Chapter 2
A Holistic Approach for Privacy Requirements Analysis: An Industrial Case Study

Mohamad Gharib, Paolo Giorgini, Mattia Salnitri, Elda Paja, Haralambos Mouratidis, Michalis Pavlidis and Jose Fran. Ruiz

2.1 Introduction

Privacy is becoming more and more a prominent concern for most countries, particularly for those of them that are moving toward the implementation of e-government [18] where software systems dealing with personal information (i.e., citizens, customers, etc.) have to be compliant with national and international privacy laws [26]. Moreover, while privacy has been frequently identified as a main concern for Public Administrations (PAs) while dealing with citizens' information for performing their activities and providing services [18, 43], several recent studies have shown that citizens might refrain from using services when their privacy is endangered [49, 56]. According to Spiekermann et al. [72], *an increasing majority of US and EU citizens say that existing laws and organizational practices do not provide a reasonable level of privacy protection and that companies share personal information inappropriately*. As an answer, the new European Privacy directives [14] introduced a number of privacy-related rules to increase the citizens' trust in PAs and their services.

Mohamad Gharib - corresponding author
University of Florence, Firenze, Italy, e-mail: mohamad.gharib@unifi.it

Paolo Giorgini
University of Trento,Trento, Italy e-mail: paolo.giorgini@unitn.it

Mattia Salnitri
Politecnico di Milano, Milano, Italy e-mail: mattia.salnitri@polimi.it

Elda Paja
University of Trento,Trento, Italy e-mail: elda.paja@unitn.it

Haralambos Mouratidis
University of Brighton, Brighton, UK e-mail: H.Mouratidis@brighton.ac.uk

Michalis Pavlidis
University of Brighton, Brighton, UK e-mail: M.Pavlidis@brighton.ac.uk

Jose Fran. Ruiz
ATOS, Madrid, Spain e-mail: jose.ruizr@atos.net

Despite this, organizations still suffer from several shortcomings that can endanger citizens' information such as bad security practices, hacker and, most importantly, insider attacks, data thefts, etc. [1].

As advocated by Privacy by Design (PbD) [37, 12], to ensure a certain level of privacy we need to adopt a systematic and holistic approach to privacy requirements engineering. More specifically, privacy requirements should be considered as first class requirements along with functional and non-functional ones [17]. For decades, privacy requirements have been considered the result of a security analysis (e.g., [82, 52, 37]) and specified as generic non-functional requirements without any clear measure for their satisfaction [3, 81, 52]. Only recently, the research community proposed a number of approaches to privacy requirements engineering [37, 12], but without showing their effectiveness in real cases and without considering privacy requirements for already existing systems (e.g., [21, 78]), as we usually have in PAs.

On the other hand, privacy is an elusive and vague concept [22, 68], and it is hard to reach consensus on its definition [22]. Although several efforts have been made to clarify this concept by linking it to more concrete concepts such as secrecy, personhood, control of personal information, etc. [69], there is no consensus on the definition of these concepts or which of them should be used to analyze privacy [69, 24]. This adds more complexity while eliciting, classifying, prioritizing, and validating privacy requirements. In addition, the relations between privacy requirements and other types of requirements have not been extensively studied, i.e., it is not clear how privacy requirements can be linked to other types of requirements.

In this chapter, we propose a holistic approach for analyzing privacy requirements specialized in their eliciting, classifying, prioritizing, and validating. The approach follows the experience we gained in the Vision Project, an H2020 innovation action funded by the European Commission (Visual Privacy Management in User Centric Open Requirements) [26]. Specifically, in the project we built on the idea that PAs can be engaged as the main source for defining the privacy requirements of users responsible for managing citizens' information, while citizens can considered as the main source for defining the privacy requirements of information owners. In other words, our approach gives citizens (information owners) a voice while specifying their privacy preferences, along those of the PAs as required by privacy norms. With our proposal, we aim at assisting software engineers in designing privacy-aware systems by providing the guidance and support while eliciting, classifying, prioritizing, and validating/consolidating privacy requirements.

The rest of the chapter is organized as follows. Section 2 presents the research baseline and we describe the VisiOn project in section 3. In section 4, we present our approach for privacy requirements specification, while section 5 discusses how the approach has been used to specify the VisiOn privacy requirements. In section 6, we discuss threats to approach validity. We present related work in section 7 and, finally, we conclude the chapter and discuss future work in section 8.

2.2 Research Baseline

A main goal of requirements engineering is discovering stakeholders' actual needs that drive to requirements for the system-to-be [8]. This requires a well-defined systematic process to be followed while dealing with such needs. Several Requirements Engineering (RE) processes for dealing with stakeholders' needs have been proposed (e.g., [57, 71, 40, 70, 74]) and most of them include the following activities: elicitation, classification, prioritization, and validation. In the rest of this section, we list and discuss the main contributions in each of these activities.

Requirements elicitation is one of the first activities in RE process and can be defined as the process of discovering, acquiring, and elaborating requirements for the system-to-be through consulting relevant stakeholders, investigating the system's documentation and/or using domain knowledge [71, 40, 70]. Usually, requirements elicitation is a complex and iterative process that starts with requirements discovery and ends with requirements documentation [70, 83]. Requirements elicitation is one of the most critical activities in the RE process (e.g., [83, 8, 70]), since getting the right requirements is considered a key factor for software development projects [36]. The main idea of requirements elicitation is gathering stakeholders' requirements concerning the system-to-be. Therefore, involving stakeholders in the process is essential for the process to succeed [8]. However, involving them is not an easy task, since stakeholders, usually, express their requirements in very general terms, they may have conflicting requirements, and they may change their requirements during the analysis process [70, 42]. Thus, involving stakeholders does not always guarantee the elicitation of the right requirements. Several requirements elicitation approaches and techniques have been proposed in the literature, including interviews [2], questionnaires [20], task analysis [11], introspection [29], laddering [34], requirements workshop [83], ethnography [6], apprenticing [9], scenarios [83], and prototyping [70]. A new trend is the use of serious games (gamifications) in which game-based elements are used during the requirements elicitation process [19, 61, 7]. Nevertheless, there is no general agreement on which elicitation technique is the best, but there is a consensus that selecting an appropriate elicitation technique greatly affects the success or failure of the requirements elicitation process [54, 32].

Requirements classification is the activity that takes an unstructured collection of requirements and groups them into coherent clusters [70]. Requirements can be classified in many different ways [5], yet they can be broadly classified under functional and non-functional requirements, where the first type refers to functionalities that the system shall deliver, and the second type refers to how the system shall deliver such functionalities [13]. More specifically, non-functional requirements are generic qualitative properties of the system as a whole, i.e., they refer to properties of the overall system, such as reliability, usability, supportability, etc. [42]. Moreover, functional requirements have clear-cut criteria for their satisfaction, while non-functional requirements can rarely be said to be accomplished or "satisfied" in a clear-cut sense [53]. Concerning security and privacy several different classifications have been proposed in the literature. For example, Singhal and Wijesekera [67] propose an ontology to classify security needs in terms of *threats*, *attacks*, *vulnerabilities*, *risks*, and *security mechanisms*. While [76] proposed to represent security needs using the following concepts: *assets*, *vulnerabilities*, *threats*, *countermeasures*, and *security policy*. Kang and Liang [38] classify security concerns into:*auditing*, *threats*, *accountability*, *non-repudiation*, *risk*, *attacks*, *availability*, *frauds*, *confidentiality*, *asset*, *integrity*, *prevention*, and *reputation*. On the other hand, Labda et al. [41] classify privacy needs in terms of *access control*, *Separation of Tasks (SoT)*, *Binding of Tasks (BoT)*, *user consent*, and *Necessity to know (NtK)*. In [37] *privacy goals* were classified under eight types namely, *authentication*, *authorization*, *identification*, *data protection*, *anonymity*, *pseydonymity*, *unlinkability*, and *unobservability*. While Solove [68] provides taxonomy for classifying privacy related problems under four main groups of possible harmful activities: *information collection*, *information processing*, *information dissemination*, and *information invasion*. Finally, other types of classifications have been proposed to sub-classify requirements such as risk [50], trust [82], information quality [23], etc.

Requirements prioritization is the activity to classify requirements on the base of their importance [70, 30], which enables for making decisions on which requirements should be implemented by the system-to-be. According to Berander and Andrews [51], prioritizing requirements allows for: deciding the core requirements of the system; selecting an optimal set requirements to be implemented, i.e., selecting a subset of requirements to realize a system that satisfies stakeholders' needs; estimating the expected users' satisfaction; and balancing the benefits of each requirement against the costs for its implementation. Several techniques for requirements prioritization has been already proposed in literature, such as: *Analytic Hierarchy Process (AHP)*, in which decision makers pair-wise compare the requirements to determine which of the two is more important [39]; *cumulative voting (the 100 point method)*, in which each stakeholder is given 100 points that he/she can use for voting in favor of the most important requirements [44]; *Bubblesort*, one of the simplest prioritization methods that sort requirements according to their priorities [39]; *ranking* [39], in which requirements are ranked based on their importance starting from the most important requirement until reaching the least important ones; *Top-Ten requirements* [42], in which stakeholders are asked to choose top-ten requirements out of all the requirements set without assigning an internal order between the requirements; or *numerical assignment (grouping)* [51], which is one of the most common prioritization technique that groups requirements into different priority groups based on their importance (e.g. critical, standard, and optional).

Requirements validation is concerned with showing that the set of requirements define the system that the stakeholders expect [40, 70]. Requirements validation is very important since detecting errors in the requirements during the design phase is much less expensive and time-consuming than discovering such errors after the system implementation [52]. One of the most well known method for requirements validation is presented in [70], which suggests five checks to be performed on the requirements: (1) *validity check* aiming at verifying the requirements with all relevant stakeholders for the system-to-be; (2) *completeness check* aiming at verifying that t requirements capture all the system functionalities and features (e.g., properties, constraints, etc.) expected by the stakeholders; (3) *consistency check* aiming at verifying that there is no inconsistencies among all requirements, i.e., there is no requirement that conflicts with any other requirement; (4) *realism check* aiming at verifying that requirements can actually be implemented; (5) *verifiability check* aiming at verifying that stakeholders and contractor(s) have the exact same understanding of the elaborated requirements, so to reduce potential disputes between them. All requirements should be written in clear way so both parties can understand and agree on them.

2.3 A Holistic Approach for Privacy Requirements Analysis

In this section, we present the approach we used to analyse the privacy requirements for the VisiOn platform. Particularly, we give an overview of the main phases of the process which will be further detailed in the next section specifically in the context of the VisiOn project [26]. The approach follows of the four general requirements engineering activities introduced in Section 2 (i.e., requirements elicitation, classification, prioritization, and validation) and proposes a process consisting of six main interrelated activities Figure 2.1.

1. Identifying the scope. Defining the scope of the project is, usually, the most appropriate way to start [62]. This is the first activity of our process aiming at determining the boundary of the system accordingly to the main objectives to be achieved [46]. In order to properly identify the scope, we have to collect as much as possible information related to the outcome of the system to be developed and its possible application domains along with its intended users. Moreover, this activity is essential for the appropriate allocation of the project resources, indeed identifying the scope correctly reduces wasting of resources and avoids unnecessary activities. The main outcome of this activity is a *system and domain analysis*.

