TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

MIKAEL VIITANIEMI

PRIVACY BY DESIGN IN AGILE SOFTWARE DEVELOPMENT

Master of Science Thesis

Examiners:
prof. Hannu-Matti Järvinen,
university teacher Marko Helenius
Examiner and topic approved on 4 January 2017

ABSTRACT

MIKAEL VIITANIEMI: Privacy by Design in Agile Software Development
Tampere University of Technology
Master of Science Thesis, 49 pages
November 2017
Master's Degree Programme in Information Technology
Major: Pervasive Systems
Examiner: professor Hannu-Matti Järvinen, university teacher Marko Helenius

Keywords: privacy, privacy by design, agile, scrum, GDPR, regulation, transparency, data protection, data security, General Data Protection Regulation

With privacy concerns on the rise, the European Commission passed the General Data Protection Regulation (GDPR) which forces all software manufacturers to employ the privacy by design principles starting from the design phase of development. The privacy by design approach has been pushed into regulation as the ultimate solution by some, but very little information is given on applying the approach in practice. Very little information is also available on enforcement of the regulatory side of privacy by design which makes evaluation of compliance difficult.

This thesis explores the state of privacy by design implementation and attempts to formulate a model for adhering to the privacy by design principles in an iterative agile software development methodology. This model is fully integrated into the Scrum software development model and provides the developers with an improved view into the compliance state of their product during development through employment of visual documentation practices. Additional focus is given to other regulatory demands of the GDPR. Compatibility with other privacy oriented development frameworks is also considered.

Furthermore, this thesis explores the criticism and benefits on privacy by design from both an implementation and regulatory point of view in Europe and in other jurisdictions. These criticisms and benefits are evaluated against the agile integrated model. The state of privacy by design in the global privacy community is a positive development, but some global privacy threats are also discussed.

TIIVISTELMÄ

MIKAEL VIITANIEMI: Privacy by Design in Agile Software Development
Tampereen teknillinen yliopisto
Diplomityö, 49 sivua
Marraskuu 2017
Tietotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Pervasive Systems
Tarkastaja: professori Hannu-Matti Järvinen, yliopisto-opettaja Marko Helenius

Avainsanat: yksityisyys, sisäänrakennettu tietosuoja, ketterä kehitys, scrum, GDPR, sääntely, avoimuus, tietosuoja-asetus

Huolen tietosuojan tasosta kasvaessa Euroopan Komissio hyväksyi yleisen tietosuoja-asetuksen, joka pakottaa kaikki ohjelmistovalmistaja noudattamaan sisäänrakennetun tietosuojan periaatteita alkaen jo kehityksen suunnitteluvaiheista. Sisäänrakennetun tietosuojan lähestymistapaa on ajettu asetuksen ja sääntelyn piiriin lopullisena ratkaisuna tietosuojaongelmiin, mutta hyvin vähän informaatiota on tarjolla lähestymistavan käytännön hyödyntämisestä. Niin ikään hyvin vähän tietoa on tarjolla asetuksen toimeenpano- ja valvontapuolesta sisäänrakennettuun tietosuojaan liittyen, mikä tekee asetuksen noudattamisen arvioinnista hankalaa.

Tämä diplomityö tutkii sisäänrakennetun tietosuojan toteutuksen tilaa ja pyrkii muodostamaan mallin jonka avulla sisäänrakennetun tietosuojan periaatteiden noudattaminen ketterässä ohjelmistokehityskehyksessä olisi mahdollista. Tämä malli on täysin integroitu Scrum-ohjelmistokehitysmenetelmään ja tarjoaa kehittäjille paremman näkyvyyden tuotteen määräystenmukaisuuteen läpi kehityksen visuaalista dokumentaatiomenetelmää käyttäen. Lisäksi malli tarjoaa etuja muiden yleisen tietosuoja-asetuksen vaatimusten noudattamiseen. Yhteensopivuus muiden tieto- ja yksityisyydensuojaan suuntautuneiden kehityskehysten kanssa huomiodaan myös.

Lisäksi tämä diplomityö tarkastelee sisäänrakennettun tietosuojaan kohdistuvaa kritiikkiä ja kehuja täytäntöönpanon sekä sääntelyn kannalta Euroopassa ja muissa hallintoalueissa. Tätä kritiikkiä ja kehuja vertaillaan tuotettua ketterää mallia vasten. Sisäänrakennetun tietosuojan kehityssuunta globaalissa tietosuojayhteisössä on positiivinen, mutta työ pohtii myös joitakin globaaleja uhkia tietosuojaa kohtaan.

PREFACE

This Master's thesis is the result of a long-term development in the data protection regulation. It has been difficult at times to follow the developments in the industry as the information about the regulation has been very scarce and intermittent, but the overall benefits to the modern society have been a huge source of motivation to keep learning more about the issues that global data protection faces.

I would like to thank the examiners Hannu-Matti Järvinen and Marko Helenius for their valuable input while I was forming the idea of how to execute this study and their feedback of the thesis in general. I am also grateful to my employer Futurice for providing the incentive to pursue this learning and for the continued support and feedback from my amazing colleagues. Lastly, I am thankful for the support my family and friends provided during the difficult time of working through the thesis.

Tampere, 15.11.2017

Mikael Viitaniemi

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ENISA	European Network and Information Security Agency
GDPR	European General Data Protection Regulation
ISO	International Organization for Standardization
LINDDUN	Privacy threat modelling framework, consisting of linkability, identifiability, non-repudiation, detectability, disclosure of information, unawareness and non-compliance [11]
OWASP	Open Web Application Security Project
PbD	Privacy by Design
STRIDE	A generic threat modelling framework, consisting of spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege

1. INTRODUCTION

Privacy is likely one of the most infringed basic human rights in the modern ubiquitous information society. A lot of public discussion has been revolving around the privacy threats of social media applications, always connected Internet of Things -gadgets and promiscuous handling of personal information, such as location based data, as a commodity. In a 2013 study, Rainie et al. found that only 24% of internet users said they are happy with the level of protection that current legislation offers to their online privacy [1] and a 2014 study by Rainie found 91% of American adults stating they had lost control over how personal information is collected and used [2].

Privacy by Design has been in the spotlight a lot recently with the passing of the European General Data Protection Regulation (GDPR). It has reached almost a mythical standing as the one true answer to privacy engineering in certain outlets. At the same time, concrete guidelines for the application of privacy by design have been very scarce. There seems to be an apparent disconnection between the ideology and practice of privacy by design.

The goal of this thesis is to explore the state of privacy by design in practice and its applicability in a software project using agile development methodologies. The approach to this goal will be through the study of existing research on the application of the privacy by design method, its problems and criticism, the regulatory aspects relating to it and the related privacy research. Based on this study, a model for the implementation of the privacy by design method in an agile workflow is created and assessed.

In terms of research questions, the following questions are set forth:

- What is the current state of privacy by design in the industry and regulation?
- Is it possible to apply privacy by design iteratively in an agile software development model?
- How relevant is the privacy by design as a privacy engineering method?
- What is the privacy threat landscape like currently?

To facilitate the exploration of these questions, in Chapter 2 the methods and practices of contemporary agile software development approaches are presented as background. Chapter 3 dives into the theory of privacy as the relevant data protection and privacy terminology is defined, the privacy by design method is opened and examined, the standing of the privacy by design method in a regulatory context relating to the GDPR is studied and the relevant privacy risks are analysed. Chapter 4 sees the agile privacy by design model and the assessment requirements behind it defined and inspected. In Chapter 5, the

research questions are evaluated critically in light of what has been found during the thesis. Finally, Chapter 6 summarizes the conclusions of the thesis.

2. AGILE SOFTWARE DEVELOPMENT

2.1 Agile development

Traditionally, software engineering projects were approached in the past using waterfall-style methodologies like many other engineering sciences projects. Characteristic for these projects is that there is a design and planning phase, and an execution or building phase, which follow chronologically with minor overlap. This means that after the plans are made, they are more or less fixed and the execution simply follows those plans with at most minor changes. In theory, this allows planning different aspects of the execution far into the future, such as material deliveries, concurrent processes or employee schedules. Ironically, anecdotal evidence has shown the opposite even in traditional manufacturing [3].

A much more realistic approach would be accepting the volatility of circumstances as a fact and using a methodology that embraces change [3]. This is the agile approach. Being agile means accepting that software engineering, information technology and business related to them are such fast-developing fields that a solution that was considered stable can become obsolete the next month as a much better technology is released or a new competitor enters the market with a ground-breaking product. As such, making far reaching plans that need to be completely revised when they become current is simply wasteful [3].

The core ideology of most agile methodologies is based in the agile manifesto, which is a set of value statements. The four statements are “individuals and interactions over processes and tools”, “working software over comprehensive documentation”, “customer collaboration over contract negotiation” and “responding to change over following a plan” [4]. These statements are also a social contract for the development team to drive the continuous improvement that makes the agile approach so special.

At the most granular level of implementation there are a multitude of different agile practices, which are activities, policies or ways of working that are commonly considered to be core building blocks to implementing the agile ideology [5]. Many agile methodologies share some of these practices and leave out others. Which practices a methodology chooses to include generally depends simply on the aspects of the project that the methodology emphasizes. Even if a practice is not core to some specific methodology, being agile means that if it is beneficial it may be worth adapting the methodology and using it anyway.

Another view of agile is that it is the practice of “continuous risk management” [5]. In a traditional waterfall model, the risk mitigation takes place mostly during the planning

phase, and the execution phase consists of taking the planned actions with hopes that they are enough to avoid the identified risks. The agile approach however, embraces risk and actively combats it through continuous demonstrations of the direction the project is heading [5]. The incremental steps allow making corrective measures if it turns out the project is headed towards the realization of an identified risk, and as decisions are not committed until there is enough information to support them, it is possible to make even large changes [3]. The agile approach also supports risk discovery through continuous communication and repeating reviews, all of which are an opportunity to steer the project away from potential pitfalls.

2.2 Scrum

Scrum is an iterative, feedback driven agile model for teams to organize around software development. The Scrum model is based on a repeating set of periodic activities that range from daily to spanning multiple weeks in duration. These activities have a defined structure and goals and their frequent repetition allows the issues that arise during the product development cycle to be tackled immediately and for the solutions to be incorporated into future cycles. [6] Figure 1 shows an overview of this iterative process.

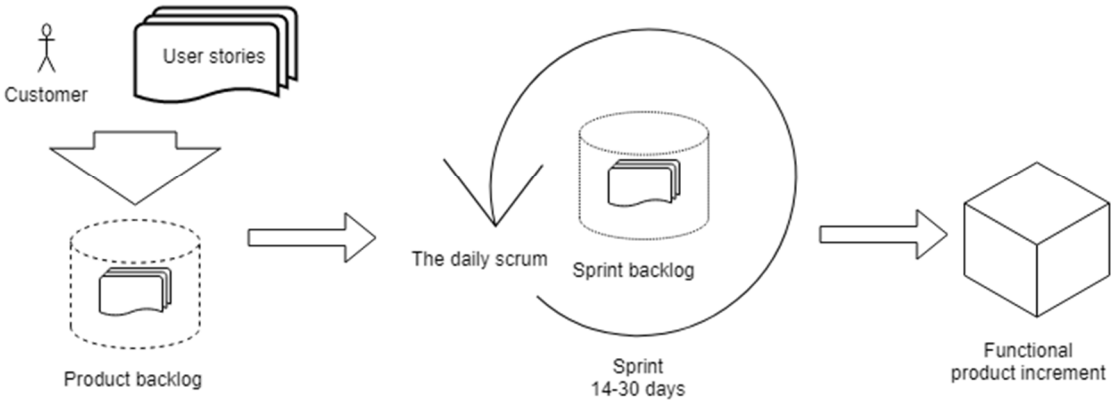

Figure 1. *Overview of the Scrum process*

2.2.1 An overview of the Scrum process

At the start of a new project, the product backlog is initialized by identifying the expectations placed on the product-to-be [6]. Usually expressed in the form of user stories as shown in Figure 1, they become items on the product backlog. The product backlog details the features to be worked on during the future development cycles [6]. These expectations can be functional or performance metrics, but also design limitations or requirements external to the product itself, if there is a strong argument for them so early in the product lifecycle. The product consists of deliverable items which represent the tangible future outcome and the scope of the project.

Transforming a rough idea of the deliverable items into a product backlog often involves multiple avenues of customer input [6]. A large amount of this input takes place at the start of the product development, but it does not end at that. The product backlog continues evolving throughout the development lifecycle of the product, as it needs to represent the changes happening in the understanding of the product requirements. New items can be added to the backlog or removed if they become invalid. The stories behind the items can be vague at first, and as the development of the product advances, more detailed input allows the stories, and thus also the backlog items, to become more detailed and atomic [7]. This process is often referred to as product discovery.

