Short Paper: Integrating the Data Protection Impact Assessment into the Software Development Lifecycle

Christopher Irvine(✉), Dharini Balasubramaniam, and Tristan Henderson

School of Computer Science, University of St. Andrews, St. Andrews KY16 9SX, UK
{cai3,dharini,tnhh}@st-andrews.ac.uk
https://www.st-andrews.ac.uk/computer-science/

Abstract. Recent years have seen many privacy violations that have cost both the users of software systems and the businesses that run them in a variety of ways. One potential cause of these violations may be the ad hoc nature of the implementation of privacy measures within software systems, which may stem from the poor representation of privacy within many Software Development LifeCycle (SDLC) processes. We propose to give privacy a higher priority within the SDLC through the creation of a confederated *Privacy-Aware* SDLC (PASDLC) which incorporates the Data Protection Impact Assessment (DPIA) lifecycle. The PASDLC brings stakeholders of the software system closer together through the implementation of multiple interception points, whilst prompting the stakeholders to consider privacy within the software system. We consider many challenges to the creation of the PASDLC, including potential communication issues from confederating the processes of a SDLC and the effective measurement of privacy as an attribute of a software system.

Keywords: Privacy · Software architecture · Software engineering lifecycle · Data protection impact assessment

1 Introduction

Recent years have seen several privacy breaches and violations. For example, on the 5th of March 2020, Virgin Media admitted a database, containing the personal details of 900,000 people, was left unsecured and accessible online for 10 months, during which this data was accessed "on at least one location" [4]. In 2019 a major breach was reported by Capital One impacting 106 million people which compromised social security numbers and bank accounts [3]. Other examples include Google ignoring user privacy preferences [23] and recent concerns that Zoom has been sharing user data with Facebook without user consent [12]. These privacy breaches and violations are all described as accidental or avoidable [3,4], which suggests there is a procedural issue with privacy in software development.

J. Garcia-Alfaro et al. (Eds.): DPM 2020/CBT 2020, LNCS 12484, pp. 219–228, 2020.
https://doi.org/10.1007/978-3-030-66172-4_13

At the time of writing, the NHS COVID-19 contact-tracing app is under investigation regarding a lack of consideration of privacy [8] and deploying the system without an approved Data Protection Impact Assessment (DPIA) [24]. The DPIA is a legal requirement under the European Union General Data Protection Regulation (EU GDPR) [10, Article 35] and the UK's Data Protection Act (2018)(DPA) [33]. A recent survey on DPIAs, performed by the European Unions Protection Supervisor, revealed that data protection officers who took part in the surveyed DPIAs believed that the DPIA processes would benefit from greater awareness and more internal support, additionally the process itself could be simpler. A recent survey of Data Protection Officers found that DPIA processes were promising, but would benefit from greater awareness, internal support and a simplification of the process itself [11].

One potential cause of a privacy breach or violation is the ad hoc nature of implementing privacy measures into software systems [17,25] due to the poor representation of privacy within the Software development LifeCycle (SDLC) [5,25]. We aim to bring clarity to the SDLC by prompting stakeholders to consider privacy as an attribute of the software system before, during and after implementation. To achieve this aim, we propose a *Privacy-Aware* SDLC (PASDLC) that combines the DPIA Lifecycle[1] with the SDLC.

The PASDLC takes into consideration legal requirements, such as those set out in the GDPR and the DPA, by regularly prompting consideration and review of the data processing that occurs within the software system being designed. To achieve this, the normally loosely related stages of a SDLC are confederated into a single governing structure where each lifecycle or process will intercept others at multiple stages, bringing the stakeholders of the software system closer together. This structure brings together both the law and computing; it has often been argued that such a multidisciplinary approach is required to address the potential harm from technology, for instance through Lessig's "pathetic dot" [21, ch. 7]. Bringing multiple disciplines together, however, may also cause communication and consistency issues impacting the overall quality of the implemented software system [19]. We discuss these challenges and how we approach them in the initial design of the PASDLC which revolves around the early processes of the SDLC, namely requirements engineering, software architecture design and implementation.

2 Background

2.1 Software Development Lifecycle

Software engineering is governed by various lifecycles and processes which guide stakeholders in developing a software system that satisfies requirements and constraints. These processes allow multiple teams of stakeholders to work on the same software system with minimal disruption [31, ch. 2]. A generic SDLC can be found in Fig. 1. Each stage within a SDLC consists of processes and lifecycles such as requirements engineering or software engineering methodologies.

