Continuous Requirements: An Example Using GDPR

Ze Shi Li
University of Victoria
Victoria, BC, Canada
lize@uvic.ca

Colin Werner
University of Victoria
Victoria, BC, Canada
colinwerner@uvic.ca

Neil Ernst
University of Victoria
Victoria, BC, Canada
nernst@uvic.ca

***Abstract*—Recently, a stringent set of privacy regulations, the General Data Protection Regulation (GDPR), was enacted in the European Union, which can be considered a privacy non-functional requirement (NFR). As a result, an organization that collects or processes data from European citizens must adhere to the GDPR. Previous studies have shown that compliance to the GDPR poses a number of challenges, which we have confirmed in our own research. In this paper, we describe our ongoing research collaboration with a startup organization that is adopting the GDPR. In addition, during the course of our research, we found that our industry collaborator, practices continuous integration (CI) like many other organizations. The number of organizations adopting CI has increased since Fowler first published his definition of CI. As such, another aspect of our current research is exploring the effects of CI on privacy NFRs and other NFRs. Finally, we describe a design science approach to iteratively learn about industry challenges in GDPR compliance, NFRs in the context of CI, as well as our ongoing work creating a tool to potentially mitigate observed GDPR compliance challenges.**

I. Introduction

The General Data Protection Regulation (GDPR) is a European Union (EU) legislation with potentially significant effects on how an organization manages personally identifiable information (PII) for EU citizens that was passed on May 24, 2016 [1]. The GDPR's mandate is to protect the privacy of EU citizens and will punish an organization that fails to comply with the GDPR, regardless of whether the organization did so with intent or negligence. Although the GDPR was not immediately enforced, an organization conducting business in the EU was expected to obtain GDPR compliance by May 25, 2018 [1]. For an organization actively working to attain compliance, the organization may operationalize each GDPR regulation into smaller work tasks. These smaller work tasks that help an organization adhere to the GDPR can be thought of as privacy non-functional requirements (NFRs).

NFRs act as characteristics that may guide the design and implementation of an organization's systems [2]. These characteristics often influence architectural decisions, which may have significant impact on a system. Architectural decisions may affect the answer to questions such as: How modular is a system? Is the system maintainable? How does the system deal with customer data? Accordingly, researchers often describe NFRs as important aspects of a system. Yet, researchers do not have a consensus on a universally accepted definition of an NFR. Moreover, the concept of NFRs is largely neglected in practice [3]. Since customers and stakeholders frequently express desired outcomes of a system in terms of functionality [4], product development focuses on satisfying the stated functions. When product design and development solely concentrate on functionality, as opposed to the underlying NFRs with long term ramifications that impact architecture, an organization may be required to re-factor a considerable amount of code.

For a startup organization that is pursuing growth and shipping features to customers as fast as possible, NFRs are not an initial priority [5]. Once a startup organization is able to hire more employees, the organization may be able to address more NFRs. However, even if a startup organization emphasizes some NFRs, such as privacy, from the onset, external forces may overwhelm an organization's initial preparation. Due to the magnitude and comprehensiveness of the GDPR, a startup organization that believes and practices the notion of protecting user privacy potentially may not have the necessary amount of resources to prepare itself for compliance.

In part due to our own interest in privacy and NFR research, as well as being approached by a local startup organization, which was required to comply with the GDPR, we began this GDPR and NFR research. Our partner organization is based in Canada, but has many European users. Therefore, the organization falls under the jurisdiction of the GDPR. Although our partner organization maintains a firm belief in user privacy, the organization experienced challenges in its initial adoption of the GDPR and desired robust processes, policies, and controls to help protect personal data and comply with the GDPR. The existence of their challenges echos the large number of organizations who reported difficulty complying by the GDPR deadline [6] [7]. To address our partner organization's problems and our research intrigue of the GDPR, we adopted a design science research methodology, which involves iterative research. The nature of iterative research allows us to build and refine our research artifacts until the completion of the research. Additionally, our research methodology contained two envisioned artifacts. The first artifact is the current state of practice of an organization adopting and complying with the GDPR and the challenges that an organization faces during

its compliance, since we did not have a clear idea of the GDPR adoption challenges nor adoption practices adhered by our partner organization. Based on our analysis of the first artifact, the second artifact contains a GDPR tool that may aid an organization's GDPR compliance.

