GDPR-Based User Stories in the Access Control Perspective

Cesare Bartolini[1], Said Daoudagh[2,3(✉)], Gabriele Lenzini[1], and Eda Marchetti[2]

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg City, Luxembourg
{cesare.bartolini,gabriele.lenzini}@uni.lu
[2] Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" (ISTI), Consiglio Nazionale delle Ricerche (CNR), via G. Moruzzi 1, 56124 Pisa, Italy
{said.daoudagh,eda.marchetti}@isti.cnr.it
[3] Computer Science Department, University of Pisa, Pisa, Italy

Abstract. Because of GDPR's principle of "data protection by design and by default", organizations who wish to stay lawful have to re-think their data practices. Access Control (AC) can be a technical solution for them to protect access to "personal data by design", and thus to gain legal compliance, but this requires to have Access Control Policies (ACPs) expressing requirements aligned with GDPR's provisions. Provisions are however pieces of law and are not written to be immediately interpreted as technical requirements; the task is thus not straightforward. The *Agile software development methodology* can help untangle the problem. It has dedicated tools to describe requirements and one of such them, *User Stories*, seems up to task. Stories are concise yet informal descriptions telling who, what and why something is required by users; they are prioritized in lists, called *backlogs*. Inspired by these Agile tools this paper advances the notion of *Data Protection backlogs*, which are lists of User Stories about GDPR provisions told as technical requirements. For each User Story we build a corresponding ACP, so enabling the implementation of GDPR compliant AC systems.

Keywords: Access Control Policy (ACP) · General Data Protection Regulation (GDPR) · User Story

1 Introduction

Nowadays, the Information Technology (IT) domain is moving towards systems with growing complexity, where digitalization, artificial intelligence, interconnection and mobility are some key factors. Indeed, in their multidisciplinary nature, they require an extensive deployment of advanced Information and Communication Technologies (ICTs), as well as the adoption of effective measures for strengthening security, trust, dependability and privacy. These aspects have to

© Springer Nature Switzerland AG 2019
M. Piattini et al. (Eds.): QUATIC 2019, CCIS 1010, pp. 3–17, 2019.
https://doi.org/10.1007/978-3-030-29238-6_1

be considered over the whole Software Development Life Cycle (SDLC), from the gathering of the requirements to the deployment and subsequent maintenance of the system.

Over the last decade, especially for small and medium enterprises, *Agile Software Development (ASD)*, first introduced in the Agile Manifesto [10], and its subsequent evolutions such as eXtreme Programming (XP) and Scrum [14] are becoming commonly-adopted software development processes. Basically, ASD is an iterative approach that focuses on incremental specification, design and implementation, while requiring a full integration of testing and development. In this development process, a common means of capturing the user needs and describing the value that the user would get from a specific functionality is the so-called *User Story* [1]. From a practical point of view, a User Story focuses on a requirement written according to a specific format (see Subsect. 2.1), and guidelines on how to implement it. Usually, depending on the granularity of the story, different names can be used for defining its contents: large ones may be known as *Epics*, and small ones as *Features*, *User Stories*, and *Tasks* [1].

However, small organisations and software development groups could not expend the effort (in terms of budget and time) needed to collect and implement all the required User Stories prior to release. When the missing stories refer to privacy requirements, the side effect is to release softwares with high privacy risks [3].

With the entering into force of the General Data Protection Regulation (GDPR) (see Subsect. 2.2) this situation is not affordable anymore, because it is changing how *Personal Data* should be processed. Indeed, the GDPR imposes several duties on the *Controller* and the *Processor*, i.e., the data managers, concerning the processing of *Personal Data*, i.e., any information related to an identified or identifiable natural person called the *Data Subject*.

Additionally, the GDPR defines a system of fines to induce controllers and processors to be compliant with its provisions. Thus, the controller and processor need to demonstrate the compliance with the GDPR as required by the *Accountability principle* (Art. 5.2). However, this is not a trivial problem as it involves the definition of specific *purposes*, the management of the *consent* given by the *Data Subject* whose personal data are referring to, and the need to demonstrate compliance with the implemented GDPR's provisions.

