Towards the weaving of the characteristics of good security requirements

Sravani Teja Bulusu, Romain Laborde, Ahmad Samer Wazan, François Barrère, Abdelmalek Benzekri

Towards the Weaving of the Characteristics of Good Security Requirements

Sravani Teja Bulusu(✉), Romain Laborde, Ahmad Samer Wazan, Francois Barrère, and Abdelmalek Benzekri

IRIT/Université Paul Sabatier, 118 Route de Narbonne, Toulouse, France
{Sravani-Teja.Bulusu,laborde,ahmad-samer.wazan,
Francois.Barrere,Abdelmalek.Benzekri}@irit.fr

Abstract. Over the past two decades, there has been a significant emphasis on the research work towards the amelioration within the discipline of security requirements engineering. Many researchers, international standards and organizations have come up with various methodologies to facilitate the elicitation and evaluation of security requirements. However, the task of deriving good quality requirements still remains challenging. One of the main reasons is that there is no consensus in defining what is a good and a bad requirement. The purpose of this paper is to provide with a survey of various quality characteristics of requirements proposed by various authors from different perspectives. Our survey analysis shows that there are a total of 20 distinctive characteristics that are defined in order to evaluate the quality aspects of requirements.

Keywords: Security requirements engineering · Requirement analysis · Requirement errors · Quality characteristics of security requirements

1 Introduction

Since early 90's many researchers and organizations have contributed their work towards the discipline of security requirements engineering. Security mainly subsumes to the three properties: availability, confidentiality and integrity. Typically, security requirements are derived on the basis of these ACI properties. From a broader perspective, all their contribution can be viewed as two parallel streams of research. One stream is towards eliciting, cataloging, evaluating and reusing of security requirements. In this context, numerous security concepts [1–3], security requirements engineering methodologies, modelling notations and security enhancements [4–7] were proposed. The second stream of research concerns with defining quality characteristics such as completeness, consistency, correctness, etc. [8]. These characteristics are used to evaluate the way requirements are derived; if they are good or bad. However, despite the research advancements, deriving good requirements still remain demanding and challenging till date. Yet many derived requirements are identified as poor requirements. The conspiracy lies within the term *good*. And ambiguity appears in answering basic questions like, what is the definition of a *good security requirement*? How can one measure a *security requirement*? How to identify a *bad security requirement*? One

of the reasons behind these ambiguities is there lacks a generic consensus or agreement in defining what are good and bad requirements.

In this context, we have made a study on the existing quality characteristics of requirements cited by different authors. We have developed a weaving strategy that allows us to provide with consolidated view of the existing characteristic definitions and their indifferences. This initiative work intends to highlight the necessity of consensus of quality characteristics for efficient and effective establishment of quality requirements.

The rest of the article is structured as follows. Section 2 of this paper briefly discusses on requirement errors. Section 3 surveys the proposed quality characteristics collected from eight sources. Our proposal is developed in Sect. 4. Finally, we conclude our work in Sect. 5.

2 Causes Behind Requirements Errors

Requirement errors are acknowledged as the most expensive errors compared to others within the whole system engineering process. Boehm [9] has stated that late correction of requirement errors could cost up to 200 times as much as correction during early stages of requirements engineering. For clear understanding of the problem, let us consider an example some requirements derived in a context to provide secure email service in an organization:

- **Req1** – Data flow between device1 and device2 shall be encrypted by a strong algorithm.
- **Req2** – Email transfers must be analysed
- **Req3** – The password recovery system must not disturb users.
- **Req4** – Analyse internal attackers not leave them

All these four requirements are prone to errors that could eventually impact the security design and implementation of the email service. To start with, first *Req1* is not clear. What is a strong encryption algorithm? Next, *Req2* has the same issue. It is not clear on what to analyze for. Here, analyzing the emails can be either detecting virus or detecting the disclosure of sensitive information. In addition, if the emails transfer data flow is encrypted because of *Req1* then it might not be possible to perform any analysis. And next *Req3* employs in terms of the negative form '*must not'*, which indicates what not to do instead of what to do. In addition it includes one more snag: how to evaluate if users are disturbed or not? Finally, *Req4* holds an ambiguity due to bad semantics. Imaginary interpretations can be made based on where a comma is placed. If interpreted like *"Analyse internal attackers, not leave them"* this demands an inspection of the internal attackers within the organization. In other hand, if we move a little bit the comma to the right of the statement, our interpretation can change completely. Indeed, *"Analyse internal attackers not, leave them"* means to ignore the internal attackers and do nothing about them. Improper verification of such requirement errors could create trouble at requirement implementation phases. However, identification of such requirement errors, particularly in the earlier stages of requirement engineering process, is known to be one of the hardest and tedious tasks. This is