Once the backlog has a sufficient number of items of varying granularity they are prioritized. Prioritization can be done based on a multitude of criteria, but the main drivers should be to have the most important business-critical core requirements and their dependencies as top priorities. This process interlinks with product discovery to an extent, as the items with more granular and detailed stories should be the ones that will be more highly prioritized. To keep the process lean, the low priority items do not need to be very detailed and too much work should not be spent on specifically detailing them before they are relevant [7]. This way the decision making is focused on items that are currently being worked on and making decisions about design details as late as possible helps avoid having to make decisions with partial information.

After the product backlog is initialized, it is time to begin the iteration through sprints. A sprint is a short timeboxed working period with its own backlog, a subset of the product backlog, whose scope is fixed at the start of the period [6]. The scope should be small enough that the development team can realistically commit to completing all the items on the sprint backlog in the allotted time. The items for the sprint backlog are prepared beforehand by selecting a set of items from the product backlog that are broken down into smaller, workable stories. This preparation is referred to as grooming and it resembles the initial forming of the product backlog, but the goal is to “prepare just enough items for the upcoming spring, just in time” [7]. The actual content of the sprint backlog is agreed on in a sprint planning meeting with the development team before the start of the sprint.

During the sprint, the project team gathers in the daily Scrum meeting. The Scrum meeting is a short stand-up meeting, where each team member spends a few minutes describing what they accomplished the previous day, what they intend to do today and if there are any obstacles preventing them from advancing as intended [6]. If any issues arise, the manager’s priority should be working with the team to remove them. The meeting should be short and the focus is on updating the team on the latest developments, not reporting to management.

At the end of the sprint, the accomplishments are presented to the team and other relevant stakeholders in a sprint review meeting. In the sprint review, the completed work items are presented and reviewed to determine if they are functional and completed to the extent

of the definition of done for the project. Incomplete items are returned to the product backlog. [7]

More importantly, however, the team organizes a sprint retrospective meeting, which is an opportunity to critically review the performance of the previous sprint and plan actions to improve the situation for the next sprint [6]. It is an opportunity to learn what worked and what did not, and to react to changes in the environment or process that affect the team performance before they have too much of an impact on the product delivery schedule. The sprint review and retrospective can be the same meeting, but in practice it is more efficient to timebox both meetings separately as they have a different agenda and focus. This also ensures one meeting does not encroach on the other.

The end of a sprint also marks the beginning of planning for the next one. This process repeats iteratively until the product backlog has been consumed or the product otherwise meets the criteria of the definition of done for all its deliverables [7]. Of course, this definition can also change during development and it should be reflected like any other changes discovered during the process.

It is often customary to bootstrap a new Scrum project by having a so-called sprint zero. The sprint zero is an opportunity to agree on the project framework details, ways of working and other project level details. The definition of done is one example of something that would be considered during sprint zero.

2.2.2 Scrum project management

Scrum defines two special roles in the team. First, the product owner is the person who is responsible for the content and prioritization of the product backlog [6]. They act as an avatar of the customer towards the project team. The second, Scrum master, is the person who is in charge of executing the Scrum activities. It does not mean that they do everything, but that they make sure it gets done. The Scrum master facilitates meetings, keeps the team to the process and acts in a manager role to remove obstacles that hinder the team's ability to perform [6].

The strength of Scrum comes from embracing change even on the process level rather than attempting to contain and minimize it. Instead of committing to a plan and a process that was made based on information that might be months or years old at worst when the time to act upon it comes around, Scrum encourages not making such detailed plans before the information is available [6]. This should not be mistaken for long term plans being completely forbidden, but rather taken as a stern reminder that the project environment is likely to evolve, sometimes very considerably, during the development cycle of a product.

The flexibility that Scrum and agile offer can become a shortcoming under certain circumstances. One of the more common worries with adopting Scrum has been recorded as the level of independence and expertise that it requires from the team to run the Scrum process without undue effort [5]. Improper application of the agile practices or seemingly applying them without a proper buy-in from the team can also lead to inefficiency and an unpleasant experience [7]. Mistaking agile or Scrum with the lack of need for planning can be another common misconception. On the contrary, in order to successfully commit and complete a timeboxed sprint, the few things that are planned need to be planned properly [5].

2.3 Other notable agile models

While the development process described in this thesis focuses on Scrum specifically, it is important to acknowledge that in addition to Scrum, there are other ways of working that have their core ideologies based in the Agile Manifesto. Such methods are for example extreme programming, Kanban and Lean. Much like in Scrum, continuous improvement is important to the core process of each one and their core ideologies are very interchangeable. The process described in this thesis for Scrum should be applicable to an extent in other agile models as well, as the agile practices which enable each model to behave like it does can be brought into other models as well.

2.3.1 Extreme programming

Like Scrum, extreme programming is timeboxed through sprints or iterations with a product backlog that is initialized at the beginning of the process. This is called an exploration phase where the team experiments with architectural choices and the backlog items are groomed [5].

After the exploration phase, comes the planning phase [5], which resembles the sprint planning of Scrum. The difference is that multiple iterations are planned ahead through estimation of when each backlog item could be addressed. In conjunction with the release planning, test scenarios are created to describe what is an acceptable product increment.

After the planning phase, the team enters an iteration cycle where they work through the planned iterations towards the product release plan. After every iteration, the release plan is reviewed and updated with the latest findings, such as new feature requirements and bugs. Eventually, the team reaches an iteration after which the product can be accepted for testing, and through the productionizing phase a release increment is produced. [5] This differs slightly from the Scrum ideal of having a working increment at the end of every sprint. Of course, both models in practice can resemble each other more than this due to team specific process modifications.

2.3.2 Kanban

Kanban originates in the 1940s factory processes of Toyota. The core idea of Kanban in software development is to treat development issues like raw stock moving through a manufacturing process. Each issue is assigned to a visual ticket which is moved through the development pipeline on a board as it progresses through various stages and teams. This board provides an instantaneous glance to the overall status of the development process. Applying limits on the amount of issues that can exist in the same stage at one time then allows to identify bottlenecks and fix them early in the process. [3]

Using Kanban with a product backlog is very similar to how Scrum is used in practice. The biggest difference is that sprints are used as timeboxed units of work in Scrum while Kanban revolves more around a continuous flow of work. Both models are largely compatible and it is often seen that the Kanban board is used to visualize work items in a Scrum sprint.

2.3.3 Lean

Lean software development takes the learnings from Toyota's product development system, also known as lean manufacturing, and applies them to the software development process. The lean system focuses on placing effort where it counts at the right time and reducing wasteful, unnecessary work or work that may need to be redone later. The lean system consists of seven principles: eliminate waste, amplify learning, decide late (often also expressed as just-in-time), deliver fast, empower people, build for integrity and consider the whole [3].

In the lean terminology, waste is used to refer to everything that does not immediately increase the value of the end product, as higher customer value directly translates to a better product [3]. Waste can be obvious things, such as making plans or designs that do not get implemented or working on features that are not required at the time, but also more integral things in the development process like multiple hand-offs between teams. The goal in the lean methodology is to make the right product with minimal waste. One key aspect to minimizing waste is to avoid making decisions that require heavy commitment equipped with only partial knowledge. This is very much like the agile ideology in general.

Software development is often problem solving at its core. To amplify learning is to enable the exploration of various solutions to a problem, before making the choice on the final method. Regardless of the exact problem solution techniques that go into arriving at the solution, allowing time for understanding the problem space is a key aspect of working towards solving a problem effectively [3]. Each potential solution is an opportunity to learn more, and the accumulated knowledge will result in a better product in the end.

Delaying decisions until they are required is key to the Scrum method in addition to Lean. The world will likely change between the introduction of a problem and the discovery of a good solution. Thus, it is only reasonable to wait until decisions do not require long-term predictions or commitments [3]. This requires preparing by building in room for change to minimize the waste produced when it eventually is required.

Continuous delivery and team-oriented processes are also very core agile practices to the Scrum process. Integrity refers to the experience of how well a product fulfils the non-verbal intimate requirements of the user and how smoothly the system works together at a conceptual level [3]. In a way, it is a measure of quality, where quality is a property that is observable only by using a product for a purpose.

Finally, considering the whole is required to keep the direction of the system coherent. It is also required to balance the effort between different aspects of development. Without careful oversight, individuals and organizations are inclined towards focusing on the area of their own expertise and neglecting other aspects of a product [3].

These principles could easily be applied to any agile model on their own or as a whole. As such, the Lean principles are not exclusive to a specific methodology but rather a set of tools, like the agile practices. They can be used if the team feels they are beneficial to the process.

3. DATA PROTECTION AND PRIVACY

3.1 A definition of privacy

Before privacy, its protection and the impact of technologies and methodologies is discussed, a sufficient understanding must be achieved on what privacy is and what the main terms related to the field are. There are multiple definitions of what privacy entails depending on the field and context of approach. The European Convention on Human Rights states “the right to respect for his private and family life” as a universal right, but does not go further into what privacy of one’s life means [8]. One broad domain definition comes from Solove [9] as “the right for a person to be let alone”. This definition may be useful in a legislative or philosophical context, but a little too imprecise and bound in the physical world for this discussion. Also from Solove, however, comes the definition that privacy is the ability to limit the scope of personal information available to others [9]. This definition is much more workable in the context of data protection and information technology.

Personal data is usually described as any information relating to an individual who is identifiable based on the given information [10], but the definition does not necessarily have to be limited to a natural person. Wuyts [11] points out an issue in this definition of personal data: under this definition, data which has been rendered unidentifiable warrants no protection. It is worth questioning if anonymous data constitutes such information, that it might be the subject of a breach of privacy, but that is a rather philosophical debate. It is however very unlikely, that in the era of big data and the ubiquitous information society meaningful data could be reliably made unidentifiable such that it cannot become linkable again [11]. Aggregation is one applicable method, but arguably the aggregated data is not the same as the source data.

Based on this definition, data protection can then be taken as the means and acts by which the individual’s control over their personal information is enabled. These means are mainly governed by data protection legislation which describes the responsibilities imposed on and rights allowed to those who are granted legitimate access to personal information by an individual directly or by proxy. Data protection encompasses both technical and non-technical measures which mitigate the threats that dealing in personal data present to the privacy of an individual.

In a perfectly private world, an individual would have complete and ultimate control over their personal information. This is not realistically possible, and exceptions to an individual’s right to control their personal information usually exist as governed by legislation, for reasons such as government databases (i.e. criminal or census registries) for example.

In other words, deconstructing the concept slightly, privacy enables anonymity. Anonymity can be defined through its opposite: identifiability, which means that the data subject can be identified from a group of potential subjects [12]. Identifiability and anonymity are not static states, but vary based on observable data and the observed group of subjects. It is safe to assume that the amount of observable information only increases, as it is unlikely for a malicious adversary aiming to break the individual's privacy to forget anything [12]. Because of this, the highest possible level of anonymity in a system is always in the initial state.

A concept related to anonymity is linkability of data. Linkability of two distinct sets of data means that an observer can determine with reasonable confidence the data to be describing the same individual or to be otherwise related, e.g. to the same transaction or role [12]. As an adversary is unlikely to forget key information, linkability is a particularly dangerous attribute especially from the perspective of data minimization which in turn is the most effective tool to preserve or enhance privacy depending on the perspective.

Linkability of data can occur regardless of whether the contents of it are observable. For example, given a messaging system where the online-status of users is disclosed to their contacts accurate to the minute, a malicious user might follow the status changes of two users to determine correlation between their usage of the system. Thus, the malicious user can determine the users involved in the exchange of a message based on unrelated data or metadata provided by the system for a different purpose, without having access to the message itself or the network traffic related to its delivery.

Pseudonymity is a slightly weaker state than anonymity. Pseudonymity refers to the usage of pseudonyms as identifiers instead of natural names or other inherently linkable identifiers [12]. A pseudonym is a replacement identifier which cannot be linked to its holder without extraneous data. As long as the pseudonym ownership data is confidential, it is in principle impossible to determine the holder's identity. This does not mean that the pseudonym could not be unmasked through linkability of data that it is used in conjunction with. Once a pseudonym is linked with an identity, so is all the data it has been used with, which makes the usage of pseudonymity a very volatile state especially as the amount of data and transactions it is used in grows. Transaction pseudonyms can be somewhat of a solution to this, where for every transaction involving the pseudonymous data, a new identifier is generated, which is at least initially unlinkable to any past identities.

The data controller, according to the European General Data Protection Regulation (GDPR), is defined as the entity who "determines the purposes and means of the processing of personal data". A data processor is an entity who performs the actual processing of data on behalf of the controller, other than the controller itself. Additionally, the data controller is required to have a contract with the processor in which the details of the processing are defined, in the case that a data processor is used. [10] The data

controller and processor are often interchangeable when discussing matters relating to the processing itself, such as accountability and confidentiality.

The modern approaches at data protection principles are largely based on the principles of the Fair Information Practice Principles, defined by the United States (US) Federal Trade Commission based on a report from 1973 [13]. The five principles defined are: notice/awareness, choice/consent, access/participation, integrity/security and enforcement/redress [14]. These principles enforce the data subject's right to privacy as the ability to control their personal data.