Within the Agile development, among the proposed solutions to tackle security issues and vulnerabilities in an efficient way, one that is currently taking place is the possibility of using security backlogs to drive the software development work. In the Agile context, a backlog represents a prioritized features list describing the functionalities to be included in the final product [1]. These backlog items are often provided in the form of User Stories [3]. The set of security backlogs is therefore a list of ready-made specification of security items (requirements and task descriptions) useful for the implementation. An example of a security User Story related to access control is reported in Subsect. 2.1.

Following this tendency, the contribution of this paper consists in three main parts: (i) introduce the concept of *Data Protection Backlog* that contains User

Stories based on GDPR requirements; (ii) map specific provisions of the GDPR to User Stories; and (iii) provide, for each User Story, the corresponding implementation guidance so as to assure a GDPR-compliant design.

Considering in particular this last point, i.e., to ensure the GDPR compliance, it would be helpful to carry out the processing of personal data automatically and in compliance with the obligations imposed by the GDPR.

To this purpose, in this paper we want to move a step towards a compliant implementation of the GDPR, by encoding the User Stories, and consequently the GDPR provisions, as Access Control Policies (ACPs). A valid solution to minimize errors and issues in the GDPR enforcement is to rely on a consolidated, verified and predefined structure of ACPs [28]. In line with this tendency, for each identified User Story, this paper provides a GDPR-based Access Control Policy (ACP) template for each provision related to access control. Indeed, the templates represent meaningful, concrete and predefined blocks for ACP specification, that can be adopted and refined for the different scenarios, so as to overcome possible misinterpretations and reduce security and privacy risks.

As a final result, the set of User Stories, associated with the proper ACP templates, would be a valid starting point for privacy requirements specification, and a generic guidance for who are facing to the problem of GDPR implementation. When a new development starts, the developer could pick up the related predefined User Story and easily implement it.

Outline. We recall User Stories and the GDPR in Sect. 2, where we also give an overview of Access Control (AC). In Sect. 3 we illustrate the related work. The proposed GDPR based User Stories model is described in Sect. 4 and in Sect. 5 we shows the process to the derive the *Data Protection Backlog* containing the User Stories and the associated ACPs. In Sect. 6 we conclude and point out the future work.

2 Background

This section introduces the main concepts used in the rest of this paper: User Stories, the GDPR and AC.

2.1 User Stories

User Stories are an important part of an Agile development process because they represent a valid means to writing simple and understandable requirements [30].

Currently, their adoption is massively growing [13] and several definitions are available [16]. However, most of them agree on the fact that commonly a User Story is a short, simple description of a feature from the perspective of a end user or customer of the system. A User Story typically presents the following structure:

> As a *[end user]*, I want to achieve *[goal]* so that *[I realize the following benefit of]*.

An example of a security User Story related to ACP, reported in [1], is as follows:

> As *[an information security manager]* I want *[that it is clearly defined which user accounts are authorised for which activities]* so that *[the effectiveness and correctness of access controls can be checked]*.

One key factor of the widespread use of User Stories is that they can be written at different levels of detail. They can cover large amount of functionalities and in this case are generally known as *Epics*. However, an epic is generally too large for being easily implementable into a single Agile iteration, thus it usually split into multiple smaller User Stories before it is worked on. Thus is for instance the case of features, User Stories, and tasks [1]. In some cases, User Stories are detailed more by adding conditions of satisfaction, i.e., a high-level description of what needs to be true after the Agile User Story is complete.

There is not a specific customer or user role for writing User Stories, but having a common set of product backlog of Agile User Stories is an important factor for a successful development. Indeed, the product backlog can be used to select and prioritize the list of the functionalities that have to be developed in different iterations of the Agile process.

2.2 General Data Protection Regulation

The General Data Protection Regulation (GDPR)[1] defines, among others, several data protection principles and Data Subject's rights. The aim of the new regulation is to strengthen the rights of individuals over their own data, and at the same time to make organizations more accountable with respect to the previous directive. In addition, the GDPR contributes to the harmonization of the previous fragmented data protection laws across the EU, so as to ensure equal right to privacy and to data protection.

The mandatory part of the GDPR is organized in chapters and contains 99 articles. Art. 4 defines *Personal Data* as "any information relating to an identified or identifiable natural person ('data subject')". This means that a *Data Subject* is a natural person (a living human being) whose data are managed by a *Controller*.