[1] As developed by the Information Commissioner's Office [14].

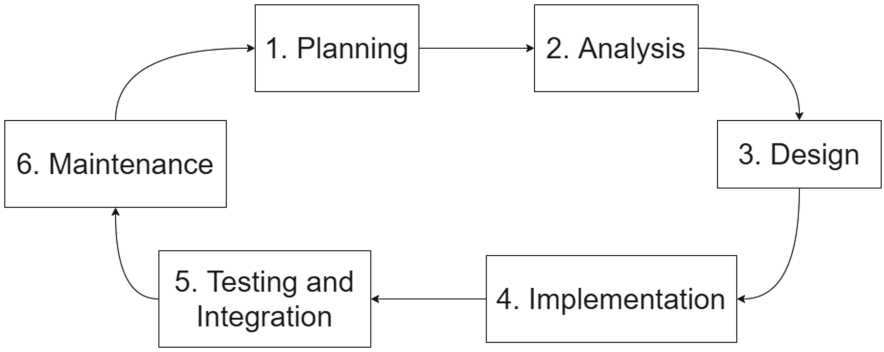

Fig. 1. A graphical representation of a software development lifecycle.

2.2 Software Architecture

Software architecture is a high level model capturing significant design decisions relating to the structure and behaviour of a software system and providing guidance to developers on how to implement and maintain the system, including details such as software components and the interactions among them [32]. Software architecture is created using design processes such as Attribute-Driven Development (ADD) [35] and evaluation processes such as the Architecture-Tradeoff Analysis Method (ATAM) [18]. Privacy is not well represented within these processes, except from using Unified Modelling Language (UML) diagrams to document privacy requirements as stated in the requirements specification [26].

2.3 Data Protection Impact Assessment

Software systems that involve the processing of personal data of EU residents are governed by the GDPR[2]. More specifically, some systems, for instance those that use automated processing that cause legal effects, or systematically monitor publicly accessible areas at a large scale, must preform a DPIA. To aid in this process, the Information Commissioner's Office (ICO) has created a suggested lifecycle for completing and updating a DPIA (Fig. 2) [14].

To be an effective impact assessment tool, the DPIA must be completed before any processing of sensitive data by the software system or any future iterations of the software system which change how data is processed.

From a software engineering perspective, the most interesting stages of the DPIA lifecycle are 7, 8 and 9. Stages 1 to 6 involve stakeholders with technical expertise from multiple disciplines who compile the DPIA document which is then signed off by the Data Protection Officer (DPO), who may be a non-engineer, in stage 7. Once the DPIA has been approved, the technical stakeholders will execute stages 8 and 9. Without a pre-established common vocabulary,

[2] We focus on the GDPR, but other similar regulations are appearing in other jurisdictions such as the California Consumer Privacy Act.

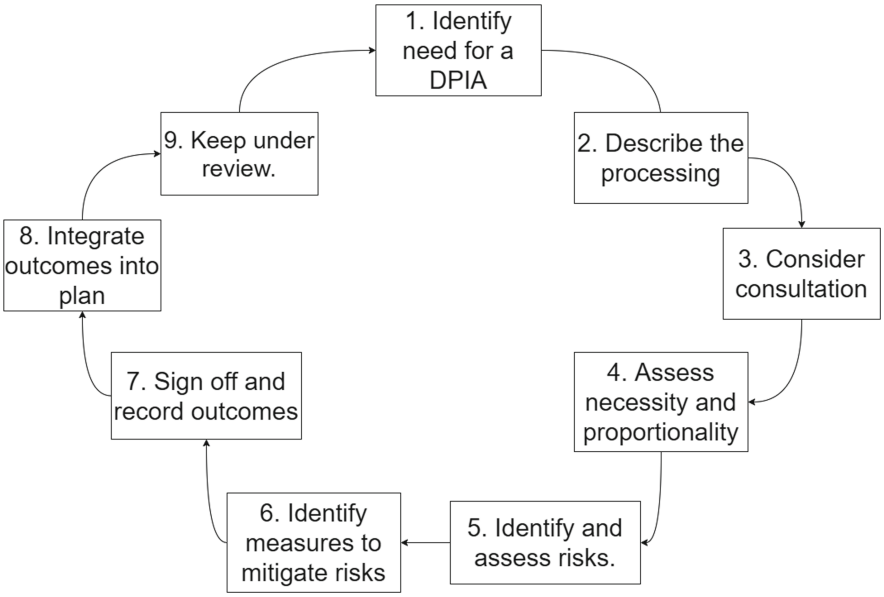

Fig. 2. A graphical representation of a data protection impact assessment lifecycle

the DPO may not fully understand the content of the DPIA leading to privacy measures being approved or rejected incorrectly.