The **notice/awareness** principle means that data subjects should be given an opportunity to make an informed decision on whether they want to disclose personal information for processing [14]. For an informed decision, the data subject needs to know what information is processed, how, by who and why. Details on other relevant information policies should also be communicated.

Choice/consent specifically means that data processing should be based on an affirmative expression of consent from the data subject [14]. The easiest way to enact this is to only process data that the subject voluntarily has disclosed for processing. In practice however, this governs the secondary usages of data for purposes other than what was necessary to fulfil the original intent.

Access/participation is the principle that ensures data subjects are able to review data about themselves. This means the data subject can verify the accuracy of information that is processed and exercise their rights to correct data or object the processing in a timely manner without hindrances [14].

Integrity/security is the data security principle. Basically, it means that the processed data stays confidential and is protected from unauthorized access, modification and destruction [14]. In addition, the principle imposes data controllers with the responsibility to keep the processed information accurate and truthful, as well as destroying or anonymizing it when it becomes obsolete.

Enforcement/redress as the final principle states that the other principles are only useful if there are mechanisms to enforce them. Data controllers must be required to self-regulate, data subjects must be able to object unlawful processing and governments must proactively protect the rights of individuals [14]. Additionally, data controllers must be properly liable for the breach of an individual's rights.

3.2 Privacy by Design

Privacy by design (PbD) is the idea that for a system to be safe in regards of privacy of personal data, privacy must be considered starting from the very earliest designs of the system. It is not enough to bring privacy into the system as an afterthought. The term was

popularized by Ann Cavoukian acting as the Information and Privacy Commissioner of Ontario, Canada. Even where it might be possible to add privacy in after the implementation of a system, it causes the overhead of the work required to assure privacy of personal data in the system to grow considerably larger. The financial implications of this overhead may jeopardize the thoroughness of the work, leading to compromises in data privacy.

The core concept of privacy by design revolves around seven so-called foundational principles as defined by Ann Cavoukian: [15]

1. proactive not reactive; preventative not remedial
2. privacy as the default setting
3. privacy embedded into design
4. full functionality – positive-sum not zero-sum
5. end-to-end security – full lifecycle protection
6. visibility and transparency – keep it open
7. respect for user privacy.

There is no universally accepted definition of what these principles mean in practice. This makes treating privacy by design as a methodology somewhat difficult. Much like agile methodologies and the agile practices that make them, there are a number of privacy practices, design strategies and technologies that could apply to a potential privacy by design method. The practical meaning of privacy by design may also vary based on the role of the definer and the relationship they have with regards to privacy. This is a common critique of privacy by design [16] along with the difficulty in interpreting these vague guidelines when attempting to put them into practice.

3.2.1 Proactive approach

Cavoukian characterizes the proactive approach as: “PbD does not wait for privacy risks to materialize, nor does it offer remedies for resolving privacy infractions once they have occurred – it aims to prevent them from occurring. In short, Privacy by Design comes before-the-fact, not after.” [15] This concept is in spirit the main driver behind privacy by design. Privacy risks are more difficult and costly to weed out from an existing system than one under design and even more so from a finished and deployed product than one still on the drawing board.

This requires adopting a defined set of privacy goals and requirements from the beginning. When the requirements are clear, the privacy enabling solutions that go into a design can be assessed against those requirements and potential weaknesses in the solutions reviewed and fixed before they become problems in production. Defining the requirements at the beginning also enables effective assessment of the privacy impact of any design choices before they are acted on, which in theory helps avoid introducing weaknesses through seemingly unrelated interactions of requirements.

However, defining such requirements at the beginning of a design process may prove to be difficult especially in an agile product development context. Doing this in practice would require knowledge about the scope and implementation of the system under design. Alternatively, the requirements would need to be very general and they could not consider the domain of the design process which they are describing. In an agile environment, the product requirements are likely to change and the same applies to privacy requirements of the product.

The application of this method may also be hard to prove, as Bier et al. [17] argue. If two functionally identical systems are inspected after implementation, it will be impossible to tell which one applied the proactive design method based purely on their functionality and outputs. The principle is easy to understand and get excited about, but without prior experience with privacy issues it may be difficult to put into practice.

The proactive principle also includes considering the technical details of the whole data lifetime ahead of time. The data lifetime spans from its introduction to the system from the initial source, through its effective usage and ends after its secure destruction when the data is no longer used. It is important to think about data destruction during design, because data that is stored, but not actively used, is still data in the system and imposes a security burden. For this reason, any technical security measures implemented should also consider the full data lifecycle.

The data lifecycle may extend to external systems as well. If a controller releases data to a third-party controller, they should also extend certain key events along the data lifecycle to those external parties as well. For instance, if a data subject exercises their right to demand an obsolete piece of data be deleted, then that demand should be relayed to other parties also using that outdated data.

Another valid criticism of the proactive method is that it essentially requires engineers to make prophetic predictions on the future. As even an engineer armed with the latest knowledge of the industry has limited capability to anticipate what the yet unmaterialized privacy threats may be [18], it does not seem like a fair requirement to place in a formal document. Additionally, developments in the collective societal opinion on what is an acceptable level of privacy may bring forth new technologies that would be considered intrusive in the current state, or worse, cause technologies that are considered acceptable currently to turn into something intrusive. The same applies to technical privacy measures, as for example the capability of an intruder to break current encryption standards is only based on current information and future discoveries may change things wildly.

The same criticism also partially applies to the privacy impact assessment process defined by the GDPR. The privacy impact assessment is a threat discovery and mitigation tool designed to assess the relative benefit of processing personal data in comparison to the

risks it causes to the rights and freedoms of the data subject. Carrying out the assessment is mandatory in certain cases involving high risk processing. The privacy impact assessment consists of the following: [10]

- a systematic description of the processing
- assessment of necessity and proportionality of processing
- management of identified risks to rights and freedoms of data subjects
- involvement of interested parties.

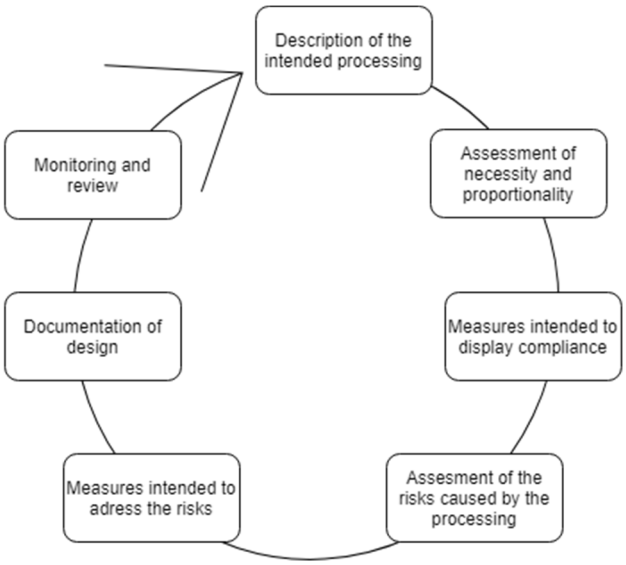

Figure 2. *Privacy impact assessment process. Adapted from [19].*

In practice, performing the privacy impact assessment is an iterative process as is shown in Figure 2. However, the privacy impact assessment must be carried out starting from the earliest designs before any data is processed as outlined in the guideline by the Article 29 Working Party [19], and as such the process practically requires the conductors of a privacy impact assessment to simply guess when it comes to implementation and operation risks. By the time a system is implemented, the information behind the privacy impact assessment which originally legitimized the system might be highly outdated. Additionally, the privacy impact assessment is not worth anything if the data controller does not stick to the promises made to counter the identified risks [13].

3.2.2 Privacy as the default

In Cavoukian's original definition, privacy as the default states that the data subject's privacy should be ensured without any action on their behalf [15]. In practice, this is often interpreted to refer to data minimization. A system should only ever require the minimum amount of personal data for the bare essential functionality [17].

According to the European Network and Information Security Agency (ENISA), data minimization requirements are poorly enforced despite them existing in the Global Privacy Standard since 2006. They go on to criticize the excessive focus on data minimisation as the ultimate solution. While it is very hard if not impossible to put technological controls on information after it has been disclosed to someone, controlled disclosure is the key to enabling many of the more engaging applications of data in the ubiquitous information society. [13] They argue for the focus to be on preserving the individual's legal rights rather than avoiding data processing. The only effective measures are then likely to be implemented through regulatory oversight. It is still difficult to argue against the fact that data that is not collected does not need any further protective measures.

Proper implementation of privacy as the default allows the individual to control how much information they wish to disclose. It allows controlled disclosure of additional information in order to provide optional but beneficial functionalities, without sacrificing future control over personal information. Effective and intuitive privacy controls are also called for in the working documents behind the GDPR [13].

Moreover, Bier et al. emphasize that even if a user's informed consent to the processing is obtained, privacy as the default needs to be fulfilled for compliance with the principle [17]. ENISA also recommends for regulatory bodies to limit the extent that user consent can be used to enable arbitrary data collection [13]. The usage of consent as a catch-all mechanism of data collection goes against the spirit of requiring the disclosure of the minimal personal information required to provide the relevant functionality. Otherwise there might be an incentive to lock all but the most basic features of a system behind a requirement of the user consenting to unlimited data collection and processing. As Bier et al. point out, users would not be likely to suffer the degraded default operation mode [17].

From a regulatory perspective, the requirement of a purpose specification has been an elementary version of privacy as the default [17]. In general, modern data protection legislation requires a specific cause for the collection and processing of personal data and having such cause is the exception. The default choice is that collection is prohibited without purpose and processing collected data is prohibited for anything but the specified purpose.

3.2.3 Embedding privacy into design

Cavoukian's original vision for embedding privacy into design was to make privacy a core feature of any product being delivered. She further states that "privacy is integral to the system, without diminishing functionality" [15].

For privacy to be fully embedded into the design of a system, the design process must identify the relevant risks that may threaten privacy of subjects in the system early on.

The idea being that the sooner in the design process risks are identified, the more comprehensive mitigation can be built against them as well as allowing unnecessary work to be avoided.

However, as the natural design process of new functionality is of an iterative nature, it may be difficult to identify risks that are specifically relevant to mitigate early on. Implementation of mitigations based on risks that were identified in early designs may end up with extra work done on outdated privacy measures for features that are no longer relevant in relation to the finished feature. To avoid this, the mitigations need to be integral to the feature rather than a separate protective layer.

As an example, location based technologies which enable reacting to the user's physical location in real time or near real time are commonly considered as processing sensitive information. To mitigate the privacy impact of using location based technologies, special consideration must be given to how that data is processed and stored. Mitigation actions for location data might include not transferring the location data out of the user's device, masking the exact location by approximating addresses, truncating data or introducing noise, and not storing the location data for longer than the minimum time required by the process. Implementation of a heavy encryption layer early on would be wasteful if the final feature does not end up transmitting the location data at all. The exact appropriate mitigation actions always depend on the application of the data, and must be built into the process.

The studies on the accuracy of the LINDDUN method by Wuyts suggest that identification of relevant risks may be difficult for non-experts in privacy issues [11]. As most new functionality in systems in general is most likely designed by people who are not experts in privacy, the sufficient execution of this principle would most of the time require a privacy impact assessment from an expert who may be external to the design process. Design by committee carries a new set of issues into the workflow with it, but it may be hard to avoid.

3.2.4 Positive-sum approach

Cavoukian describes this principle as "Privacy by Design seeks to accommodate all legitimate interests and objectives in a positive-sum win-win manner, not through a dated, zero-sum approach, where unnecessary trade-offs are made" [15]. The related ideal is that it is possible to find a common ground where both parties benefit without making undue compromises or sacrifices to e.g. individual privacy and implement systems in this space.

When it comes to implementing privacy by design, it has been critiqued for its positive-sum win-win pitch being too idealistic. Attempting to make sure that both parties, the service provider and the user, benefit from a data transaction, may lead to an incentive to upsell features in order to collect more data. Schaar argues that instead of avoiding trade-

offs between conflicting interests, the focus on preserving privacy should always come first [20]. Using the German electronic health card project as an example, he speaks for allowing the cardholder to have ultimate choice over what data is stored on the card and what data is available to service providers. This would mean that a service provider can never simply dictate what data it requires, but has to work with what the cardholder is comfortable with.

Bier et al. have a different interpretation of what positive-sum means in the context of privacy by design. In their view, positive-sum does not refer to conflicting interests of separate parties but the internal conflict of adding functionality and reducing privacy in a system [17]. It essentially states that a functionality may only be added if it has no negative privacy effects no matter how small.

A problem with this kind of definition is that applying it requires functionality and privacy to be quantifiable. Quantifying abstract concepts like this would require defining a measure for them which would without a doubt be a subjective process. Methods such as k-anonymity have been used in research [11], but they depend on the specific anonymity group for the quantification and are difficult to generalize. Furthermore, the definition is a logical contradiction from the sense of building something new. Strictly following it would forbid ever adding personal information to a system as it would lower privacy from the total anonymity of the initial state, as Bier et al. themselves point out [17].