2. Stakeholder analysis aims at identifying all stakeholders that may influence, or that can be influenced, by the system. Stakeholders are then classified in coherent groups so to generalize their needs and expectations. This activity mainly focus on the identification of two groups of stakeholders: stakeholders who own personal information (legitimate information owners) and stakeholders who deal with/manage such information. Both of these groups play main roles while eliciting, classifying, and prioritizing privacy requirements. This activity is composed of two sub-activities:

2.1 Stakeholder identification & classification take the *system and domain analysis* as input and identify an initial list of stakeholders, which is further analyzed to identify any other relevant stakeholder. Since an inadequate stakeholders identification leads to inadequate stakeholder analysis [45], an accurate list of all possible stakeholders has to be produced at this stage. Then, identified stakehold-

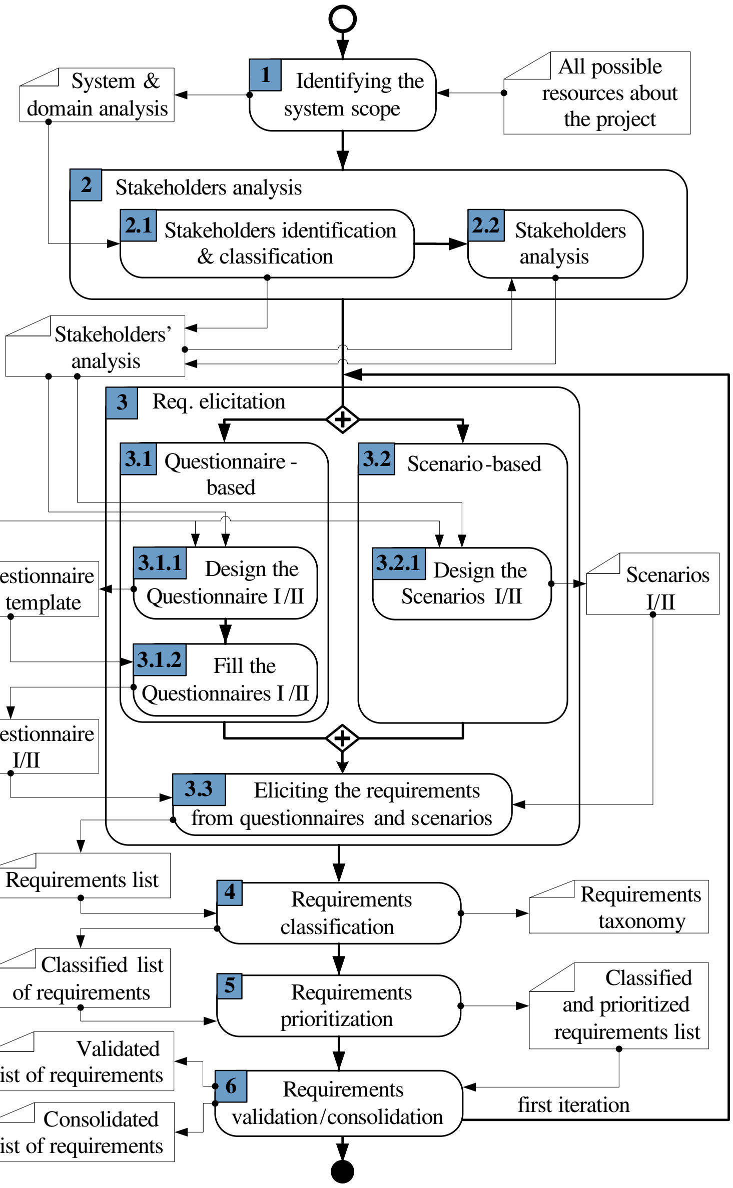

System & domain analysis
1 Identifying the system scope
All possible resources about the project
2 Stakeholders analysis
2.1 Stakeholders identification & classification
2.2 Stakeholders analysis
Stakeholders' analysis
3 Req. elicitation
3.1 Questionnaire - based
3.2 Scenario-based
3.1.1 Design the Questionnaire I/II
3.2.1 Design the Scenarios I/II
Questionnaire I/II template
Scenarios I/II
3.1.2 Fill the Questionnaires I/II
Questionnaire I/II
3.3 Eliciting the requirements from questionnaires and scenarios
Requirements list
4 Requirements classification
Requirements taxonomy
Classified list of requirements
5 Requirements prioritization
Classified and prioritized requirements list
Validated list of requirements
6 Requirements validation/consolidation
first iteration
Consolidated list of requirements

Fig. 2.1 The process for the specification of privacy requirements

ers are further grouped into coherent groups[1] to better communicate with them, learn about their needs, integrate them into RE activities (e.g., in the require-

[1] These groups are not mutually exclusive, i.e., a stakeholder may belong to all of them

ments consolidation activity), and prioritizing their needs accordingly to their importance [45].

2.2 Stakeholder analysis aims at analyzing each single stakeholder that has been identified in terms of its needs and expectations. We use Socio Technical Security - modelling language (STS-ml) [55], a goal based modeling language, to represent and analyze stakeholders and their objectives. We restricted the choice to goal-based modeling languages because they allow for an explicit representation of stakeholders' objectives and their needs. In particular, we selected STS-ml because it allows for a clear representation of stakeholders' objectives and their relation.

3. Requirements elicitation. Once stakeholders are identified, the process of requirements elicitation begins. The elicitation process is all about determining stakeholders' needs and it is performed incrementally and iteratively adding details about requirements [83]. Adopting the right elicitation technique from existing ones (e.g., interviews, questionnaires, task analysis, scenarios, prototyping, etc.) greatly affects the success of the requirements elicitation process [54, 32]. Moreover, many projects adopt more than one technique for requirements elicitation [33, 83] to improve the reliability and quality of elicited requirements. In this context, we have adopted two different techniques, namely *questionnaire-based* and *scenario-based*. These two techniques complement each other: the former has been adopted because it allows for collecting multiple stakeholders' requirements simultaneously, eliciting the actual stakeholders' requirements, and most importantly its flexibility in contacting the stakeholders; the latter has been chosen because it allows for interactively involving stakeholders during the requirements elicitation process, which is essential for privacy requirements due to the vague nature of such requirements [24]. The requirements elicitation activity is repeated twice, with the main purpose of eliciting more detailed privacy requirements in the second iteration and it is composed of three main sub-activities:

3.1 Questionnaire-based requirements elicitation. This activity is composed of two main activities *3.1.1 Design the questionnaire* and *3.1.2 Filling the questionnaire*, where the first aims at designing the questionnaire template and the last aims at sharing the questionnaires with the stakeholders and receive their feedback. In the first iteration, the first questionnaire is designed and filled by stakeholders, while in the second iteration, the questionnaires is refined in more detailed questions accordingly to the feedback collected in the first iteration and filled again by stakeholders.

3.2 Scenario-based requirements elicitation. This activity aims at defining several scenarios where personal information become critical for stakeholders. They are defined by stakeholders assisted by the analyst responsible of the requirements elicitation activity. Each scenario is then used to elicit stakeholders' privacy requirements. Similar to activity 3.1, this activity is repeated twice, where scenarios I and scenarios II are modeled in the first and second iteration of the activity, respectively.

3.3 Extracting privacy requirements. In this activity, the analyst extracts requirements from questionnaires and scenarios and integrate them in a consistent and coherent list. Practically, the analyst identifies explicit and implicit requirements from the answers of stakeholders and their expectations highlighted in the scenarios. Duplicated requirements are eliminated and conflictual situations are solved.

4. Requirements classification. Relying on existing requirements taxonomies, this activity aims at classifying requirements into coherent groups with closely related characteristics. When existing taxonomies do not cover types for some requirement, new classifications/sub-classifications are introduced. A taxonomy can reduce or remove any vagueness while dealing with requirements, and in turn, it contributes to better understanding of how requirements can be realized. Note that, the two stakeholders groups, namely information owners and who manage personal information, should be actively involved in this activity; particularly while extending the taxonomy to cover the privacy related concerns. In summary, this activity takes the requirements list and relays on existing requirement taxonomies to classify them accordingly. The taxonomy is refined iteratively with the active participation of stakeholders until it covers all types of requirements. After that, each requirement is assigned to a classification/sub-classification, the list of classified requirements is shared with all stakeholders so to receive their feedback and possibly revise further the taxonomy.

5. Requirements prioritization aims at prioritizing requirements based on their importance. Requirements should be prioritized mainly on the base of stakeholders' suggestions. Different weights can be associated to different groups of stakeholders so to reflect the importance and relevance of their feedback [45] and allowing for more accurate prioritization. In addition, the prioritization process should also consider requirements interdependencies that are largely [31]. Considering requirements interdependencies enable for better decisions concerning requirements implementation. For instance, a requirement might be classified as a low priority based on the feedback of the stakeholders (might not be implemented), yet it is *required* by a high priority requirement(s) (should be implemented). In such case, the latter requirement cannot be achieved without implementing the former one, therefore, such requirement should be implemented even it has been classified as a low priority requirement by the stakeholders.

6. Requirements validation/consolidation aims at verifying that the list of requirements captures all functionalities and qualities required by the stakeholders, the requirements are consistent one another, and a real-world solution can be used to implement each of these requirements [70, 74]. Our approach adopts the validation method proposed in [70], which performs five checks to validate the requirements, namely, *validity*, *completeness*, *consistency*, *realism*, and *verifiability* checks. On the other hand, requirements consolidation is the final activity of the process and it replicates the same activities performed during the validation, but in a more binding way since it produces the final list of requirements, i.e., no more modification or refinement for any of the requirements will be further performed.

2.4 Analyzing Privacy Requirements for the VisiOn Platform

This section gives a detailed description of how we applied our approach to analyzing the VisiOn stakeholders' privacy requirements. The process follows what has been proposed in the previous section and illustrated in Figure 2.1 and consists of five main interrelated activities (classification, prioritization and validation the VisiOn requirements are combined in one single activity). In the rest of this section, we describe each of these activities.

1. Identifying the VisiOn project scope. In order to get a better understanding the overall scope of the VisiOn project, we depended on the VisiOn proposal and all available documentations that have been obtained from the partners[2] as input to analyze the scope of the VisiOn project. This activity produced the *VisiOn project & domain analysis*, which is used for the *VisiOn stakeholders analysis* activity.

2. VisiOn stakeholders analysis. This section summarizes our activities for identifying, classifying and analyzing the stakeholders of the VisiOn platform.