because, most of the information during the earlier stages will be in the form of either abstract ideas or discussions or some rough drafts of old documents with some definitions etc.

GS Walia *et al.* [10] have classified the causing factors which could lead to requirement errors into three types, see Fig. 1. Human based errors correspond to shortcomings in the knowledge acquisition on domain environment, or stakeholder needs, and improper communication. Process based errors correspond to inadequate planning and implementation of requirements engineering process. Finally, documentation errors correspond to bad documentation of the elicited stakeholder needs or objectives that could lead to either missing or misrepresentation of requirements.

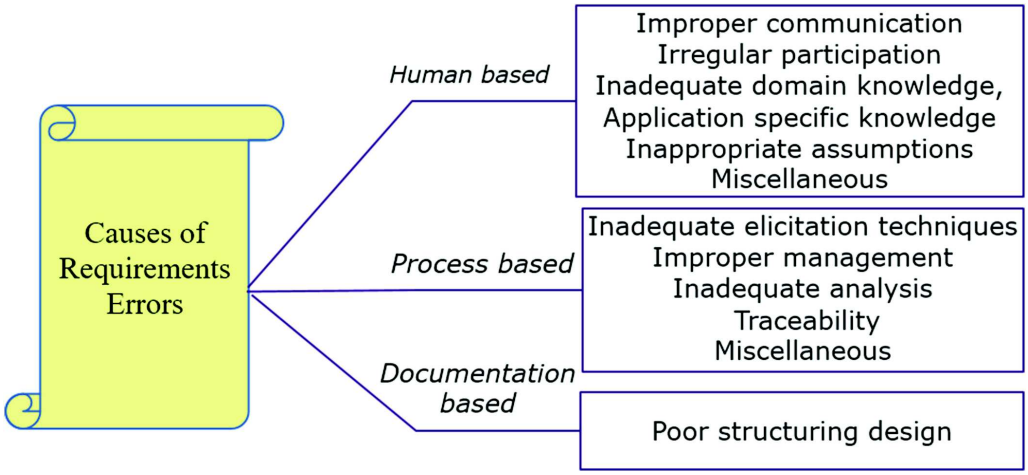

Fig. 1. Requirement errors: causing factors [10]

Although it is not only sufficient to analyse security requirements at the early stages, explicitly defining requirements errors is also mandatory. We need to characterize good requirements to minimize risks pertaining to bad quality requirements. And from security engineering perspective, these characteristics help to measure the quality of security requirements.

3 Characteristics of Good Requirements

This section provides the literature on the various quality characteristics gathered from related works. We have projected below the characteristics proposed by each of them. Different sources have listed different set of criteria defining different characteristics of good requirements. However, some of their criterion definitions share similar meaning with similar characteristic name. In such cases, to avoid reputation and to reduce space, we have included the definition only once. In all other cases we have included the respective definitions as given in the respective sources.

3.1 ISO29148

The international standard for requirements engineering [1] has defined a total of 12 characteristics to measure the quality of requirements. The definitions as per the standard are as follows:

1. **Complete:** for a singular requirement, complete means that the requirement needs no further amplification because it is measurable and meets stakeholder needs. For a set of requirements, it means that the selected requirements contain everything pertinent to the system to be built.
2. **Consistent:** The stated requirement must be free of conflicts with other requirements.
3. **Feasible:** The stated requirement must be technically achievable within the technological constraints of the system (e.g., cost, schedule, legal, regulatory, etc.) with acceptable risks.
4. **Affordable:** The set of defined requirements must be feasible within a given system life cycle constraints.
5. **Traceable:** The stated requirement is traceable upward to specific documented stakeholder needs. And also must traceable to downward to low end requirement specifications or design artefacts.
6. **Implementation Free:** The requirement must state only what is required when exhibiting the necessary characteristic and not how the requirement is met or achieved.
7. **Unambiguous:** The requirement must be stated that it does not lead to more than one interpretation of the same.
8. **Necessary:** a requirement is considered necessary when in cases it is removed; it will raise a deficiency in the system to be built.
9. **Bounded:** The set of requirements must maintain the identified scope for the intended solution without increasing beyond what is required.
10. **Singular:** The stated requirement must define only one need at a time with no use of conjunctions (i.e. atomic).
11. **Verifiable:** The requirement must possess means to prove that the system satisfies the specific requirement. This is enhanced when the requirement is measurable.
12. **Requirement language criteria:** vague and general terms used for the description of requirements are to avoid such as superlatives, subjective language, and vague pronouns.
13. **Attributes:** Requirements should have descriptive attributes defined to help in understanding and managing requirements. Requirement attributes may include stakeholder priority, requirement identification, risk related information etc.

3.2 Axel Van Lamsweerde

This source [4] has proposed 11 characteristics of requirements. In this work, there is no explicit mentioning of the applicability of those characteristics to a singular or to a set of requirements. However, new conceptual elements are considered in characteristic

definitions are domain properties (e.g. physical laws) and assumptions. The characteristics *completeness, consistency, unambiguity, traceability* and *feasibility* share same meaning as the ISO29148, we list the remaining characteristics in below:

1. **Adequacy:** The requirements translation to specifications must ensure that the actual needs of the new system are completely satisfied.
2. **Measurability:** The requirements must be formulated at a level of precision that enables people such as analysts, developers, users to verify and evaluate if the requirements really meets what is needed.
3. **Pertinence:** The requirements and assumptions must at least contribute to the satisfaction of one or several objectives.
4. **Comprehensibility:** The stated requirements must be comprehensible to the respective people who need to use them.
5. **Good Structuring:** The requirements document should be organized in a structured manner for clear understanding. For example: the definition of a term must precede its use.
6. **Modifiability:** The requirements document should be flexible to revise and adapt to any changes or modifications.

3.3 Donald Firesmith

This source [8] has mentioned a total of 15 characteristics. In this list, *completeness, consistency, feasibility,* and *lack of ambiguity* are similar to aforementioned works. The remaining characteristics are as follows:

1. **Metadata:** Individual requirements should have metadata (i.e., attributes or annotations) that characterizes them. This metadata can include (but is not limited to) acceptance criteria, allocation, assumptions, identification, prioritization, rationale, schedule, status, and tracing information.
2. **Cohesiveness:** Individual requirement should be cohesive. The requirements are considered cohesive if all its parts (data, interface, functions and quality) belong together.
3. **Validatability:** Individual requirements must actually fulfil the needs and desires of their primary stakeholders.
4. **Customer/User Orientation**: Individual requirements should be defined in a way that they are understandable and validatable around the customers and users. They should not include any technical jargon of the development team.
5. **Usability:** Stated individual requirements must be understandable and reusable by numerous stakeholders.
6. **Mandatory:** Individual requirements should be necessary and required to fulfil the organizational objectives.
7. **Relevance:** Some identified and specified “requirements” actually turn out to be outside of the scope of the current endeavour. Thus, it is important to ensure that individual requirements are relevant.

8. **Correctness:** Individual requirements should be semantically and syntactically correct. It should be the accurate elaboration of high level goal or high level requirement.
9. **Currency:** The requirements document should be updated when in need of changes or modifications
10. **Verifiability:** Requirements always have sources, and it is important that requirements are consistent with them. Similarly, requirements need to be consistent with the standards, guidelines, and templates that are used in their preparation. Thus, individual requirements should be verifiable.
11. **External Observability:** Requirements should not unnecessarily specify the internal architecture and design of an application or component. Thus, individual requirements should only specify behaviour or characteristics that are externally observable.

It should be noted that the characteristics of *cohesiveness* and *relevance* are ambiguous. Additionally, some characteristics encompass other characteristics, such as *completeness* that refers to *traceability* and *language criteria*.

