictor of the adoption of ethical engineering in general, and privacy and security engineering specifically [99]. Inquisitiveness may affect the adoption of security tools [111]. People with pronounced imagination and emotionality, low immoderation [39], high proactiveness and reactiveness [49], and good soft skills [52], such as communication and people skills, context awareness, and service orientation [51], are more likely to become successful security advocates. One study mentioned the religiousness can positively affect developer's ethical decision-making process and consequently, privacy related decisions [99].

## 4 Discussion and conclusions

Prior work provides a wealth of insights about the barriers and enablers to adoption and implementation of privacy and security practices in software development. 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.

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 behaviour change theory [77; 78]. The COM-B model describes a framework, in which Capability, Opportunity, and Motivation are three essential conditions for generating behaviour. 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 (such as engineers, companies' management, policymakers, etc.).

### 4.1 Leveraging Motivation

To increase companies' motivation to take privacy and security seriously, policy-makers and industry associations need to create and enforce regulations that require 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 the regulations (§3.1.1). Penalties for violations of users' privacy and security further impact product-related factors, such as tension between privacy/security and business priorities (§3.4.2), making it more costly for the organizations to ignore privacy and security aspects of the products and services they create, and including potential costs of violations into the profit calculus, thereby moving privacy and security objectives up in the list of business priorities.

The demonstration of evidence that privacy and security of software are enforced can be a valid competitive advantage (§3.4.3) that may attract and retain users. In contrast, violations of privacy and security may repel users and harm companies' reputation (§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 done 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 (§3.1.2), and disseminate the results, to raise the awareness, dispel potential misconceptions about users' 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 users' 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 (§3.4.1). However, practically no software is safe from privacy/security risks. Threat modelling exercises (e.g. using the Security Cards[1]), regular vulnerability discovery activities (e.g., bug bounty, penetration testing, threat analysis, etc.), and less formal activities (e.g. hackathons) can help engineers to correctly assess the relevance of privacy/security issues and vulnerability of the systems.

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 (§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 security team to do the reviews and making them fully responsible for privacy and security aspects of the system [116]. To leverage the instrumental attitudes (§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 (§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 (§3.5.4) by making solving privacy and security issues

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

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 (§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 [53]. By motivating and encouraging such champions, companies can gradually shift and proliferate positive privacy and security culture to the rest of organization [16; 103]. (See more strategies for promoting organizational privacy culture in Tahaei et al. [103].) 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 [6; 52; 96; 103], thus more research on this topic is encouraged.

## 4.2 Leveraging Capability

COM-B model defines capability as the physical and psychological capacity to perform the behavior, e.g. engage in the necessary thought processes, for instance, comprehension or reasoning [77; 78]. To leverage developers' capability to engage in user privacy and security protection it is important to ensure the adequate level of their privacy expertise and knowledge (§3.5.2), for example, through training, peer support, other experts in the company exchanging their knowledge, and other resources (§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 [6; 114] or validation of security code libraries by peers before implementation [29]. While peer support, Q&A websites (e.g. StackOverflow), and other media resources can be more engaging than formal documentation, these sources of information may be less reliable [2].

Requirements engineering tools [118] may enable the identification and prioritization of privacy and security requirements and strategies (§3.3.2). It is also important to provide engineers an appropriate level of authority, autonomy and control (§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 the appropriate tools and libraries, but also make them easy to use (§3.3.3). Similarly, internal organizational documentation and procedures (§3.3.1) and external privacy and security guidelines and documentation should be readily available to the developers, comprehensive, useful, and easy to understand [71]. It can be achieved by providing security reference guidelines that are adapted to non-experts [62], and by providing reputable interactive third-party security implementations and tools, which could free developers from writing complex security code from scratch [53] and help to reduce programming errors [115].

## 4.3 Leveraging Opportunity

It is important to create opportunities for implementing privacy and security in software development, for example, by providing management support (§3.2.4), including security and privacy in the board's agenda [11], clearly communicating to employees their support of security advocates [52], and dedicating to privacy and security sufficient financial and human resources (§3.2.2), and time (§3.3.3), e.g. 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 (§3.2.7) to facilitate communication between teams about privacy and software, increase the diversity of opinions in order 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 [22] into the 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 (§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, quantify the relative impact of the factors, and relationships among them.