2.1 Stakeholders identification & classification. Depending on the *VisiOn project & domain analysis*, all identified stakeholders can be described as: stakeholders who represent legitimate owners of personal information), stakeholders who deal with/manage personal information, or stakeholders who are responsible for providing components for the VisiOn Privacy Platform (VPP) – VPP will be developed by integrating the partners existing software and tools. Therefore, we classify the stakeholders into three main groups (roles): *(1) Citizen*, people that will use VisiOn to define, visualize and control how their personal information are used by others (e.g., PAs); *(2) PA*, organizations that will use VisiOn to visualize, manage and control how the citizens' personal information are used and for which reasons by their own services and those provided by others[3]; and *(3) Component provider*, representing a VisiOn's partner that provides technical components for the final VisiOn platform. They contributed with requirements of each component and information of the integration among the components of the VPP.

2.2 Stakeholders analysis. After identifying the three main types of VPP stakeholders, we analyzed each of them in terms of their objectives related to the VisiOn's scope. Figure 2.2 shows the main stakeholder types along with their objectives represented with STS-ml [55]. STS-ml is a modeling language focused on social/organizational interactions between entities in socio-technical systems, i.e., systems where humans and technical components interact with each other to achieve common objectives. PA systems are an example of socio-technical systems: they are composed of technical components, such as the software services use to manage fees or payment, and humans, such as the citizens and the employees of the PAs. In STS-ml, autonomous entities (both humans and technological

[2] Partners refer to the full consortium of the VisiOn project

[3] Citizens and PAs roles can be generalized to a User stakeholder role

components) are called *actors*. Actors can be specified in an STS-ml diagram as *roles*, to represent a set of autonomous entities, or as *agents*, to represent a specific entity. For example, “Citizen" is considered a role, because it represents the set of people in a given country, while “George" is considered an agent since it is a single person. Roles are graphically represented with a pink solid circle, with a half circle in the lower part, while agents are represented with the same solid pink circle but with a segment in the upper part. Objectives that a stakeholder aims to achieve, are called *Goals* and are graphically represented as green solid ovals. Examples of goals are “PA trusted", which consists in building confidence in PA, or “software provided", which consists in providing software tools. The oval shapes attached to actors represent their *scopes*: the set of goals positioned inside a scope is assigned to the actor and specifies that the actor is in charge for fulfilling them. For example, in Figure 2.2 *Citizen* is in charge of the goal *PA trusted*. Goals can be refined through “and-decomposed” or “or-decomposed” into subgoals, where in and-decomposition all subgoals must be achieved to fulfill the main goal, while only one of the subgoals must be achieved to fulfill the main goal in the or-decomposition. Figure 2.2 shows the objectives of stakeholders. Citizens have the main objective of trusting the PA. This goal can be split in three subgoals that must be reached in order to trust the PA that are: *Privacy issues shown*, *Consent managed* and *Sensitive data protected*. The first goal is and-decomposed in two subgoals: *show privacy violation*, which consists in promptly receiving information about violation of privacy in PA’s systems, and *threats visualized*, which consists in receiving information from the PA about the threat to privacy on citizens information. *Consent managed* goal consists in reading and signing consents, while *Sensitive data protected* is and-decomposed in tree sub-goals: *Privacy req. specified*, *Privacy req. Visualized* and *Privacy req. enforced*. The three goals are achieved respectively if citizens’ requirements are specified by the citizens and PA, shown by the PA to the citizens and enforced in PA system.

3. (I) Eliciting the VisiOn requirements (first iteration). In what follows, we describe the activities we performed to elicit VisiOn user requirements during the first iteration.

3.1 (I) VisiOn Requirements Questionnaire I. Contains two sub-activities that describe how the first VisiOn requirements questionnaire was designed and filled by the partners respectively.

3.1.1 (I) Designing the VisiOn Questionnaire I (Q1). The requirements for building a system can be elicited from several sources [48], including stakeholders, users, documentation, and other existing systems [83]. Therefore, the first VisiOn Questionnaire template was designed to elicit requirements from the following three main sources: *(1) application domains,* which should be explored together with its political, organizational, social aspects, constraints that may influence the system [35, 83]; *(2) stakeholders,* are the entities who can influence, or are being influenced by the system, where analyzing the stakeholders of the system is a key factor for the success of the overall requirements

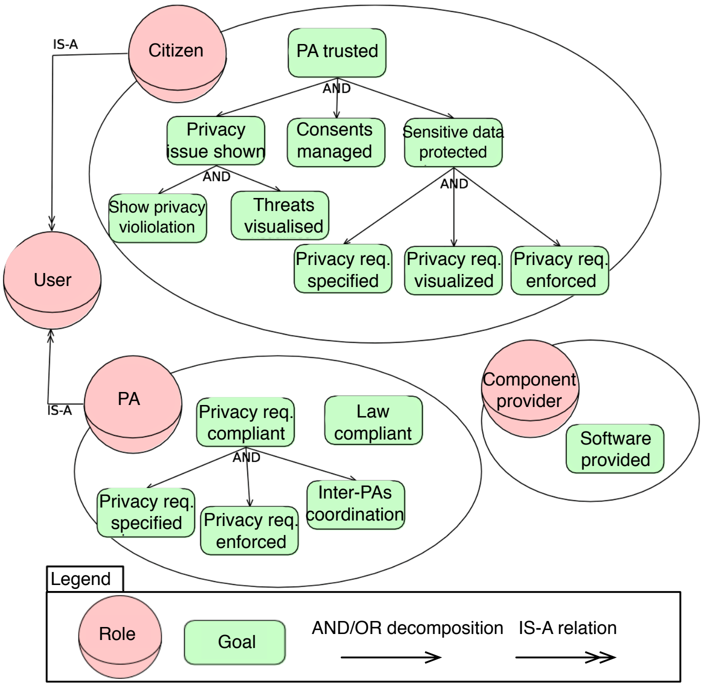


Fig. 2.2 Objectives of stakeholders of VisiOn

elicitation process [8]; and *(3) intended users,* are the entities who directly interact with the system to perform their work, and they play a central role in the requirements elicitation process as some requirements can be defined only by them (e.g., usability, supportability) [54]. In this context, the questionnaire contains four main sections to be filled by the partners concerning: *(1) application domains*, *(2) stakeholders*[4], *(3) intended users*, and *(4) examples of usage* that identify at least three possible scenarios in the application domains, where users may use the VisiOn platform. This may reveal new requirements that the partner forgets to mention while compiling the previous sections.

3.1.2 (I) Filling and refining the VisiOn Questionnaire I (Q1). The questionnaire template has been shared with four End-User (E-U) partners that represent both PAs and citizens, and we asked them to fill and return. Few days after sharing the questionnaire, the partners started contacting us asking for some clarifications about some concerns related to their input. We have answered each of the raised concerns, and support them with more information when it is required. Once we received the filled questionnaires, we analyzed them

[4] To extend our knowledge about the stakeholders analysis (activity 2), and uncover any stakeholder that has not been identified so far

appropriately adding our comments wherever a clarification is needed from the partner. In several cases, we supported our comments with general examples to assist the partner in replying to them. And then, we sent back the questionnaire to the partners to refine their input. In some cases, the questionnaire was sent back and forth to the partner several times until their input is clear and understandable. The returned questionnaires were carefully analyzed, and we have identified 32 stakeholders and 12 users of VPP along with their objectives, excepted functionalities and qualities.

3.2 (I) Modeling and analyzing the scenarios I. The consortium of the VisiOn project is composed by two type of partners: technical partners, who provide the software and create the VisiOn platform, and the pilot partners, who are PAs and use their premises to evaluate and validate the platform. The pilot partners are one Spanish hospital, one Italian hospital, one Italian ministry and one Greek company who manage the municipality of Athens services. In this chapter, we use the latter as a running example, we did not include the other case studies for space limit. During the initial part of the VisiOn project, we asked the Pilot partners to define at least three scenarios each, where the management of personal information is critical, in terms of privacy, for both Citizens and PAs. We asked them to use STS-ml [55] for modeling these scenarios. STS-ml requirements models are created by the construction of three complementary views:

- **The social view** (shown in Figure 2.3) is built on three concepts: *actor* that can be divided into a *role* (e.g., *Citizen*) or an *agent* (e.g., *Management system*), *goal* (e.g., “Birth certificate obtained”), and *document* that is a tangible supporting materials (e.g., “Birth certificate”). A goal may *produce* a document, i.e., the document is created when the goal is achieved (e.g., “Birth certificate issued” will *produces* the document “Birth certificate”). It may *read* a document, i.e., the actor linked to the goal needs to read the document in order to achieve the goal (e.g., the actor "Citizen" needs to read the “Birth certificate” to achieve the goal “Birth certificate obtained”). An actor can also *modify* a document to achieve a goal. The interactions between actors are represented with two relations, *transmission* and *delegation*. The former represents the *transmission* of a document between two actors. For example, in Figure 2.3 “Citizen" *transmits* the “ID copy” to “Citizen Registry". *Delegation* of a goal represents the assignment of an objective from an actor to another actor, i.e., with a delegation the responsibility of achieving a goal in transferred to another actor. For example, in Figure 2.3 “Citizen" *delegates* the goal of “Birth certificate issued” to “Citizen Registry". STS-ml permits to specify security and privacy requirements on *transmissions* and *delegations*. For example, Figure 2.3 three of them are shown: *integrity*, represented with a "Int" string inside a pink box, *confidentiality*, represented with a "Con" string inside a brown box, and *authentication*, represented with a "Auth" string inside a yellow box. *Integrity* can be specified on *transmissions* and it means that the document received is the same as the document sent. *Confidentiality* is specified on *transmissions* and it means that only authorized users can read the document that is sent.

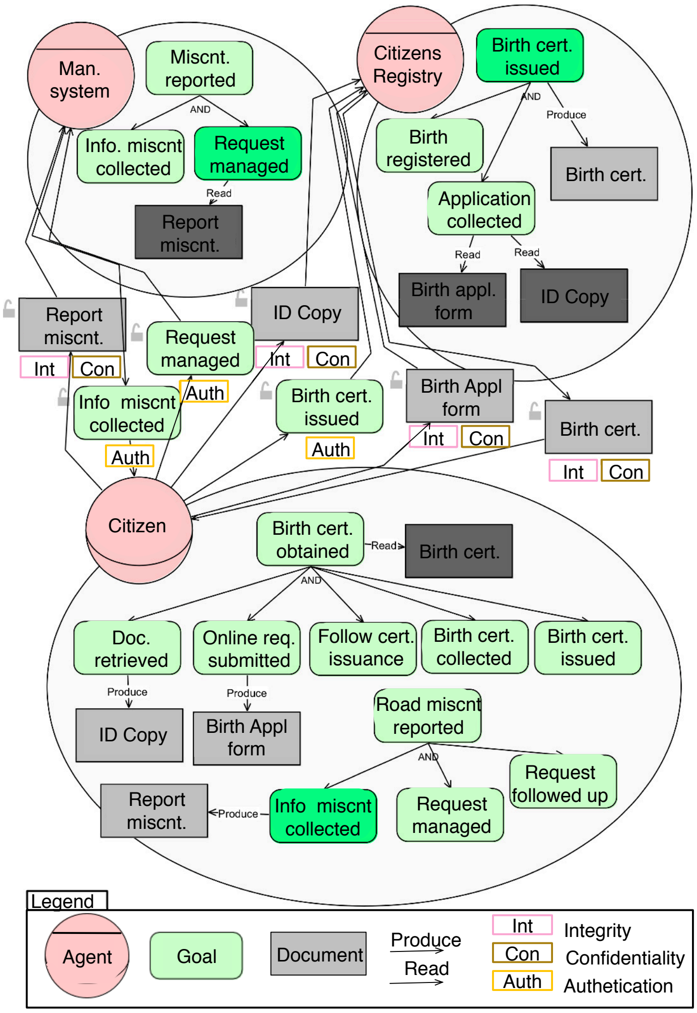


Fig. 2.3 STS-ml social view

Authentication can be specified on *delegations*, and indicates that the source and destination actors must prove their identity, e.g., using an authentication security mechanism.