The novelty of our research is to identify *what* a developer did to prepare for the GDPR and *how* the GDPR affects a developer. In particular, learning about industrial practices strengthens the knowledge base of specific challenges that plague an organization's GDPR compliance.

While our research is currently ongoing, we will discuss our preliminary results in this paper, which includes parts of the first artifact. Currently, our iterative research is in the midst of producing the second artifact. When we first began our research we quickly found another intriguing aspect that was ubiquitous to our partner organization's work; although we had not initially considered the effect of continuous integration (CI) [8] as part of our research. We found that CI had a large impact on our partner's planning, development, testing, and deployment. Hence, we could not, and cannot, disregard the importance of CI on our partner organization, especially the impact of CI on the organization's treatment of NFRs and compliance with the GDPR.

We argue that this paper presents three main contributions

- detailed analysis of environmental context for a startup dealing with GDPR, in a CI context,
- methodology for investigating CI and GDPR in practice, and
- preliminary results on developer awareness of privacy and GDPR compliance.

The rest of the paper is structured as follows. In Section II, an overview of NFRs, the GDPR, CI, and related work in privacy is covered. In Section III, we provide details on our research methodology. In Section IV, we share some insights from our initial findings. In Section V, we conclude the paper and highlight our future work.

II. Background and Related Work

A. Non-Functional Requirements

As previously mentioned, the GDPR may be considered as a single or a series of privacy NFRs. A common definition of an NFR is provided by Glinz [2] as "an attribute of or constraint on a system". In essence, an NFR, also known as a quality attribute, is a characteristic of a system that may help guide architectural decisions. In practice, NFRs are often ignored or not prioritized by both customers and developers [9] or prioritized lower than functional requirements [10].

Deciding which NFRs are important to a particular system is a challenging task, as decisions must be made with trade-offs between various NFRs. In addition, the lack of any sort of prioritization is also a form of a trade-off. Whether willingly or not, when an organization settles on an NFR trade-off, the organization may inadvertently accumulate more technical debt. As the amount of technical debt swells, the organization's ability to develop new features may be limited by the insurmountable technical debt. Eventually, such an organization may reach a point where substantial code re-factoring is necessary for the organization to continue to deliver new features. As a result, an organization that neglects NFRs when developing and designing a system may need to significantly re-factor its system, if at all possible, later in the system's life cycle [4]. For example, an NFR, such as privacy, can be especially difficult to implement at a later stage in a product's life cycle and can lead to a "long and unhappy history of incremental patching and retrofitting that characterizes the current Internet architecture" [11]. Furthermore, attempting to retrofit privacy requirements may be flawed without initial consideration of security as security and privacy are often intertwined [12].

As privacy safeguards and procedures are a fundamental aspect of a system, privacy measures should inherently exist as part of the product from inception. On the contrary, an organization may perceive benefit if an appropriate amount of importance is placed on NFRs early in a product life cycle, which can help propel and shape the architectural design and implementation [13].

B. General Data Protection Regulation

Prior to the passing of the GDPR, our partner organization already implemented controls to protect personal data, but as the most comprehensive privacy law, the GDPR affected everything from development to storage of data. In addition to the GDPR, other major privacy regulations have been or will be introduced, including the California Consumer Privacy Act (CCPA),[1] as well as Vermont's Data Broker Regulations.[2] In addition to being a complicated NFR to handle, privacy is one NFR that applies to most organizations as the GDPR applies to any organization that is based in the EU or collects personal data from identifiable EU citizens [1], which certainly is applicable to a significant number of organizations.