There are a number of limitations. Firstly, 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 the factors already 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 includes all 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, judgement is inevitably involved in categorizing the factors derived from the reviewed studies. 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.

## Acknowledgments

This project has received funding in part from the EU Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 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 Programme to Lero – the Irish Software Research Centre (www.lero.ie); in part by a grant from the Center for Long-Term Cybersecurity (CLTC) at U.C. Berkeley, by National Science Foundation grants CNS-1514211 and CNS-1528070, 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.

## References


[1] Abramov, J., Anson, O., Dahan, M., Shoval, P., and Sturm, A. A methodology for integrating access control policies within database development. *computers & security 31*, 3 (2012), 299–314.

[2] Acar, Y., Backes, M., Fahl, S., Kim, D., Mazurek, M. L., and Stransky, C. You get where you're looking for: The impact of information sources on code security. In *2016 IEEE Symposium on Security and Privacy (SP)* (2016), IEEE, IEEE, pp. 289–305.

[3] Acar, Y., Stransky, C., Wermke, D., Weir, C., Mazurek, M. L., and Fahl, S. Developers need support, too: A survey of security advice for software developers. In *2017 IEEE Cybersecurity Development (SecDev)* (2017), IEEE, pp. 22–26.

[4] Ajzen, I. The theory of planned behavior. *Organizational behavior and human decision processes 50*, 2 (1991), 179–211.

[5] Alkussayer, A., and Allen, W. H. The isdf framework: Integrating security patterns and best practices. In *Advances in Information Security and Its Application* (Berlin, Heidelberg, 2009), J. H. Park, J. Zhan, C. Lee, G. Wang, T.-h. Kim, and S.-S. Yeo, Eds., Springer Berlin Heidelberg, pp. 17–28.

[6] Assal, H., and Chiasson, S. "think secure from the beginning" a survey with software developers. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–13.

[7] Ayalon, O., Toch, E., Hadar, I., and Birnhack, M. 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* (2017), ACM New York, NY, USA, pp. 135–138.

[8] Baldassarre, M. T., Santa Barletta, V., Caivano, D., and Scalera, M. Privacy oriented software development. In *International Conference on the Quality of Information and Communications Technology* (2019), Springer, pp. 18–32.

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

[10] Balebako, R., Marsh, A., Lin, J., Hong, J. I., and Cranor, L. F. The privacy and security behaviors of smartphone app developers. In *USEC 2014, Workshop on Usable Security* (2014), Citeseer.

[11] Bamberger, K. A., and Mulligan, D. K. Privacy on the ground: Driving corporate behavior in the united states and europe (chapter 1). *Privacy on the Ground: Driving Corporate Behavior in the US and Europe (MIT 2015)* (2015).

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

[13] Bartsch, S. Practitioners' perspectives on security in agile development. In *2011 Sixth International Conference on Availability, Reliability and Security* (2011), IEEE, pp. 479–484.

[14] Becker, I., Parkin, S., and Sasse, M. A. Finding security champions in blends of organisational culture. *Proc. USEC 11* (2017).

[15] Bednar, K., Spiekermann, S., and Langheinrich, M. Engineering privacy by design: Are engineers ready to live up to the challenge? *The Information Society 35*, 3 (2019), 122–142.

[16] Beris, O., Beautement, A., and Sasse, M. A. 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* (New York, NY, USA, 2015), Association for Computing Machinery, pp. 73–84.

[17] Botta, D., Werlinger, R., Gagné, A., Beznosov, K., Iverson, L., Fels, S., and Fisher, B. Towards understanding it security professionals and their tools. In *Proceedings of the 3rd symposium on Usable privacy and security* (New York, NY, USA, 2007), Association for Computing Machinery, pp. 100–111.

[18] Byers, D., and Shahmehri, N. Design of a process for software security. In *The Second International Conference on Availability, Reliability and Security (ARES'07)* (2007), IEEE, pp. 301–309.

[19] Byrne, J. A. Improving the peer review of narrative literature reviews. *Research integrity and peer review 1*, 1 (2016), 12.

[20] Camp, J., Henry, R., Kohno, T., Mare, S., Myers, S., Patel, S. N., and Streiff, J. Toward a secure internet of things: Directions for research. *IEEE Security & Privacy* (2020).