- **The information view** (shown in Figure 2.4) is built on two concepts: *document* and *information*. The latter represents intangible data, such as name, surname, bank account details that is stored in one or more document. The relation *Tangible By* connects an *information* to a *document* and specifies that the information is stored in that document. For example, in Figure 2.4 informa-

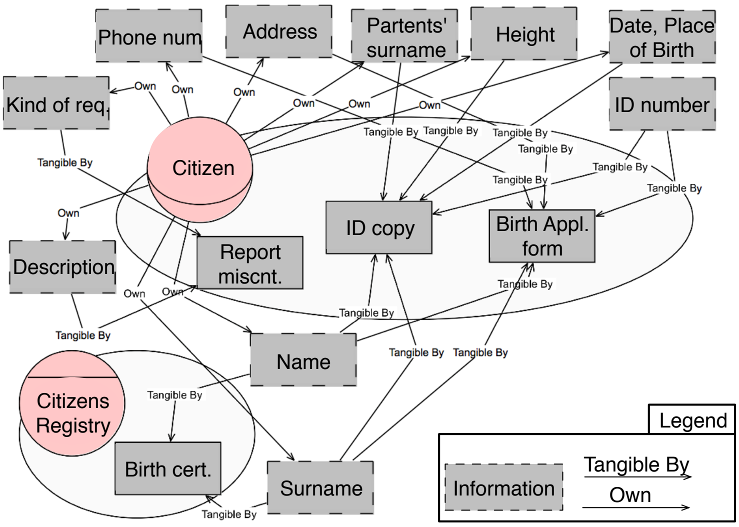


Fig. 2.4 STS-ml information view

tion "Name" that is stored in "Birth certificate". Information can be possessed by at most one actor. This is represented with the *Own* relation that connects an actor to an information and specifies that the actor is the legitimate owner of information. For example, in Figure 2.4 the "Citizen" role *own* information "Name".

- **The authorization view** (shown in Figure 2.5) is used to represent the authorizations that actors grant to one another over their information. The authorization relation connects two actors and it consists of three parts: (i) a set of authorizations, i.e., **R**ead, **M**odify, **P**roduce and **T**ransmit; (ii) a set of information, i.e., the target of the authorizations; and (iii) the scope of the authorization, i.e., the sot of goals for which the authorization is granted. For example, in Figure 2.5 the authorization relation between *Citizen* and *Management system* authorizes the latter to read and transmit "Picture", "Description", "Location details" and "Kind of request" information. Since the scope part is empty, the authorization does not specify any constraint for what concerns the scope. Each partner was assisted by a modeling expert while modeling its scenarios. The models have been refined iteratively through several modeling sessions. The resulting models were analyzed by STS-tool [66], a software framework which supports STS-ml, to detect any modeling deficiencies and inconsistencies. Once they were verified, they were used to generate VisiOn user requirements. This feature of the tool automatically derives requirements form the models, based on the goals, the dependencies and interaction between actors and security constraints defined is the diagrams.

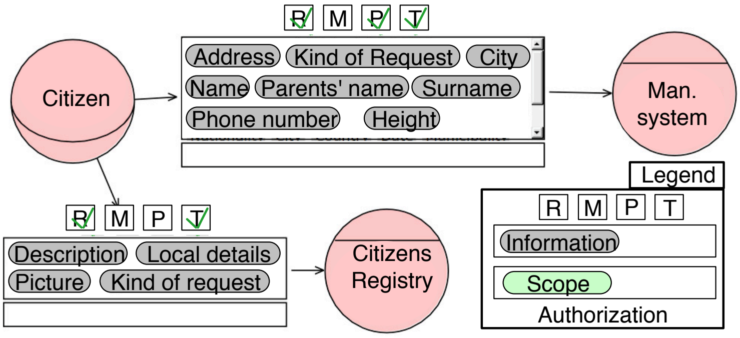


Fig. 2.5 STS-ml authorization view

3.3 (I) Eliciting the VisiOn requirements from questionnaires I and scenarios I. Once all the questionnaires are filled and refined, and all the scenarios are modeled and analyzed, we have used them to elicit the stakeholders' needs. In summary, 91 stakeholders' needs were elicited, where these needs have been used to elaborate the first set of the VisiOn user requirements (99 requirements). In particular, when the stakeholder's need is clear enough, it is considered as a requirement. While when the need is not clear, it is refined into a requirement or more. Note that adopting two different techniques for requirements elicitation, significantly improved the quality of the requirements we elicited. For example, the same requirement might be elicited by the two techniques, yet it is unlikely that both techniques elicit the exact same requirement. Therefore, the two versions of the requirement can be used to produce more detailed requirement. In addition, we assigned different groups to perform the two elicitation techniques in order to reduce the impact that one technique might have on the other, and in turn, might influence the quality of the elicited requirements. Finally, each of the identified needs and requirements has been assigned a unique identifier that specifies the source where the requirement has been first identified. This is particularly important for requirements traceability reasons, i.e., it enables for tracing requirements back to their original sources and identify what kind of modifications have been applied to them. The list of VisiOn requirements has been shared with the partners to receive their feedback, which we took into account while revising the requirements.

4. Classifying, Prioritizing and Validating the VisiOn requirements. This section describes how we classify, prioritize and validate the VisiOn requirements elicited during the first iteration of the requirements elicitation activity.

- *Design the VisiOn requirements classification, prioritization and validation questionnaire.* The questionnaire presents a table that contains the elicited requirements, where each requirement has been assigned a *type* based on our proposed classification, a *priority* of the requirement to be filled by the partner (1 low - 5 high), and a text box to add any *comment/suggestion* concerning the requirement. In particular, 17 individuals from nine different partners have participated in this activity, and we asked them to analyze the table, to provide priorities, to check the classification/sub-classification we assigned to the requirements and revise if needed. Moreover, we provided them with a table that contains a *mapping* between the requirements and the VisiOn components that will realize them, and we asked them to provide feedback. This section has been added to help component developers to understand better their responsibilities, and how they should extend their component(s) to realize the requirements. Moreover, it facilitates performing the *requirements realism check*[5]. In addition, we ask them to check the requirement carefully and, possibly, to extend the list with other relevant requirements if required.

- *VisiOn requirements classification.* As previously mentioned, RE community broadly classifies requirements under functional and non-functional requirements, where the first have clear-cut criteria for their satisfaction, and the last do not have such criteria [53, 70, 13]. In this context, we proposed a taxonomy that differentiates between two main types of requirements, functional and non-functional requirements. More specifically, when the requirements have clear-cut criteria for their satisfaction, they are classified as functional requirements; otherwise, they are classified as non-functional requirements. In addition, we provide a classification/sub-classification for both functional and non-functional requirements based on the related literature to covers all types of the elicited requirements. In particular, non-functional requirements have been further sub-classified under four types, namely usability, reliability, performance, and supportability [53]. Functional requirements were further sub-classified under four types of requirements namely, privacy requirements, security requirements, IQ requirements and trust requirements. Since no existing work proposes a well-defined taxonomy to classify privacy requirements, they have been classified based on the common aspects of privacy identified based on the feedback we received from the stakeholders taking into consideration the five components of VisiOn Platform[6] (privacy assessment, privacy requirements, privacy specification, privacy runtime, and privacy transparency visualization). To this end, *privacy requirements* have six main sub-categories, 1- information ownership, 2- information control (authentication), 3- information usage, 4- information transmission, 5- privacy assessment, and 6- privacy verification.
- *Security requirements* have been considered to capture the main stakeholders' security concerns. In our taxonomy, security requirements have six main sub-categories that have been chosen based on the best practices concerning capturing security requirements in the literature [73, 52, 82, 50], namely 1- confidentiality, 2- integrity, 3- availability, 4- vulnerability, 5- threat, and 6- attack. While *Trust requirements* have been considered to capture the actors' expectations in one another

[5] Requirements realism will be discussed later in this section

[6] Next chapters provide more information about VisiOn components

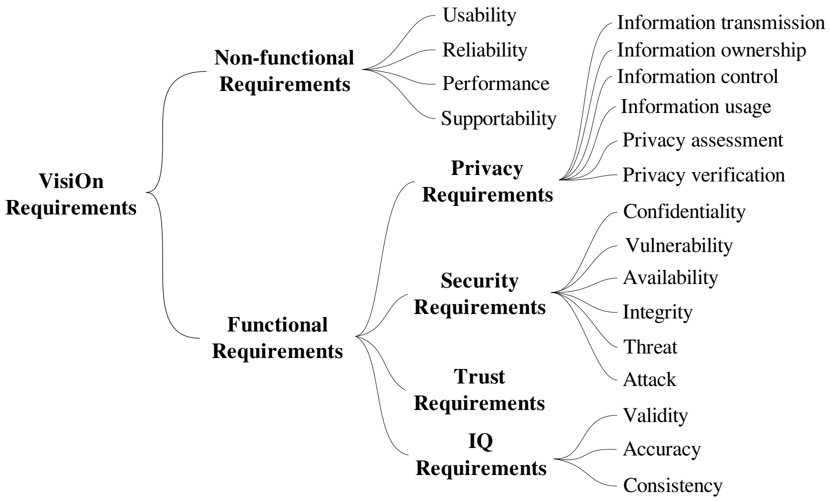


Fig. 2.6 VisiOn requirements taxonomy

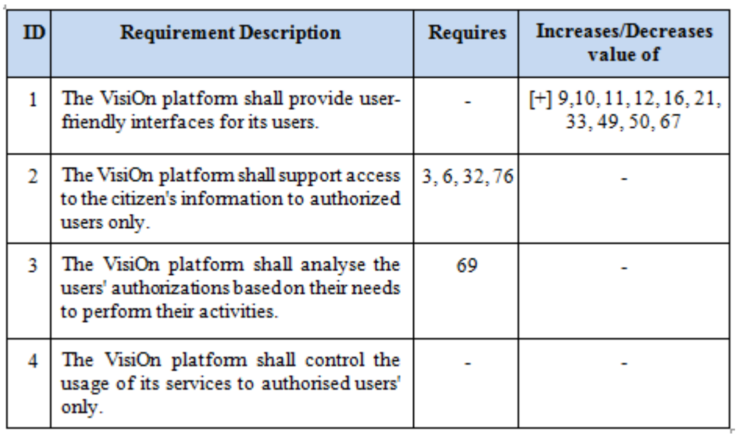
concerning their objectives and dependencies [52, 82]. Finally, *IQ requirements* [23, 10] have been considered to address stakeholders' needs concerning information accuracy, validity, and consistency. Therefore, IQ requirements have three main sub-categories, 1- accuracy, 2-validity, and 3- consistency. Once the taxonomy is considered complete, i.e., it covers all types of VisiOn requirements, we provide the partners with a table that contains the list of requirements, which have been assigned types based on our proposed classification, and we asked them to provide feedback. The returned feedback was carefully examined while producing the final taxonomy of the VisiOn requirements that is depicted in Figure 2.6.