2.4 Related Work

Privacy engineering aims to create techniques that decrease privacy risks and increase effective privacy controls within software systems [9], integrating Privacy Enhancing Technologies (PETs) such as anonymisation. Software engineers who use the PASDLC will be able to use Privacy Engineering techniques to implement the planned privacy measures during the implementation and design stages of the PASDLC.

Privacy by Design (PbD) [7] and Data Protection by Design (DPbD)[3] serve as principles to guide the development activities of software engineers towards creating software systems with increased privacy awareness. Hadar et al. find that developers may be actively discouraged from PbD processes due to organisational norms or lack of knowledge [13]. We propose to integrate the DPIA (and DPbD) into the organisation through the PASDLC.

Some PbD/DPbD activities encourage stakeholders to integrate privacy into the architectural specification [29]. This is done either by integrating specific privacy enhancing methods into the architectural specification [20] or the creation of specific software architectural privacy views. Sion proposed that DPbD should

[3] Data Protection by Design is specific to GDPR (Article 25).

have a dedicated architectural view supported by data flow diagrams to instruct engineers how to model the flow of data between software components [30].

To test whether the PASDLC improves privacy within a software system, we need to be able to measure privacy. There are multiple privacy metrics available which measure different data ranging from the estimated effort required for a third party to breach a database to the gain the third party would receive for completing the breach [34]. Each metric is individually useful to the stakeholders, however, there is no overall measurement of privacy within a software system. Zhao and Wagner recommend combining metrics into a *metric suite*, which is specific to the software system, as a method of measuring overall privacy of the software system [36].

Sedano et al. and Sievi-Korte et al. note communication issues have been amplified by the rising level of outsourcing in the software engineering industry, resulting in increased design deviations [27,28]. Current solutions revolve around categorising the causes of the communication issues – such as time zones and response delays – and then creating a mitigation strategy for each category. These strategies often rely on the use of third party instant messaging, video conferencing and organisation tools [16], which, as the Berlin data protection authority outlines, may themselves introduce data protection risks [6].

Whilst this research is concerned with the ICO's methodology for generating and maintaining a DPIA, we note that other methods may be used, such as the model-based approach proposed by Ahmadian in [1].

3 Approach

We hypothesise that a confederated PASDLC which combines the SDLC and the DPIA lifecycles, as discussed in Sect. 2, can improve privacy within the developed software system. The PASDLC goes beyond integrating the DPIA lifecycle into regular procedure, providing multiple intersection points between each of the stages within the PASDLC that allow stakeholders of the software system to address concerns mid-iteration.

At this point our focus is on the initial stages of developing the PASDLC: requirements engineering, software architecture design & evaluation and implementation to act as a proof of concept. See Fig. 3 for a high level view of the PASDLC.

Using the NHS COVID-19 contact-tracing app as a case study (see Sect. 1) we discuss the PASDLC further. The requirements will be agreed with the clients, the NHS and the UK Government, and the need for a DPIA is established due to the sensitive health and location data processed by the app [10, Article 35]. The stakeholders will describe in detail the processing necessary for the app to function. At this point external consultants may be employed, such as data protection lawyers, to assist with the DPIA risk assessment later in the process. Once the requirements engineering processes have ended, the necessity and proportionality of the processing is assessed to ensure it is vital to the functionality of the software system. For the contact-tracing app, processing