[21] Casola, V., De Benedictis, A., Rak, M., and Villano, U. A novel security-by-design methodology: Modeling and assessing security by slas with a quantitative approach. *Journal of Systems and Software 163* (2020), 110537.

[22] Cavoukian, A. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada 5* (2009).

[23] Chehrazi, G., Heimbach, I., and Hinz, O. The impact of security by design on the success of open source software.

[24] Chen, W., Huang, G., Miller, J., Lee, K.-H., Mauro, D., Stephens, B., and Li, X. "as we grow, it will become a priority": American mobile start-ups' privacy practices. *American Behavioral Scientist 62*, 10 (2018), 1338–1355.

[25] Colesky, M., and Caiza, J. C. A system of privacy patterns for informing users: Creating a pattern system. In *Proceedings of the 23rd European Conference on Pattern Languages of Programs* (New York, NY, USA, 2018), EuroPLoP '18, Association for Computing Machinery.

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

[27] Connolly, L., Lang, M., and Tygar, J. D. Investigation of employee security behaviour: A grounded theory approach. In *IFIP International Information Security and Privacy Conference* (2015), Springer, pp. 283–296.

[28] Consortium, C. C. *Engineering Privacy*. No. 3 in Privacy by Design. 2015.

[29] Craggs, B. A just culture is fundamental: extending security ergonomics by design. In *2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)* (2019), IEEE, pp. 46–49.

[30] Cranor, L. F. Wanted: Privacy engineers. Tech. rep., April 2015. Accessed: 14 December 2018.

[31] Cranor, L. F., and Sadeh, N. A shortage of privacy engineers. *IEEE Security & Privacy 11*, 2 (2013), 77–79.

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

[33] Diamantopoulou, V., Argyropoulos, N., Kalloniatis, C., and Gritzalis, S. Supporting the design of privacy-aware business processes via privacy process patterns. In *2017 11th International Conference on Research Challenges in Information Science (RCIS)* (2017), IEEE, pp. 187–198.

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

[35] Egele, M., Brumley, D., Fratantonio, Y., and Kruegel, C. An empirical study of cryptographic misuse in android applications. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (2013), pp. 73–84.

[36] Fahl, S., Acar, Y., Perl, H., and Smith, M. 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* (2014), pp. 507–512.

[37] Fahl, S., Harbach, M., Perl, H., Koetter, M., and Smith, M. Rethinking ssl development in an appified world. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (2013), pp. 49–60.

[38] Frijns, P., Bierwolf, R., and Zijderhand, T. Reframing security in contemporary software development life cycle. In *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)* (2018), IEEE, pp. 230–236.

[39] Gabriel, T., and Furnell, S. Selecting security champions. *Computer Fraud & Security 2011*, 8 (2011), 8–12.

[40] Garousi, V., Tarhan, A., Pfahl, D., Coşkunçay, A., and Demirörs, O. Correlation of critical success factors with success of software projects: an empirical investigation. *Software Quality Journal 27*, 1 (2019), 429–493.

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

[42] Georgiev, M., Iyengar, S., Jana, S., Anubhai, R., Boneh, D., and Shmatikov, V. 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* (2012), pp. 38–49.

[43] Glisson, W. B., and Welland, R. Web development evolution: The assimilation of web engineering security. In *Third Latin American Web Congress (LA-WEB'2005)* (2005), IEEE, pp. 5–pp.

[44] Green, B. N., Johnson, C. D., and Adams, A. Writing narrative literature reviews for peer-reviewed journals: secrets of the trade. *Journal of chiropractic medicine 5*, 3 (2006), 101–117.

[45] Greene, D., and Shilton, K. Platform privacies: Governance, collaboration, and the different meanings of "'privacy" in ios and android development. *new media & society 20*, 4 (2018), 1640–1657.

[46] Greenemeier, L. Nsa efforts to evade encryption technology damaged us cryptography standard. *Sep..[Online]. Available: https://www. scientificamerican. com/article/nsa-nistencryption-scandal* (2013).

[47] Guan, H., Chen, W., Liu, L., and Yang, H. Environment-driven threats elicitation for web applications. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications* (2011), Springer, pp. 291–300.

[48] Gurses, S., and Van Hoboken, J. Privacy after the agile turn.

[49] Haber, E., and Kandogan, E. Security administrators: A breed apart. *SOUPS USM* (2007), 3–6.