3.4 Ian Sommerville

This source [11] has defined a list of 7 characteristics. Among them *completeness, consistency, verifiability, traceability, comprehensibility, adaptability (modifiability) and realism (feasibility)* are similar to aforementioned works. The remaining characteristic is as follows:

1. **Validity** – The requirements should provide the functions which ***best*** support the customer's needs.

It should be noted that, this characteristic definition is complex and ambiguous. The author uses an ambiguous term "*best support*" which can be interpreted in different ways.

3.5 R R Young

The author [12] has proposed a list of 15 characteristics. Among them *complete*, *consistent, feasible, traceable, unambiguous, necessary, written using standard construct* and *devoid of escape clauses (language criteria)*, *design independent (implementation free)* and *verifiable* are similar to aforementioned works. The remaining characteristics are as follows:

1. **Allocated:** The requirement is assigned to a component of the designed system.
2. **Non-redundant:** The stated requirement is not a duplicate one.
3. **Assigned a unique identifier:** Each requirement should be identified uniquely.
4. **Concise:** The stated requirement must be simple.
5. **Correct:** The facts related to requirement are accurate, and it is technically and legally possible.

Again in this work, some of the defined characteristics are not clear. The author has used the term *simple* and *accurate* to describe the characteristics.

3.6 E Hull *et al.*

The authors [13] have proposed a list of 14 characteristics. Among them *complete*, *consistent, feasible* and *verifiable*, *structured*, *unique, legal, abstract* (implementation free) and *non-redundant* are similar to aforementioned works. The remaining characteristics are as follows:

1. **Atomic:** The stated requirement must carry a single traceable element,
2. **Clear:** The stated requirement must be clearly understandable,
3. **Precise:** The requirement statement must be precise and concise.
4. **Modular:** The set of requirements must belong together or close to one another.
5. **Satisfied/qualified:** The requirements document must achieve the appropriate degree of traceability coverage.

3.7 Karl *et al.*

The authors [14] have proposed a list of 10 characteristics. Among them *complete*, *consistent, feasible, traceable, unambiguous, necessary,* and *verifiable*, *modifiable* and *correct* are similar to aforementioned works. The remaining characteristic is as follows:

1. **Prioritize:** The requirement stated must be assigned with an implementation priority

4 Towards the Weaving of the Characteristics of Good Security Requirements

In our study on the existing characteristics of good requirements, we have identified a total of 20 distinctive criteria definitions. However, we have observed many variations in their corresponding definitions. Our objective is to define an exhaustive list of the existing characteristics that can be integrated to any security requirement engineering process. This entails defining a weaving strategy that we present in the next section.

4.1 Weaving Methodology

To avoid confusions and misinterpretations, we have decided to:

1. Give a unique reference to each characteristic of good requirements. In previous section, we noticed that different authors, for similar criteria, have defined different names. We use the term criterion to refer to each characteristic name. As result, we have named 20 criteria and referred to them as C1 to C20.

Table 1. Survey on quality characteristics of requirements

No	Abstract criterion definition	Characteristics fetched from the works of different authors							Applicability	Credibility
		ISO29148 (3.1)	Lamsweerde (3.2)	Firesmith (3.3)	Sommerville (3.4)	R R Young (3.5)	Hull et al (3.6)	Karl et al (3.7)		
C1	**All requirements are included and meet the stakeholder needs**	Complete	Complete	Complete	Complete	Complete	Complete	Complete	All	High
C2	**Compatible, non-contradictory requirements**	Consistent	Consistent	Consistent	Consistent	Consistent	Consistent	Consistent	All	High
C3	***Accomplishable within the given financial, time, legal, technological constraints***	Feasible/ Affordable	Feasible	Feasible	Realism	Feasible	Feasible/legal	Feasible	All	High
C4	Requirements needs to be well documente	--	Well Structured	--	--	--	Structured	--	All	Low
C5	***Requirement should be able to refer back to its objective. Dependency or reference links between requirements should be explicity defined.***	Traceable	Traceable	Cohesiveness	Traceable	Traceable/ Allocated	Satisfied/ Qualified/ Modular	Traceable	All	High
C6	*Requirements should state what is needed but not how it is met*	Implementation Free	--	External Observability	--	Design Independent	Abstract	--	All	Medium
C7	*Documented requirements must be easily adaptable to new changes*	--	Modifiable	--	Adaptability	--	--	Modifiable	All	Medium
C8	No redundant requirements	--	--	--	--	Non-redundant	Non-redundant	--	All	low
C9	Every requirement is uniquely identified	--	--	--	--	Unique	Unique	--	All	low
C10	*Stakeholders needs are sufficienly expressed*	--	Adequacy	Validatability	Validity	--	--	--	Each	Medium

