Applying Privacy by Design in Software Engineering - An European Perspective

Karin Bernsmed
Department of software engineering, safety and security
SINTEF ICT
Trondheim, Norway
karin.bernsmed@sintef.no

Abstract— **Privacy by Design (PbD) is an approach to protect privacy by embedding it into the design specifications of technologies, business practices, and physical infrastructures. However, despite its many advantages, many organizations struggle with incorporating these practices in their existing software engineering processes. This paper evaluates the current state-of-the-art related to PbD in software engineering and analyzes the impact of the proposed European data protection legislation on this process. We propose four key viewpoints of PbD and discuss how these can be applied in a software engineering process. We then translate these viewpoints into a self-assessment method that can be used to evaluate to what degree an organization has managed to adopt the PbD mindset in their software engineering projects.**

Keywords-privacy; PbD; privacy engineering; personal data; EU data protection law

I. INTRODUCTION

Privacy and personal data protection issues have been frequently in the news during the last few years, in particular in the context of social networking, big data and cloud computing. Consumer profiling by online advertising companies is a huge market and the loss of privacy is the price that consumers have to pay for the free services that they utilize. At the same time, the right to data protection is a highly developed area of law in Europe. Creating and maintaining software that is compliant with European data protection laws are therefore crucial for organizations that want to do business in Europe.

Broadly speaking, personal data means any kind of information that can be used to identify an individual. Some obvious examples include someone's name, address, national identification number, credit card number or a photograph. Less obvious examples are metadata in electronic documents, log files and system configurations and IP addresses. Personal data is not just information that can be used to identify an individual directly; information that can be used to single out a person from a group of people using a combination of information (or other identifiers) will also fall in the category of personal data. Almost all software that provides services targeted towards individual end-users will therefore collect personal data and hence be subject to applicable data protection law.

Privacy by Design (PbD) is an approach to protect privacy by embedding it into the design specifications of technologies, business practices, and physical infrastructures. PbD consists of seven foundational principles [1]:

1. *Proactive not Reactive; Preventative not Remedial*, which means to anticipate and prevent privacy invasive events before they happen.
2. *Privacy as the Default Setting*, to ensure that personal data are automatically protected in any given IT system or business practice. No action is required by the user – privacy is built in by default.
3. *Privacy Embedded into Design*, not bolted on as an add-on. Privacy becomes an essential component of the core functionality being delivered.
4. *Full Functionality — Positive-Sum, not Zero-Sum*, meaning that one seeks to accommodate all legitimate interests and objectives in a positive-sum "win-win" manner. The purpose is to avoid dichotomies, such as privacy vs. security or privacy vs. functionality.
5. *End-to-End Security — Full Lifecycle Protection*, to ensure that all data are securely retained throughout its entire lifecycle, and then securely destroyed at the end of the process, in a timely fashion.
6. *Visibility and Transparency — Keep it Open*, to assure all stakeholders that whatever the business practice or technology involved, it is in fact, operating according to the stated promises and objectives.
7. *Respect for User Privacy — Keep it User-Centric*, which requires architects and operators to keep the interests of the individual uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options.

PbD hence implies a proactive integration of technical privacy principles in the design of a system or software (such as privacy default settings or end-to end security of personal data) as well as the recognition of privacy in a company's risk management processes [2]. According to Ann Cavoukian, the Ontario Canada information and privacy commissioner who first coined the term, PbD can thus be defined as "an engineering and strategic management approach that commits to selectively and sustainably minimize information systems' privacy risks through technical and governance controls" [1].

PbD is often presented as the solution to the digital world's privacy problems. However, despite the obvious advantage with adopting the PbD approach, many organizations still struggle with how to incorporate these practices in their existing software engineering processes [3]. The seven principles are expressed in abstract terms and there are many open challenges that need to be addressed. *Privacy engineering* has emerged as a concept for transforming the PbD principles into a framework for

 ISBN: 978-1-61208-458-9

implementing privacy in system design and development processes. As concluded from the 2014 NIST Privacy Engineering Workshop [4], there is currently a communication gap around privacy between the legal and policy, design and engineering, and product and project management teams, which makes it difficult to understand and manage privacy risks. Moreover, there is a need for tools that measure the efficiency of existing privacy practices in organizations.