A requirement like the positive-sum approach is difficult to formalize, and it is easy to join in the critique of it being very idealistic and providing little gain from the perspective of the individual's privacy. It may be best to forgo it as a redundant reminder that functionalities should be constructed in such a way that they cause minimal negative impact on privacy and that added functionality should ultimately be to the benefit of the data subject.

3.2.5 Full lifecycle security

"Privacy by Design ensures cradle to grave, secure lifecycle management of information, end-to-end", Cavoukian describes the full lifecycle principle. She goes on to point out that no privacy can exist if there is no security to ensure confidentiality. [15] It is easy to agree that data security is an important aspect of privacy even without privacy by design.

It is not defined how privacy by design ensures this full lifecycle security for data, however. This principle seems to act more as a reminder that data security still matters, rather than a design guiding ideal like the other principles. It may not always be possible to foresee the full lifecycle for data in a system, considering that systems may be very complex with multiple in and outflows of data and external dependencies, which may also make measuring success difficult.

Bier et al. point out that while it may be possible to formally prove the security of an individual low-level module, it is a completely different undertaking to prove the security of a system built from those modules. [17] Thus, verifying that this principle has been followed may be difficult if not impossible in the end other than through a thorough review of the implemented security measures.

It is important however, to not get too hung up on technical protections as a guarantee of confidentiality. Privacy-enhancing technologies alone cannot guarantee privacy, especially if they are merely a few components embedded as a part of a large infrastructure in a system [13]. While data security is in large part the application of correct technologies, the policies that govern the operative side are equally important to confidentiality. This side of a system is even more difficult to formally validate.

3.2.6 Transparency

Cavoukian states that "Privacy by Design seeks to assure all stakeholders that whatever the business practice or technology involved, it is in fact, operating according to the stated promises and objectives, subject to independent verification" [15]. This principle is very well in line with current data protection legislation in general and the concepts of informability and controllability.

As privacy was defined as the ability for a person to retain control over the disclosure of their personal information, naturally exercising this ability requires awareness of this personal information being processed in the first place. Informing the data subject of the processing when the processing happens as a direct consequence of the subject's actions is fairly straightforward, but it can get more difficult in indirect cases. The GDPR, for one, specifies that data subjects must be given specific information on how they can exercise their rights [10].

ENISA outlines the avenues of transparency in addition to the data subject as data processors being accountable to their controllers and data protection authorities who need to supervise compliance and enforce regulations [13]. They deem the current standard level of transparency insufficient and point out that the systems processing personal information are growing more complex. In the era of the ubiquitous information society, the amount of personal data and applications that rely on it to function are likely to continue growing.

Transparency is a special case also in the sense that instead of relying on technological solutions, it is based wholly on the correct information being available to be shared. Additionally, the personal impact of the communication may vary a lot from person to person, as well as preferences on the scope of the communication. Transparency-enhancing technologies can aid in communicating about the facts of the processing at several different levels of granularity as well as empowering data subjects to control their own privacy

through intuitive dashboards for example. [13] Formal audits can also be a method to communicate transparency, depending on who the stakeholders of the information are.

Information about responsibilities, requirements and the evaluation criteria and methods for measuring compliance should be openly available. Freedom of information enables independent review of the design and its privacy features as well as the policies surrounding data privacy in the system. While this level of transparency is not a requirement of the GDPR for example, it is heavily encouraged. However, sometimes regulatory demands are at odds with each other which prevents transparency from working, as an example law enforcement databases [13].

When data processing is performed based on the consent of the data subject, transparency becomes especially important and is required for the subject to be able to give informed and free consent at all. The GDPR expands on the specifics of consent and in their opinion (15/2011) the EU Article 29 working party have outlined that only informed consent is considered valid. In specific, this means that "all the necessary information must be given at the moment the consent is requested, and that this should address the substantive aspects of the processing that the consent is intended to legitimise" [21].

3.3 Privacy by Design in legislation

The same values that are behind the privacy by design ideology have been influencing the direction of evolution legislation and regulation related to data protection in recent years. Language similar to parts of Cavoukian's privacy by design manifesto requiring sufficient technical safeguards be implemented by certain data controllers can be found in various US and EU legislative acts or directives. [18] Upcoming legislative data protection reforms in the EU will take this trend one step further by directly influencing the software and systems design and manufacture processes with almost universal requirements to employ privacy enhancing methodologies and technologies during design and implementation of systems that involve personal data [10].

Lack of such legal empowerment of the privacy by design methodology in the design process has been one source of criticism of privacy by design as a true solution [13]. Without emphasis on design, policies and overarching consideration for privacy as the right of a person, privacy solutions in the industry have tended to gravitate towards the implementation of easily marketable and replicable technical solutions for data security while ignoring the political and moral aspects of data protection and privacy [18]. This can be somewhat seen in the recent uproar of public debate and controversy relating to mass surveillance conducted by technical means as a valid tool for e.g. police work and calls for backdoors or reduced-effectiveness encryption in consumer devices. Consumer privacy advocate organizations such as the Electronic Frontier Foundation have been calling attention to the privacy violations regularly occurring at border crossings in the US where border control agents can arbitrarily coerce travellers to bypass technical privacy

measures through blackmail with the threat of detention [22]. Issues like these paint a view of the current political and regulatory landscape of privacy as a right of an individual as compared to a collection of technical measures required by industry standards.

The UK Information Commissioner's Office started a program in 2008 to encourage privacy by design adoption in private and public organisations. They call for a higher mandate to apply privacy by design and privacy impact assessments during design and operation of systems. Furthermore, they speak for promoting privacy by design to also consider organizational changes. [16] The details of how such evolution of the principles might happen are still open. ENISA, likewise, petitions policymakers to push for periodic privacy assessments by independent agencies [13]. They also urge for the development of privacy engineering tools that are easier and more intuitive to apply and better defined.

In April 2016, The European Commission approved the new General Data Protection Regulation (GDPR) which is going to replace the current Data Protection Directive [10]. The regulation is a result of the data protection reform that has been in works since early 2012. The core drivers of the reform were to unify the regulation and supervisory bodies across EU nations, unify and improve the rights of EU citizens, bring additional transparency to data processing. Additionally, a "one-stop shop" approach was pitched to provide support and ease of regulation for emerging technologies and digital commerce. The regulation directly supersedes the old directive as well as national legislation as the authoritative source of data protection legislation. It will be directly enforceable in all EU nations after May 2018 as well as certain foreign entities when they process personal data of EU citizens [10].

While the broadest terms of the regulation, such as the definition of personal data, are compatible with current legislation in Finland, many changes are also to be implemented. Such changes at a glance include more strict requirements for consent, requirement for the establishment of a data protection officer in certain organizations, EU-wide unified sanctions for incompliance and the inclusion of systems development and vendors directly into the data protection process.

A piece of legal criticism on privacy by design that the GDPR does not directly answer concerns the translation of law into code. There is an apparent disconnection between the law text that describes the principles and processes having to be specific enough to be translatable into a concrete implementation in code or design by an engineer and yet flexible and broad enough that it will sufficiently answer future technical and sociological issues that may threaten privacy [18]. It is however worth questioning if the translation of the spirit of law text into code should even be attempted. The law is always interpreted by a human reader to judge whether it has been breached, and it is not likely that computer assisted judgements, case summaries or profiling would stray far from this requirement in the near future. As such, it might be sufficient for the law governing privacy preserving systems design and implementation to give borderline requirements and define the criteria

through which an educated human reader can infer the absolute requirements concerning their system under design. Writing law text so specific that it is translatable to code without room for interpretation would most likely also cause conflicts with many unforeseen domain specific issues. These issues would then need exceptions implemented into the law, and the maintenance burden on further additions to this law might quickly grow out of hand. The GDPR does reserve the option for amendments to be passed into the regulation with more specific current implementation guides and requirements in response to emerging privacy threats, that are in line with the framework of the law [10].

Another noteworthy problem with enforcing privacy by design as a framework defined in law is assignment of accountability and liability. If a data controller does not design their own systems, but outsources their implementation, they might not know the details of the manufacture process. They might be simply users of a system sourced from a completely different legal environment [18]. As the ultimate victim of a data breach would most likely be a person who may not even be a direct customer of the data controller, tracing down the responsible party may lead into a complex web of contracts between private enterprises.

Placing the conformance liability with manufacturer would simplify the handling of manufacturing defects and errors that lead to regulation violations. This would, however, cause jurisdictional issues as in the case of Software as a Service, for example, the software with the defect could be provided for use from a jurisdiction where the liability scheme is different. It would also place domestic businesses at a disadvantage in the global market.

The other option is placing the exclusive and ultimate responsibility with the data controller. In many cases this means that the controller will still depend on assurances from the manufacturer to display conformance with design requirements in the case of sourced systems. This simplifies the overall liability from the point of view of the law, as the data controller must organize the processing in such a way that they can show that privacy by design aspects have been considered regardless of how the processing is implemented. The privacy by design requirements would still transitively fall to the manufacturer through business requirements and this would enable the industry to practice a form of self-regulation.

The GDPR takes somewhat of a hybrid approach to liability. The data controller is ultimately liable, but data processors commissioned by a controller to perform the processing will be equally liable under the law. This sort of a business relationship must be based in a written contract that specifies the nature of the processing and the relationship of the entities. Additionally, there are references to a certification framework, which would allow businesses to proactively show conformance to the process. [10] The details of the certification framework still seem to be in a state of flux regarding who would perform them and what the certification process requires.

How specifically a data controller must prove usage of privacy by design is still also an important open question regarding liability. No precedent exists yet for how the design and implementation process must be documented to show application of the privacy principles. Likewise, no precedent exists for how these privacy qualities should be presented in a system. Certification guidelines may be one answer, when the certification framework gets defined further. For now, it is important to be prepared to show best effort compliance through employing a defined process and documentation artefacts.

It is also unknown what are the specific processes that are acceptable to base decisions about privacy enhancing technologies on. Many processes exist for threat discovery, such as STRIDE or LINDDUN, but these may not discover all relevant threats even when employed by experts [11]. The Article 29 Working Party provides one interpretation of the impact assessment process in their 2017 guideline publication, but go on to state that it is not formally defined in the GDPR [19].

3.4 Current top risks to privacy

Visibility into the state of the industry is important when discussing measures that might impact the situation positively or negatively. The Open Web Application Security Project (OWASP) maintains a list of the top 10 risks to privacy in web applications and the risks ordered as found on the 2014 revision are as follows: [23]

1. web application vulnerabilities
2. operator-sided data leakage
3. insufficient data breach response
4. insufficient deletion of personal data
5. non-transparent policies, terms and conditions
6. collection of data not required for the primary purpose
7. sharing data with a third party
8. outdated personal data
9. missing or insufficient session expiration
10. insecure data transfer.

Vulnerabilities in application implementation take the first spot deservedly, even if it stands here as a catch-all category of sorts. No matter how bespoke the security mechanisms in an application are, they are not worth much if they are improperly implemented. The basics of privacy are laid in data security. Likewise, even the best policies will do little to mitigate risks if they are not exercised in the implementation. Having sound operational practices and ways to audit their effectivity will also go a long way towards mitigating this risk.

Several of these risks can be directly linked to a failure to employ proper data minimisation practices, either through collecting and storing extraneous data or releasing too much data through poor access control. Almost all of the issues listed are related to improper or insufficient employment of data protection principles in general. Many of these also have

a regulatory impact through compliance failures or bad governance. The risks that are discussed here in further detail are operator-sided data leakage, insufficient data breach response, storage of extraneous data and lack of transparency in policies as they have a strong presence in the privacy by design and data protection principles of the GDPR.

With cloud based PaaS and IaaS adoption growing, an increasing amount of systems is relying on external operators to hold up their end of the privacy requirements. This increases the likelihood of an error on the operator's side. Of course, with more data resting in the operator's systems there is also more opportunity for malicious activity from an internal actor. If relying on an external operator is the only option, the main mitigation for this risk is proper vetting of the operator's reputation and procedures [23]. Many cloud service providers base the viability of their operation on access to a large amount of capacity in server farms, which may be located in various jurisdictions [24]. Acting as a data controller for a system built on a platform like this may be a precarious undertaking without full understanding of the various aspects of international data protection and a level of control over the implementation of the platform's operation.

3.4.1 Data migration to external operators

In reference to external operators, it is worthwhile to discuss the regulatory aspects of global operation from a European perspective in relation to the vast amount of cloud service providers located in the US. On October 6, 2015, the European Court of Justice judged that the safe harbour statutes are invalid due to internal conflicts with the European data protection directive. The safe harbour statutes had been the source of legitimacy for businesses located in the US who provide services requiring the transfer of personal information from the European economic area. [25] This judgement has shown that mere regulatory compliance based on the existence of international frameworks governing privacy protection is not sufficient proof that the entities taking part in processing under the framework are complying with the limitations outlined within. In this case specifically, there was a concern that the US-based businesses could not guarantee the privacy of European data subjects. Due to the domestic authority of the US National Security Agency to perform covert intelligence operations, operator-sided data leakage which could be considered unauthorized from the data subject's jurisdiction could not be ruled out [25].