Table 1. Continued

No	Abstract criterion definition	Characteristics fetched from the works of different authors							Applicability	Credibility
		ISO29148 (3.1)	Lamsweerde (3.2)	Firesmith (3.3)	Sommerville (3.4)	R R Young (3.5)	Hull et al (3.6)	Karl et al (3.7)		
C11	*Requirements defined are simple using common terminology and non-technical jargon.*	--	Comprehensibility	Customer / User Orientation	Comprehensibility	Concise	Clear	--	Each	Medium
C12	*Requirements are defined precisely not leading to multiple interpretations*	Unambiguou	Unambiguou	Lack of Ambiguity	--	Unambiguou	Precise	Unambiguous	Each	High
C13	Requirements defined allows evaluation - quantifiable values	--	Measurable	--	--	--	--	--	Each	low
C14	*Every requirement has a purpose*	Necessary/Bounded	Pertinence	Mandatory/ Relevance	--	Necessary	--	Necessary	Each	Medium
C15	Requirement should correctly represent th facts and needs. Syntactically and semantically	--	--	Correctness/ Currency	--	Correct	--	Correct	Each	Medium
C16	*Non conjunctive requirements*	Singular	--	--	--	--	Atomic	--	Each	low
C17	*Should define some means to prove the compliance or satisfaction of requirement with stakeholder needs, standards and constraints.*	Verifiable	--	Verifiability	Verifiability	Verifiable	Verifiable	Verifiable	Each	high
C18	*Formulation of Requirement statements must follow specific criteria*	*Requirement language criteria*	--	--	--	Devoid of escape clauses/ Standard Construct	--	--	Each	low
C19	Requirements must be reusable by numerc stakeholders	--	--	Usability	--	--	--	--	Each	low
C20	*Individual requirements should be defined with some attributes or annotations that characterizes them*	Attributes	--	Metadata	--	--	--	--	Each	low

2. Colour different notable special cases. Same author has defined different characteristic names for similar criterion definition for which we have highlighted the characteristic name in orange colour. Another interesting case is to show the list criteria proposed by all the authors. This list is highlighted in bold. Furthermore, similar criterion definitions are named differently by different authors for which we have projected the variation in italic. And finally, we have used the blue colour to indicate the case where a criterion is proposed by only one author.
3. Give one-line definition to each criterion. If the characteristic is defined in ISO29148, we give their definition. Otherwise, we give the definition of the respective authors if the characteristic description is clear. Finally, when the characteristics description is ambiguous, we give our own interpretation. In this way, we link the different characteristics to each other and thereby address the ambiguities.
4. Distinguish the applicability of each characteristic to one requirement or to a set of requirements or to a requirements specification document as a whole. We have projected this difference in the *Applicability column* in the Table 1.
5. Define credibility scores in terms of the frequency of mentions of each criterion. Credibility *high* corresponds to criterion proposed by at least six authors; medium corresponds to criterion proposed by at least three authors; low corresponds to criteria proposed by less than three authors.

4.2 Weaving Results

We have highlighted our observations of the aforementioned variations in Table 1.

(a) **Criteria**: Criteria used by all the authors [C1, C2, C3 and C5]
(b) Criteria: Single criterion defined by only one author – [C13 and C19]
(c) *Characteristic name:* Different names used for same criterion definition by different authors [C3, C5, C6, C7, C10, C11, C12, C14, C16, C17, C18, C20]
(d) Characteristic Name: Different names defined by single author maps to single criterion [C3, C5, C11, C14, C15 and C17]
(e) Applicability – **All**: Applies to whole set of requirements or to Requirement specification document
(f) Applicability – **Each**: Single or set of requirements corresponding to a particular stakeholder needs

4.3 Discussion

The essence of requirements engineering deals with managing the evolution of business objectives, from abstract ideas in to an aggregated set of requirement specifications. The resulting requirement specifications document serves as a baselined source which fills the communication gap between stakeholders and system developers. Here comes the role of quality characterises which are used to evaluate the integrity and reliability of these specified requirements in terms of expression. In Table 1 significant amount of contribution can be observed from eight different authors. Many interesting aspects and

arguments were discussed in the respective criterion definitions from multiple perspectives. In below, we briefly discuss some notable propositions as well as notable indifferences within the characteristics definitions.