As a novel and trailblazing set of regulations that strive to protect the privacy of individual personal data from abuse, the GDPR stipulates tough financial penalties for violators and imposes stringent regulations to an organization's data collection and process practices. Consequently, many organizations have adopted the GDPR as their default privacy standard, as the GDPR is generally regarded as the most strict privacy policy released and enforced to date [14].

However, as the deadline approached, many organizations were still not yet GDPR compliant [6] [7]. Leading up to the 2018 deadline, a non-GDPR compliant organization had two options: 1) turn off aspects of their system that were not GDPR compliant or 2) continue with normal operations in hope of not being caught for non-compliance. Unfortunately, statistics are unavailable on the percentage of organizations that risked being found in contempt of the GDPR and continued with their usual operations despite being non-compliant. Finally, the crux

[1] https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375

[2] https://legislature.vermont.gov/Documents/2018/Docs/ACTS/ACT171/ACT171%20As%20Enacted.pdf

of the GDPR is the emphasis on being compliant *all* of the time. Instead of having the luxury to have discretion on when to comply with the GDPR, our partner organization must be continually compliant.

C. Continuous Integration

To better help our partner organization, we had to first understand their development process. However, we not only considered their development process, but also planning, testing, and deployment processes. Our partner organization is a proponent of CI. CI is a practice that involves the use of a.) automated builds and tests, b.) each developer committing at least once a day, and c.) quickly fixing broken builds [8]. A characteristic of an organization that practices CI is that the organization quickly releases new updates to users and rapidly receives feedback from users [8].

In the current competitive business environment, it is difficult for an organization to compete if the organization is unable to quickly adapt to changes in the market and release products that match customer demands. As such, an organization may opt to adopt CI. The roots of CI originated from Agile development, Lean development, and Extreme Programming [8] and has extended to other "continuous" practices such as continuous delivery (CD) [15]. The appeal for an organization to adopt continuous practices, includes fast release of software to customers, reliably releasing to production, and high customer satisfaction [16].

CI aims to reduce a developer's time spent on manual tasks, such as building and testing software [8]. Similarly, an organization practicing CD not only practices CI, but also maintains a high-level of confidence that the organization's software is in a production-ready state at all times [15]). CD aims to maintain a high-level of code quality and also enables an organization to shorten a customer's wait time for receiving the latest updates and remove the hardships of preparing software for release [17].

Unfortunately, there have been conflicting reports on operationalizing NFRs in the context of "continuous". One study has suggested it is difficult to automate testing of NFRs [17] as creating validation criteria for NFRs is not clear-cut. Practitioners in a recent summit indicate that NFRs, such as privacy, are not the concern of every developer [18]. Nevertheless, if an organization is serious about GDPR compliance, a privacy NFR, then the organization may require a privacy-conscience effort from every employee.

Furthermore, considering that the GDPR expects continuous compliance, it may be prudent for an organization's continuous pipeline to run a series of GDPR-related tests on every commit to ensure that privacy is not compromised. However, we do not know the ideal frequency at which an organization should test privacy. Moreover, a frequency that is optimal for one organization may not translate to another organization. A goal of our research is to help contextualize the frequency and depth of testing the practitioner's perspective. If an organization forgoes the notion of NFRs and develops software without consideration of constraints or attributes of a system, the organization may accumulate substantial amounts of technical debt [19]. When an organization accumulates a crushing amount of technical debt, the software may experience catastrophic challenges, such as the inability to re-factor architecture and loss of usability [20].