The purpose of this paper is to help organizations apply the Privacy by Design concept in their software engineering lifecycle by providing support for analyzing the current situation and practical guidance for building in PbD data protection practices that are compliant with European Data protection legislation. The paper is organized as follows. Section II summarized existing guidelines, tools and research related to engineering Privacy by Design. In Section III, we discuss the legislative aspects of PbD in Europe. Section IV outlines our approach to integrating PbD in a software engineering process and Section V presents a self-assessment method for PbD. Finally, Section VI concludes our work.

II. STATE OF THE ART

In this section, we summarize existing work related to PbD. We pay particular attention to the papers and reports that provide practical guidance on how to operationalize PbD, i.e., how to integrate the principles into existing software engineering processes. We also provide an overview over relevant ongoing research efforts in Europe.

A. Reports and Guidelines from the Software Industry

The report "Operationalizing Privacy by Design: A Guide to Implementing Strong Privacy Practices" from 2012 [5] gives a thorough introduction to the seven principles of PbD and provides practical advices for how each of the different principles can be implemented in an organization and by whom (i.e., the management, the application and program owners or the software engineers). For each of the PbD principles, the report also outlines a number of different case studies from different domains and explains how this particular principle has been implemented in practice. The report represents an overview over the work that has been performed at the Information and Privacy Commissioner in Ontario, Canada.

The OASIS Privacy by Design for Software Engineers (PbD-SE) Technical Committee has developed a draft specification to help document software engineering make privacy-informed decisions about a system's architecture. Their Privacy Management Reference Model and Methodology (PMRM) [6] intend to help system architects to analyze the system from a privacy point of view and to help them identify necessary technical and process mechanisms that must be implemented to support existing privacy policies in the organization. The methodology is based on defining and analyzing how actors and systems integrate in use-cases and the report contains a number of illustrative examples of how this can be done. PMRM is primarily specified with the Fair Information Practice Principles (FIPPs) [24] in mind, however it also supports parts of the PbD concept since it encourages building privacy in already from day one of a system design.

Microsoft's guidance document "Privacy Guidelines for Developing Software Products and Services" [7] includes an overview of basic concepts and definitions that are related to software security and provides guidelines for how the principles notice, choice, onward transfer (to third parties), access, security, and data integrity should be implemented. The document includes several practical examples (figures) showing how many of the concepts, for example explicit consent and opt-in, have been implemented in Microsoft's own software portfolio.

Finally, the position paper "Privacy Engineering & Assurance" written by NOKIA in 2014 [10] presents a process consisting of a set of proactive engineering activities. The activities include identifying the privacy impact of a given object, designing controls and mitigations to ensure appropriate Privacy by Design, and then verifying that the implementation is complete and operational, while documenting evidence of this state for reference of regulatory compliance and in the event of a privacy breach.

B. Relevant Research on Privacy by Design

The paper "Engineering Privacy" from 2009 by Spiekermann and Cranor [2] provides a structured overview over the different topics included under the term privacy engineering. The paper introduces the term privacy spheres to categorize the collection of personal data w.r.t whether the data are stored at the users' own devices under their own control (the so called "user sphere"), in back-end servers and networks under the service providers' control (the "recipient sphere"), or a combination thereof where users have some control over their personal data (the "joint sphere"). Spiekermann and Cranor recognize the necessity to consider the users' privacy expectations as well as possible regulatory issues when analyzing how system activities will impact privacy and they point out a number of different privacy issues that the needs to consider when designing IT systems. They also give some guidance for how to design a "privacy friendly" system, based on the degree of identifiability that will be required by its users, and provides some practical advices for how to maximize privacy for different types of systems. The paper also provides a nice overview over the existing research disciplines in the field of information system privacy.