- *VisiOn requirements prioritization.* Requirements prioritization is the process of classifying the requirements based on their importance [70, 30], which enables system developers to make decisions on which requirements should be implemented. Among the existing requirements prioritizing techniques (e.g., Analytic Hierarchy Process (AHP), Cumulative Voting, Ranking), we have adopted the numerical assignment (grouping) [51], which is the most common prioritization technique and it is standardized (see IEEE Std. 830-1998 [15]). In the numerical assignment, requirements are classified into different priority groups based on their importance (e.g. critical, standard, and optional). In particular, we asked the partners to prioritize each of the requirements on an ordinal scale from 1-5, where 1 is the least important and 5 is the most important. After that, we classified the partners based on their role in the VisiOn project under End-Users (E-U) (i.e., Citizens and/or PAs), System Integrators (SID) and Research and Academic (R-C). Then we calculated the requirements priority value for each of three partners' types. This was followed by assigning qualitative values instead of the numbered ones to enable qualitative reasoning concerning requirements prioritization. In particular, priority is **H**igh if its priority is at least four, it is **M**edium if its priority is at least three and less than four, and the priority is **L**ow if it is less than three. Furthermore, following [59], we assigned different weights to the input received from the different partners' categories. More specifically, the input received from E-U partners was considered as the most relevant since they represent the actual users of VPP (Citizens and PAs), followed by the SID partners input since they have experience in developing and commercializing software products, while the least important is the R-C partners input since they have experience in developing software products. Table 2.1 shows how we determine the priority of the requirements based on the input from the different partners. The priority values are evaluated qualitatively as follows: we have priority **H** when the priority expressed by E-U is **H**, while both of the priority values expressed by SID and R-C are at least **M**. The priority value is **M** if it was expressed by E-U as **M**, and both of the priority values expressed by SID and R-C are at least **M**. Finally, the priority is **L** if the priority expressed by E-U is **L** regardless of the input provided by SID and R-C, or when the priority expressed by E-U is **M**, and at least one of the SID and R-C has expressed it **L**. On the other hand, requirement dependency is gaining more attention in requirements prioritization lately [63, 51, 31]. According to Carlshamre et al. [11] only fifth of the requirements are not related to or influenced by other requirements. Therefore, the final decision concerning requirements implementation should not only depend on their assigned priorities, but also on its relation(s) with other requirements. Following [11], we have considered three different relations among requirements:

- *Requires,* which implies that the fulfillment of one requirement depends on the fulfillment of another one. Usually, such relation is used to describe that if one requirement is to be included into the system, it requires another requirement to be included as well, i.e., a requirement is a pre-requisite or pre-condition for another one. For example, if one requirement states that the system should include web-access, a network connection is required.
- *Conflicts_with,* which means that a requirement is in conflict with another one, if they cannot exist at the same time, i.e., fulfilling one of them decreases or even prevent the fulfillment of the other. For example, if one requirement states that the system should include web-access, and another one state that no access to the system should be allowed from the web, we say that these two requirements are conflicting, i.e., the system will not be able to fulfill both of them.

Table 2.1 Priority Matrix

Priority	H	M	M	M	L	L	L
E-U	H	M	H	H	L	M	M
SID	M	M	L	-	-	L	-
R-C	M	M	-	L	-	-	L



ID	Requirement Description	Requires	Increases/Decreases value of
1	The VisiOn platform shall provide user-friendly interfaces for its users.	-	[+] 9,10, 11, 12, 16, 21, 33, 49, 50, 67
2	The VisiOn platform shall support access to the citizen's information to authorized users only.	3, 6, 32, 76	-
3	The VisiOn platform shall analyse the users' authorizations based on their needs to perform their activities.	69	-
4	The VisiOn platform shall control the usage of its services to authorised users' only.	-	-

Fig. 2.7 A Snapshot of the relations among the VisiOn requirements

- *Increases/Decreases_value_of,* occurs when choosing one requirement for implementation increases or decreases the value to the customer of another requirement(s), i.e., implementing a specific requirement may have a positive or negative influence on the customer value of some requirements. For example, providing a context-dependent notification system to the software will increase the customer satisfaction, since the software will automatically modify the notification means without any involvement at the customer side when the context changes.

Figure 2.7, shows a snapshot of the table that captures the relations among the VisiOn requirements[7]. Identifying such relations is essential to specify how some requirements are essential for the satisfaction of other ones and how some requirements add value to other ones, which helps in deciding which requirements should be implemented. In other words, considering requirement dependencies enables for avoiding situations where some requirements have been classified as low priority based on the feedback of the stakeholders, and they are *required by*/*increases_value_of* high priority requirements.

- *VisiOn requirements validation.* The first elicited set of VisiOn requirements was validated by the feedback received from the partners and the individual meetings we arrange with them during the Technical Meeting[8]. In Particular, we ask them to check the requirement carefully and provide a feedback concerning them.

[7] Conflicts_with relations are not shown in the table since we already resolve all the inconsistencies that use to exist among the requirements

[8] Occurred in Rome with the participation of all VisiOn partners

5. (II) Eliciting the VisiOn User Requirements (second iteration). This section describes our activities for eliciting the VisiOn user requirements during the second iteration of this activity.

5.1(II) VisiOn Requirements Questionnaire II. In what follows, we describe how the second VisiOn requirements questionnaire II was designed and filled.

5.1.1 (II) Design the VisiOn Questionnaire II (Q2). Q2 was designed with the main objective of eliciting more detailed requirements from the two types of VisiOn users (PA and citizen) concerning their functionalites and qualities, how they are expected to interact with VisiOn to perform such functionalities, and how the platform is expected to realize their defined qualities. In addition, Q2 was designed in a way to link the users' feedback with the different components of the VPP, which enable the component developers to better understand how they can modify and extend their tools/components to meet the defined functionalities and qualities. Therefore, we provided a specialized version of the questionnaire for each partner taking into consideration his/her input in Q1. In particular, Q2 template includes two sub-questionnaires specialized for the two types of VisiOn users (PA and citizen), to be filled by the partner for each PA and citizen users identified by them in Q1. In what follows, we describe each of the sub-questionnaire. Each of these sub-questionnaire contains six sections. The first section is different between the two sub-questionnaires, in the *PA user questionnaire* it aims to describe the *(1) system analysis,* that captures the interaction between the VPP and the system(s) that is/are using the citizens' information, and in the *citizen user questionnaire* it aims to describe the *(1) privacy requirements - identification,* that captures how the citizen is expected to interact with the VPP to specify its privacy requirements and how the VPP is expected to assist him/her during the process, etc. While the two questionnaires share the same following five sections, *(2) privacy requirements - visualization,* to capture what kind of information a PA/citizen might need to visualize, how it needs to visualize it, etc.; *(3) privacy requirements analysis,* to capture what kind of analysis the VPP should provide, what is the expected output of such analysis, etc.; *(4) privacy requirements analysis at run-time,* to capture what kind of analysis the VPP should perform at run-time, what is the expected output for such analysis, etc.; *(5) Privacy Level Agreement (PLA),* to capture the PA/citizen expectations about the PLA, which enable us to extend our knowledge concerning the PA/citizen objectives; and *(6) examples of usage,* to elicit requirements of the PA/citizen that the partner might forget to mention while compiling the previous sections.

5.1.2 (II) Filling and refining the VisiOn Questionnaire II (Q2). In line with what we did for Q1, we shared Q2 with three E-U partners and we asked them to fill and return. Similar to the Q1 filling and refining process, we assist them during this process. After receiving the filled questionnaires, we analyzed them, and we contacted some partners to refine their input until it is clear. In summary, very detailed needs of six PAs and three Citizens concerning the VPP were identified.

5.2 (II) Modeling and Analyzing the Scenarios II. Similar to what we did in the first iteration of this activity, we asked the VisiOn partners to enrich the STS-ml models of the scenarios they created in the previous iteration. We organized a workshop, which lasted a day, in which one expert modeler and two domain experts, per pilot partner, analyzed and extended the previously created STS-ml models. After the workshop the domain experts kept updating the diagrams without the help of the modeling experts. This led to the creation of more complete models that cover, with great details, the scenarios. We used such models to identify the related VisiOn user requirements, on the part of the system included in the scenarios.

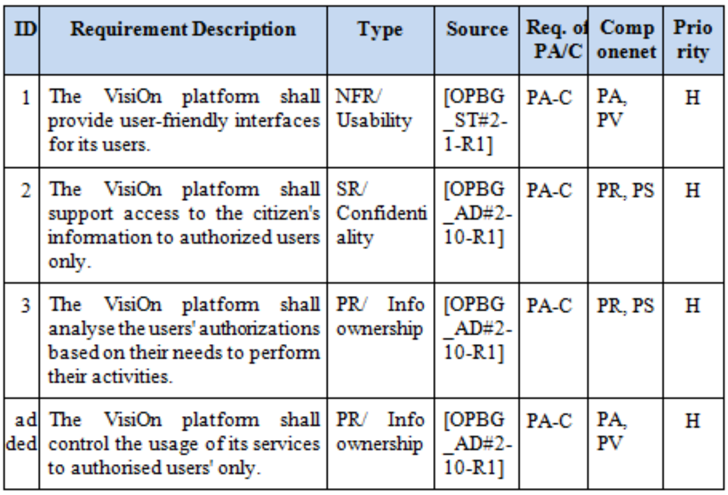
5.3 (II) Eliciting the VisiOn user requirements from questionnaires II and scenarios II. Similar to the first iteration of this activity, we used both of the questionnaires and scenarios to elicit the second set of VisiOn user requirements, which have been used to refine and extend the already elicited requirements to produce the final list of VisiOn user requirements (41 new requirements).

6. Consolidating the VisiOn requirements[9]. Requirements validation is very important activity, since detecting errors in the requirements during the design phase is much less expensive and time-consuming than discovering such errors after the system implementation [52]. Following [70], we performed five checks (validity, completeness, consistency, realism, and verifiability) to validate the VisiOn requirements. In what follows, we discuss how each of these checks has been performed to produce the final consolidated list of VisiOn requirements:

- *VisiOn requirements validity check,* aims to verify the elaborated requirements with all the stakeholders of the system-to-be. We performed this check by sharing the VisiOn requirements with all the partners, and we asked them to carefully check the requirements and provide us with their feedback. The feedback contains suggestions to revise and refine some requirements in order to better define the functionalities/features they require the system to deliver.
- *VisiOn requirements completeness check,* aims to verify that the elaborated requirements capture all the functions, features, constraints, etc. that are expected by the system users. We performed the completeness check by asking the End-User (E-U) partners that represent both PAs and citizens to check the elaborated list of requirements and whether they describe all the functionalities and features they expect the system to deliver. Some partners asked to add new requirements to the list that were not included in the requirements we elaborate.
- *VisiOn requirements consistency check,* aims to verify that the elaborated requirements are consistent with one another, i.e., no inconsistency should exist among them. The consistency check was able to detect some conflicts among the requirements. However, we manage to solve this issue by revising the conflicting requirements with the help of the partner(s) who identify such requirements[10].