D. Related Privacy Work

Seminal works in privacy include Deng et al's [21] LINDDUN methodology that can help identify privacy threats in an organization's system and map those threats to privacy requirements, but the methodology abstracts threats at a high level. Nonetheless, just as Deng et al. suggested solution strategies for privacy requirements [21], our partner organization shut off parts of their EU services before the implementation of the GDPR. While previous studies on helping organization assess GDPR compliance exist, such as a tool based approach to conduct Data Protection Impact Assessments [22], there is a shortage of studies analyzing GDPR adoption *challenges* and *practices*. In Sirur et al.'s [23] work, it was found that large organizations did not experience significant obstacles with GDPR compliance. Alternatively, smaller organizations without significant prior emphasis on security felt GDPR compliance was onerous [23] indicating there was insufficient privacy by design [24]. Furthermore, the surveys leading up to the compliance deadline [6] [7] indicated that a substantial number of organizations were not ready for the GDPR in time. Regardless, there appears to be a major disconnect between privacy concerns, such as the GDPR, and implementing engineering solutions to satisfy privacy concerns [25] [26]. Methodologies do exist that focus on engineering with privacy concerns, such as privacy by design; however, privacy by design revolves around the ability to develop a system from the onset, as opposed to retrofitting privacy into an existing, perhaps legacy, system [27]. An organization adopts, often due to convenience, the less reliable "privacy-by-policy" approach as opposed to the more reliable "privacy-by-architecture" [28]; however, "privacy-by-architecture" is difficult to apply to an existing project, much like "privacy-by-design", as re-factoring a business model may be even more challenging than solely a system.

Ultimately, regardless of how a system is designed or built, there is a need to ensure that the system remains continuously compliant with the GDPR mandate. One study proposes reusing static code analysis tools to define and discover potential GDPR compliance violations [29]. Unfortunately, static code analysis is limited to analyzing only code, whereas GDPR compliance encompasses many more components, such as infrastructure, architecture, and databases. Another proprietary solution, IBM Security Guardium Analyzer, is able to analyze a database and classify a datum as personally identifiable information [30]. Hewlett Packard Enterprises offers a similar data classification tool as part of their GDPR Starter Kit[3].

[3]https://www.hpe.com/us/en/newsroom/news-advisory/2017/05/hpe-software-launches-gdpr-starter-kit-to-expedite-and-simplify-compliance.html

However, simply raising awareness by identifying potentially personally identifiable information is only one, albeit quite large, aspect of GDPR compliance. This lack of an all-encompassing continuous GDPR compliance tool motivates our research ambition to collaborate with industrial partners in a design science methodology to help design, implement, and assess a continuous GDPR compliance tool.

III. Design Science Methodology

The purpose of our study is to increase the understanding of the GDPR and privacy NFRs in an industrial setting. In particular, we observe and analyze how an organization practicing CI manages and handles NFRs. As previously discussed, the current state of knowledge of an organization practicing continuous practices suggests that the organization may not clearly define or test NFRs. Therefore, one of the main goals of our research is studying how an organization deals with privacy NFRs that the organization must prioritize and comply. During our research, we are documenting GDPR adoption challenges of our partner organization to increase the awareness of adoption pitfalls. More importantly, we want to help alleviate some of these challenges; as such, we are designing and developing a GDPR compliance tool to mitigate some GDPR compliance challenges and ensuring the tool is felicitous for practitioners. The nature of our research started with the exploration of NFRs (specifically privacy) during our partner's adoption and compliance with the GDPR. Since we began the research without knowing the specific challenges and practices of our collaborating organization, we are conducting design science research, which relies on iterative cycles of building and refining our design science artifacts [31].

Our research methodology consists of two major parts with a total of four design science iterations as shown in Fig. 1. *Part A* involved interviewing developers from our collaborating organization and contained one design science iteration. *Part B* involves designing, developing, deploying, and synthesizing the results of a GDPR tool that aids our collaborating organization. To build a tool that benefits our collaborating organization, we observe and reflect on each iteration from both parts *A* and *B*. The first iterative cycle was a problem investigation cycle, which forms the foundation for, and informs the work of, our subsequent cycles. The primary focus of the first cycle was to identify the standard of practice of defining and testing NFRs observed in practice and explained by our collaborating organization. Furthermore, we interviewed our collaborating organization to gain insight on their GDPR adoption practices and challenges as a specific example of an NFR in practice. Finally, we conducted our research mindful of CI. We observed our collaborating organization's development process and also included questions about CI in our interviews. In short, the design science artifact produced from the first cycle is the breadth of knowledge that represents an organization's definition and treatment of NFRs, as well as challenges experienced by practitioners when complying with the GDPR.