The paper "Engineering Privacy by Design" by Gurses et.al, [3] points out data minimization as the necessary first step in order to create systems that are in line with the PbD concept. The authors point out the lack of concrete guidance of how to actually implement the PbD principles and they further argue that the FIPPs' focus on control and transparency, and the European data protection regulation's focus on purpose specification and user consent, are not sufficient to protect the individuals' privacy. The paper

 ISBN: 978-1-61208-458-9

presents two case studies where the authors show how privacy risks can be heavily reduced when data minimization is applied. They generalize their findings into five main steps for system design that should be taken to reduce privacy risks: 1) Functional Requirements Analysis (the necessary system functionality is clearly described), 2) Data Minimization (for each functionality, the data that are absolutely necessary to fulfil the functionality is analyzed), 3) Modelling Attackers, Threats and Risks (models of attackers and threats are developed, and the likelihood and impact of the threats are used to do a thorough risk analysis), 4) Multilateral Security Requirements Analysis (to ensure that the security and correct behavior of the system), and 5) Implementation and Testing of the Design (making sure that the system fulfils the integrity requirements revealing the minimal amount of personal data and that the functional requirements are fulfilled). Finally, the authors point out the need for experts trained in privacy engineering methodologies that also have a basic understanding of legal requirements related to personal data protection.

The paper "Privacy Design Strategies" by Hoepman [11] presents eight privacy design strategies, which are derived from legal requirements from the European data protection legislation. The strategies are derived both from a data oriented perspective (focusing on the principles minimize, hide, separate and aggregate) and from a process oriented perspective (focusing on the principles inform, control, enforce and demonstrate). For each of the eight strategies, the author has identified a number of privacy design patterns that can be applied to implement the strategies. The paper represents work in progress and the author state that further research will be performed to classify existing privacy design patterns into privacy design strategies, and to describe these design patterns in more detail.

Privacy by Design is also a topic of investigation in several ongoing European FP7 research projects; the most prominent being CIPHER [15], which will analyze security and trust in information systems that process personal data, and provide a methodological framework and a global European regulatory and technological roadmap, PRIPARE [16], which will deliver a privacy and security-by-design software and systems engineering methodology, A4Cloud [17], which will (amongst other things) deliver a Privacy Impact Assessment tool for cloud services and USEMP [18], which aim to empower users with control over the sharing of their personal data. In particular, PRIPARE is relevant to our work since they aim to deliver a methodology for Privacy and Security by Design that can be embedded into current methodologies for ICT systems and software [12].

Our analysis of the existing work in this section concludes that either the existing guidelines on PbD do not consider the strict EU personal data legislation in their guidance documents [2][3][5][6][7] or (implicitly) they assume that the organization that will operate the software develops its own software [4][10][11]. Even though there are promising ongoing research efforts, much work remains to be done. In particular, there is currently a gap of knowledge in how PbD can be built in the procurement phase of IT systems for organizations that engage consultancies or external software development companies and that have little or no knowledge of how to derive security and privacy requirements and how to impose such requirements on their software vendors. In the next section, we will present the main implications of the existing personal data protection legislation is Europe, before we proceed with presenting our approach for applying PbD.

III. PbD IN EUROPEAN DATA PROTECTION LEGISLATION

The processing of personal data in Europe is regulated by the implementation of the Data Protection Directive ("the Directive") [19], which ensures that personal data can only be collected and used legally under strict conditions, for a legitimate purpose, and that the data subject, who is an identified or identifiable natural person, must always be informed about the intention to collect and use his/her data. According to the Directive, the person, or organization, that is defined as the data controller, i.e., the entity that determines the purposes and means of the processing of personal data, will (in most cases) be held responsible and accountable to the data subject for ensuring that personal data are processed according to the rules in the Directive. Even though the Directive aims to protect the privacy of individuals, it only supports a limited part of PbD, and to a very limited extent. For example, as pointed out in the RAND report [8], while privacy policies are considered to be an acceptable way to meet the legislative requirements of obtaining consent and providing transparency, these policies are rarely read and even if they are, they appear to serve little useful purpose for the data subject due to their length, complexity and extensive use of legal terminology.

However, with the evolution of regulation, PbD has received more attention. In 2012, the Commission proposed a major reform of the EU legal framework on the protection of personal data (the "proposed Regulation") [9]. The European Commission has explicitly stated that the Proposed Regulation will embrace the concept of Privacy by Design [20]. Unfortunately, the current version of the Proposed Regulation is still quite general and vague. The most relevant part of the Proposed Regulation, from the PbD perspective, is its Article 23 - Data protection by design and by default. The first paragraph in this article states that "*the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organizational measures and procedures...*". Even though this statement indicates that privacy must be considered both when the system is to be designed ("the time of the determination of the means") and when it is operating ("the time of the processing itself"), nothing is said of how these requirements should be implemented in practice. Further,