[9] Requirements consolidation is used to refer to the validation of the final list of VisiOn user requirements

[10] We depend on STS-ml to analyze the consistency of some of the functional requirements (e.g., security, trust, etc.)



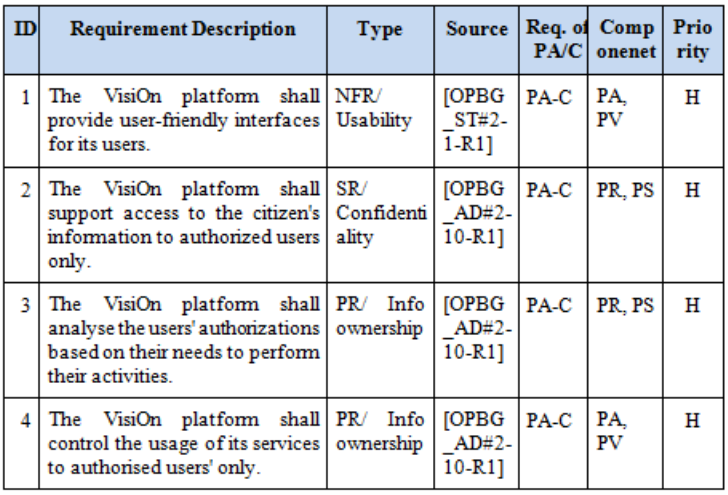
ID	Requirement Description	Type	Source	Req. of PA/C	Componenet	Priority
1	The VisiOn platform shall provide user-friendly interfaces for its users.	NFR/ Usability	[OPBG_ST#2-1-R1]	PA-C	PA, PV	H
2	The VisiOn platform shall support access to the citizen's information to authorized users only.	SR/ Confidentiality	[OPBG_AD#2-10-R1]	PA-C	PR, PS	H
3	The VisiOn platform shall analyse the users' authorizations based on their needs to perform their activities.	PR/ Info ownership	[OPBG_AD#2-10-R1]	PA-C	PR, PS	H
added	The VisiOn platform shall control the usage of its services to authorised users' only.	PR/ Info ownership	[OPBG_AD#2-10-R1]	PA-C	PA, PV	H

Fig. 2.8 A Snapshot of the requirements table shared with the partners

- *VisiOn requirements realism check,* aims to verify that the requirements can actually be implemented. We performed this check by sharing the requirements list with the partners that are responsible for developing the components of the VPP, and we ask them to carefully check the requirements list and provide us with their feedback. The feedback contains suggestions to revise several requirements, and mark 15 of them as out of the VPP scope. In addition, we had teleconference meetings with them to discuss the requirements one-by-one. After the meeting, the requirements list was revised accordingly. A snapshot of the shared requirements table is shown in Figure 2.8.
- *VisiOn requirements verifiability check,* aims to ensure that the requirements are documented in a clear and understandable way so that they can be verifiable by the different stakeholders of the system, which reduce any potential dispute among the stakeholders concerning the requirements. This check was performed by sharing the final list of requirements with End-Users (PAs and Citizens) and Component developers, i.e., both of them were able to check and provide their feedback concerning the same requirements list. Both of them verify that the requirements are clear, understandable and describe all the functionalities and features they expect the system. Moreover, we kept records of all the documents we shared with the different partners along with their feedback on these documents, which enables for resolving any potential dispute between the two sides.

A snapshot of the table that contains the consolidated VisiOn user requirements is shown in Figure 2.9, where each requirement is described with the following attributes:

- *Req. ID:* A unique identifier for each requirement.



ID	Requirement Description	Type	Source	Req. of PA/C	Comp onenet	Prio rity
1	The VisiOn platform shall provide user-friendly interfaces for its users.	NFR/ Usability	[OPBG _ST#2-1-R1]	PA-C	PA, PV	H
2	The VisiOn platform shall support access to the citizen's information to authorized users only.	SR/ Confidentiality	[OPBG _AD#2-10-R1]	PA-C	PR, PS	H
3	The VisiOn platform shall analyse the users' authorizations based on their needs to perform their activities.	PR/ Info ownership	[OPBG _AD#2-10-R1]	PA-C	PR, PS	H
4	The VisiOn platform shall control the usage of its services to authorised users' only.	PR/ Info ownership	[OPBG _AD#2-10-R1]	PA-C	PA, PV	H

Fig. 2.9 A Snapshot of the consolidated VisiOn user requirements

- *Description:* a textual description of the requirement, and a clarificatory text for some requirement.
- *Type:* the type of the requirement based on our taxonomy.
- *Source:* used for traceability reasons, requirement source is represented with a unique identifier that specifies the source where the requirement has been elicited from.
- *Req. of (PA/C):* whether it is a requirement for Public Administration (PA) and/or for Citizen (C).
- *Component:* it identifies the component(s) that will realize such requirement, where we have five VisiOn components, Privacy Assessment (PA), Privacy Requirements (PR), Privacy Specification (PS), Privacy Run-Time (PRT), and Privacy Transparency Visualization (PTV).
- *Priority (H/M/L):* indicates how important the requirement is in order to achieve the objectives of the project: 1- (H)igh: Must have, 2- (M)edium: Should have, and 3- (L)ow: Nice to have.

2.5 Approach threats to validity

After presenting and discussing our approach, we discuss the threats to its validity. Following [79], we classify threats to validity under four types:

Threats to construct validity concerns the relationships between theory and observation, i.e., to what extent a test measures what it claims to be measuring [64, 79]. We have identified the following two threats: *(1) Hypothesis guessing,* occurs when a participant of the experiment is able to guess the desired result, which may influences his/her response [75]. To mitigate this threat, the different questionnaires concerning requirements elicitation, prioritization, classification and validation/consolidation were designed carefully in order not to influence nor guide the participants. *(2) Experimenter expectations,* occurs when experimenter's expectations are communicated unintentionally to participants [75]. To avoid such threat, we shared all the questionnaires with the partners who are not participants and ask them to check whether the questionnaire is properly designed, i.e., it does not communicate any information that might reveal the experiment expectations to the participants.

Threats to internal validity concerns with external factors that have not been considered in the study, and they could have influenced the dependent variables in the study [75]. We have identified one internal threat. *Researcher bias,* occurs when the researcher influences the outcome of the study. To reduce the probability of such threat, the role of the researchers during all the activities that involve participants were limited only to assist them when needed without influencing their decisions. Moreover, we followed clear criteria while dealing with the participants' feedback concerning requirements elicitation, prioritization, classification and validation/consolidation.

Threats to external validity concerns the ability to generalize the results of the study. We have identified the following external threat: *Extensive evaluation,* the approach has been applied to only one project that concerns different application areas. This may threaten *the generalization of our findings*. However, we aim to better validate the approach by applying it other projects in different application domains.

Threats to reliability validity concerns the relationship between the treatment and the outcome, i.e., to what extent the study is dependent on the researcher(s), i.e., if another researcher(s) conducted the same study, the result should be the same. Detailed information concerning all the performed activities/adopted techniques (e.g., questionnaires, scenarios) for eliciting, classifying, prioritizing and validating/consolidating the VisiOn requirements are available at [28], and the overall process can be repeated. However, repeating these activities may not return the exact same results, but it presents a strong evidence about the reliability of the approach application.

2.6 Related work

Several approaches for Privacy Requirements Specification have been proposed in the literature. For instance, Spiekermann and Cranor [72] propose a framework that enables system analysts to build privacy friendly information systems. The framework contains high-level responsibilities for system analysts that stem from well-accepted definitions of privacy. In particular, according to the authors there are privacy issues in cases of data storage, transfer, and processing and the system analysts are instructed to understand privacy expectations of users. Four levels of system privacy friendliness are presented along with guidelines on how each level can be achieved. A basic principle of this approach is that the less easy is to identify a user based on some data the more privacy-friendly the system is.

A threat-based approach to elicit privacy requirements, named LINDDUN, is proposed by Deng et al. [16], which includes a systematic methodology and catalog of privacy related threat tree patterns. In particular, the authors propose a mapping of privacy threat types to system components that are modeled with Data Flow Diagrams (DFDs). Once privacy threat types are identified then they are further refined with the help of privacy threat tree patterns specifically developed for each threat type. Finally, the authors present a mapping of privacy requirements to existing Privacy Enhancing Technologys (PETs) in order to support analysts that are not experts in privacy technologies.

PriS [37] is a privacy requirements engineering method that allows system analysts to identify privacy requirements from the early stages of software development. Privacy requirements are considered organizational goals that must be satisfied by system under development. In particular, the method is based on the Enterprise Knowledge Development (EKD) framework where system requirements are modeled as goals and a goal hierarchy of the system is built. In turn, the analyst needs to identify processes that realize the goals. Similarly, privacy requirements are modeled as privacy goals, which may cause the modification of existing goals or the creation of new ones. Then, respective privacy processes have to be identified, which can be carried out with the support of a set of privacy-process patterns that the authors describe. Furthermore, appropriate privacy-enhancing technologies can be identified that support the business processes with regards to privacy.

The OASIS Privacy Management Reference Model and Methodology (PMRM) [65] focuses on the management of privacy requirements and risks. It contains a series of steps that guide analysts in the identification and scoping of use cases and mapping of privacy policies to privacy controls, both technical and procedural.

The PReparing Industry to Privacy-by-design by supporting its Application in REsearch (PRIPARE) methodology [60] is the result of a European Union funded project, which aims to integrate existing practices and research proposals on privacy engineering. It contains seven phases that enable the analyst to consider privacy issues, from the analysis phase where privacy requirements need to be identified to the decommission phase where personal data needs to be protected when the system is dismantled. During each phase, a number of different modeling languages and techniques are proposed for the analyst to employ, such as Unified Modelling Language (UML) or LINDDUN.

Radics et al. [58] present a framework that is mostly focused on the privacy requirements elicitation stage. In particular, it guides the analyst on the collection of relevant data that are used for the elicitation of privacy requirements through the identification of privacy related patterns in the collected data.

Some of the approaches reviewed above although they cover all the phases that are required for the specification of privacy requirements, they offer only high-level support to the analyst. On the other hand, there are approaches that include detailed steps for the completion of tasks but they do not cover all the tasks that are part of the privacy requirements specification. The approach that we employed in this chapter contained guidelines for all the required tasks, such as elicitation, classification, prioritization, and validation of privacy requirements.

General privacy taxonomies such as (Anton et al. [4]), (Solove et al. [68]), and (Wuyts et al. [80]) can serve as a general knowledge repository for a knowledge-based privacy goal refinement. However, they lack though a systematic process that can be followed in order to specify privacy requirements.