Currently, we are past the first iteration cycle and are working on *Part B* as shown in Fig. 1. Although the relevance cycle of our research has been established, the rigor cycle and evaluation of our research artifacts are still ongoing. Based on the insights gained from the first iteration, we are iteratively developing, and deploying a GDPR tool as part of the second, third, and fourth iterations. After each deployment of the GDPR tool, we further observe and interview developers to understand the positive impact of the tool; any suggestions for improvement or observed deficiencies are documented and congregated into the subsequent iteration. Similar to why our partner organization practices CI, we can quickly adapt our tool based on feedback after deployment. As a result, the third and fourth iterations update the GDPR tool by reflecting on the tool through the tool's execution. Thus, we are producing two artifacts, 1) the knowledge-base built during our initial iteration and 2) a tool that helps assess the understanding of our knowledge-base in a practical industry setting.

The interviews and observations from *Part A* contain various themes that represent practices and challenges. To reduce misinterpretations, our primary form of evaluation is performed by validating our interpretation with our collaborating organization. As part of our ongoing work regarding *Part B*, we are creating a GDPR tool that is being deployed in each design cycle iteration. As a result, the GDPR tool has three versions, each derived and built upon its predecessor. After deploying each version, our collaborating organization analyzes the results of the tool and submits feedback regarding the strengths and weaknesses of the tool. Similarly, we observe the results of the tool to form potential ideas for future enhancements. Our tool is evolving through iterative deployment and evaluation, the result of which enables us to produce an artifact that is both relevant to industrial practices and novel to the current knowledge base.

IV. Preliminary Results

As previously described, a major aspect of our first iterative cycle involved observing and interviewing our collaborating organization. In our initial analysis, our data suggests that the scope of the GDPR was far greater than expected and represented an immense undertaking for a startup organization. A large organization may be able to designate a group of employees to work predominantly on GDPR compliance work, but our collaborating organization (less than 50 employees) was restricted by the amount of resources it could allocate towards GDPR compliance. Furthermore, we noted in our observations and interviews that a developer was usually busy with various tasks, which further complicated understanding the GDPR, especially given the size of the GDPR. The benefit of CI towards facilitating rapid releases and quick feedback seemed to come at the cost of the thorough definition and characterization of an NFR. Notwithstanding a developer's exuberant effort, a developer is not likely to be a legal expert nor trained in privacy law. Hence, the constant affair of balancing different tasks and lack of GDPR training impeded our partner organization's ability to sufficiently comply. Moreover, upon