 ISBN: 978-1-61208-458-9

the second paragraph of Article 23 states that "*The controller shall implement mechanisms for ensuring that, by default, only those personal data are processed which are necessary for each specific purpose of the processing...* " and "*... those mechanisms shall ensure that by default personal data are not made accessible to an indefinite number of individuals*". Here we note that, even though the "by default" part of PbD is supported, the Proposed Regulation does not aim to minimize the purpose of data collection at all; it merely states that the default setting should be to only process data for a specific purpose. This requirement already exists in the current EU Data Protection legislation. Further we note that the Proposed Regulation only points out the controller as being responsible for implementing these mechanisms. In many practical cases settings (for example when public cloud services are adopted) the controllers will not be involved in neither the design nor the implementation of the system. In our opinion, even though the European Commission has emphasized that the Proposed Regulation will support PbD; it is unclear to whether it will have any impact at all on existing software engineering processes. This view is also shared by Koops and Leenes, who argue that Article 23 cannot, and should not, "be read as a procedural requirement to embed data protection rules as much as possible in system design, but instead as a substantive requirement calling upon data controllers to consistently keep privacy at the front of their minds when defining system requirements" [14].

Even though PbD is vaguely described in Article 23, there are other parts of the Proposed Regulation, which will strengthen the rights of the data subjects. One example is Article 17, which emphasizes the data subjects right to "be forgotten", meaning that the controller must be prepared to erase all links to, copies of and replications of the data subject's personal data. Another example is Article 18, which specifies that a data subject has the right to obtain a copy of all his/her personal data that has been collected.

Awaiting the Proposed Regulation, several of the European Data Protection Authorities (DPAs) have started to promote the PbD concept, for example the British ICO [21] and the Norwegian Datatilsynet [22]. However, similar to the SOTA presented in Section III, there is a clear gap between the advices provided by these authorities and the concrete mechanisms that must be implemented in the software in order to be compliant with the Proposed Regulation.

IV. INTEGRATING PBD IN THE SOFTWARE SYSTEM ENGINEERING PROCESS

It is a non-trivial path for an organization with little knowledge of security and limited funds to go ahead and implement the best practices presented in Section II and the requirements that steam from the regulation presented in Section III. Very little research has been done to address the real world challenges of using the proposed methods in organizations, apart from the large software companies. This is especially the case where the organization has no dedicated software security or privacy group, which is often the case in, for example, SMEs and the public sector where few, if any, dedicated developers are employed. Instead, procurement and integration of commercially available (or open source) software into the enterprise architecture is more common, often along with custom built software components for integration and various functionality "plumbing". Consultants are commonly used for development and integration, making it hard to establish privacy and security engineering practices within the organization. In cases like these, the data management lifecycle, which spans from the moment personal data are gathered by the organization until the moment they have been destroyed (i.e., the retention period), is in the hands of the organization itself whereas the software engineering lifecycle, which spans from the early design phase until the software is fully installed and operating, is managed by the consultants.

Moreover, implementing PbD in the software engineering lifecycle is in itself a multidisciplinary exercise, comprising technical, organizational and legal concerns. A properly defined set of security and privacy policies must for instance be in place for application owners and developers to elicit specific sets of security and privacy requirements. On the other hand, true support is a matter of the management in the procuring organization, ensuring that the organization has the capabilities needed to accomplish its mission.

Figure 1. The stakeholders involved in the software engineering process.

An organization that wants to implement the seven PbD principles therefore needs to concretize them into a set of actions that the organization needs to consider internally, as well as into a set of well-formulated privacy requirements that they will need to impose on their consultancies and/or or software vendors during the analysis, design and implementation phases of the development of the software itself. This is a process that will need the involvement of a wide range of stakeholder (illustrated in Fig. 1).

When analyzing the seven PbD principles (from a software engineering point of view) and the different documents that were reviewed in Sections II-III, we have concluded that there are four distinct viewpoints of PbD,

 ISBN: 978-1-61208-458-9

which is top-down in the sense of involving both the organization as well as the actual software engineering process and that will require the involvement of the stakeholders identified in Fig. 1. These four viewpoints will be presented in the next section, along with an introduction to the accompanying PbD self-assessment tool that we have created.