The *purpose* of the *processing* of personal data is determined by the controller, and this "processing shall be lawful only if and to the extent that at least one of the" six legal bases "applies" (Art. 6). In particular, one of those legal bases is the consent given by the data subject "to the processing of his or her personal data for one or more specific purposes" (Art. 6.1(a)). *Consent* is defined as "any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her" (Art. 4.11).

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

However, the GDPR sets the *Conditions for Consent* in Art. 7. On one hand, "the controller shall be able to demonstrate that the data subject has consented to processing of his or her personal data", and this is line with the *Accountability* principle defined in Art. 5.2. On the other hand, "the data subject shall have the right to withdraw his or her consent at any time" and "it shall be as easy to withdraw as to give consent"(Art. 7.3).

The GDPR also sets other fundamental rights of the data subject, such as the right of access (Art. 15) and the right to data portability (Art. 20), and several principles that the controller and processor shall abide to. For instance, the "integrity and confidentiality" principle imposes the controller to use "appropriate technical or organisational measures" to "ensure appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage" (Art. 5.1(f)). Paragraph 2 of the same article introduces the Accountability principle, according to which "[t]he controller shall be responsible for, and be able to demonstrate compliance with, paragraph 1".

Finally, the controller shall "taking into account the state of the art [...] both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures [...] to integrate the necessary safeguards into the processing in order to meet the requirements of this regulation and protect the rights of data subjects." (Art. 25.1). This is the so-called principle of data protection by design.

These brief excerpts from the GDPR show the complexity of the regulation and hint at the subsequent difficulty to introduce such legal concepts into software development environment. A set of User Stories summarizing the main important tasks and features to be implemented in the different contexts can be a valid solution to comply the GDPR requirements. Furthermore, having a formalized representation of the legal concepts and the relations among them could help and facilitate the definition of a more consistent set of policies governing a GDPR compliant data access. Hence, the proposal of this paper: a way to represent GDPR requirements as User Stories organized in a *Data Protection Backlog*, i.e., *Privacy Backlog*, and the associated ACPs in a uniform, simple and processable format.

2.3 Access Control

Access Control is a mechanism used to restrict access to data or systems, based ACPs, i.e., a set of rules that specify who (e.g., Controller, Processor or Data Subject) has access to which resources (e.g., Personal Data) and under which circumstances [24]. One of the emerging AC models is the Attribute-Based Access Control (ABAC) model.

The basic idea of ABAC is to employ attributes (characteristics or properties) of different entities to make access control decisions regarding a subject's (e.g., user o process) access on an object (e.g., file or database) in a system. The AC decisions are evaluated based on authorization policies specified by an

administrator using a policy specification language. ABAC authorization policies are a set of rules defined based on the attributes of subjects and objects as well as other attributes, such as contextual attributes.

The National Institute of Standards and Technology (NIST) defines Attribute-Based Access Control (ABAC) as "[a]n access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions" [12].

This definition lists various key concepts. In particular, *attributes* are characteristics of the subject or object, or environment conditions, containing information given by a name-value pair. A *subject* is a human user, legal entity (e.g., Data Subject or Controller) or an abstract entity, such as a device that issues access requests to perform operations on objects/resource (e.g., Personal Data). Subjects are assigned one or more attributes.

An *object* is a resource for which access is managed by the ABAC system, such as devices, files, records, tables, processes, programs, networks, or domains containing or receiving information. In the context of the GDPR, objects can be either Personal Data or records of processing activities.

An *operation* is the execution of a function upon an object. Operations include read, write, edit, modify and erase. *Environment conditions* represent the operational or situational context in which access requests occur (e.g., current time or location of a user). By referring to the GDPR we assume that the consent is a contextual information. The *policy* is the representation of rules or relationships that allow to determine if a requested access should be allowed, given the values of the attributes of the subject, object, operation and environment conditions.

3 Related Work

In this section, we provide a non-exhaustive overview of the proposals dealing with the main topics of this paper: representing security and privacy by means of User Stories, and how to put in relation AC environment and the GDPR.

3.1 Security and Privacy by Means of User Stories

An important innovation for speeding up the development of software has been the introduction of Agile development and the Scrum methodology. Over the last years, literature has moved an important criticism to these kind of approaches because they mostly ignore the security risk management activity [1,4,23,29]. Thus the concepts of security should be considered during all stages of the software development life cycle, in Agile environment this commonly means integrating security principles in terms of security backlog [3,4]. The security backlog is a set of ready-made User Stories that can be used to cover the security requirements [25]. This new backlog can be used to manage and mitigate the security risks associated with the software [23,29].