There are also approaches that consider privacy as part of security requirements. For example, Liu et al. [47] present a methodological framework that enables the identification of security and privacy requirements by employing a set of analysis mechanisms. These are applied within the i* modeling language, where security and privacy are considered as soft goals, and lead to the systematic extraction of security and privacy threats and related countermeasures.

Van Lamsweerde [77] present an extension of the KAOS framework for elaborating security requirements. In this method the software engineer constructs two models, an intentional model of the system under development and an anti-model that contains vulnerabilities and capabilities for achieving the anti-goals that threaten the systems security goals. Apart from confidentiality, integrity, availability, authentication, and non-repudiation, privacy is also considered as a security goal.

Giorgini et al [27] introduce the concepts of ownership, provisioning, trust, and delegation, in order to enable software engineers to consider security issues throughout the whole development process. By employing the aforementioned concepts the authors claim that privacy requirements can be captured. Mouratidis and Giorgini [52] propose a security-oriented methodology is presented where a security requirement is considered as a restriction that can influence the analysis and design of a system under development by restricting some alternative design solutions, by conflicting with some of the requirements, or by refining the objectives of the system. Such restrictions can be in terms of integrity, availability, and privacy.

The above approaches treat privacy mainly as confidentiality protection of personal data. However, privacy goals include also anonymity, unlinkability, unobservability, and pseudonymity among others. Without appropriate techniques that force the software engineer to look at these aspects of privacy it is very likely that important privacy requirements will be omitted for the system under development.

2.7 Conclusions and future work

In this chapter, we have presented a holistic requirements engineering approach for eliciting, classifying, prioritizing and validating privacy requirements. In particular, it combines several existing requirements engineering activities that have been adapted in order to deal with privacy requirements. The approach has been successfully used to elicit, classify, prioritize and consolidate the VisiOn platform requirements. In particular, the consolidated list of requirements is the result of an iterative and incremental process that has intertwined the use of state of the art techniques for requirements elicitation with a close interaction with stakeholders and users, where these requirements have been used to define the main functionalities and qualities of two types of VPP users (e.g., PAs and citizens). In addition, the requirements have been used by component developers to identify how their tools need to be extended and integrated into the VPP. This approach has been developed to be used for real world projects (e.g., industry). Therefore, the process underlining the approach has been designed carefully to assist software engineers during the overall process for specifying privacy requirements. Moreover, each of the process activities has been accompanied with a detailed description of how it can be performed.

For future work, we are investigating how the proposed taxonomy of privacy requirements can be further refined into more concrete concepts depending on [25], and how privacy requirements are linked to other types of requirements such as security and trust. Moreover, we aim to better investigate the inter-dependencies between the requirements activities and especially between the two different requirements elicitation activities. In addition, we intend to provide a more expressive analysis of the mapping between requirements and the component of the system that will realize them. Finally, we aim to better validate the approach by applying it to other similar projects that belong to different domains.

References

[1] Alessandro Acquisti, Allan Friedman, and Rahul Telang. "Is there a cost to privacy breaches? An event study". In: *Fifth Workshop on the Economics of Information Security* (2006), pp. 1–20. DOI: 10.1.1.73.2942.

[2] Ritu Agarwal and Mohan R. Tanniru. "Knowledge Acquisition Using Structured Interviewing: An Empirical Investigation". In: *Journal of Management Information Systems* 7.1 (1990), pp. 123–140. ISSN: 0742-1222. DOI: 10.1080/07421222.1990.11517884. URL: http://www.tandfonline.com/doi/full/10.1080/07421222.1990.11517884.

[3] Annie I. Antón, Julia B. Earp, and Angela Reese. "Analyzing Website privacy requirements using a privacy goal taxonomy". In: *Proceedings of the IEEE International Conference on Requirements Engineering*. Vol. 2002-Janua. IEEE. 2002, pp. 23–31. ISBN: 0769514650. DOI: 10.1109/ICRE.2002.1048502.

[4] AnnieI. Antï£¡n and JuliaB. Earp. "A requirements taxonomy for reducing Web site privacy vulnerabilities". In: *Requirements Engineering* 9.3 (2004), pp. 169–185. ISSN: 0947-3602. DOI: 10.1007/s00766-003-0183-z. URL: http://link.springer.com/10.1007/s00766-003-0183-z.

[5] Ayb??ke Aurum and Claes Wohlin. "Requirements engineering: Setting the context". In: *Engineering and Managing Software Requirements*. Springer,

2005, pp. 1–15. ISBN: 9783540250432. DOI: 10.1007/3-540-28244-0_1. arXiv: arXiv:1011.1669v3.

[6] LINDEN J. BALL and THOMAS C. ORMEROD. "Putting ethnography to work: the case for a cognitive ethnography of design". In: *International Journal of Human-Computer Studies* 53.1 (2000), pp. 147–168. ISSN: 10715819. DOI: 10.1006/ijhc.2000.0372. URL: http://linkinghub.elsevier.com/retrieve/pii/S1071581900903720.

[7] Kristian Beckers and Sebastian Pape. "A Serious Game for Eliciting Social Engineering Security Requirements". In: *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference, RE 2016*. IEEE. 2016, pp. 16–25. ISBN: 9781509041213. DOI: 10.1109/RE.2016.39.

[8] H. Belani, K. Pripuzic, and K. Kobas. "Implementing web-surveys for software requirements elicitation". In: *Proceedings of the 8th International Conference on Telecommunications, 2005. ConTEL 2005.* 2005, pp. 465–469. ISBN: 953-184-081-4. DOI: 10.1109/CONTEL.2005.185931. URL: http://ieeexplore.ieee.org/document/1458610/.

[9] Hugh R. Beyer and Karen Holtzblatt. "Apprenticing with the customer". In: *Communications of the ACM* 38.5 (1995), pp. 45–52. ISSN: 00010782. DOI: 10.1145/203356.203365. URL: http://portal.acm.org/citation.cfm?doid=203356.203365.

[10] B. Bovee, M., Srivastava, R., and Mak. "A conceptual framework and belief-function approach to assessing overall information quality. Paper presented at the Proceedings of the 6th International Conference on Information Quality". In: *International Journal of Intelligent Systems* 18.1 (2001), pp. 311–328.

[11] P??r Carlshamre et al. "An industrial survey of requirements interdependencies in software product release planning". In: *Proceedings of the IEEE International Conference on Requirements Engineering*. IEEE. 2001, pp. 84–91. ISBN: 0-7695-1125-2. DOI: 10.1109/ISRE.2001.948547.

[12] Ann Cavoukian. "Privacy by Design: Origins, Meaning, and Prospects". In: *Privacy Protection Measures and Technologies in Business Organizations: Aspects and Standards: Aspects and Standards* (2011), p. 170.

[13] Lawrence Chung et al. "Non-Functional Requirements in Software Engineering". In: *Conceptual modeling: Foundations and applications* (1999), p. 472.

[14] European Commission. *European Data Protection Supervisor*. 2016. URL: http://ec.europa.eu/justice/data-protection/bodies/supervisor/index{_}en.htm.

[15] IEEE Computer Society Software Engineering Standards Committee. "Recommended Practice for Software Requirements Specifications". In: *IEEE Std 830-1998*. Institute of Electrical and Electronics Engineers. 1998. ISBN: 0738103322.

[16] Mina Deng et al. "A privacy threat analysis framework: supporting the elicitation and fulfill...: EBSCOhost". In: *Requirements Engineering* 16.1 (2011), pp. 1–27. URL: http://web.b.ebscohost.com.library.capella.edu/ehost/pdfviewer/pdfviewer?sid=e7ebe3bc-59f7-43a0-ace9-60485dc3acd3@sessionmgr111{\&}vid=1{\&}hid=118.

[17] George Duncan. “Engineering Privacy by Design”. In: *Science (New York, N.Y.)* 317.5842 (2007), pp. 1178–1179. ISSN: 0036-8075. DOI: 10.1126/science.1143464.

[18] Zakareya Ebrahim and Zahir Irani. “E-government adoption: architecture and barriers”. In: *Business process management journal* 11.5 (2005), pp. 589–611.

[19] J??ao Fernandes et al. “IThink : A game-based approach towards improving collaboration and participation in requirement elicitation”. In: *Procedia Computer Science* 15 (2012), pp. 66–77. ISSN: 18770509. DOI: 10.1016/j.procs.2012.10.059. arXiv: 11/09 [ACM 978-1-4503-0816-8].

[20] William Foddy. *Constructing Questions for Interviews and Surveys: Theory and Practice in Social Research*. Cambridge university press, 1993. URL: https://books.google.co.uk/books?hl=en{\&}lr={\&}id=tok{_}OKwywQIC{\&}oi=fnd{\&}pg=PR7{\&}dq=questionnaires+in+social+research{\&}ots=Tybbm2R3LP{\&}sig=Mqh2DafK5wKDOkcDuvOgIj3hl6s.

[21] David Garlan, Robert Allen, and John Ockerbloom. “Architectural mismatch or why it’s hard to build systems out of existing parts”. In: *Proceedings of the 17th international conference on Software engineering - ICSE ’95*. ICSE ’95. ACM, 1995, pp. 179–185. ISBN: 0897917081. DOI: 10.1145/225014.225031. URL: http://portal.acm.org/citation.cfm?doid=225014.225031.

[22] Robert Gellman. “Privacy , Consumers , and Costs - How The Lack of Privacy Costs Consumers and Why Business Studies of Privacy Costs are Biased and Incomplete”. In: *Ford Foundation*. March. 2002, pp. 1 –37.

[23] Mohamad Gharib and Paolo Giorgini. “Modeling and Reasoning About Information Quality Requirements”. In: *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer. Springer, 2015, pp. 49–64. ISBN: 978-3-319-19237-6. DOI: 10.1007/978-3-319-16101-3_4. URL: http://link.springer.com/10.1007/978-3-319-16101-3{_}4.

[24] Mohamad Gharib, Paolo Giorgini, and John Mylopoulos. “Ontologies for Privacy Requirements Engineering: A Systematic Literature Review”. In: *arXiv preprint arXiv:1611.10097* (2016). arXiv: 1611.10097. URL: http://arxiv.org/abs/1611.10097.

[25] Mohamad Gharib, Paolo Giorgini, and John Mylopoulos. “Towards an Ontology for Privacy Requirements via a Systematic Literature Review”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 10650 LNCS. Springer, Cham, 2017, pp. 193–208. ISBN: 9783319699035. DOI: 10.1007/978-3-319-69904-2_16. URL: http://link.springer.com/10.1007/978-3-319-69904-2{_}16.

[26] Mohamad Gharib et al. “Privacy Requirements: Findings and Lessons Learned in Developing a Privacy Platform”. In: *Proceedings - 2016 IEEE 24th International Requirements Engineering Conference, RE 2016*. IEEE. 2016, pp. 256–265. ISBN: 9781509041213. DOI: 10.1109/RE.2016.13.