V. A SELF-ASSESSMENT METHOD FOR PBD

The self-assessment method that we propose consists of four different viewpoints. First and foremost, we maintain that *privacy must be acknowledged in the organization*. This viewpoint implies that privacy must be taken seriously by the management and that a privacy mindset should be adopted by those who are responsible for the systems that process personal data. In our view, acknowledging privacy means, for example, that the organization has appointed a privacy officer who is accountable for privacy protection, a privacy policy has been established and approved by the management and that PIAs, or privacy risk assessments, are regularly performed within the organization.

Secondly, organizations need to be transparent about their privacy practices; any organization that processes personal data needs to inform the data subjects about the processing of their personal data. The *privacy policy* (or set of privacy policies) is the statement that discloses the details of what data will be collected, how it will be used and with whom it may be shared. The organization's privacy policy must be compliant with data protection legislation and it must be actively enforced in all its IT systems, including the software that is to be developed. Unfortunately, due to their complexity, difficult language and sheer length, users tend to neither read nor understand the policies prior to acceptance [8][13]. Having adopted a PbD mindset, the organization's privacy policy should therefore be clearly written and easy to access, contain no ambiguous language, and be as restrictive as possible in terms of how much data that will be collected and how it will be used.

Having acknowledged privacy in the organization and having a proper privacy policy in place are two fundamental cornerstones that the organization needs to have in place before the software system procurement phase starts. The former will ensure that sufficient attention and resources are put in place to protect privacy and the latter will serve as a basis for deriving appropriated privacy requirements when the software development process starts. These two viewpoints will need the involvement of business owners, regulators, 3rd parties and application developers.

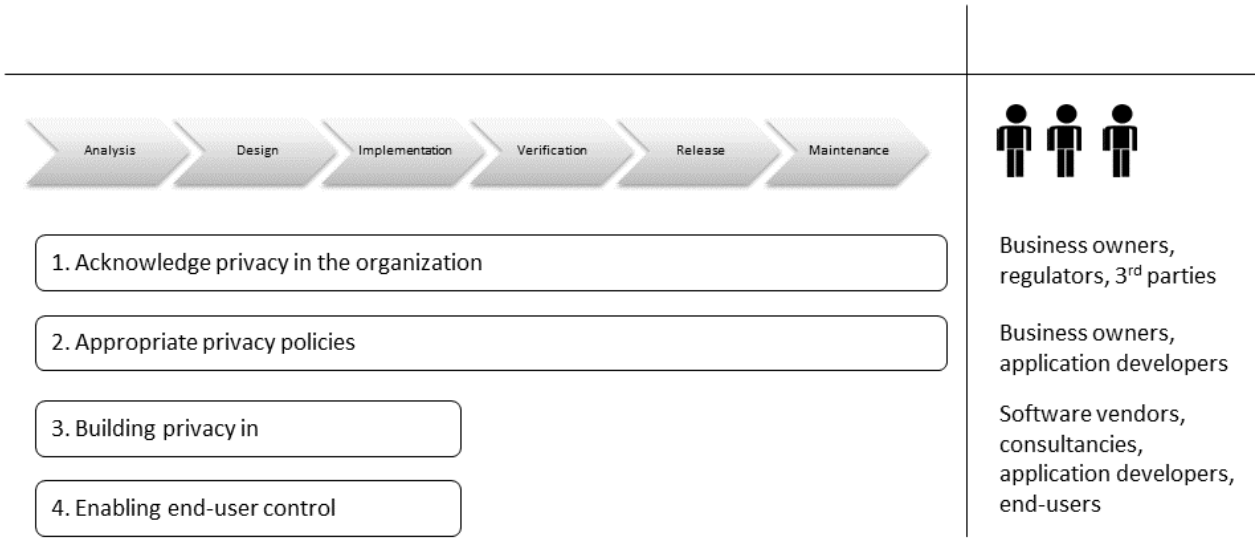

Figure 2. The role of the four viewpoints in the different phases of the standard waterfall software development process.

Once the software development processes has started, the third viewpoint, *building privacy in* is invoked. This viewpoint aims to ensure that privacy is integrated into the early phases of the software engineering process, in particular during the analysis, design and implementation phases. Software specific privacy requirements will be elicited from relevant stakeholders (business and application owners, regulators and the intended end-users), the privacy requirements must validated towards the organization's privacy policy and existing PbD best-practices are incorporated into the code by the software development team.