Since the invalidation of the framework coincided with the regulatory work of creating the GDPR, the Article 29 Working Party clarified that similar mechanics to the ones governing the relationship of a data controller and a data processor might be employed to satisfy the court's standard [25]. These mechanisms are the standard contractual clauses and binding corporate rules, which are an organisation-level contract to accept the regulatory liabilities for processing personal data in a manner compliant with the European legislation [24].

The GDPR does allow international transfers under certain terms, even without standard contractual clauses or binding corporate rules, to countries which have been deemed to have sufficient levels of data protection in their own legislation in a manner compatible with the European legislation. Notably, the US does not have a national data protection law and thus cannot be adequate under this rule. [24] Even if the US had a national data protection law, it might not be adequate due to the authority of the domestic intelligence agencies which exceeds the rights of private individuals. As a partial resolution, Varotto [25] calls for the founding of an inter-governmental body to regulate data protection internationally and to act as an accountability agency for countries acting under the adequacy rule.

To summarize, when considering external operators, individual research and audits are good practice. Acceptance of external operators should not be based purely on compliance with framework agreements, but individual contracts between organisations. These contracts must outline the liabilities that arise from handling personal information.

3.4.2 Data breach response

In the event a data breach does occur, the data controller's response can do much to aid in either mitigating the impact or worsening it. Under the GDPR, a personal data breach is defined as "a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed" [10]. There are four phases to a proper response to a data breach: detection, notification, mitigation and improving.

Detection of a data breach may happen multiple ways. In the best-case scenario, proactive intrusion detection systems and audit log monitoring alert operators to the fact that a breach is ongoing. A notification may also come from an external party, such as a security researcher or a data protection authority, in which case it needs to be verified. The worst-case scenario is that suspicions of a data breach arise after leaked data is discovered already in public circulation. Early detection is important in order to allow fast response and effective mitigation. Without detection, a data breach cannot be responded to at all.

Notification of a breach must be given to the supervisory authority within 72 hours of detection under the GDPR [10]. Technically the notification can be avoided if the data controller can determine that the breach will not result in a "risk to the rights and freedoms of natural persons" [10], but it is unlikely that an organization could perform an exhaustive verification of the risks a breach imposes so quickly [24]. There is no harm in giving a notice to the supervisory authority in either case. In certain cases, if there is a high risk to a person, the notice should also be given directly to the data subjects affected [10].

The notification itself must describe the breach, giving as accurate details as possible on the type and volume of data involved as well as the amount of affected data subjects.

Additionally, the notice must include information and contact details on where more information about the breach can be inquired. The notice must include an assessment of the consequences that the breach is likely to cause as well as the measures that were or will be taken in response to the data breach, such as mitigative actions to the identified risks. [10] The notice may be amended iteratively as more details come available. This process is also in line with the implementation of an incident response plan as required by standards such as ISO 27001, ISO 22301 and Payment Card Industry Data Security Standard for example. [24] For an organization prepared to meet the responsible incident management criteria of these standards, the notification phase should be an easy addition.

Mitigation of the data breach begins with the notification, as awareness of loss of control over personal data enables the affected individuals to respond accordingly. Other mitigative actions depend largely on the type and amount of data that was affected by the breach. Another important factor in deciding the appropriate mitigative actions is the length of time from the breach to detection, as the longer the breach took to detect, the more likely it is that it has led to further exploitation. Using breach assessment frameworks such as ENISA's severity assessment method may be helpful to understand the proportions and potential of a breach [26]. The data controller must evaluate the risks to data subjects and others and act accordingly to mitigate the effects and prevent potential follow-up abuse of leaked data for example. These actions need to be documented and they need to have someone named responsible for overseeing their implementation.

Improving policies and processes to learn from breaches to avoid them in the future is the last thing to do after the breach has been resolved. This means organizing a retrospective debriefing with everyone involved in the spirit of continuous improvement, after the acute issues have been taken care of and some time for monitoring has been allowed. [27] The key questions to look at are what happened, why it was possible, how well was the response executed and could anything have been done better either before, during or after the breach.

3.4.3 Lack of compliance

Failure to remove obsolete data and collection of extraneous data both lead to data existing in a system without a legitimate basis for it. Such data, in addition to being a legal burden, poses a huge risk in case of a data breach. In the case of failure to delete, there are two clear options of how this risk might realize: the system holds personal data that is not known about and thus not deleted, or the system holds personal data that is known, but proper deletion is neglected.

In the case that data is not known about, there has been an issue with data flow and lifecycle management, as the personal data affected has entered a state that was not originally planned. This is the case even if there was no plan. This is one of the situations that proper employment of data minimization and other privacy by design practices during system

design is meant to mitigate. The general mitigation against it is simply ensuring storage or transfer of personal data in unintended states is guarded against by reviews of design, implementation and documentation.

In the case that destruction of obsolete data was knowingly ignored or that extraneous data was processed against the specified purpose, there may be a conflict between the actual intended purpose of the system and the documented purpose specification. Alternatively, there may be a willing failure to comply with regulation, but that discussion is not in the scope here. The implementation of a system must be reviewed against the stated purpose specification and in the case of a conflict, the purpose specification must be updated. There should rarely be a need to update the implementation due to a purpose specification change, as the purpose specification is a tool for explaining the actual intended functionality of a system from the data subject's perspective. It has been a common practice in data-intensive modern services to make purpose specifications as vague as possible to enable the processing of arbitrary data that has business value, but this is likely to be the target of regulatory intervention at least in the EU with the enforceability of the GDPR beginning. Lack of transparency over the data lifecycle can thus become a business risk.

The lack of transparency in policies and terms of service, creating a disconnect between the stated purpose of a system and the actual usage of data, has been a source of a lot of critique of data protection regulation [13] [18]. Either through lack of communication or communication being difficult to understand, the data subject's ability to exercise their right to control their personal information is limited. Large operators especially in the social media sector have begun developing their privacy controls into a direction that makes them easier to understand through the use of proper localization to the data subject's native language and usage of descriptive iconographic language. [13] This is a promising development and it is in line with the intent of transparency drivers in the GDPR.

As an alternative approach Anthonysamy et al. [28] present a method for analysing the relationship between the stated privacy policies and the privacy controls they offer to data subjects by extracting and mapping action statements in policy to operations available in controls. This sort of analysis can reveal potential differences between the communicated rights and freedoms of the data subject and the actual privacy controls that are implemented to allow exercising those rights. Additional findings can include inconsistencies in stated data sources or types.

In their study of social network operators using their described method, Anthonysamy et al. also call attention to issues of information asymmetry and default opt-ins, which reduce the control a data subject has [28]. Information asymmetry here refers to the usage of vague and general terms which make it difficult for the data subject to learn how their data is used. This paired up with default opt-ins, which the users may not be aware of, can make users agree to usages of their data they do not like because they do not know

how to decline. The practice of default opt-in in relation to privacy controls is very questionable at the least under the interpretation of informed consent as specified by the Article 29 Working Party [21].

3.5 Other privacy model frameworks

While many frameworks have been created for the benefit of data security and verification of controls such as authorization, relatively few specialize in privacy threats or issues. A recent development in privacy threat modelling methodology that warrants a mention is the LINDDUN method.

The LINDDUN method is a privacy threat modelling methodology. It has been built as a complementary support technique for privacy by design implementation. The name comes from the privacy threat categories that are contained in the methodology: linkability, identifiability, non-repudiation, detectability, disclosure of information, unawareness and non-compliance [11]. It is strongly inspired by the STRIDE methodology.

The LINDDUN process consists of six steps. The first step is creation and examination of a data flow diagram describing the overall system. The second step is mapping each entity of the data flow diagram to the corresponding threat categories and identification of potential threats. The third step is examination of the identified threats to validate them and figure out which ones are actually applicable to the system. This is done through a collection of pre-fabricated attack path threat trees that describe common privacy vulnerabilities relating to each specific threat category. [11]

The fourth step of the LINDDUN process is prioritization of the analysed threats according to their risk level. In the fifth step, the privacy requirements concerning each threat are identified and mapped to the threats to facilitate the choice of the mitigative solution as the sixth and final step of the process. [11]

The STRIDE method is a generic security threat modelling method. Its name stands for spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege, each also referring to the risk categories the method consists of [29]. It also encourages the use of a data flow diagram as a view to the system and the examination of entities found in the diagram to identify threats related to them.

4. AGILE PRIVACY BY DESIGN

4.1 Agile privacy by design principles

To combine the privacy by design principles into an iterative development model the approach needs to reach beyond simply choosing which technical measures to implement in order to sufficiently satisfy the privacy by design principles. All relevant privacy principles need to be considered and built in in the design phase of the system. [18] The design phase in an iterative development process happens gradually, often and repeatedly as new options are explored and new knowledge acquired. For these reasons, it is unrealistic to incorporate a heavy ahead-of-time process to accomplish the privacy work in the iterative workflow.

Instead, it would make sense to embrace the iterative process and approach the data protection tasks also from that perspective. Proactively ensuring that each gradual increment individually follows the privacy by design principles is much more realistic than implementing features first and then verifying each design as a whole through a separate threat discovery process or performing huge data protection design up front prior to implementation work in an iteration. Not only would such an approach hinder the flow of the agile project, it would still involve adding privacy solutions in after the fact or disconnected from the actual implementation. The data protection measures required need to be designed and built in together with the features they relate to.

The Scrum model has two logical points where the design choices made in a sprint are present, which are the sprint planning and sprint review. The sprint planning meeting decides what is worked on in a sprint, which directly affects the scope of the design work to be done. The sprint review marks the ending of a sprint and is where tasks are showcased and accepted or rejected. Thus, it makes sense to focus the process-oriented aspects here.

There is also a common misconception that documentation is somehow anti-agile, but it is not true. A key characteristic of the agile approach is delaying major decisions until their effects are imminent. This reduces unnecessary up-front design and avoids decisions made with partial information. It means more of the large considerations happen later in the process, which should also be reflected in the agile privacy by design model. [7] In conjunction with this, the connotation should be that producing excessive documentation up-front is against the agile principle. Producing documentation just-in-time however is perfectly fine, and avoids the problem of producing documentation that will be outdated before the implementation is even relevant [6]. Following this line of thought, it makes sense to produce the data protection related documentation iteratively parallel to the implementation of the solutions they describe.

4.2 Requirements

To begin exploring the application of privacy by design in the agile design process, some requirements are needed to evaluate the model with. The main goal in this section is to identify a set of requirements sufficient to evaluate whether the produced model could reasonably be employed to aid in complying with the requirements of the GDPR as far as the design process is concerned.

As the goal is to form a process compatible with the privacy by design ideology as it is seen in the context of the GDPR, the core concepts to consider as data protection issues are selected from both the requirements of the GDPR and the core privacy by design principles. These aspects are very closely related to the minimum information that needs to be communicated to a data subject before processing data. The following list contains the identified aspects of the system design process that have regulatory requirements or relate to the principles of privacy by design:

- data storage, lifetime and amount/specificity
- legitimacy of processing
- application of data, purpose and intent
- confidentiality
- integrity
- accountability, auditability
- portability
- risk and impact assessment, and appropriacy of protection
- data subject's rights, informability
- ability to react to incidents
- compliance.

Data minimization. All the aspects identified have to do with personal data directly or indirectly. To properly design for these issues, a data controller needs to have information on what personal information actually exists in the system, in storage or through transit. Reasonable effort must be made to ensure only the minimal data necessary exists in the system and for the minimal time.

Legitimacy. Every piece of personal data collected should be covered by the claim of legitimacy for processing data, i.e. the purpose specification. This claim represents the broadest category of the purpose personal data is used for in a system. Any data not covered should be considered extraneous, or the specification should be changed. The basis for legitimacy may be different for different users of a system. It is a tool for accountability and used to communicate to the data subject the reason for processing their personal data.

Purpose. Much like the general-purpose specification, the purpose or intent of processing any specific data should be accounted for. Processing personal data for purposes other than their specified original purpose must not be allowed if it is not directly aligned with

the original purpose [13]. In a very small or narrow scope project the purpose may be the same for every piece of data and it does not need to be documented very granularly, but every distinct function performed based on the data should be covered by the claim of the intended purpose. In many cases, the purpose can simply be providing a specific service or function to a user that would be impossible without the data in question.

Care must be exercised when data collection is planned based on a design from a previous iteration of a system. Schaar points out as an example in the German electronic proof of earnings project (ELENA) that data collection needs were initially based on paper forms that made up the previous system with little more than a cursory review, but after more in depth analysis it was discovered that there the necessity of certain data fields was questionable and they should most likely be removed [20]. This goes to show, that proper application of privacy by design requires a certain amount of scepticism towards the data collection requirements even when they are coming from domain experts describing the necessary features.