[27] Paolo Giorgini, Fabio Massacci, and Nicola Zannone. “Security and Trust Requirements Engineering”. In: *Foundations of Security Analysis and Design* 3 (2005), pp. 237–272. ISSN: 16113349. DOI: 10.1007/11554578_8. URL: papers3://publication/uuid/70b6c194-8401-45ee-bc52-17b6aecd336c.

[28] Paolo Giorgini et al. *D2.2 Citizens and public administration privacy requirements {V} 2.0*. Tech. rep. Universitá degli studi di Trento, 2016.

[29] J.A. Goguen and C. Linde. “Techniques for requirements elicitation”. In: *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering* 93 (1993), pp. 152–164. ISSN: 0740-7459. DOI: 10.1109/ISRE.1993.324822. URL: http://ieeexplore.ieee.org/document/324822/.

[30] Markus Helfert and Clemens Herrmann. “Proactive Data Quality Management for Data Warehouse Systems - A Metadata based Data Quality System”. In: *4th International Workshop on Design and Management of Data Warehouses (DMDW 2002)*. Vol. 2002. 2002, pp. 97–106. URL: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-58/herrmann.pdf.

[31] a Herrmann and M Daneva. “Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research”. In: *Proc. 16th IEEE Int’l Conf. Requirements Eng.* iv. IEEE. 2008, pp. 125–134. ISBN: 0-7695-1980-6. DOI: 10.1109/RE.2008.48.

[32] A N N M Hickey and Alan M Davis. “A Unified Model of Requirements Elicitation”. In: *Journal of Management Information Systems* 20.July 2015 (2014), pp. 65–84. DOI: 10.1080/07421222.2004.11045786.

[33] Ann M Hickey and Alan M Davis. “Requirements Elicitation and Elicitation Technique Selection : A Model for Two Knowledge-Intensive Software Development Processes Unsolved Problem Software Development Software Solutions”. In: *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*. C. IEEE. 2002, pp. 2005–2010. ISBN: 0769518745.

[34] Dennis Neil Hinkle. “The change of personal constructs from the viewpoint of a theory of construct implications”. PhD thesis. Ohio State University Columbus, 2010, pp. 1–61.

[35] Michael Jackson. “The World and the Machine”. In: *1995 17th International Conference on Software Engineering*. IEEE. 1995, pp. 1–10. ISBN: 0-89791-708-1. DOI: 10.1145/225014.225041.

[36] C Jones. *Applied Software Measurement-Assuring Productiviety and Quality*. New York, NY, USA: McGraw-Hill, Inc., 1991. ISBN: 0-07-032813-7.

[37] Christos Kalloniatis, Evangelia Kavakli, and Stefanos Gritzalis. “Addressing privacy requirements in system design: The PriS method”. In: *Requirements Engineering* 13.3 (2008), pp. 241–255. ISSN: 09473602. DOI: 10.1007/s00766-008-0067-3.

[38] Wentao Kang and Ying Liang. “A security ontology with MDA for software development”. In: *Proceedings - 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2013*.

IEEE. 2013, pp. 67–74. ISBN: 9780768551067. DOI: 10.1109/CyberC.2013.20.

[39] Joachim Karlsson, Claes Wohlin, and Björn Regnell. "An evaluation of methods for prioritizing software requirements". In: *Information and Software Technology* 39.14-15 (1998), pp. 939–947. ISSN: 09505849. DOI: 10.1016/S0950-5849(97)00053-0. URL: http://linkinghub.elsevier.com/retrieve/pii/S0950584997000530.

[40] Gerald Kotonya, Ian Sommerville, and Gerald Kotonya. *Requirements engineering: processes and techniques*. 1st. Wiley Publishing, 1998. ISBN: 0471972088, 9780471972082.

[41] Wadha Labda, Nikolay Mehandjiev, and Pedro Sampaio. "Modeling of privacy-aware business processes in {BPMN} to protect personal data". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM. 2014, pp. 1399–1405.

[42] Soren Lauesen. *Software requirements: styles and techniques*. Pearson Education, 2002.

[43] Karen Layne and Jungwoo Lee. "Developing fully functional E-government: A four stage model". In: *Government information quarterly* 18.2 (2001), pp. 122–136.

[44] Dean Leffingwell and Don Widrig. *Managing software requirements: a unified approach*. Addison-Wesley Professional, 2000.

[45] Soo Ling Lim and Anthony Finkelstein. "StakeRare: using social networks and collaborative filtering for large-scale requirements elicitation". In: *Software Engineering, IEEE Transactions on* 38.3 (2012), pp. 707–735.

[46] Soo Ling Lim, Daniele Quercia, and Anthony Finkelstein. "StakeNet: using social networks to analyse the stakeholders of large-scale software projects". In: *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*. ACM. 2010, pp. 295–304.

[47] Lin Liu, Eric Yu, and John Mylopoulos. "Security and privacy requirements analysis within a social setting". In: *11th International Requirements Engineering Conference*. IEEE. 2003, pp. 151–161.

[48] Pericles Loucopoulos and Vassilios Karakostas. *System requirements engineering*. McGraw-Hill, Inc., 1995.

[49] Antoni Martinez-Balleste, Pablo Perez-martinez, and Agusti Solanas. "The pursuit of citizens' privacy: a privacy-aware smart city is possible". In: *Communications Magazine, IEEE* 51.6 (2013), pp. 136–141.

[50] Nicolas Mayer. "Model-based management of information system security risk". PhD thesis. University of Namur, 2009.

[51] Nancy R Mead and Software Engineering. "Requirements Prioritization Introduction". In: *Framework*. September. Springer, 2008, pp. 1–7.

[52] H Mouratidis and P Giorgini. "Secure Tropos: A security-oriented extension of the Tropos methodology". In: *Journal of Software Engineering and Knowledge Engineering* 17.2 (2007), pp. 285–309.

[53] J Mylopoulos, L Chung, and B Nixon. “Representing and using nonfunctional requirements: A process-oriented approach”. In: *IEEE Transactions on Software Engineering* (1992), pp. 483–497.

[54] Bashar Nuseibeh and Steve Easterbrook. “Requirements engineering: a roadmap”. In: *Proceedings of the Conference on the Future of Software Engineering*. ACM. 2000, pp. 35–46.

[55] Elda Paja, Fabiano Dalpiaz, and Paolo Giorgini. “Modelling and reasoning about security requirements in socio-technical systems”. In: *Data & Knowledge Engineering* 98 (2015), pp. 123–143.

[56] Constantinos Patsakis et al. “Interoperable privacy-aware e-participation within smart cities”. In: *Computer* 48.1 (2015), pp. 52–58.

[57] Klaus Pohl. *Requirements engineering: An overview*. RWTH, Fachgruppe Informatik, 1996.

[58] Peter J Radics, Denis Gracanin, and Dennis Kafura. “Preprocess before you build: Introducing a framework for privacy requirements engineering”. In: *Social Computing (SocialCom), 2013 International Conference on*. IEEE. 2013, pp. 564–569.

[59] Björn Regnell et al. “An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software”. In: *Requirements Engineering* 6.1 (2001), pp. 51–62.

[60] Principles Report. *PReparing Industry to by supporting its Application in REsearch*. Tech. rep. 2014, pp. 1–60.

[61] Claudia Ribeiro et al. “Gamifying requirement elicitation: Practical implications and outcomes in improving stakeholders collaboration”. In: *Entertainment Computing* 5.4 (2014), pp. 335–345.

[62] Suzanne Robertson and James Robertson. *Mastering the requirements process: getting requirements right*. Addison-Wesley, 2012.

[63] Günther Ruhe, Armin Eberlein, and Dietmar Pfahl. “Trade-off analysis for requirements selection”. In: *International Journal of Software Engineering and Knowledge Engineering* 13.04 (2003), pp. 345–366.

[64] Per Runeson and Martin Höst. “Guidelines for conducting and reporting case study research in software engineering”. In: *Empirical software engineering* 14.2 (2009), pp. 131–164.

[65] J Sabo et al. “Privacy Management Reference Model and Methodology”. In: *OASIS PMRM TC Standards Track Committee Specification* (2013).

[66] Mattia Salnitri et al. “STS-Tool 3.0: Maintaining Security in Socio-Technical Systems.” In: *CAiSE Forum*. 2015, pp. 205–212.

[67] Anoop Singhal and Duminda Wijesekera. “Ontologies for modeling enterprise level security metrics”. In: *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*. ACM. 2010, p. 58.

[68] Daniel J. Solove. “A Taxonomy of Privacy”. In: *University of Pennsylvania Law Review* 154.3 (2006), p. 477. ISSN: 00419907. DOI: `10.2307/40041279`. arXiv: `arXiv:1011.1669v3`. URL: `http://www.jstor.org/stable/10.2307/40041279?origin=crossref`.

[69] Daniel J Solove. “Conceptualizing privacy”. In: *California Law Review* (2002), pp. 1087–1155.

[70] Ian Sommerville. *Software engineering 8*. pearson Education limitd, 2007.

[71] Ian Sommerville and Pete Sawyer. *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.

[72] Sarah Spiekermann and Lorrie Faith Cranor. “Engineering privacy”. In: *Software Engineering, IEEE Transactions on* 35.1 (2009), pp. 67–82.

[73] British Standard. “Information security managementï£¡Part 1: Code of practice for information security management”. In: *British Standard BS7799-1* 1999 (1999).

[74] A Terry Bahill and S J Henderson. “Requirements development, verification, and validation exhibited in famous failures”. In: *Systems Engineering* 8.1 (2005), pp. 1–14.

[75] W Trochim and J P Donnelly. *The Research Methods Knowledge Base*. Cengage Learning, 2006. isbn: 9781592602919.

[76] Bill Tsoumas and Dimitris Gritzalis. “Towards an ontology-based security management”. In: *20th International Conference on Advanced Information Networking and Applications (AINA)*. Vol. 1. IEEE. 2006, pp. 985–992.

[77] Axel Van Lamsweerde. “Elaborating security requirements by construction of intentional anti-models”. In: *Proceedings of the 26th International Conference on Software Engineering*. IEEE Computer Society. 2004, pp. 148–157.

[78] Gan Wang, Ricardo Valerdi, and Jared Fortune. “Reuse in systems engineering”. In: *Systems Journal, IEEE* 4.3 (2010), pp. 376–384.

[79] Claes Wohlin et al. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[80] Kim Wuyts et al. “Linking privacy solutions to developer goals”. In: *Availability, Reliability and Security, 2009. ARES’09. International Conference on*. IEEE. 2009, pp. 847–852.

[81] E Yu and L Cysneiros. “Designing for privacy and other competing requirements”. In: *2nd Symposium on Requirements Engineering for Information Security (SREIS’02), Raleigh, North Carolina*. Citeseer. 2002, pp. 15–16.

[82] Nicola Zannone. “A requirements engineering methodology for trust, security, and privacy”. PhD thesis. University of Trento, 2006.

[83] Didar Zowghi and Chad Coulin. “Requirements elicitation: A survey of techniques, approaches, and tools”. In: *Engineering and managing software requirements*. Springer, 2005, pp. 19–46.