Criterion C1 ensures that final set requirement specifications sufficiently express all the needs of stakeholders, respecting all the considerable aspects and scenarios. In a way, this criterion insists on efficient requirements elicitation and risk analysis process. The difficulty in fulfilling this criterion lies in identifying all considerable aspects such as stakeholder security and risk management objectives. The common keyword (characteristic name) used to represent this criterion is *complete* with credibility *high.*

Criterion C2 ensures that all requirements are compatible and consistent with one another. Accordingly, this criterion **C2** insists on verifying if there exist any conflicts in terms of contradicting requirement statements, improper representation of viewpoints, or possibility of incompatible interpretations of a statement, etc. The difficulty in fulfilling this criterion corresponds to establishment of right level of trade-off as highlighted in related works [15–17]. This criterion indirectly contributes to the fulfilment of requirement completeness. The common keyword used is *consistent* with credibility as *high*.

Criterion C3 ensures that all those derived requirements are accomplishable within the given constrains. Constraints can be viewed in two ways, one as they are imposed by stakeholders and the other based on operational context. On a whole, this criterion insists on identifying and acquiring all the possible constraints in terms of financial or technological implementations. ISO defines some of the considerable constraints such as time, cost, and process control, financial, technical, legal, and regulatory. In addition, dependency constrains and domain constraints [4] can also be considered. The common keyword used is *feasible* with credibility *high.* And other keywords used are *affordable* (3.1), *realism* (3.4) and *legal* (3.6).

Criterion C4 ensures that all requirements within the document are well categorized and well documented in a structured manner so that it is maintainable with fewer changes. Credibility of this criterion is *low* and common keyword used is *structured*.

Criterion C4 ensures that all requirements within the document are prioritized and well documented in a structured manner. Credibility of this criterion is *low* and common keyword used is *structured*.

Criterion C5 ensures that specified within the document are traceable in both forward and backward ways. Credibility of this criterion is *high* and common keyword used is *traceable*. Some sources have highlighted different aspects in the same context; hence different keywords were used accordingly. The keywords are *cohesiveness* (3.3), *allocated* (3.5), *satisfied/qualified* (3.6).

Criterion C6 ensures requirements derived do not specify the implementation details of the solution instead it specifies what is needed. Credibility of this criterion is *medium* and the keywords used are *implementation free* (3.1), *external observability* (3.2)*, design independent* (3.5) and *abstract* (3.6).

Criterion C7 ensures that the document containing all set of derived requirements is modifiable and adaptable to changes. It is to note that this is like a Meta characteristic to criterion C4 (well structured). Credibility of this criterion is *medium* and the keywords used are *modifiable* (3.2 and 3.7), *adaptability* (3.4).

Criterion C8 ensures that there is no redundancy of information corresponding requirement needs. It insists during the requirements elicitation process, one must clearly be able to distinguish between redundant stakeholder needs and non-redundant stakeholder needs. Credibility of this criterion is *low* and the common keyword used is *non-redundant*.

Criterion 9 ensures that all the requirements in the document are uniquely identifiable. This criterion helps to achieve the traceability feature (C5). Credibility of this criterion is *low* and the common keyword used is *unique*.

Criterion C10 ensures that completeness feature of an individual requirement. In a way it insists on verifying if the stakeholder need is sufficiently elicited. Credibility of this criterion is *low* and the keywords used are *adequacy* (3.2) and *validatability* (3.3).

Criterion C11 ensures that requirements are derived using simple terminology without usage of technical jargon. Technical jargon corresponds to terminology used by different teams working in different areas of business operational environments. For example, terminology used in software development environment is difficult to be understood by individuals belonging to organizational environment. Hence, this criterion enforces that the derived requirement must be comprehensible to all the readers of the document with in the business environment. Credibility of this criterion is *medium* and the common keyword used is *comprehensibility*. Some sources have highlighted different aspects in the same context; hence accordingly different keywords used. They are *customer or user orientation* (3.3) and *clear* (3.6).