Confidentiality. When collecting data, the data controller implicitly accepts responsibility for controlling access to the data they are trusted to process. Failure to do so would effectively take away the data subject's ability to control their privacy. In this sense, data confidentiality is the primary guard against loss of control. Enabling confidentiality has two elements to it: the policy aspect and the technical aspect. On the technical side, all sensitive data should be appropriately protected at rest and in transport to mitigate large scale data breach effects, but leakage also needs to be considered during active processing. The policy aspect is parallel to the technical side as the implementation of access controls and other relevant policies requires technical enablers, but using those technologies requires human choices. To enable assessing the adequacy of the confidentiality mechanisms, they need to be observable and that essentially means they need to be covered by documentation.

Integrity. For the result of processing data to be useful, it generally needs to be based on data that is correct and up to date. When working with linked systems, propagation of faulty data is an inherent risk. Especially when decisions are based on profiling information, decisions based on outdated or false data may be very damaging. To mitigate this, the authority of any sources of personal data should be reviewable as well as the sources of modifications to stored personal data. In addition to basic data security issues, integrity is very much an audit trail and access control problem.

Accountability is the secondary mechanism of enabling a data subject to control their privacy. For a data controller to be accountable for the storing and processing personal data, the data subject first needs to be informed of the processing taking place and the details of the circumstances of the processing. Additionally, the data subject needs to be able to verify that their rights have been taken into account and that the responsibilities of the data controller have been fulfilled. In practice, this means transparency about the

issues discussed in this chapter on the side of the data controller and sufficient auditable records of activities on personal data.

Portability requires consideration in design to identify what data can be exported and what additional requirements exporting it places on the system. The right to data portability is very closely related to the right to access collected data. At the bare minimum, this means that any barriers to data portability that are apparent in the design should be considered issues that need to be worked around. The portable data, according to GDPR, needs to be provided in a generally recognized applicable electronic form [10], which is a rather loose definition as far as requirements go. Properly implementing a portability mechanism should also minimize any issues arising from requests to inspect collected data.

The **privacy impact assessment**, also known as data protection impact assessment by the GDPR [19], is a tool to describe the intention for processing personal data and assess the need for collecting the data as well as gauging the risks it may cause to the privacy or other rights of an individual. Lastly it helps design the mitigative actions to minimize the impact on the individual's privacy and other rights. It is required by regulation in some cases, but the Article 29 Working Party recommends privacy impact assessments as a method to display compliance with the risk-based aspects of the privacy by design process [19]. There is no universal strict formal process that performing a privacy impact assessment entails. The guidelines by the Article 29 Working Party include a checklist which expands slightly on the basic requirements of the GDPR [19].

In order to comply with the GDPR, all the implemented privacy solutions should be designed based on a risk assessment or a privacy impact assessment. Approaching the implementation from the mitigated risk perspective also helps assess that the proper privacy by design principles are followed instead of shopping for solutions after the fact.

Data subject's rights. Depending on the specifics of the law, the data subject has a number of rights concerning the data collected by a data controller. Such rights might be for example the right to be informed, access and inspect, transfer, correct, restrict or object processing and request erasure [10]. Implementing enablers for these rights will likely require consideration in the architecture of a system.

Incident response. The risk based approach mentioned under other aspects also applies for the ultimate risk: data breach. Every system that includes personal data is susceptible to it, and thus it makes sense to give special care to mitigating it. It is such a significant matter that it occupies the third place on the OWASP privacy threats list [23]. The GDPR requires data controllers to notify a supervisory authority of breaches without delay, and to notify data subjects directly where the breach might result in a significant risk to their rights [10].

From the data subject's point of view, the notification of a data breach is an important tool for mitigating the secondary risks associated with a data breach. Identity theft is a globally applicable follow-up risk, but actual threats may even include physical harm [26].

4.3 Preparatory work

The Scrum process begins with the sprint zero, where matters such as ways of working are defined. It also includes bootstrapping the product backlog, which means the most important features to build have been discovered, described and prioritized. After all this, the project would be ready to start planning the first sprint.

The sprint zero is also the time to agree on the privacy goals for the project. Bier et al. present the concept of data protection targets as a practical set of principles that describe the data protection aspects that should be considered when making design decisions on functionality. They define confidentiality, integrity, availability, transparency, unlinkability and intervenability as dimensions that should be documented for each major design decision. [17] Schaar presents an alternative set of data minimization, controllability, transparency, data confidentiality, data quality and possibility of segregation [20]. These should be documented and visibly present in the everyday work for the project team.

Bier et al.	**Schaar**
confidentiality	data confidentiality
integrity	data quality
availability	
transparency	transparency
unlinkability	
intervenability	controllability
	data minimization
	possibility of segregation

Table 1. *Data protection target candidates from Bier et al. and Schaar*

Table 1 shows the data protection targets chosen by both Schaar and Bier et al. with synonymous concepts paired up. The lists have many of the same principles as their basis, expressed from a slightly differing point of view. For example, intervenability is a more adversarial approach to discussing the individual's rights than controllability, which emphasizes the empowerment of individuals by providing them with tools to control their personal data.

Availability stands out on the side of Bier et al. as more of a data security oriented target, but it is reasonable to argue that failure to ensure availability might endanger privacy when it comes to for example privacy controls and the operability of the privacy tools provided for the individual. The same applies to informing the individual of processing when the notice is latent.

Likewise, possibility of segregation may seem like a goal that is tied to implementation of a specific technology, but it is important to notice that it refers to the existence of an architectural enabler for a technical solution. The system architecture must enable data minimization in such a way that segregated processes can be used for completion of unrelated tasks without being able to access data from other contexts [20].

There are many other candidates that could be on the list of potential data protection targets in addition to the ones listed in Table 1. For example, many of the privacy by design principles could be considered data protection targets, such as proactive mitigation, portability and full lifecycle security. The project team should be free to consider whatever data protection aspects are relevant to the domain of the project under development.

Regardless of what set of data protection targets is selected as the initial basis, they must be able to evolve throughout the future iterations of the project. Deciding on a fixed set of requirements beforehand would be anti-agile even when it comes to privacy and data protection. Furthermore, settling on a set of targets and complacently executing them fails to consider that the data protection landscape is in a constant state of flux. Not only do the project requirements change during the project, new operational and technological threats may arise from outside of the project that require immediate attention.

Bier et al. also introduce definition of a purpose specification as a requirement for starting a system design process. The purpose specification is a documented statement of a system's concrete purpose. They claim that changing the purpose specification in any way after design has started would require the initiation of a whole new design process starting from the purpose specification. [17] Fulfilling this requirement in an actual agile product development project seems very difficult. Not only does making such a statement at the start of a project go against the spirit of agile development, there is a high risk that such statement would then need to be so generic that it gives little guidance on the actual design work. Restarting the design process with an existing system as its basis would not have a large privacy impact, as the design needs to be consistently and continuously reviewed either way and the sort of minutiae this rule is meant to catch are likely to go unnoticed when restarts become mundane in an agile process. Without the requirement to start a new design process, this concept may be workable into a high-level goal, but other steps during the design process will likely not depend on it.

Sprint zero should also include setting up all the global documents that the team will access and update throughout the development process. Such documents are at least the data flow diagram as an architectural overview and a data storage diagram overviewing where the personal data used in the system resides. Additional documents may include a data lifecycle plan for example, outlining the actions that need to be performed in order to take data in, protect it and then ensure it is purged from the system.

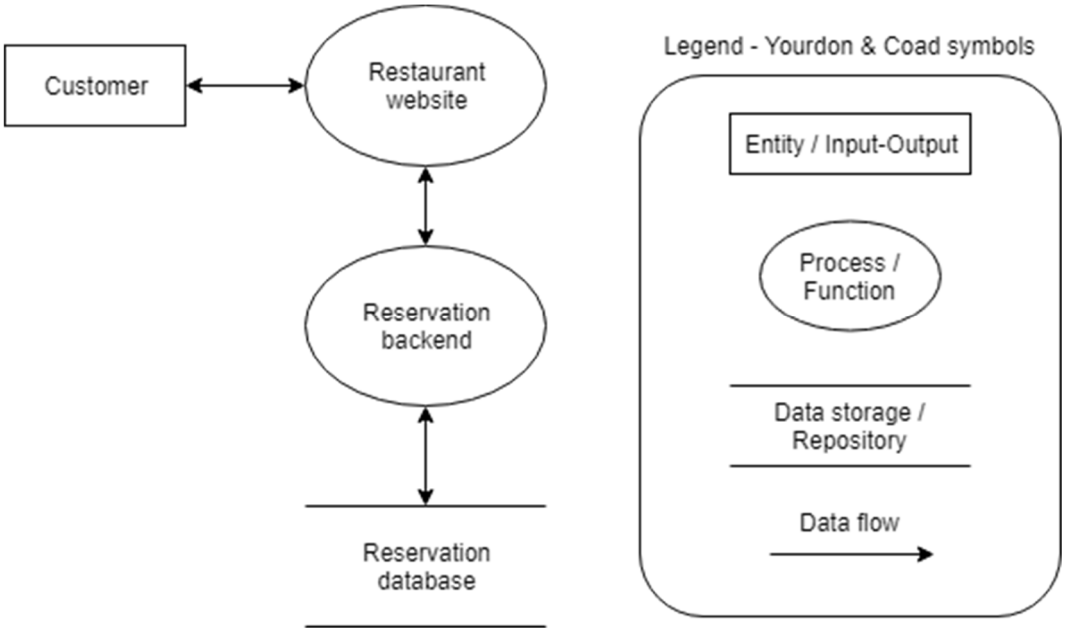

Figure 3. *A simple data flow diagram. Adapted from [11].*

The data flow diagram is a tool to chart the movement of data within a system. Its purpose is to assist in keeping track of the overall design and discovering additional data flows in a system as it is iteratively designed [30]. Figure 3 shows a simple example of a data flow diagram depicting a restaurant reservation system embedded onto the restaurant's website. The reservation system allows a customer to search for an open time-slot, make a reservation and potentially verify the details of the reservation. The customer is the sole input source and output destination in this diagram from the system's perspective. An additional detail to be added as the system grows is descriptions attached to the data flows describing the type of data exchanged.

As is apparent from Figure 3, all the data on way to or from the reservation database must travel through two distinct systems represented as processes. The longer the path of a flow, i.e. the more systems it passes through during its lifetime, the larger the data protection footprint of that data is. However, the quantitative length of a data flow is not that important, but the nature of the processes it passes through.

The elements of the data flow diagram will likely end up being more of a logical representation of the system's components as it evolves, rather than a physical or structural representation. Using logical components allows for a more granular expression where necessary and to group physical processes and systems into process blocks elsewhere, if it improves readability of the diagram. Eventually in larger projects it may become necessary to split the diagram into multiple independently maintained parts. The important part is that the diagram is a truthful and inclusive representation of the data flows between subsystems.

The data flow diagram is notably used as a key tool in threat discovery methods. Both the general-purpose method STRIDE and the privacy oriented LINDDUN method base their view of the system in a data flow diagram to begin the threat discovery process. [11] Thus, it makes sense to use it as an architectural tool in any agile development process as well and keep it updated as the development proceeds in order to enable usage of such threat discovery tools during the iterative development.

The data storage diagram is an approach at the implementation of a data protection practice known as data mapping. The idea is to help retain in a simple and accessible format the information on what personal data may exist in a system, where it exists and what conditions govern its existence. While there is no explicit requirement to keep this information documented in such a way, it may prove to be impossible to comply with all the requirements of the GDPR without engaging at least some form of data mapping. [24] In an agile environment, making the data mapping work a part of the iterative workflow is crucial to retaining accurate information with the least effort spent on documentation.