[50] Hadar, I., Hasson, T., Ayalon, O., Toch, E., Birnhack, M., Sherman, S., and Balissa, A. Privacy by designers: software developers' privacy mindset. *Empirical Software Engineering 23*, 1 (2018), 259–289.

[51] Haney, J. M., and Lutters, W. G. Skills and characteristics of successful cybersecurity advocates. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)* (Santa Clara, CA, July 2017), USENIX Association.

[52] Haney, J. M., and Lutters, W. G. Motivating cybersecurity advocates: Implications for recruitment and retention. In *Proceedings of the 2019 on Computers and People Research Conference* (New York, NY, USA, 2019), SIGMIS-CPR '19, Association for Computing Machinery, p. 109–117.

[53] Haney, J. M., Theofanos, M., Acar, Y., and Prettyman, S. S. "we make it

a big deal in the company": Security mindsets in organizations that develop cryptographic products. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)* (Baltimore, MD, Aug. 2018), USENIX Association, pp. 357–373.

[54] Hartzog, W. *Privacy's blueprint: The battle to control the design of new technologies.* Harvard University Press, 2018.

[55] Hawkey, K., Botta, D., Werlinger, R., Muldner, K., Gagne, A., and Beznosov, K. Human, organizational, and technological factors of it security. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08. Association for Computing Machinery, New York, NY, USA, 2008, p. 3639–3644.

[56] Hein, D., and Saiedian, H. Secure software engineering: Learning from the past to address future challenges. *Information Security Journal: A Global Perspective 18*, 1 (2009), 8–25.

[57] Herath, T., and Rao, H. R. Encouraging information security behaviors in organizations: Role of penalties, pressures and perceived effectiveness. *Decision Support Systems 47*, 2 (2009), 154–165.

[58] Hoepman, J.-H. Privacy design strategies. In *ICT Systems Security and Privacy Protection* (Berlin, Heidelberg, 2014), N. Cuppens-Boulahia, F. Cuppens, S. Jajodia, A. Abou El Kalam, and T. Sans, Eds., Springer Berlin Heidelberg, pp. 446–459.

[59] Hoffman, D. Privacy is a business opportunity. *Harvard Business Review 18* (2014), 2–7.

[60] ISO/IEC 27701:2019(en), y. . . Security techniques — extension to iso/iec 27001 and iso/iec 27002 for privacy information management — requirements and guidelines.

[61] Jones, R. L., and Rastogi, A. Secure coding: building security into the software development life cycle. *Inf. Secur. J. A Glob. Perspect. 13*, 5 (2004), 29–39.

[62] Kanniah, S. L., and Mahrin, M. N. A review on factors influencing implementation of secure software development practices. *International Journal of Computer and Systems Engineering 10*, 8 (2016), 3032–3039.

[63] Kish-Gephart, J. J., Harrison, D. A., and Treviño, L. K. 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.

[64] Kleidermacher, D., and Wolf, M. Using static analysis to improve communications infrastructure. In *2008 IEEE/AIAA 27th Digital Avionics Systems Conference* (2008), IEEE, pp. 1–D.

[65] Kneuper, R. Integrating data protection into the software life cycle. In *International Conference on Product-Focused Software Process Improvement* (2019), Springer, pp. 417–432.

[66] Krumay, B., and Oetzel, M. C. Security and privacy in companies: State-of-the-art and qualitative analysis. In *2011 Sixth International Conference on Availability, Reliability and Security* (2011), IEEE, pp. 313–320.

[67] Landau, S. Educating engineers: Teaching privacy in a world of open doors. *IEEE Security & Privacy 12* (2014).

[68] Langheinrich, M., and Lahlou, S. Troubadour approach to privacy. *Ambient Agoras Report 15*, 1 (2003), 2–29.

[69] Leon, P. G., Cranor, L. F., McDonald, A. M., and McGuire, R. 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* (2010), ACM New York, NY, USA, pp. 93–104.

[70] Leontiadis, I., Efstratiou, C., Picone, M., and Mascolo, C. Don't kill my ads! balancing privacy in an ad-supported mobile application market. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications* (2012), pp. 1–6.

[71] Li, T., Agarwal, Y., and Hong, J. I. 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.

[72] Loruenser, T., Pöhls, H. C., Sell, L., and Laenger, T. Cryptsdlc: Embedding cryptographic engineering into secure software development lifecycle. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (2018), pp. 1–9.