Finally, the fourth viewpoint *enabling end-user control* will ensure that the intended users of the software (i.e., the individuals who will be the data subjects of the personal data that will be collected) will be in control over his/her personal data. This viewpoint will ensure that the users are empowered with mechanisms to change their privacy settings, give and withdraw consent, and view, correct and delete personal data that have already been collected.

 ISBN: 978-1-61208-458-9

Fig. 2 illustrates how these four viewpoints relate to a standard waterfall software development process and what stakeholders that will be involved in each of the viewpoints. As indicated in the figure, acknowledging privacy and appropriate privacy policies are continuous processes that need to be in place before the software development activities starts and that will persist during the lifetime of the software. These processes will involve business owners, regulators, 3rd parties (with whom the data may be shared) and application developers in the organization. On the contrary, building privacy in and enabling end-user control consist of activities that will be accomplished during the analysis, design and implementation phase and that will involve the software vendors and prospective consultancies, the application developers in the organization and representatives of the end users who will be data subjects when the software is operating.

In the rest of this section, we present the self-assessment method, which has been organized as a checklist (Table I-IV) that has been derived from the four viewpoint introduced in the previous section. The checklist has gone through several iterations with security and privacy experts, before converging to 43 questions to be treated as recommendations, (i.e., answering "yes" is better than answering "no"). We then introduce a simple tool for analyzing the results of applying the checklist to an ongoing or finalized software project. Note that the tool itself is an adapted version of the security checklist for water network operators, originally developed by Jaatun et.al [23].

In our checklist, we have prepared three possible answers; "yes", "partly" and "no", however, it is of course also possible to use for example a sliding scale to indicate to what degree the organization that is being assessed is compliant with the different statements. We do not stipulate what methods the organization should apply to answer the individual checkpoints, but envision a combination of interviews, document analysis and testing as being an appropriate approach.

TABLE I. ACKNOWLEDGING PRIVACY IN THE ORGANISATION

Checkpoint	Yes	Partly	No
The organization has appointed a privacy officer, who is accountable for privacy protection			
A privacy policy has been established and approved by the management			
PIAs, or privacy risk assessments, are regularly performed within the organization			
Privacy audits are regularly performed within the organization			
Notice of personal data processing has been given to all the relevant DPAs			
Data processing agreements have been established with all 3rd parties that will process personal data			
The organizations' software and infrastructure regularly undergoes security risk and threat analysis			
The organization has a privacy education/awareness training program			
The organization is prepared to handle security incidents affecting personal data			

TABLE II. APPROPRIATE PRIVACY POLICIES

Checkpoint	Yes	Partly	No
The amounts of personal data that can be collect have been minimized			
The purpose for data collection has been defined to be as specific as possible			
Any sharing of personal data to 3rd parties has been clearly specified			
The retention date is no longer than necessary to fulfil the purpose of data collection (or to comply with existing legislation)			
The privacy policy clearly states who are responsible for the personal data and how they can be contacted			
The privacy policy is clearly written, to make it easy to understand by the intended end-users			
The length of the privacy policy is not excessive, but kept to a minimum			
The privacy policy can easily be retrieved by customers and end-user at all times			

TABLE III. BUILDING PRIVACY IN (SOFTWARE SUPPORT)

Checkpoint	Yes	Partly	No
Software specific privacy requirements have been elicited from relevant stakeholders (business and application owners, regulators and the intended end-users)			
The privacy requirements are consist with the organizations' privacy policy			
The privacy requirements have been incorporated in code developed by the software engineers			
The software only collect the personal data necessary to deliver its intended functionality			
The software includes appropriate mechanisms for obtaining end-user consent			

 ISBN: 978-1-61208-458-9

Checkpoint	Yes	Partly	No
The software has mechanisms in place to limit the use of personal data to the specific purpose for which it was collected			
The software has mechanisms in place to avoid future data linkage			
The software will encrypted all personal data by default using standardized encryption mechanisms with securely managed encryption keys			
All personal data are anonymized whenever possible			
There is an expiry date associated with all personal data that are collected			
All collected personal data will be properly deleted after they expire			
The software provides audit trails showing how personal data have been collected, processed and deleted			
The software has been subject to a thorough security risk and threat assessment			
The focus on privacy has not been traded against functionality			