The introduction of GDPR requirements in the secure software development adopted into the Agile processes for discovering and solving security threats is not sufficient anymore to guarantee the required privacy level, and few proposal are recently targeting this issue. Among these, in [23] the authors propose a Threat Poker method to exercise both security risks and privacy risks and evaluate the effort needed to remove the corresponding vulnerabilities in the software developed. However, the proposal is mainly focused on the estimation of the seriousness of security and/or privacy risks during software development. Similarly, in [17] the authors present an Agile process for the definition of security and privacy in terms of User Stories, in order to develop a framework to manage Personal Health Information. In particular, the authors highlight the need for suitable policies and procedures for data security and privacy management, so as to make the framework compliant with regulations.

This paper has similar aims, i.e., to contribute to the definition of privacy-related User Stories, but in addition it also provides a practical means for defining concrete privacy policies specification.

3.2 Access Control and the GDPR

Several works use AC as the main means of protecting personal data. For example, an initial proposal [7] for an automatically-enforceable policy language for access and usage control of personal information aims at transparent and accountable data usage. Authors in [27] give a formal definition of the consent as stated in the GDPR, applied in an IoT context.

The work in [11] presents an ABAC model for smart cars ecosystem to take into account the individual user privacy preferences for allowing or denying service notifications, alerts and operations to on-board resources.

Unfortunately, the proposals are fragmented and only consider a few aspects that can be traced back to some of the principles introduced by the GDPR. This paper would like to exploit the ideas proposed by the aforementioned papers towards ACPs specification for GDPR compliance, and to provide a systematic approach to gather as many GDPR requirements as possible, providing corresponding GDPR-compliant ACPs. This paper is inspired by a proposal [26] that describes a new semi-formalized, constrained natural language format for User Stories. The format uses variables to precisely correlate various parts of the story with a predefined format, to express strictly-defined operators in a (almost) natural language. The authors also showed a possible way to extract access control information for role-based access control from this format.

4 GDPR-Based User Stories Conceptual Model

In an attempt to comply with the principle of data protection by design, laid out by Art. 25.1 of the GDPR, we detail a methodology for defining privacy-based User Stories and gathering them to ACPs requirements directly from the GDPR. From a practical point of view, this means first extracting, in an Agile

perspective, User Stories that represent atomic privacy or legal requirements to be implemented so as to comply by-design with the GDPR. Then, considering systems that enforce an AC, defining an actionable list of simple AC system specifications which address the core requirements demanded by the GDPR.

The proposal would like to contribute to: (i) an incremental development of the AC system, by guaranteeing that, by design, it maintains compliance with the GDPR; (ii) the Data Protection Impact Assessment (DPIA) along the development of the system; (iii) a mapping between the implemented functionalities and the corresponding GDPR provisions. This will help to create a traceability mechanism useful for demonstrating GDPR compliance, as required by the *Accountability* principle.

The User Stories are built taking in consideration the GDPR concepts of *Data Subject*, *Controller*, *Processor*, *Data Protection Officer (DPO)*, and *Personal Data*.

The conceptual model for User Stories, used for the derivation of the actionable list, is shown in Fig. 1. It is composed of three sub-models: the GDPR Model, User Stories Model and AC Model. The sub-models are combined into the process followed for going from the definition of the User Stories to specification of AC policies.

The sub-models have been voluntarily kept separated to increase the possible generalization of the paper proposal. Indeed, the GDPR Model and AC Model could be replaced by any other legal regulation or legislation which is suited for automatic enforcement.

The remainder of this section provides specific details about these sub-models.