Criterion C12 ensures that the derived requirements are precise enough and does not lead to any misinterpretations. It is to note that this criterion is different from the previous one C11 (comprehensibility). C11 insists on the aspect that there is no difficulty in the comprehension of the text (phrase or sentence), in the way it was written (focus on terminology). And C12 insists on the aspect that the content of the text maintains careful precision while expressing the idea so that it does not lead to misinterpretation of the idea. In end, this criterion emphasizes on the verification that comprehension of the text is not wrong. It focuses on punctuation and meaning of terminology or vocabulary used. In a way, this criterion can be viewed as a meta-characteristic of the criterion C11. Credibility of this criterion is *high* and the common keyword used is *unambiguous.* Another keyword used is *precise* (3.6).

Criterion C13 it ensures that requirements derived can be measured with some quantifiable values. For example, consider a requirement need "*a service must be available to all the customers"*. This need cannot be measured and while eliciting such needs, it is important to elicit measurable information. For this derived requirement for this need can say *"a service must be available on an average to 'x' number of customers at 't' units of time"*. This way, the requirements can be measured. Credibility of this criterion is *low* and the common keyword used is *measurable* (3.2).

Criterion C14 ensures that the derived requirement specifies what is needed and it has not got any unnecessary information. It is to note that this criterion complements the criterion C10 (adequacy). Credibility of this criterion is *medium* and the common keywords used are *necessary* and *mandatory*. Some sources have highlighted different aspects in the same context; hence accordingly different keywords used. They are *bounded* (3.1), *pertinence* (3.2) and *relevance* (3.3).

Criterion C15 ensures that requirement must possess accurate and up to date information. Credibility of this criterion is *medium* and the common keyword used is *correct*. Firesmith [8] has highlighted another aspect within the same context with a key word *currency* (3.3).

C**riterion C16** ensures that one requirement derives one need. For example, if a requirement need says *"entrance to aircraft allowed to customers with boarding pass and special emergency pass"*. This is not singular or atomic in nature. It is speaking allowing customers of two different types. One can split this into two as: "*entrance to aircraft allowed to customers with boarding pass" and "entrance to aircraft allowed to customer's special emergency pass"*. This way it helps to defined more precisely what does it mean by saying special or emergency. Accordingly, we can say that this criterion C16 contributes towards C13 (measured). Credibility of this criterion is *low* and the keywords used are *singular* (3.1) and *atomic* (3.6).

Criterion C17 it ensures that each of the requirements is verifiable against the constraints, standards and regulations to ensure the correctness of the requirements. This criterion somewhere again falls between C10 (adequacy) and C15 (Accurate). Credibility of this criterion is *high* and the common keyword used is *verifiability.*

Criterion C18 it ensures the formulation of requirement must follow some standard so that they are understandable globally. Credibility of this criterion is *low* and the common keywords used are *requirement language criteria* (3.1) and *devoid of escape clauses* (3.5).

Criterion C19 it ensures that requirements must be formulated in such a way that they are reusable. This criterion emphasizes on the using some common pattern for similar type of requirement needs. Credibility of this criterion is *rare* and the common keyword is *usability* (3.3).

Criterion C20 it ensures that every requirement should be identified with some metadata such as attributes, acceptance criteria. This way, it facilitates in their validation and evaluation. Credibility of this criterion is *rare* and the keyword used is *Metadata* (3.3).

In our survey we have identified that criteria *complete*, *consistency, feasibility, traceability, verifiability and unambiguous* holds high credibility. However, apart from the credibility factor, the respective criterion definitions are ad hoc and they lack consensus. Inharmonious proposition of various aspects, concerning a characteristic definition, could result in missing or inadequate knowledge acquisition, vague comprehension or misinterpretation, etc. Quality criteria definitions in general are written in natural language and it is generally difficult to identify how failing of one criterion could impact the fulfilment of other criteria. Therefore, it is required to first obtain consensus in order to define what a good requirement is.