TABLE IV. ENABLING END-USER CONTROL

Checkpoint	Yes	Partly	No
The default privacy settings in the software are as restrictive as possible			
The user can change the settings that control what kind of personal data are collected			
The user can change the settings that control for what purpose personal data are collected			
The user can view what personal data have been collected			
The user can view who has access to the personal data that will be collected			
The user can view who has accessed the personal data that have been collected			
The user can make corrections to personal data that have been collected			
The user can export a copy of all personal data that have been collected			
The user can request personal data to be immediately deleted			
The user's personal data is not shared with 3rd parties, unless the user specifically agrees to this ("opt-in")			
The user can choose not to share personal with 3rd parties ("opt-out")			
The user's privacy settings are valid across different platforms and persist over time			

If the checklist is used to evaluate a software engineering process that has already started, or software that is already operating, the answers can be visualized in order to show to what degree the PbD concept has been adopted. We have implemented a simple Excel-based tool and applied it to a case study that we are working on. The case study involves a public organization in Scandinavia, which currently is preparing a pilot study of the usage of cloud-based software for remote monitoring of health-care patients in their homes. Security and privacy are high on the agenda for this organization and since the software will collect large amounts of (sensitive) personal data, they need to be compliant to the existing privacy legislation in Europe, as well as to the upcoming privacy regulation, in order to succeed with their project. (For confidentiality reasons we are not allowed to reveal any technical details about the case study.) The result from the first viewpoint for this organization is illustrated in Fig. 3.

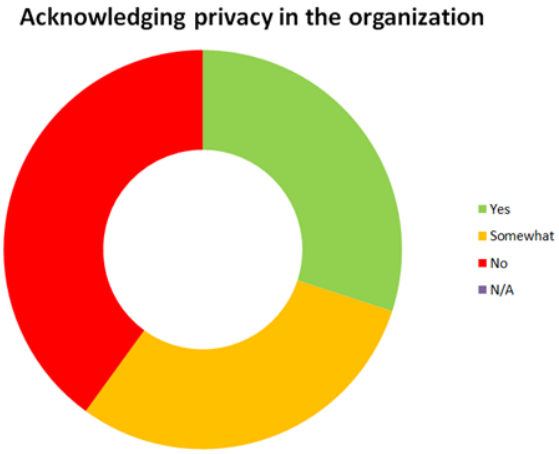

Figure 3. Visualizing to what degree privacy has been acknowledged in the organization

In the figure, the colors green, yellow and orange have been used to visualize the ratio of answers that have been selected as "yes", "partly" and "no", respectively. From the figure, we can see that, even though this particular organization have fulfilled some of the identified checkpoints, they still have a long way to go before privacy has been fully acknowledged.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented four viewpoints of Privacy by Design and our approach to translate these into a list of checkpoints. The intention of our approach is to clarify what the PbD concept means in a software engineering context. We also aim to help organizations that are involved in personal data processing to adopt a privacy mindset and to make sure that their software is compliant with the vision of PbD. In the next step, we will compile a best-practices document that includes existing privacy design patterns, strategies, mechanisms and tools, and map

 ISBN: 978-1-61208-458-9

these to the checkpoints in our self-assessment checklist in order to identify whether there are any gaps that current technology cannot fulfil. We believe that a combination of technical mechanisms (PETs) and organizational measures will be necessary in order to fully adopt the PbD concept.

ACKNOWLEDGMENT

This work has been partly funded from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement no: 317631 (OPTET) and 317550 (A4CLOUD).

REFERENCES

[1] A. Cavoukian, "Privacy by Design Curriculum 2.0", 2011. [Online]. Available from: https://www.ipc.on.ca/ [retreived: 2015-10-29]

[2] S. Spiekermann, and L. Faith Cranor, "Engineering Privacy". IEEE Trans. Softw. Eng. 35 (1), pp 67-82, January 2009, doi=10.1109/TSE.2008.88

[3] S. F. Gürses, C. Troncoso, and C. Diaz, "Engineering Privacy by Design", Computers, Privacy & Data Protection, 2011. [Online] Available from: http://www.cosic.esat.kuleuven.be/publications/article-1542.pdf [retreived: 2016-01-12].