[73] Maher, Z. A., Shaikh, H., Khan, M. S., Arbaaeen, A., and Shah, A. Factors affecting secure software development practices among developers-an investigation. In *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (2018), IEEE, pp. 1–6.

[74] Martín, Y.-S., Del Alamo, J. M., and Yelmo, J. C. Engineering privacy requirements valuable lessons from another realm. In *2014 IEEE 1st International Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE)* (2014), IEEE, pp. 19–24.

[75] McGraw, G. Software security. ieee security privacy 2, 2 (march 2004), 80–83, 2004.

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

[77] Michie, S., Atkins, L., and West, R. *The Behaviour Change Wheel: A Guide To Designing Interventions.* Silverback Publishing, 2014.

[78] Michie, S., van Stralen, M. M., and West, R. The behaviour change wheel: A new method for characterising and designing behaviour change interventions. *Implementation Science 6*, 1 (Apr. 2011), 42.

[79] Morales-Trujillo, M. E., and Garcia-Mireles, G. A. Extending iso/iec 29110 basic profile with privacy-by-design approach: A case study in the health care sector. In *2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC)* (2018), IEEE, pp. 56–64.

[80] Notario, N., Crespo, A., Martín, Y.-S., Del Alamo, J. M., Le Métayer, D., Antignac, T., Kung, A., Kroener, I., and Wright, D. Pripare: integrating privacy best practices into a privacy engineering methodology. In *2015 IEEE Security and Privacy Workshops* (2015), IEEE, pp. 151–158.

[81] Nurgalieva, L., O'Callaghan, D., and Doherty, G. Security and privacy of mhealth applications: A scoping review. *IEEE Access 8* (2020), 104247–104268.

[82] Oftedal, E. M., Foss, L., and Iakovleva, T. Responsible for responsibility? a study of digital e-health startups. *Sustainability 11*, 19 (2019), 5433.

[83] Okubo, T., Kaiya, H., and Yoshioka, N. Mutual refinement of security requirements and architecture using twin peaks model. In *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops* (2012), IEEE, pp. 367–372.

[84] Paré, G., Trudel, M.-C., Jaana, M., and Kitsiou, S. Synthesizing information systems knowledge: A typology of literature reviews. *Information & Management 52*, 2 (2015), 183–199.

[85] Payne, J. Integrating application security into software development. *IT professional 12*, 2 (2010), 6–9.

[86] Raghavan, V., and Zhang, X. Building security in during information systems development. *AMCIS 2009 Proceedings* (2009), 687.

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

[88] Rubinstein, I. S., and Good, N. Privacy by design: A counterfactual analysis of google and facebook privacy incidents. *Berkeley Tech. LJ 28* (2013), 1333.

[89] Rygge, H., and Jøsang, A. Threat poker: solving security and privacy threats in agile software development. In *Nordic Conference on Secure IT Systems* (2018), Springer, pp. 468–483.

[90] Senarath, A., and Arachchilage, N. A. G. Why developers cannot embed privacy into software systems? an empirical investigation. EASE'18, Association for Computing Machinery, p. 211–216.

[91] Shaw, T. R. The moral intensity of privacy: an empirical study of webmaster'attitudes. *Journal of Business Ethics 46*, 4 (2003), 301–318.

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

[93] Siljee, J. Privacy transparency patterns. EuroPLoP '15, Association for Computing Machinery.

[94] Sion, L., Dewitte, P., Van Landuyt, D., Wuyts, K., Emanuilov, I., Valcke, P., and Joosen, W. An architectural view for data protection by design. In *2019 IEEE International Conference on Software Architecture (ICSA)* (2019), IEEE, pp. 11–20.

[95] Smith, H. J. *Managing privacy: Information technology and corporate America.* UNC Press Books, 1994.

[96] Sommestad, T., Hallberg, J., Lundholm, K., and Bengtsson, J. Variables influencing information security policy compliance. *Information Management & Computer Security* (2014).

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

[98] Spiekermann, S. *Ethical IT innovation: A value-based system design approach.* CRC Press, 2015.

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

[100] Suphakul, T., and Senivongse, T. Development of privacy design patterns based on privacy principles and uml. In *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (2017), IEEE, pp. 369–375.

[101] Szekely, I. 10 what do it professionals think about surveillance? *Internet and surveillance: The challenges of Web 2.0 and social media 16* (2013), 198.