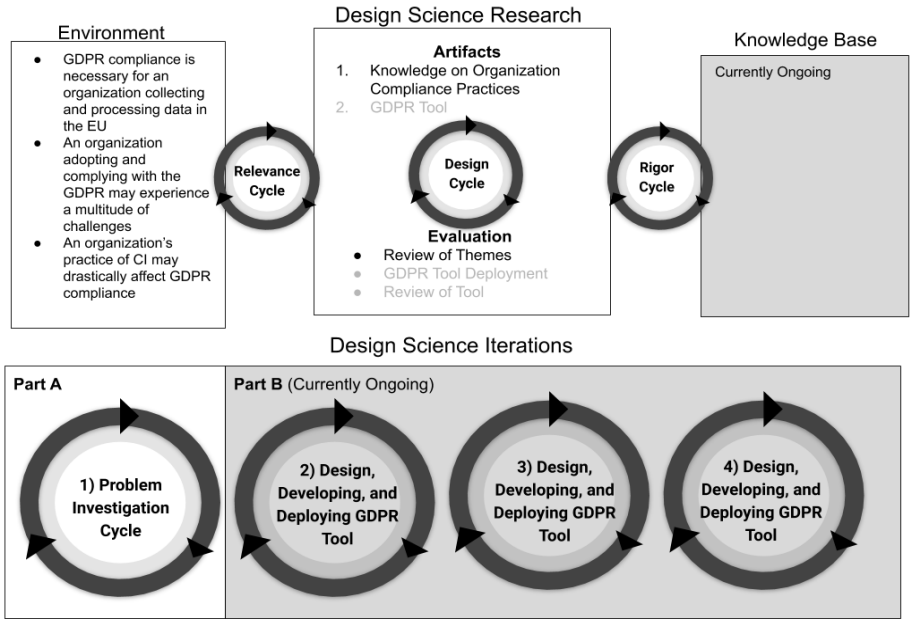

Fig. 1: Design Science Methodology

Note: The greyed parts of the Design Science Research diagram represent ongoing work. The Design Science Iterations diagram illustrates the four cycles of our research that will help produce the two artifacts

close investigation, we found that a developer typically had little influence or relationship to privacy in his or her work, yet the adoption and compliance to the GDPR did affect the same developer's work. This led to a disconnect between a developer's previous work and the developer's ideal GDPR compliant work. Although change was minimal for the vast majority of developers in our collaborating organization, the fact remained that almost every developer had to conduct some level of research for the GDPR. Furthermore, most developer agreed that finding a GDPR exposure in their system is difficult when considering the size of their system and the GDPR.

Our results shed light on the fact that challenges do exist in our collaborating organization's GDPR compliance, but also provide specific challenges that we can help tackle. Knowing that time is of essence for developers, it may be possible for our GDPR tool to help flag GDPR exposures so that a developer does spend an exorbitant amount of time researching and finding specific GDPR exposures in their system.

V. Conclusions

In this paper, we highlight our current study on how an organization deals with the GDPR, NFRs, and CI in practice. In particular, we highlight the need to further study those topics due to the enactment of the GDPR and increasingly widespread use of CI. We also proposed and discussed a design science approach that we currently conduct with a collaborating organization that practices CI. Furthermore, we provide details on some of our observations and interview results from our initial work. As described in Section III and Fig. 1, our approach has four iterative cycles split into two parts *A* and *B*. *Part A* had one iterative cycle and involved identifying how practitioners handle and comply with the GDPR. The first iterative cycle not only included studying practitioner practices, but also learning and identifying challenges experienced by practitioners when complying with the GDPR. In contrast, *Part B* relies on the findings in *Part A* and has three iterative cycles. The purpose of *Part B* is to iteratively design and build a GDPR that assuages some compliance challenges identified in *Part A*.

The current state of knowledge on the relationship between NFRs and CI indicates little NFR prioritization and definition. Yet, we are auspicious that our research will help bridge the knowledge gap between the GDPR, NFRs, and CI. We believe the discovered GDPR compliance challenges and our potential GDPR tool will be beneficial contributions to the research and industry community. Nonetheless, more work is necessary to investigate our initial findings.

References

[1] *Data protection in the EU*. en. Text. URL: https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en (visited on 04/06/2019).

[2] M. Glinz. "On Non-Functional Requirements". In: *15th IEEE International Requirements Engineering Conference (RE 2007)*. Oct. 2007, pp. 21–26. DOI: 10.1109/RE.2007.45.

[3] J. Eckhardt, A. Vogelsang, and D. M. Fernández. "Are "Non-functional" Requirements really Non-functional? An Investigation of Non-functional Requirements in Practice". In: *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. 2016, pp. 832–842. DOI: 10.1145/2884781.2884788.

[4] L. Cao and B. Ramesh. "Agile Requirements Engineering Practices: An Empirical Study". In: *IEEE Software* 25.1 (Jan. 2008), pp. 60–67. ISSN: 0740-7459. DOI: 10.1109/MS.2008.1.