[4] NIST Privacy Engineering Objectives and Risk Model Discussion Draft, April 2014. [Online]. Available: http://www.nist.gov/itl/csd/upload/nist_privacy_engr_objectives_risk_model_discussion_draft.pdf [retreived: 2016-01-12].

[5] A. Cavoukian, "Operationalizing Privacy by Design: A Guide to Implementing Strong Privacy Practices". Information and Privacy Commissioner, Ontario, Canada, December 2012.

[6] OASIS. "OASIS. Privacy Management Reference Model and Methodology (PMRM) Version 1.0.", March 2012. [Online]. Available from https://www.oasis-open.org/ [retreived: 2016-01-12]

[7] Microsoft. "Privacy Guidelines for Developing Software Products and Services, Version 3.1", September, 2008. [Online]. Available: http://www.microsoft.com/en-us/download/details.aspx?id=16048 [retreived: 2016-01-12]

[8] N. Robinson, H. Graux, M. Botterman, and L. Valeri, "Review of the European Data Protection Directive", 2009, RAND Corporation. [Online]. Available: http://www.rand.org/pubs/technical_reports/TR710.html [retreived: 2016-01-12]

[9] Proposal for a Regulation of the European Parliament and of the Council on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation), COM (2012) 11 final (25 January 2012).

[10] NOKIA, "Privacy Engineering & Assurance. The Emerging Engineering Discipline for implementing Privacy by Design", NOKIA Position Paper 1/10, 2014-09-08. [Online]. Available: http://www.w3.org/2014/privacyws/pp/Hirsch.pdf [retreived: 2016-01-12]

[11] J-H. Hoepman, "Privacy Design Strategies". ICT Systems Security and Privacy Protection, IFIP Advances in Information and Communication Technology Volume 428, 2014, pp 446-459, 2014.

[12] "PRIPARE: A New Vision on Engineering Privacy and Security by Design", PRIPARE position paper, April 2014. [Online] Available from: http://prip areproject.eu/research/ [retreived: 2016-01-12]

[13] L. Faith Cranor, P. Guduru, and M. Arjula, "User interfaces for privacy agents". ACM Trans. Comput.- Hum. Interact., 13(2):135–178, 2006.

[14] B-J. Koops and R. Leenes, "Privacy regulation cannot be hardcoded. A critical comment on the 'privacy by design' provision in data-protection law". Int. Rev. Law Comput. Technol. 28 (2), May 2014, pp. 159-171. doi=http://dx.doi.org/10.1080/13600869.2013.801589

[15] "CIPHER: Integrated Cybersecurity framework and roadmap". [Online] http://cipherproject.eu/ [retreived: 2016-01-12]

[16] "PRIPARE: Preparing Industry to Privacy-by-design by supporting its Application in Research". [Online] http://prip areproject.eu/ [retreived: 2016-01-12]

[17] "Cloud Accountability project". [Online]. http://www.a4cloud.eu/ [retreived: 2016-01-12]

[18] "USEMP: User Empowerment for enhanched online management". [Online]. http://www.usemp-project.eu/ [retreived: 2016-01-12]

[19] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

[20] European Commission, "Progress on EU data protection reform now irreversible following European Parliament vote", Strasbourg, 12 March 2014. [Online]. Available: http://europa.eu/rapid/press-release_MEMO-14-186_en.htm [retreived: 2016-01-12]

[21] Information Commissioner's Office (ICO), "What is 'privacy by design'?" [Online] Available: https://ico.org.uk/for-organisations/guide-to-data-protection/privacy-by-design/ [retreived: 2016-01-12]

[22] The Norwegian Data Protection Authority. http://www.datatilsynet.no/English/ [retreived: 2016-01-12]

[23] M.G. Jaatun, J. Røstum, S. Petersen, and R. Ugarelli, "Security Checklists: A Compliance Alibi, or a Useful Tool for Water Network Operators?", Procedia Engineering, Volume 70, 2014, Pages 872-876, ISSN 1877-7058, http://dx.doi.org/10.1016/j.proeng.2014.02.096.

[24] "Privacy Online: A Report to Congress. Federal Trade Commission", June 1998. [Online]. Available from: https://www.ftc.gov/reports/privacy-online-report-congress [retreived: 2016-01-12]

 ISBN: 978-1-61208-458-9