[102] Tacconelli, E. Systematic reviews: Crd's guidance for undertaking reviews in health care. *The Lancet Infectious Diseases 10*, 4 (2010), 226.

[103] Tahaei, M., Frik, A., and Vaniea, K. Privacy champions in software teams: Understanding their motivations,strategies,and challenges. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.*

[104] Tahaei, M., Frik, A., and Vaniea, K. Privacy champions in software teams: Understanding their motivations, strategies, and challenges.

[105] Tøndel, I. A., and Jaatun, M. G. Towards a conceptual framework for security requirements work in agile software development. *International Journal of Systems and Software Security and Protection (IJSSSP) 11*, 1 (2020), 33–62.

[106] Vaidya, J., Shafiq, B., Lorenzi, D., and Badar, N. Incorporating privacy into the undergraduate curriculum. In *Proceedings of the Information Security Curriculum Development Conference 2013* (2013), ACM.

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

[108] Waldman, A. E. Designing without privacy. *Hous. L. Rev. 55* (2017), 659.

[109] Walker, R., Cooke, M., Henderson, A., and Creedy, D. K. Characteristics of leadership that influence clinical learning: a narrative review. *Nurse Education Today 31*, 8 (2011), 743–756.
[110] Wirtz, R., and Heisel, M. Managing security risks: template-based specification of controls. In *Proceedings of the 24th European Conference on Pattern Languages of Programs* (2019), pp. 1–13.
[111] Witschey, J., Xiao, S., and Murphy-Hill, E. Technical and personal factors influencing developers' adoption of security tools. In *Proceedings of the 2014 ACM Workshop on Security Information Workers* (2014), ACM New York, NY, USA, pp. 23–26.
[112] Witschey, J., Zielinska, O., Welk, A., Murphy-Hill, E., Mayhorn, C., and Zimmermann, T. Quantifying developers' adoption of security tools. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (2015), pp. 260–271.
[113] Wohlgemuth, S. Adaptive user-centered security. In *International Conference on Availability, Reliability, and Security* (2014), Springer, pp. 94–109.
[114] Xiao, S., Witschey, J., and Murphy-Hill, E. Social influences on secure development tool adoption: why security tools spread. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (2014), ACM New York, NY, USA, pp. 1095–1106.
[115] Xie, J., Lipford, H., and Chu, B.-T. Evaluating interactive support for secure programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), pp. 2707–2716.
[116] Xie, J., Lipford, H. R., and Chu, B. Why do programmers make security errors? In *2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2011), IEEE, pp. 161–164.
[117] Zhioua, Z., Roudier, Y., and Ameur-Boulifa, R. Formal specification of security guidelines for program certification. In *2017 International Symposium on Theoretical Aspects of Software Engineering (TASE)* (2017), IEEE, pp. 1–8.
[118] Zia, T. A., and Rizvi, A. Source code embedded (scem) security framework.
[119] Zuccato, A., Daniels, N., and Jampathom, C. Service security requirement profiles for telecom: how software engineers may tackle security. In *2011 Sixth International Conference on Availability, Reliability and Security* (2011), IEEE, pp. 521–526.

# A Narrative review process

**Table 1: A guiding framework for narrative review**

| Review step | Description |
|---|---|
| 1. Developing a concept | 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 | a) Search strategy was developed, and included the following: defining relevant venues and research groups publishing on the topic, online electronic database search, and secondary search via relevant journal and reference list.<br>b) Search strategy resulted into 99 documents.<br>c) Repetition of ranked papers indicated saturation.<br>d) 54 papers identified as relevant. |
| 3. Categorisation process and development of the initial model | 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 of paper 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 | 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 into 99 documents.<br>c) Duplicates, dissertation, and grey literature (commercial reports, policy statements, or editorial papers) were excluded.<br>d) 47 papers identified as relevant based on their abstract and were read in full.<br>e) 26 papers were included for the validation and refinement of the model. |
| 5. Refining the model | The factors extracted from the systematic search at the 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. |