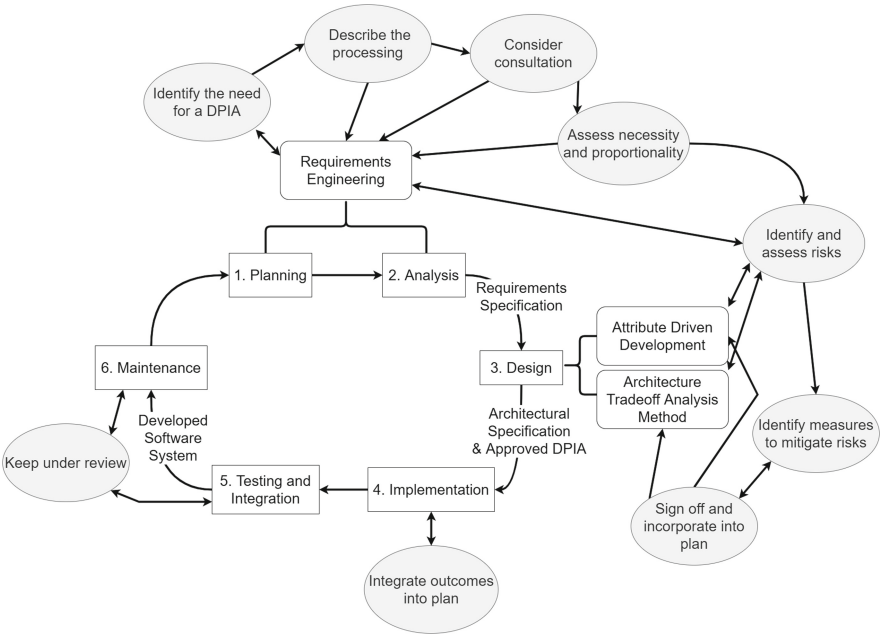

Fig. 3. A high level view of the PASDLC; the steps of the DPIA are in grey ovals, and the steps of the SDLC are in white rectangles, with suggested processes for the design step in rectangles with rounded corners. The arrows signify the order in which processes should be carried out by stakeholders.

sensitive data is vital to the functionality, therefore the DPIA process moves on to the risk assessment stages.

During the design stage of the PASDLC a variety of methodologies to develop (ADD) and evaluate (ATAM) a software architecture can be used. Regardless of the methodology used, as part of the DPIA, a privacy risk assessment will be performed by the stakeholders of the software system. An example risk for the app may be an unauthorised access to the NHS patient records which could affect millions of people. Risk mitigation methods are then integrated into the requirements and software architecture specifications, for example, limiting the data access to the NHS patient records to only COVID-19 related data.

The software system is implemented using the approved requirements and architecture specifications controlled by the software engineering methodology the stakeholders choose. A primary goal when testing the software system will be to ensure that the software system adheres to the approved DPIA by checking that all implementable privacy measures have been implemented. After passing the testing processes, the software system is deployed and remains in the maintenance stage of the PASDLC until new features are added. Requiring the approval of the DPO before the implementation stage of the PASDLC reduces the risk of deploying a software system or integrating a new feature into an existing software

system without an approved DPIA, as was the case for the contact-tracing app. By integrating both and making it clear that this is an ongoing and repeated lifecycle, we also hope to prevent a mismatch between DPIA and released system, as was also the case for the NHS app, with a DPIA only being released for an initial pilot test and not for the final system.

In lower level views of the PASDLC, specific processes, such as scrum or waterfall (for the S.D. 4), ADD and ATAM (for S.D. 3) and requirements engineering (for S.D. 1 and 2), will be inserted into the corresponding stage of the PASDLC. Each activity within these processes will be mapped to the appropriate DPIA activities, providing an easy to use framework for engineers and non-engineers alike to follow the development of a *Privacy-Aware* software system.

The PASDLC will become an engineering privacy tool box which will not only be compatible with PETs, PbD/DPbD and standards such as ISO/IEC 29110 [15,22] or the generally accepted privacy principles [2], it will prompt to the user to consider the inclusion of relevant standards, processes or technologies at the appropriate points. The PASDLC will not prescribe to the user any one given standard, technology or processes and will encourage the user to research the best standard, technology or process for the software system being developed.

This research will address three main challenges: measuring privacy, managing communication issues and evaluating the PASDLC proof of concept. As discussed in Sect. 2.4, Metric suites may be the solution to measuring privacy within software systems and evaluating the effectiveness of the PASDLC.

Requiring stakeholders from different disciplines to work closer together through the non-linear nature of the PASDLC may exacerbate existing communication issues – such as the DPO not understanding technical terminology within the DPIA – or create new ones. Part of this research will investigate the potential for communication issues and explore mitigation techniques, such as establishing a common vocabulary or defining system documentation, that can be utilised by stakeholders to counter their adverse effects on the software system. Successful mitigation techniques will be incorporated into the PASDLC either as a step (such as in the case of establishing a common dictionary) or highlighting existing steps to encourage users of the PASDLC to deploy the appropriate mitigation technique.