[5] C. Gralha, D. Damian, A. I. T. Wasserman, M. Goulão, and J. Araújo. "The Evolution of Requirements Practices in Software Startups". In: *Proceedings of the 40th International Conference on Software Engineering*. ICSE '18. event-place: Gothenburg, Sweden. New York, NY, USA: ACM, 2018, pp. 823–833. ISBN: 978-1-4503-5638-1. DOI: 10.1145/3180155.3180158. URL: http://doi.acm.org/10.1145/3180155.3180158 (visited on 07/09/2019).

[6] C. F.i.S. o. May 16, 2018, and. A. Pst. *Only 36% of firms will be fully compliant with GDPR by its deadline*. en. URL: https://www.techrepublic.com/article/only-36-of-firms-will-be-fully-compliant-with-gdpr-by-its-deadline/ (visited on 04/06/2019).

[7] H. N. Security. *Only 20% of companies have fully completed their GDPR implementations*. en-US. July 2018. URL: https://www.helpnetsecurity.com/2018/07/16/complete-gdpr-implementation/ (visited on 04/06/2019).

[8] M. Fowler. *Continuous Integration*. URL: https://martinfowler.com/articles/continuousIntegration.html.

[9] F. Buschmann, D. Ameller, C. P. Ayala, J. Cabot, and X. Franch. "Architecture Quality Revisited". In: *IEEE Software* 29.4 (2012), pp. 22–24. ISSN: 0740-7459. DOI: 10.1109/MS.2012.77.

[10] L. Chung and B. A. Nixon. "Dealing with Non-functional Requirements: Three Experimental Studies of a Process-oriented Approach". In: *Proceedings of the 17th International Conference on Software Engineering*. ICSE '95. Seattle, Washington, USA: ACM, 1995, pp. 25–37. ISBN: 0-89791-708-1. DOI: 10.1145/225014.225017. URL: http://doi.acm.org/10.1145/225014.225017.

[11] M. Ambrosin, A. Compagno, M. Conti, C. Ghali, and G. Tsudik. "Security and Privacy Analysis of National Science Foundation Future Internet Architectures". In: *IEEE Communications Surveys Tutorials* 20.2 (2018), pp. 1418–1442. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2798280.

[12] G. Kalogridis, M. Sooriyabandara, Z. Fan, and M. A. Mustafa. "Toward Unified Security and Privacy Protection for Smart Meter Networks". In: *IEEE Systems Journal* 8.2 (2014), pp. 641–654. ISSN: 1932-8184. DOI: 10.1109/JSYST.2013.2260940.

[13] L. Chung and J. C. S. do Prado Leite. "On Non-Functional Requirements in Software Engineering". In: *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*. Ed. by A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 363–379. ISBN: 978-3-642-02463-4. DOI: 10.1007/978-3-642-02463-4_19. URL: https://doi.org/10.1007/978-3-642-02463-4_19.

[14] *What is GDPR, the EU's new data protection law?* 2019. URL: https://gdpr.eu/what-is-gdpr/.

[15] L. Chen. "Continuous Delivery: Overcoming adoption challenges". en. In: *Journal of Systems and Software* 128 (June 2017), pp. 72–86. ISSN: 01641212. DOI: 10.1016/j.jss.2017.02.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0164121217300353 (visited on 01/14/2019).

[16] L. Chen. "Towards Architecting for Continuous Delivery". In: *2015 12th Working IEEE/IFIP Conference on Software Architecture*. May 2015, pp. 131–134. DOI: 10.1109/WICSA.2015.23.

[17] L. Chen. "Continuous Delivery: Overcoming adoption challenges". en. In: *Journal of Systems and Software* 128 (June 2017), pp. 72–86. ISSN: 01641212. DOI: 10.1016/j.jss.2017.02.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0164121217300353 (visited on 01/14/2019).

[18] C. Parnin, E. Helms, C. Atlee, H. Boughton, M. Ghattas, A. Glover, J. Holman, J. Micco, B. Murphy, T. Savor, M. Stumm, S. Whitaker, and L. Williams. "The Top 10 Adages in Continuous Deployment". In: *IEEE Software* 34.3 (May 2017), pp. 86–95. ISSN: 0740-7459. DOI: 10.1109/MS.2017.86.