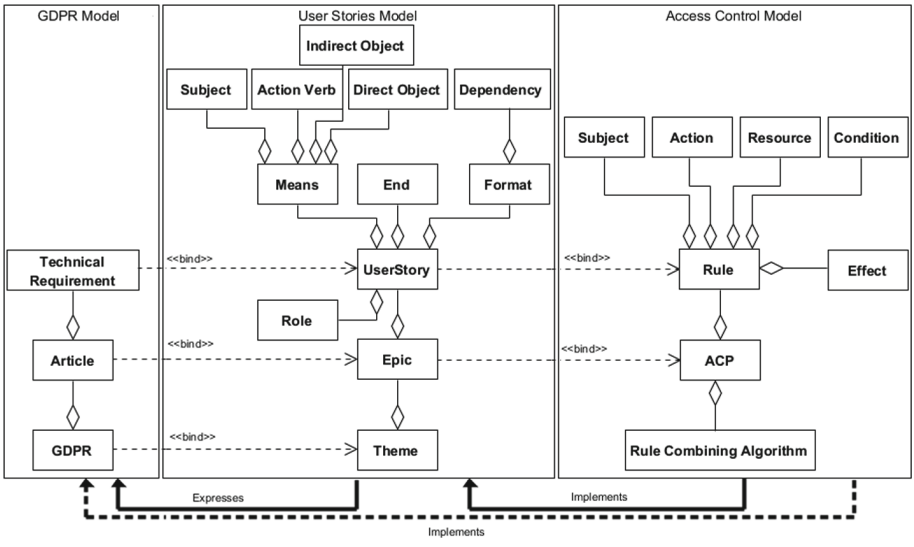

Fig. 1. The conceptual model of GDPR-focused user stories.

4.1 User Stories Model

The User Story used in our model is a modified version of that introduced in [15]. More precisely, we do not consider the *Clarification* and the *Quality* elements of the *End* component; we eliminate the *Adjective* element from the *Means* component; and finally, we introduce the *Theme* component as abstract level to better bind the User Stories to the GDPR.

As depicted in Fig. 1, a User Story always includes one relevant *Role*, which is associated with the stakeholder or legal entity that expresses the need. Currently, due to the complexity of the GDPR text, the number of proposals trying to provide a conceptual model of the regulation [6,21,22] is increasing in literature. Among those available, and in order to relying on a formal base for the role specification, in this paper we rely on the formalization provided by the Privacy Ontology (PrOnto) [19,20]. The details of this ontology are out of the scope of this paper; suffice it to say that the stakeholders that we use in the proposal are *Controller*, *Processor*, *Data Protection Officer* (*DPO*), *Data Subject* and *Supervisory Authority*.

The *Format* of the User Story is a predefined template in which the role, means, and optional end(s) are specified. As described in Sect. 2 we refer to the most widespread format introduced in [8] which consist of:

> As a [type of user], I want [goal], so that [some reason].

Differently, *Means* can have different structures that can be used to represent different types of requirements. Means have three common grammatical elements: (i) a subject with an aim; (ii) an action verb that expresses the action related to the feature being requested; and (iii) a direct object (and optionally an indirect object) on which the subject executes the action.

The End of a User Story explains why the means are requested. However, User Stories often include other types of information, such as dependency on another functionality, i.e., implicit references a functionality which is required for the means to be realized. This is useful in the context of the regulations since legal text often use the cross-reference mechanism between articles.

In the GDPR context, a possible User Story related to Art. 30.4 could be:

> As a *[Supervisory Authority]*, I want *[to access the record of processing activities]*, so that *[I can monitor those processing operations]*.

4.2 The GDPR Model

In this study we model the GDPR only from a structural point of view. As described in Sect. 2, the mandatory part of the GDPR is composed of ninety-nine articles organized in chapters; some chapters are then broken in sections. The GDPR's articles present a structure that involves at least other two levels (paragraphs and letters). Finally, each article may include one or more technical requirements.

In order to be aligned with the structure defined in User Stories model, we model the GDPR as an aggregation of articles. More precisely, we do not consider

the recitals, and we collapsed all the aforementioned complex structure of the regulation in a more simple one that includes only three levels: *GDPR* → *Article* → *Technical Requirement*.

This simple structure helps in binding the GDPR core code with the concept of Theme in Agile terminology; then, the articles represent Epics which contain one or more small and manageable technical requirements, each expressed by means of a User Story.

4.3 The Access Control Model

An ACP defines the AC requirements of a protected system, i.e., a set of AC *Rules* that specify who (e.g., Controller, Processor or Data Subject) has access to which resources (e.g., Personal Data) and under which circumstances [24]. The AC rule is often specified using Natural Language Access Control Policy (NLACP)), that presents the following structure: *[Subject] can [Action] [Resource] if [Condition]* [12].

The Access Control Model used in this proposal is a simplified version of the Policy Language Model provided by the eXtensible Access Control Markup Language (XACML) standard [18]. Even simple, the model captures all the essential concepts for the design of both simple and more complex ACPs.