The final challenge is the evaluation of the PASDLC proof of concept. Case studies will have their software architecture redeveloped using the PASDLC processes. The amount of privacy in both the original and redeveloped architectures will be measured where we expect to see an increase in privacy within the redeveloped architecture.

4 Conclusion

This work aims to address the insufficient privacy measures implemented into software systems, potentially caused by the poor representation of privacy within many SDLC processes. We hypothesise that this problem can be addressed by integrating the DPIA lifecycle with the SDLC creating the PASDLC.

We will evaluate the developed PASDLC proof of concept by redeveloping the software architectures of case studies using the PASDLC where we expect to see an increase in privacy in the redeveloped architecture as measured by privacy metrics. We will further investigate the PASDLC for potential communication issues. Strategies to mitigate these issues will be developed to reduce consistency problems across multiple artefacts and stakeholders of the software system.

The next steps are the development and evaluation of the proof of concept PASDLC which will expand into the creation of an engineering privacy toolbox which is both compatible and promotes the use of privacy standards, practices and technologies.

Through the creation of an effective PASDLC we hope to see a reduction in privacy breaches and violations that can cause financial and reputational harm to the stakeholders of software systems which process sensitive data.

References

1. Ahmadian, A.S., Strüber, D., Riediger, V., Jürjens, J.: Supporting privacy impact assessment by model-based privacy analysis. In: Proceedings of the ACM Symposium on Applied Computing, pp. 1467–1474 (2018). https://doi.org/10.1145/3167132.3167288
2. American Institute of Certified Public Accountants, Inc. and Canadian Institute of Chartered Accountants: Generally Accepted Privacy Principles and Criteria (August), pp. 1–84 (2009). http://www.aicpa.org/InterestAreas/InformationTechnology/Resources/Privacy/GenerallyAcceptedPrivacyPrinciples/DownloadableDocuments/GAPP_PrinciplesandCriteria.pdf
3. BBC News: Capital One data breach: Arrest after details of 106m people stolen - BBC News. https://www.bbc.co.uk/news/world-us-canada-49159859 (2019)
4. BBC News: Virgin Media data breach affects 900,000 people - BBC News. https://www.bbc.co.uk/news/business-51760510 (2020)
5. Beckers, K.: Comparing privacy requirements engineering approaches. In: Proceedings - 2012 7th International Conference on Availability, Reliability and Security, ARES 2012, pp. 574–581 (2012). https://doi.org/10.1109/ARES.2012.29
6. Berliner Beauftragte für Datenschutz und Informationsfreiheit: Hinweise für Berliner Verantwortliche zu Anbietern von Videokonferenz-Diensten. https://www.datenschutz-berlin.de/fileadmin/user_upload/pdf/orientierungshilfen/2020-BlnBDI-Hinweise_Berliner_Verantwortliche_zu_Anbietern_Videokonferenz-Dienste.pdf, July 2020
7. Cavoukian, A., et al.: Privacy by design: The 7 foundational principles. Information and privacy commissioner of Ontario, Canada 5 (2009)
8. Dearden, L.: Coronavirus: NHS contact-tracing app must not be released to public without privacy protections, MPs say — The Independent. https://www.independent.co.uk/news/uk/home-news/coronavirus-nhs-contact-tracing-app-covid-19-uk-release-date-privacy-protection-a9503321.html
9. Dennedy, M.F., Fox, J., Finneran, T.R., Bonabeau, E.: The Privacy Engineer's Manifesto: Getting from Policy to Code to QA to Value. Apress, Berkely (2014)
10. EU: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/4. Official J. European Union (OJ) **59**, 1–88 (2016)