# B 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. | [11; 12; 15; 20; 24; 28; 45; 60; 72; 87; 92; 95] |
| | Perceived social norms and user expectations about privacy and security | [11; 12; 15; 20; 50; 53] |
| | Competition and reputation | [6; 53; 98] |
| **Organizational factors** are aspects pertinent to the company. | The proliferation of privacy/security knowledge within the organisation:<br>• Privacy/security education and training;<br>• Peer support;<br>• The role of privacy/security champions;<br>• Q&A and code sharing websites;<br>• Media and other resources. | [2; 3; 7; 10; 14; 16; 20; 27; 51–53; 62; 69; 71; 85; 86; 90; 92; 99; 104; 114] |
| | Privacy and security culture:<br>• Organisational security culture;<br>• Organisational privacy culture; | [6; 7; 11; 15; 50; 53; 62; 99; 108; 114] |
| | Organizational maturity | [24; 53; 79; 87; 87] |
| | Financial and human resources | [6; 10; 11; 13; 41; 47; 53; 62; 66; 83; 95; 99; 114; 119] |
| | Management support | [6; 11; 41; 43; 50; 52; 56; 57; 61; 62; 66] |

Continued on next page

**Table 2 – continued from previous page**

| Categories | Sub-categories | References |
|---|---|---|
| | Organizational incentives | [6; 12; 20; 29; 50; 52; 57; 62; 96] |
| | Organizational team structure:<br>• Siloed teams;<br>• Team diversity. | [1; 5; 10; 11; 15; 18; 28; 38; 41; 43; 48; 51; 52; 62; 66; 72; 87; 88; 97; 99; 108; 114; 118] |
| **Process-related factors** are the ones that can have an impact throughout the process (such as internal organisational documentation), or at a specific stage of the software development life cycle. | Internal organisational documentation and procedures:<br>• Availability;<br>• Awareness;<br>• Perceived usefulness. | [6–8; 10; 14–16; 24; 25; 28; 29; 33; 41; 46; 48; 50; 53; 62; 66; 71; 88; 90; 94; 96; 99; 99; 108; 114; 115] |
| | Requirements stage factors:<br>• Difficulties with defining privacy and security concepts and requirements;<br>• Tension between privacy/security and other technical and system requirements. | [6; 12; 14; 15; 17; 20; 23; 28; 45; 48; 50; 53; 55; 58; 62; 65; 68; 71; 72; 74; 79; 80; 87; 88; 90; 95; 97; 108; 113; 118] |
| | 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. | [6; 7; 9–11; 15; 20; 21; 28; 34; 35; 37; 41; 42; 48; 50; 53; 62; 64–66; 71; 80; 83; 87–90; 94; 97; 99; 108; 110; 113; 114; 116; 117] |
| | Review and evaluation stage factors:<br>• Evaluation process and metrics | [10; 11; 15; 21; 24; 28; 48; 53; 58; 65; 71; 81; 90; 92; 97; 100; 108; 114; 116] |
| **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 | [6; 10; 12; 20; 24; 28; 50; 53; 82; 87; 114] |
| | Tensions between privacy/security and business priorities | [10; 15; 24; 28; 48; 50; 66; 70; 88; 99; 107] |
| | Competitive advantage | [12; 24; 28; 45; 48; 53; 54; 59; 62; 66; 87; 92; 97; 99] |
| **Personal factors** include developers' personal characteristics and backgrounds. | Position and role:<br>• Hierarchical position;<br>• Perceived personal responsibility;<br>• Autonomy and control. | [6; 10; 11; 15; 18; 24; 27; 28; 45; 50; 62; 66; 68; 71; 85; 87; 91; 95–97; 99; 101; 114] |
| | Privacy expertise and knowledge | [2; 6; 10; 12; 13; 15; 16; 20; 30; 31; 36; 49; 51–53; 62; 67; 71; 72; 87; 88; 90; 93; 99; 106] |
| | Instrumental privacy/security attitudes | [4; 6; 7; 15; 23; 52; 62; 66; 71; 87; 90; 99; 112] |
| | Experimental privacy/security attitudes | [6; 15; 16; 52; 99] |
| | Prior experiences with privacy/security violations in the software development context | [6; 7; 114] |
| | Personality traits | [39; 49; 51; 52; 63; 99; 111] |

# C Search Queries

## C.1 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 into 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 into 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 into 84 documents

## C.2 IEEE Xplore

(((("Abstract":Privacy OR security) AND "Abstract":Development OR developer) AND "Abstract":Software) AND "Abstract":"by design")) - 18 May, resulted into 43 documents.