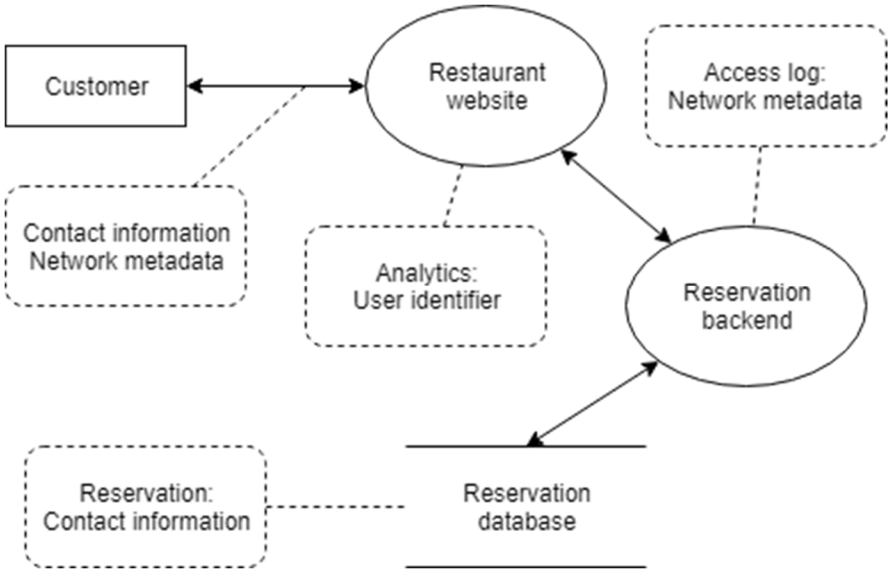

Figure 4. *Collected data overlaid on data flow diagram.*

The data storage diagram can be a simple extension of the data flow diagram or a document of its own that only references the data flow diagram. The main purpose is to indicate the logical location where potentially privacy sensitive information is processed and stored. Figure 4 shows an example of collected data overlaid on the data flow diagram defined previously. The customer provides their contact information in order to make a reservation, but as a side-effect of the web platform their network access details also get transmitted. The website uses an analytics tool to gather business intelligence which needs to combine events that occur during the customer's session into one batch, which requires storage of a pseudonymous user identifier. The reservation system on the other hand stores some access details data in order to provide an audit trail in case of misuse of the system. Ultimately the user's contact details get saved into the reservation database

to improve their service. A seemingly simple transaction includes a lot of secondary data storage.

The key questions that the maintenance of these diagrams is trying to answer are where does the personal data come from, where is it stored, and for how long. Additionally, the accountability and liability issues should be clear for all collected data, i.e. what process requires the collected data and who accesses it [24]. Finally, the documentation should help describe where data goes from the system, whether it is shared with any external parties and how it eventually gets destroyed.

These diagrams should be accessible by and editable to the whole development team. Minimizing the obstacles to keeping them up to date is in practice a key requirement of using such documents as a part of an agile process. Developers are likely to neglect parts of the process that are seen as a hindrance without value, which in truth are the elements that the Scrum retrospective process and continuous improvement are supposed to weed out [31]. However, it also cannot be ignored that introducing a privacy centric workflow into a team requires non-trivial effort from all members.

4.4 Per-iteration work

The Scrum sprint begins with the team committing to completing the planned work in the timeboxed sprint duration. All features included in the sprint should be fully finished at the end of sprint. This may be verified with the help of a definition of done -specification which outlines the conditions for considering a feature done [7].

In order to embrace the existing framework, put in place by the Scrum method, it would make sense to consider the privacy by design aspect as a part of the definition of done for each work item. Like everything else, this aspect of the definition of done could be fulfilled incrementally, and in the simplest case it can simply be agreed that a specific work item has no privacy impact whatsoever and be done with that. It is, of course, important to do this only when there can be no question whether the work item has any adverse effects.

To minimize overhead added by considering the privacy impact for every work item, the process can be broken down into several phases. The first phase happens during the sprint planning. For every work item accepted into the sprint a decision must be made, whether this work item includes aspects related to privacy sensitive areas or not. To aid with this decision, a simple decision tree can be used to quickly process all items, as is shown in Figure 5. The evaluation step can be further expanded with the specific questions that the evaluation should consist of at minimum, which have been left out here for brevity.

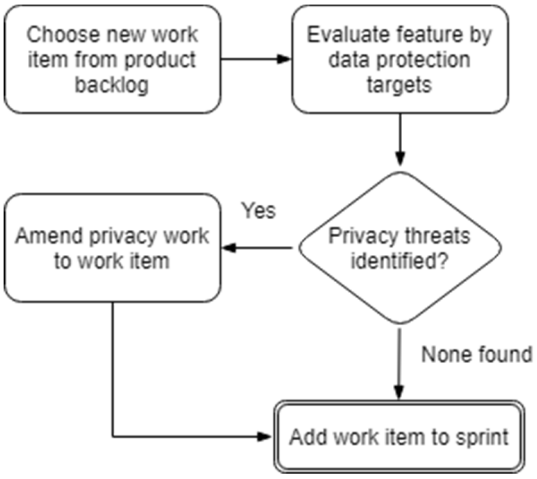

Figure 5. *Phase 1: Sprint planning*

The second phase includes work during the sprint as normal. Additionally, for those items that were deemed privacy critical, the previously agreed privacy targets are considered during development. This includes updating the relevant privacy documentation with changes made or threats introduced. In case a new threat is opened, a task for mitigating it must also be included. If during the development of an item originally not deemed to have a privacy impact it turns out the opposite is true, development can continue with this knowledge as if the item had originally been identified as a privacy critical item, assuming the time-boxing of the sprint does not jeopardize the thoroughness of the related privacy work. Otherwise, the item must be considered blocked and moved to the next sprint with the privacy work included.

At the end of a sprint, the data protection related documents and tasks should be up to date if the process is followed through. This leaves the product backlog ready for the next sprint planning. If, however, tasks are returned to the backlog, the visibility to the scope of the tasks may not be perfect [7]. Thus, from a project management visibility perspective, there is a strong motivation to be thorough when assessing tasks before beginning the sprint.

The third and last phase happens during the sprint review. In the sprint review, work items that fulfil the definition of done and their description are accepted, and defective or in-progress items are moved to the next sprint. The goal is to verify that the produced product increment fulfils the sprint goals agreed on at the beginning of the sprint. [7] Figure 6 shows the review process at a glance from the privacy point of view. The review process is in part a superset of the review process used before the sprint, but whereas before the sprint only the work item descriptions were evaluated, in the sprint review the actual implementations are available for review.

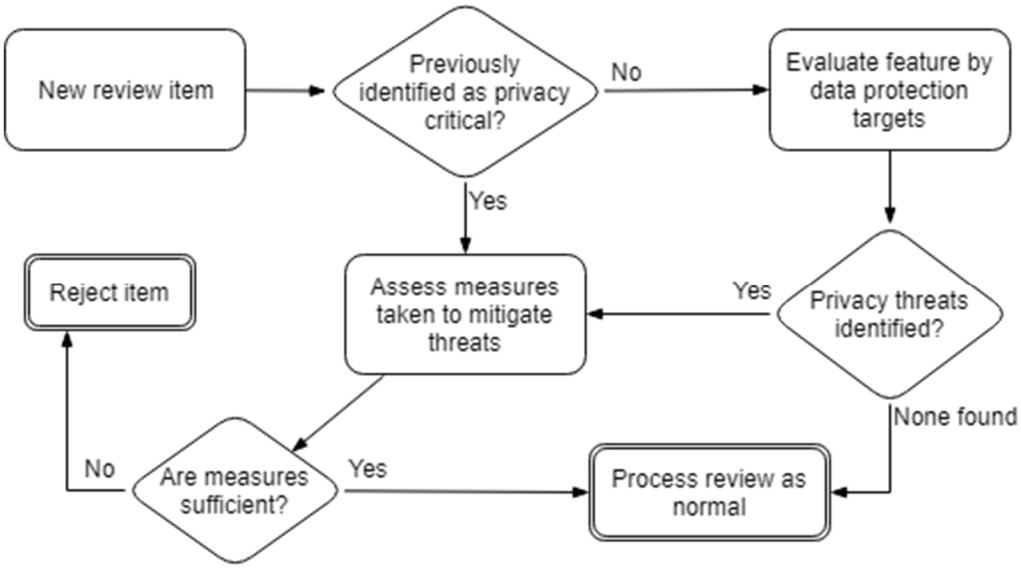

Figure 6. *Phase 3: Sprint review*

The criteria used to determine which work items are privacy critical should be based in the data protection targets. The team may adapt these targets into a workflow that supports the review instead of inhibiting it. To facilitate a thorough but efficient review, some specific questions can be identified in a checklist type manner. These questions should be used to complement the evaluation process as seen in Figure 5 and Figure 6. Example questions identified are:

- Does this feature directly process personal data?
- Does this feature affect the observability of personal data?
- Does this feature affect the linkability of data into personal data?
- Does this feature change the scope of data collected elsewhere?
- Does this feature change the lifecycle of personal data collected elsewhere?
- Does this feature introduce personal data to a module that did not previously process any personal data?
- Does this feature deprecate some personal data collected elsewhere?
- Does this feature affect the agreed data protection targets?
- Does anything prevent the affected data from being effectively exported on demand?

An affirmative answer to any question warrants closer inspection of the work item. For example, in the reservation system shown in Figure 4, the website analytics might be a work item to consider. In this case, the analytics most likely would not process personal data directly, but they definitely affect the observability of personal data in case the analytics system used is provided by an external manufacturer. It may also affect data linkability depending on what is collected, as well as the data protection targets if review of external systems has not been considered before.

For the work items that had included the data protection work in their implementation already through the definition of done, instead of going through the privacy impact assessment, the implementation of the protective measures should be reviewed. Focus should be on verifying that the identified privacy threats have been properly mitigated, and that identification of privacy threats to mitigate has been thoroughly carried out. In practice, this means verifying that the relevant documentation has been updated with the identified threats and the measures taken to mitigate them, and that the actual implementations of such measures are sufficient. For this the team must have sufficient expertise in privacy threats, or expertise in applying relevant methods such as LINDDUN. Again, incomplete or defective items must be returned to the product backlog and worked on in the next sprint.

Changing the data protection targets between sprints is possible, and in a long running project it should most likely be expected. As Scrum is about continuous iterative improvement and agile is about embracing change, it would be counterproductive to not update the process requirements to match the implementation requirements. It would also be naïve to assume the information available at the start of a project is enough to arrive at the perfect full coverage set of data protection targets that are valid for the rest of the project's duration.

Altering the targets requires that the proper attention is given to ensuring that the current product passes the changed acceptance criteria or application of the criteria would be pointless in the first place. If the system does not match the new criteria, it should be treated like any other core defect and fixed in the same sprint the criteria is changed or before the change. The more complex and further in development a project gets, the more overhead there will be to ensure that the current system matches the new criteria. Because of this, while the agile nature allows changing the targets, it should not be done lightly.

4.5 Deliverables produced

The product of following the outlined process is a set of documentation that should aid in assessing whether it meets the requirements defined. This documentation includes the data flow diagram, the privacy threats identified during development and the measures that were taken to mitigate them. Additionally, a data storage diagram will outline where in the system sensitive data is introduced and stored.

These documents make for a very good entry point into a formal audit process through a method like STRIDE, LINDDUN or other similar options. It could also be argued that these documents can be used as part of a proof that privacy by design has been exercised and privacy has been considered proactively and built in to the design, although the legal side is still unclear on it. Finally, these documents enable the operator of the system to produce accurate communication on what processing takes place and how in order to

fulfil the transparency required. It is still important not to be mistaken that the mere existence of these documents automatically means that everything is in order and the privacy by design process has been properly conducted.

The requirements that were set forth previously need to be reviewed to achieve confidence in the value of doing this work. The requirements were related to both legal compliance and sound privacy practice. The identified aspects were data minimization, legitimate processing, purpose specification, confidentiality, integrity, accountability, portability. Additionally, there should be enablers for privacy impact assessment, fulfilling the data subject's rights and incident response.

The iterative review process is a clear enabler for data minimization, as any data introduced to the system should be subject to scrutiny as long as the team properly vets the work items before the sprint. Similarly, a legitimate purpose specification should be retrievable from the work item documentation. The review process should also be sensitive to the scope of data being changed which is key to avoiding breach of the purpose specification. Of course, the risk that changes go unnoticed in the review still exists, either through unintended changes or because complex systems can have complex interactions that are difficult to foresee.

Confidentiality, integrity and accountability are data security aspects that should be considered during implementation and the sufficiency of the measures to protect the data must have been reviewed. As the measures are built parallel to the implementation of the features that rely on them for protection, it is less likely that protective measures are applied without actual risk-based assessment of their effectiveness. Moreover, consideration for integrity and accountability should end up being built into the features that concern mutation or access of personal information. Again, the documentation of the work item should reflect this and as such it should be possible to extract a description of how and why these technical measures were implemented and what threats they mitigate.

If portability is on the data protection targets, it gets considered for every feature that deals with data that might be exported. While this process might not directly drive the development towards enhancing data portability, it is an aspect that the development team should include in their data protection targets if they wish to be compliant. The process does provide an additional enabler in the sense of the data flow and storage diagrams which make it easier to plan how to implement potential portability support features.

The iterative review process most likely does not fulfil all the requirements of a comprehensive formal privacy impact assessment without extra review and documentation effort, but it serves as a small-scale variant of one. It should be straightforward to apply what is learned during the iterative process to perform a full scope impact assessment if the need arises. At the very least, the resulting documentation can be transformed into a privacy

impact assessment format. The privacy impact assessment is not a strict requirement unless the data processing constitutes a high-risk category.

The technical implementation of making it possible for a data subject to exercise their rights needs to be considered during the iterative review, and issues related these should be found in the documentation. However, there is an operational side to the individual's rights that the process cannot enforce. Similarly, incident response requires technical enablers in the form of intrusion detection capability and accountability measures, but there is an operational side that the process cannot guarantee. The documentation produced will however be a valuable asset in the operational execution of these aspects too, especially in relation to producing an incident response notification for the supervisory authority.