As in Fig. 1, the model consists of *Rule* class, which represents the most elementary unit of policy enforceable by an Access Control System (ACS). The rule is composed of one single *Subject*[2], one single *Action*, one single *Resource* and one single optional *Condition*. The *Effect* associated with the rule represents the rule-designer's intended consequence of a *True* evaluation for the rule. The usual two values allowed for the rule's effect are: *Permit* and *Deny*. As depicted in Fig. 1, the rule represents an expression of an atomic technical requirement described by a User Story. The ACP class is a composition of rules and *Rule Combining Algorithm* which defines strategy by which the results of evaluating the rules are combined when the ACS evaluates the policy.

5 User Stories Related to Access Control

The process we used to define the set of User Stories, related to the provisions of the GDPR and the AC rules, is composed of three steps (see Fig. 2): (1) *GDPR Articles Selection*; (2) *User Stories Definition*; (3) *GDPR AC Rules Definition*.

GDPR Articles Selection. The input of the process is the GDPR text. Firstly, we selected only the mandatory part of the GDPR which consists of

[2] Note that the Subject expressed in this model is different from the one defined in the User Stories Model: the Subject in that model represents a grammatical function in the formulation of the means; while Subject in the AC domain represents an active entity which covers a role. The Subject in this model is an entity that can semantically be correlated with the Role entity in the User Stories Model.

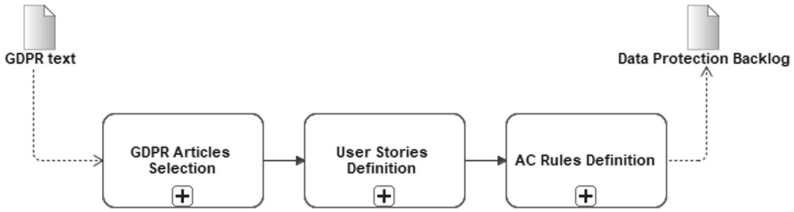

Fig. 2. GDPR-focused user stories definition process.

ninety-nine articles; for each article, we decided whether is related to AC concept, i.e., AC language or AC mechanism, and consequently we created an Epic associated to the current article. The result of this step was the section of forty-one Epics (GDPR articles) related to AC. Specifically, three of them were concerning only AC mechanism; eight were referring only ACPs, and thirty articles related to both ACPs and AC mechanism. For more details about this step we refer to our previous work in [5].

User Stories Definition. For each article identified in the previous phase, we extracted one or more technical requirements and defined a specific User Story for each of them. Thus, the User Stories were added to the Epic associated with the current article. In order to trace the covered GDPR's articles during the Agile development process, we defined a for each Epic an identifier (named EpicID[3]) able to find the GDPR's article the Epic is referring to. Similarly, we defined an identifier for each User Story (called UserStoryID[4]) with the purpose to the specific part of the GDPR's article the User Story related to (e.g., the paragraph or the letter of the article).

GDPR AC Rules Definition. The final step deals with the translation of the technical requirements associated with the AC language, and consequently we defined an AC rule for each User Story conceived in the previous step. It is out of the scope of this paper going into details of the procedure of extracting ACPs. In literature there exist different proposals for the derivation of ACPs from the natural language [2,31] or controlled natural language [9]. In our previous work [5] we defined a systematic approach for deriving ACPs directly from the GDPR, and we refer to it for more details about this step.

As in Fig. 2, the result of this process is a *Data Protection Backlog*, i.e., a Privacy Backlog containing a set of AC rules organized in User Stories, Epics and Theme. This is a ready solution to be used during the Agile development of an ACs system aligned with the GDPR requirements.

[3] The identifier EpicID has the following structure: GDPR.Epic.Article.[articleNumber].

[4] The identifier UserStoryID has the following structure: [EpicID].[Paragraph Number].[letter].US.[progressiveNumber].

For space limitation, in Table 1 we present an extract of the defined Data Protection Backlog. The User Stories are reported from both the perspective of the Data Subject and the Controller.

The table is composed of three columns: the column **Article** (first column) contains the GDPR's articles. The column **User Story** contains the GDPR-based User Stories defined. Finally, the third column contains the AC rules related to the User Stories.