5 Conclusion

The importance of eliciting and evaluating requirements is largely recognized now. Different approaches, inspired by the domain of requirement engineering, have proposed methods to express and analyse requirements. These methods can help to

structure the early phases of requirements specification. However, what makes a requirement good is still an open question. Many quality characteristics have been proposed to describe the good quality of requirements. Nonetheless, there is no one complete and consistent list of quality characteristics. In this article, we have proposed a comprehensive survey on these characteristics showing that if some characteristics are common, other have the same name but different meanings or conversely different names for the same meaning, etc. Based on this analysis, we built a unified list of characteristics for good quality requirements.

In practice, it may seem not always possible that security requirements fulfill all these quality criteria; for instance achieving both anonymity and accountability security objectives. If there is no revocable anonymity scheme available, then it is not possible to identify the malicious users in case of any misuse (such as in case of preventing double spending of anonymous eCash [18]). Therefore, some trade-offs between the anonymity and accountability objectives need to be found in such a way that the final set of security requirements derived to address both security objectives should be non-conflicting. Therefore, although the link between quality criteria and security requirement engineering is not commonly seen, it is indeed important to consider the quality characteristics in order to derive good quality security requirements.

For future works, we plan to integrate these quality characteristics in the process of security requirements engineering. This will encompass providing a meta-model of security requirements including the quality characteristics. Also, we will have to link the meta-model to the risk management process as well as the processes of verification and validation of security requirements.

Acknowledgement. This work is part of project IREHDO2 funded by DGA/DGAC. The authors thank M. Michalski and Eric Lacombe, security experts at Airbus, for their useful comments. Finally, we would like to thanks the anonymous reviewers for their valuable inputs.

References

1. ISO, I., IEC, IEEE: ISO/IEC/IEEE 29148:2011 Systems and software engineering – Life cycle processes – Requirements engineering. International Organization for Standardization (2011)
2. Pohl, K.: Requirements Engineering: Fundamentals, Principles, and Techniques. Springer Publishing Company, Incorporated (2010)
3. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Eng. **11**, 102–107 (2006)
4. Van Lamsweerde, A.: Requirements engineering: from system goals to UML models to software specifications (2009)
5. Mouratidis, H., Giorgini, P.: Secure tropos: a security-oriented extension of the tropos methodology. Int. J. Softw. Eng. Knowl. Eng. **17**, 285–309 (2007)
6. Hatebur, D., Heisel, M., Schmidt, H.: A pattern system for security requirements engineering. In: The Second International Conference on Availability, Reliability and Security, 2007, ARES 2007, pp. 356–365. IEEE (2007)

7. Graa, M., Cuppens-Boulahia, N., Autrel, F., Azkia, H., Cuppens, F., Coatrieux, G., Cavalli, A., Mammar, A.: Using requirements engineering in an automatic security policy derivation process. In: Data Privacy Management and Autonomous Spontaneous Security, pp. 155–172. Springer, Heidelberg (2012)
8. Firesmith, D.: Specifying good requirements. J. Object Technol. **2**, 77–87 (2003)
9. Mills, H.D.: Software Engineering Economics by Barry W. Boehm (1982). Comments on
10. Walia, G.S., Carver, J.C.: A systematic literature review to identify and classify software requirement errors. Inf. Softw. Technol. **51**, 1087–1109 (2009)
11. Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. Wiley, Hoboken (1997)
12. Young, R.R.: The Requirements Engineering Handbook. Artech House (2004)
13. Hull, E., Jackson, K., Dick, J.: Requirements Engineering. Springer Science & Business Media (2010)
14. Wiegers, K.E.: Writing quality requirements. Softw. Develop. **7**, 44–48 (1999)
15. Egyed, A., Grunbacher, P.: Identifying requirements conflicts and cooperation: how quality attributes and automated traceability can help. IEEE Softw. **21**, 50–58 (2004)
16. Ciechanowicz, Z.: Risk analysis: requirements, conflicts and problems. Comput. Secur. **16**, 223–232 (1997)
17. Massacci, F., Zannone, N.: Detecting Conflicts between Functional and Security Requirements with Secure Tropos: John Rusnak and the Allied Irish Bank. MIT Press, Cambridge (2008). Social modeling for requirements engineering
18. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy (SP), pp. 397–411. IEEE (2013)