5. EVALUATION

5.1 Reviewing research goals and limitations

The goals of the thesis were to explore how it would be possible to fulfil the privacy by design process requirements introduced in the GDPR while using an agile software development process, what the state of privacy by design as a privacy framework is and what the privacy threat landscape looks like in 2017. This was done through review of existing research on the privacy by design principles and other methods to approach the issue of building privacy integrally into systems. The data protection requirements of the GDPR were also reviewed and used as a basis while forming the general requirements for the agile privacy by design model.

Based on these requirements and principles, the agile privacy by design model was introduced in Chapter 4. The introduced model was based on the combination of multiple different interpretations of privacy by design, as discussed in Chapter 3, as well as documented real case examples, such as described by Schaar [20]. Chapter 4.2 discussed the individual requirements and principles which the model was based on.

The outcome of this study process was the agile privacy by design model, overlaid on the Scrum method. The model consists of a set of tasks which produce documentation about the usage of privacy by design principles as well as other privacy and data protection related requirements such as data portability. To be clear, this documentation is the result of considering the data protection targets during development, but it cannot be reliably used to prove that the privacy by design principles were followed. As the existence of documentation is only the end state, it is impossible to show how the process arrived at that state.

The fact that there are no universally acknowledged criteria for a privacy by design process makes formal evaluation of the agile privacy by design process somewhat difficult. However, with proper selection of the data protection targets and assuming the process is fully adopted, the process is semantically identical to exercising privacy by design without the agile model. Another issue related to the lack of specificity in the privacy by design implementation is that there is no well-defined universal process which to follow in order to practice privacy by design. The model solves this by leaving the data protection target selection up to the project team, which is also a weakness as it requires the team to have sufficient privacy expertise to understand what choices to make.

While not critique of the agile privacy by design model specifically, the fact that privacy by design does not guarantee protection from future threats still applies to it as well. The accuracy of any predictions made depend entirely on the expertise of the team employing

the method. Similarly, the quality of the privacy work and the usefulness of the documentation that is produced as a result of following the process depend entirely on the project team's ability to understand and assess privacy issues. In order to implement privacy by design well, the team needs to have an understanding of privacy beforehand. The agile model can help in paying attention to things at the correct time, but it cannot replace the lack of experience.

The requirement of privacy experience was seen in other privacy tools as well. For example, the studies on the LINDDUN method [11] showed that in order to fully benefit from such a framework, the user must have sufficient expertise and experience in handling privacy related matters.

Other security frameworks such as Microsoft's Security Development Lifecycle also include privacy as a part of them, but their focus is heavily on the data security aspect instead of privacy as a right of an individual. This has been a common development direction in the software service industry. In this sense, employing the agile privacy by design model can help thinking of the human side of privacy issues. It may be beneficial to combine the frameworks to achieve a balance of focus between operational security, data security and privacy.

The model is as of yet untested in a large-scale project undertaking. Lack of details from the Article 29 Working Party make predicting how it would perform difficult. With further information on how to interpret the design requirements of the GDPR, more research could be done to learn how to support teams in fulfilling those requirements while also making the privacy enhancement industry better for everyone.

As for the state of privacy by design as a privacy engineering methodology, there is no general consensus on the best way to advance apart from the fact that privacy needs to be pushed further as an integral part of the software development flow. This is clear from the problems with the evaluation of the agile privacy by design model. Many advocates and researchers in Europe have pushed for privacy by design to be adopted as a global standard, but there is still much work to be done until it is a universally understood, well-defined tool for privacy engineering. As ENISA stated in their 2014 report, policy makers, researchers, media and legislators all need to pitch in to provide the software development industry with the kinds of practical tools that enable the intuitive adoption of privacy engineering through privacy by design principles [13]. Better practical and regulatory guidelines need to be developed still and privacy engineering needs a stronger legislative mandate to push compliance [13]. Privacy standards need to be globally unified to enable actors in the global marketplace to interoperate and collaborate to build better privacy for the products and services of tomorrow.

The privacy landscape in general is looking brighter by the day, but the few dark clouds around are darker than one would hope. Governments need to start enforcing on the systematic breach of civilians' privacy by authorities and intelligence agencies. A proper balance must be defined between the rights of the individual and the benefit of the majority. The situation in Europe is not as grim as globally and the global development should be lobbied to follow the same progress with regard to the protection of individuals' rights.

5.2 Future research

A trial run of the agile privacy by design model in a controlled project environment would be a great first study to continue on. Validation of the model is still an open question. Perhaps real project experience on the issues and strengths of the model might aid in empirically validating or invalidating it in its proposed form.

It would also be an interesting study to attempt to gain statistically significant feedback on the implementation of privacy by design in real projects. Privacy by design is a repeating topic in privacy research, but not a lot of work exists trying to quantify the experience of applying it. It would also be a great opportunity for a comparative study between projects that employ a defined privacy by design framework and projects that do not follow a framework model.

Additionally, research on the developer and project management perception of applying agile privacy by design would be beneficial for further development of the model. As it stands, a large part of the model is reliant on the privacy expertise of the developer. The model could be more widely useful if it was possible to enhance it so that it provides more guidance towards a safe default set of data protection targets for example.

After the enforcement of the GDPR begins and development teams gain experience in the unique challenges it brings forth, it would be also an opportunity to research the application of the model globally. Whether the success or failure of the privacy by design approach as a legislative measure causes pressure to adopt or abandon development of further regulatory measures like it is also a long-term research opportunity.

6. CONCLUSIONS

In this thesis, it has been shown that it is possible to apply the privacy by design principles in a defined iterative framework. The adaptation of agile practices for specialized purposes is an effective tool to guide the focus of the workflow. This framework does not solve issues that are inherent with privacy by design itself, but it eases the introduction of other related privacy and threat discovery tools into the agile development workflow. The application of the framework will also leave the project team with a better understanding of the structure of their system and provide a better starting point for thinking about operative privacy issues than using no framework at all.

The background research on the issues related to implementing privacy controls into complex systems has shown that privacy has often been sacrificed over functionality. The application of technical measures as a panacea or so-called shopping for privacy can mask underlying structural issues in development practices. Privacy is a complex field and the lack of enforcement action on the regulations gives too much incentive for non-compliance. The adversarial views between privacy and security, privacy and functionality or privacy and business have been problematic for the development of more open and fair information processing practices. Also, the importance of transparency and visibility into the details of data processing cannot be overstated.

It has also been learned that there are multiple open issues in applying privacy by design as a regulated mandatory design methodology and that those issues might not have simple solutions at all. However, these issues may be ameliorated with the introduction of more accurate guidelines on the application of privacy by design in practice. It is important to remember that the GDPR is the first time that privacy by design is mandated by regulation at such scale and it is to be expected that there will be practical issues to resolve.

Finally, it can be said that the development of privacy regulation has been moving in a better direction in the last few decades both globally and in Europe. The internet society is still a relatively young phenomenon. Governments and legislators need to work faster than ever in order to protect the rights and freedoms of individual citizens in the ubiquitous information society. A lot of work is required to provide governments and businesses with the necessary tools to fully embrace the opportunities of the modern era in a privacy positive manner.

REFERENCES

[1] L. Rainie, S. Kiesler, R. Kang ja M. Madden, ”Anonymity, Privacy, and Security Online,” Pew Research Center, [Online]. Available: http://www.pewinternet.org/2013/09/05/anonymity-privacy-and-security-online/. [Haettu 13 October 2017].

[2] L. Rainie, “The state of privacy in post-Snowden America,” Pew Research Center, [Online]. Available: http://www.pewresearch.org/fact-tank/2016/09/21/the-state-of-privacy-in-america/. [Accessed 13 October 2017].

[3] M. Poppendieck ja T. Poppendieck, Lean Software Development: An Agile Toolkit, Addison Wesley, 2003.

[4] W. Cunningham, “Manifesto for Agile Software Development,” [Online]. Available: http://agilemanifesto.org/. [Accessed 3 June 2017].

[5] G. Smith ja A. Sidky, Becoming Agile in an imperfect world, Greenwhich: Manning, 2009.

[6] K. Pries ja J. Quigley, Scrum Project Management, Boca Raton: CRC Press, 2011.

[7] R. Pichler, Agile Product Management with Scrum, Boston: Pearson Education, 2010.

[8] European Council, “Convention for the Protection of Human Rights and Fundamental Freedoms,” [Online]. Available: https://ec.europa.eu/digital-single-market/sites/digital-agenda/files/Convention_ENG.pdf. [Accessed 14 May 2017].

[9] D. J. Solove, Understanding Privacy, Harvard University Press, 2008.

[10] European Council, “General Data Protection Regulation (Regulation 2016/679),” 4 May 2016. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN. [Accessed 17 January 2017].

[11] K. Wuyts, Privacy Threats in Software Architectures, Heverlee: KU Leuven - Faculty of Engineering Science, 2015.

[12] A. Pfitzmann and M. Hansen, “A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management,” August 2010. [Online]. Available: http://dud.inf.tu-dresden.de/Anon_Terminology.shtml. [Accessed 2017 July 2017].

[13] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. Le Métayer, R. Tirtea ja S. Schiffner, ”Privacy and Data Protection by Design - from policy to engineering,” European Union Agency for Network and Information Security (ENISA), Brussels, 2014.

[14] M. Landesberg, T. Levin, C. Curtin and O. Lev, “Privacy Online: A Report to Congress,” June 1998. [Online]. Available: https://www.ftc.gov/sites/default/files/documents/reports/privacy-online-report-congress/priv-23a.pdf. [Accessed 19 July 2017].

[15] A. Cavoukian, ”Privacy by Design - The 7 foundational principles,” Ontario Information and Privacy Commissioner, Ontario, 2011.

[16] I. Kroener and D. Wright, “A Strategy for Operationalizing Privacy by Design,” *The Information Society,* vol. 30, no. 5, pp. 355-365, 2014.

[17] C. Bier, P. Birnstill, E. Krempel, H. Vagts and J. Beyerer, “Enhancing Privacy by Design from a Developer’s Perspective,” *Privacy Technologies and Policy. APF 2012. Lecture Notes in Computer Science,* vol. 8319, pp. 73-85, 2012.

[18] D. Klitou, “A Solution, But Not a Panacea for Defending Privacy: The Challenges, Criticism and Limitations of Privacy by Design,” *Privacy Technologies and Policy. APF 2012. Lecture Notes in Computer Science,* vol. 8319, pp. 86-110, 2014.

[19] Article 29 Data Protection Working Party, ”Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is “likely to result in a high risk” for the purposes of Regulation 2016/679,” European Commission, Brussels, 2017.

[20] P. Schaar, ”Privacy by Design,” *Identity in the Information Society,* osa/vuosik. 3, nro 2, pp. 267-274, 2010.

[21] Article 29 Data Protection Working Party, ”Opinion 15/2011 on the definition of consent,” European Commission, Brussels, 2011.

[22] A. Schwartz and S. Cope, “Pass the Protecting Data at the Border Act,” 13 October 2017. [Online]. Available: https://www.eff.org/deeplinks/2017/10/pass-protecting-data-border-act.

[23] Open Web Application Security Project, “Top 10 Privacy Risks Project,” 8 April 2016. [Online]. Available: https://www.owasp.org/images/0/0a/OWASP_Top_10_Privacy_Countermeasures_v1.0.pdf. [Accessed 13 June 2017].

[24] IT Governance Privacy Team, U General Data Protection Regulation (GDPR) - An Implementation and Compliance Guide, IT Governance, 2016.

[25] S. Varotto, “The Schrems decision, the EU-US Privacy Shield and the necessity to rethink how to approach cross border personal data transfers at global level,” *Communications Law,* vol. 21, no. 3, pp. 78-87, 2016.

[26] European Union Agency for Network and Information Security, ”Recommendations for a methodology of the assessment of severity of personal data breaches,” European Union Agency for Network and Information Security (ENISA), Brussels, 2013.

[27] E. Derby ja D. Larsen, Agile Retrospectives: Making Good Teams Great, Pragmatic Bookshelf, 2006.

[28] P. Anthonysamy, P. Greenwood and A. Rashid, “A Method for Analysing Traceability between Privacy Policies and Privacy Controls of Online Social Network,” *Privacy Technologies and Policy. APF 2012. Lecture Notes in Computer Science,* vol. 8319, pp. 187-202, 2012.

[29] S. Hernan, S. Lambert, T. Ostwald and A. Shostack, “Threat modeling - uncover security design flaws using the STRIDE approach,” *MSDN Magazine,* pp. 68-75, January 2006.

[30] C. Larman, Applying UML and Patterns, Prentice Hall, 2004.

[31] J. Seppänen, Scrum - From theory to practce in software development, M.Sc. thesis, Tampere University of Technology, 2016.