Table 1. GDPR-focused user stories: controller and data subject perspectives

Article	User story	AC rule
Art. 6.1(a)	As a [Controller], I want [to process Personal Data only if Data Subject has given consent for one or more specific purpose], so that [the processing shall be lawful].	[Controller] can [Process] [Personal Data] If [PersonalData.purpose = Processing.purpose AND PersonalData.purpose.consent = TRUE]
Art. 7.3	As a [Data Subject], I want [to withdraw my consent], so that [I can exercise my right as stated in Art. 7.3]	[Data Subject] can [Withdraw] [PersonalData.purpose.consent] If [PersonalData.owner = DataSubject AND PersonalData.purpose.consent = TRUE]
Art. 15.1	As a [Data Subject], I want [to access my Personal Data and all the information], so that [I can be aware about my privacy]	[Data Subject] can [Action = access] [PersonalData] AND [Resource = PersonalData.purposes] AND [Resource = PersonalData.categories] if [PersonalData.owner = Data Subject]
Article 15.3	As a [Data Subject], I want [to download a copy of my Personal Data], so that [I can check their correctness]	[Data Subject] can [download] [Personal Data] If [PersonalData.owner = Data Subject]

6 Conclusions and Future Work

This paper presents an Agile methodology to gather access control requirements from the GDPR by using the concept of User Stories. This methodology is a first step towards a formal definition of access control solutions addressing GDPR requirements in Agile environment. To the best of the authors' knowledge, an Agile methodology for the specification of User Stories, organized in Data Protection Backlog, i.e., Privacy Backlog, aimed at extracting legal ACPs from the GDPR is novel. Although grounded in a domain-related implementation (i.e., the GDPR), the Agile methodology yields a more general spectrum, since it can be applied to different data protection legislation that encodes ACPs specification.

In our case, the generation of a set of ACPs aligned with the GDPR was conceived in three phases: the selection of GDPR's articles related to access

control; the definition of a Data Protection Backlog containing User Stories extracted from the selected GDPR's articles; and finally, the definition of access control rules, each related to a specific User Story. Having a User Story (and consequently an access control rule) related to a specific GDPR provision helps to detect the rules that need to be updated when the regulation changes.

As a future work, we are planning to consider the GDPR requirements referring access control mechanisms, i.e., requirements from the architectural point of view. Future work includes also the validation of the User Stories by different Agile development teams in the context of an ongoing European project that addresses key regulations such as the GDPR.

References

1. Ahola, J., et al.: Handbook of the secure agile software development life cycle. University of Oulu (2014)
2. Alohaly, M., Takabi, H., Blanco, E.: Automated extraction of attributes from natural language attribute-based access control (ABAC) policies. Cybersecurity **2**(1), 2 (2019)
3. Asthana, V., Tarandach, I., O'Donoghue, N., Sullivan, B., Saario, M.: Practical security stories and security tasks for agile development environments, July 2012
4. Azham, Z., Ghani, I., Ithnin, N.: Security backlog in scrum security practices. In: 2011 Malaysian Conference in Software Engineering, pp. 414–417. IEEE (2011)
5. Bartolini, C., Daoudagh, S., Lenzini, G., Marchetti, E.: Towards a lawful authorized access: a preliminary GDPR-based authorized access. In: 14th International Conference on Software Technologies (ICSOFT 2019), Prague, Czech Republic, 26–28 July 2019, pp. 331–338 (2019)
6. Bartolini, C., Giurgiu, A., Lenzini, G., Robaldo, L.: Towards legal compliance by correlating standards and laws with a semi-automated methodology. In: Bosse, T., Bredeweg, B. (eds.) BNAIC 2016. CCIS, vol. 765, pp. 47–62. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67468-1_4
7. Cerbo, F.D., Martinelli, F., Matteucci, I., Mori, P.: Towards a declarative approach to stateful and stateless usage control for data protection. In: WEBIST, pp. 308–315. SciTePress (2018)
8. Cohn, M.: User Stories Applied: For Agile Software Development. Addison-Wesley Professional, Boston (2004)
9. Fatema, K., Debruyne, C., Lewis, D., O'Sullivan, D., Morrison, J.P., Mazed, A.: A semi-automated methodology for extracting access control rules from the European data protection directive. In: 2016 IEEE SPW, pp. 25–32, May 2016
10. Fowler, M., Highsmith, J., et al.: The agile manifesto. Softw. Dev. **9**(8), 28–35 (2001)
11. Gupta, M., Benson, J., Patwa, F., Sandhu, R.: Dynamic groups and attribute-based access control for next-generation smart cars. In: CODASPY 2019, Richardson, TX, USA, 25–27 March 2019 (2019)
12. Hu, C.T., et al.: Guide to attribute based access control (ABAC) definition and considerations [includes updates as of 02-25-2019]. Technical report (2019)
13. Kassab, M.: The changing landscape of requirements engineering practices over the past decade. In: 2015 IEEE EmpiRE, pp. 1–8, August 2015
14. Kniberg, H.: Scrum and XP from the Trenches (2015). Lulu.com

15. Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M., Brinkkemper, S.: Improving agile requirements: the quality user story framework and tool. Requirements Eng. **21**(3), 383–403 (2016)
16. Lucassen, G., Dalpiaz, F., Werf, J.M.E.M., Brinkkemper, S.: The use and effectiveness of user stories in practice. In: Daneva, M., Pastor, O. (eds.) REFSQ 2016. LNCS, vol. 9619, pp. 205–222. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30282-9_14
17. McCaffery, F., et al.: A process framework combining safety and security in practice. In: Larrucea, X., Santamaria, I., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2018. CCIS, vol. 896, pp. 173–180. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97925-0_14
18. OASIS: eXtensible Access Control Markup Language (XACML) Version 3.0, January 2013. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
19. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Legal ontology for modelling GDPR concepts and norms. In: Legal Knowledge and Information Systems: JURIX 2018, vol. 313, p. 91. IOS Press (2018)
20. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: PrOnto: privacy ontology for legal reasoning. In: Kő, A., Francesconi, E. (eds.) EGOVIS 2018. LNCS, vol. 11032, pp. 139–152. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98349-3_11
21. Pandit, H.J., Fatema, K., O'Sullivan, D., Lewis, D.: GDPRtEXT - GDPR as a linked data resource. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 481–495. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_31
22. Pandit, H.J., Lewis, D.: Modelling provenance for GDPR compliance using linked open data vocabularies. In: PrivOn@ ISWC (2017)
23. Rygge, H., Jøsang, A.: Threat poker: solving security and privacy threats in agile software development. In: Gruschka, N. (ed.) NordSec 2018. LNCS, vol. 11252, pp. 468–483. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03638-6_29
24. Sandhu, R.S., Samarati, P.: Access control: principle and practice. IEEE Commun. Mag. **32**(9), 40–48 (1994)
25. Siiskonen, T., Särs, C., Vähä-Sipilä, A., Pietikääinen, A.: Generic security user stories. In: Pekka, P., Juha, R. (eds.) Handbook of the Secure Agile Software Development Life Cycle. University of Oulu, Oulu (2014)
26. Sobieski, Ś., Zieliński, B.: User stories and parameterized role based access control. In: Bellatreche, L., Manolopoulos, Y. (eds.) MEDI 2015. LNCS, vol. 9344, pp. 311–319. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23781-7_25
27. Ulbricht, M.-R., Pallas, F.: YaPPL - a lightweight privacy preference language for legally sufficient and automated consent provision in IoT scenarios. In: Garcia-Alfaro, J., Herrera-Joancomartí, J., Livraga, G., Rios, R. (eds.) DPM/CBT -2018. LNCS, vol. 11025, pp. 329–344. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00305-0_23
28. Wachter, S.: Normative challenges of identification in the internet of things: privacy, profiling, discrimination, and the GDPR. Comput. Law Secur. Rev. **34**(3), 436–449 (2018)
29. Wang, W., Gupta, A., Niu, N.: Mining security requirements from common vulnerabilities and exposures for agile projects. In: 2018 IEEE 1st International Workshop on Quality Requirements in Agile Projects (QuaRAP), pp. 6–9, August 2018

30. Wang, X., Zhao, L., Wang, Y., Sun, J.: The role of requirements engineering practices in agile development: an empirical study. In: Zowghi, D., Jin, Z. (eds.) Requirements Engineering. CCIS, vol. 432, pp. 195–209. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43610-3_15
31. Xiao, X., Paradkar, A., Thummalapenta, S., Xie, T.: Automated extraction of security policies from natural-language software documents. In: Proceedings of the ACM SIGSOFT FSE 2012, FSE 2012, pp. 12:1–12:11. ACM, New York (2012)