11. European Data Protection Supervisor: EDPS Survey on Data Protection Impact Assessments under Article 39 of the Regulation 1, 1–31 (2020). https://edps.europa.eu/data-protection/our-work/publications/reports/edps-survey-data-protection-impact-assessments-under
12. Foltyn, T.: Zoom's privacy and security woes in the spotlight (2020). https://www.welivesecurity.com/2020/04/03/zoom-privacy-security-spotlight/
13. Hadar, I., Hasson, T., Ayalon, O., Toch, E., Birnhack, M., Sherman, S., Balissa, A.: Privacy by designers: software developers' privacy mindset. Emp. Softw. Eng. **23**(1), 259–289 (2017). https://doi.org/10.1007/s10664-017-9517-1
14. Information Commissioner's Office: Data protection impact assessments — ICO (2018). https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/accountability-and-governance/data-protection-impact-assessments/
15. ISO/IEC 29110-2-1:2015: Software engineering - Lifecycle profiles for Very Small Entities (VSEs) - Part 2–1: Framework and taxonomy. Standard, International Organization for Standardization, November 2015
16. Jaanu, T., Paasivaara, M., Lassenius, C.: Effects of four distances on communication processes in global software projects. In: Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 231–234 (2012)
17. Kalloniatis, C., Kavakli, E., Gritzalis, S.: Dealing with privacy issues during the system design process. In: Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology 2005, pp. 546–551 (2005)
18. Kazman, R., Klein, M., Clements, P.: Atam: method for architecture evaluation. Tech. Rep. CMU/SEI-2000-TR-004, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (2000). http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5177
19. Khan, A.A., Basri, S., Dominc, P.: A proposed framework for communication risks during RCM in GSD. Procedia Soc. Behav. Sci. **129**, 496–503 (2014). https://doi.org/10.1016/j.sbspro.2014.03.706
20. Kung, A.: PEARs: Privacy Enhancing ARchitectures. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **8450**, 18–29 (2014). https://doi.org/10.1007/978-3-319-06749-0_2
21. Lessig, L.: C o d e, 2nd edn. New York, New York, USA (2006)
22. Morales-Trujillo, M.E., Garcia-Mireles, G.A.: Extending ISO/IEC 29110 basic profile with privacy-by-design approach: A case study in the health care sector. Proceedings - 2018 International Conference on the Quality of Information and Communications Technology, QUATIC 2018, pp. 56–64 (2018). https://doi.org/10.1109/QUATIC.2018.00018
23. Nakashima, R.: AP Exclusive: Google tracks your movements, like it or not (2018). https://apnews.com/828aefab64d4411bac257a07c1af0ecb
24. Page, C.: Test and trace initiative faces legal challenge in the U.K. over data collection (2020). https://www.forbes.com/sites/carlypage/2020/06/01/nhs-faces-legal-challenge-over-rushed-test-and-trace-initiative/#73c74e2f673f
25. Peixoto, M., Ferreira, D., Cavalcanti, M., Silva, C., Vilela, J., Araújo, J., Gorschek, T.: On understanding how developers perceive and interpret privacy requirements research preview. In: Madhavji, N., Pasquale, L., Ferrari, A., Gnesi, S. (eds.) REFSQ 2020. LNCS, vol. 12045, pp. 116–123. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44429-7_8

26. Sachitano, A., Chapman, R.O., Hamilton, J.A.: Security in software architecture: a case study. In: Proceedings fron the Fifth Annual IEEE System, Man and Cybernetics Information Assurance Workshop, SMC, pp. 370–376. IEEE, West Point, NY, USA (2004). https://doi.org/10.1109/iaw.2004.1437841
27. Sedano, T., Ralph, P., Péraire, C.: Software development waste. In: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pp. 130–140 (2017)
28. Sievi-Korte, O., Richardson, I., Beecham, S.: Software architecture design in global software development: An empirical study. J. Syst. Softw. **158** (2019). https://doi.org/10.1016/j.jss.2019.110400
29. Sion, L., Van Landuyt, D., Yskout, K., Joosen, W.: Sparta: security privacy architecture through risk-driven threat assessment. In: 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), pp. 89–92 (2018)
30. Sion, L., et al.: An architectural view for data protection by design. In: Proceedings - 2019 IEEE International Conference on Software Architecture, ICSA 2019, pp. 11–20. No. i, IEEE, Hamburg, Germany (2019). https://doi.org/10.1109/ICSA.2019.00010
31. Sommerville, I.: Software engineering (10th edition) (2016)
32. Taylor, R.N., Medvidovic, N., Dashofy, E.: Software Architecture: Foundations, Theory and Practice (2010)
33. UK Government: Data Protection Act 2018 (2018)
34. Wagner, I., Eckhoff, D.: Technical privacy metrics: a systematic survey. ACM Comput. Surv. **51**(3) (2018). https://doi.org/10.1145/3168389
35. Wojcik, R., et al.: Attribute-driven design (add), version 2.0. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST (2006)
36. Zhao, Y., Wagner, I.: Using metrics suites to improve the measurement of privacy in graphs. IEEE Trans. Dependable Secure Comput. pp. 1 (2020). https://doi.org/10.1109/tdsc.2020.2980271