[19] N. A. Ernst. "On the Role of Requirements in Understanding and Managing Technical Debt". In: *Proceedings of the Third International Workshop on Managing Technical Debt*. MTD '12. Zurich, Switzerland: IEEE Press, 2012, pp. 61–64. ISBN: 978-1-4673-1749-8. URL: http://dl.acm.org/citation.cfm?id=2666036.2666047.

[20] W. Cunningham. "The WyCash Portfolio Management System". In: *SIGPLAN OOPS Mess.* 4.2 (Dec. 1992), pp. 29–30. ISSN: 1055-6400. DOI: 10.1145/157710.157715. URL: http://doi.acm.org/10.1145/157710.157715.

[21] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen. "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements". en. In: *Requirements Engineering* 16.1 (Mar. 2011), pp. 3–32. ISSN: 1432-010X. DOI: 10.1007/s00766-010-0115-7. URL: https://doi.org/10.1007/s00766-010-0115-7 (visited on 07/26/2019).

[22] J. Coles, S. Faily, and D. Ki-Aries. "Tool-Supporting Data Protection Impact Assessments with CAIRIS". In: *2018 IEEE 5th International Workshop on Evolving Security Privacy Requirements Engineering (ESPRE)*. Aug. 2018, pp. 21–27. DOI: 10.1109/ESPRE.2018.00010.

[23] S. Sirur, J. R. C. Nurse, and H. Webb. "Are we there yet? Understanding the challenges faced in complying with the General Data Protection Regulation (GDPR)". In: *arXiv:1808.07338 [cs]* (Aug. 2018). arXiv: 1808.07338. URL: http://arxiv.org/abs/1808.07338 (visited on 04/06/2019).

[24] A. Cavoukian. "Privacy by design: the definitive workshop. A foreword by Ann Cavoukian, Ph. D". In: *Identity in the Information Society* 3.2 (2010), pp. 247–251.

[25] Y. Martin and A. Kung. "Methods and Tools for GDPR Compliance Through Privacy and Data Protection Engineering". In: *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*. 2018, pp. 108–111. DOI: 10.1109/EuroSPW.2018.00021.

[26] A. Senarath and N. A. G. Arachchilage. "Why Developers Cannot Embed Privacy into Software Systems?: An Empirical Investigation". In: *Proceedings of the 22Nd International Conference on Evaluation and Assessment in Software Engineering 2018*. EASE'18. Christchurch, New Zealand: ACM, 2018, pp. 211–216. ISBN: 978-1-4503-6403-4. DOI: 10.1145/3210459.3210484. URL: http://doi.acm.org.ezproxy.library.uvic.ca/10.1145/3210459.3210484.

[27] S. Gürses, C. Troncoso, and C. Diaz. "Engineering privacy by design". In: *Computers, Privacy & Data Protection* 14.3 (2011), p. 25.

[28] S. Spiekermann and L. F. Cranor. "Engineering Privacy". In: *IEEE Transactions on Software Engineering* 35.1 (Jan. 2009), pp. 67–82. ISSN: 0098-5589. DOI: 10.1109/TSE.2008.88.

[29] P. Ferrara and F. Spoto. "Static Analysis for GDPR Compliance". In: *ITASEC*. 2018.

[30] D. Shah, L. Lindsay, J. Diaz, S. Shechter, and A. Becher. "IBM Security Guardium Analyzer Bootcamp". In: *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering*. CASCON '18. Markham, Ontario, Canada: IBM Corp., 2018, pp. 380–382. URL: http://dl.acm.org.ezproxy.library.uvic.ca/citation.cfm?id=3291291.3291349.

[31] A. R. Hevner, S. T. March, J. Park, and S. Ram. "Design Science in Information Systems Research". In: *MIS Quarterly* 28.1 (2004), pp. 75–105. ISSN: 02767783. URL: http://www.jstor.org/stable/25148625.