

Trustworthiness Attributes and Metrics for Engineering Trusted Internet-Based Software Systems

Nazila Gol Mohammadi¹(✉), Sachar Paulus², Mohamed Bishr¹,
Andreas Metzger¹, Holger Kōnnecke², Sandro Hartenstein²,
Thorsten Weyer¹, and Klaus Pohl¹

¹ Paluno – The Ruhr Institute for Software Technology,
Duisburg-Essen University, 45127 Essen, Germany
{nazila.golmohammadi, mohamed.bishr, andreas.metzger,
thorsten.weyer, klaus.pohl}@paluno.uni-due.de

² Department of Economics, Brandenburg University of Applied Sciences,
14770 Brandenburg, Germany
{sachar.paulus, holger.koennecke,
sandro.hartenstein}@fh-brandenburg.de

Abstract. Trustworthiness of Internet-based software systems, apps, services and platform is a key success factor for their use and acceptance by organizations and end-users. The notion of trustworthiness, though, is subject to individual interpretation and preference, e.g., organizations require confidence about how their business critical data is handled whereas end-users may be more concerned about usability. As one main contribution, we present an extensive list of software quality attributes that contribute to trustworthiness. Those software quality attributes have been identified by a systematic review of the research literature and by analyzing two real-world use cases. As a second contribution, we sketch an approach for systematically deriving metrics to measure the trustworthiness of software system. Our work thereby contributes to better understanding which software quality attributes should be considered and assured when engineering trustworthy Internet-based software systems.

Keywords: Trust · Trustworthiness · Trustworthiness attributes · Socio-Technical Systems · Information and communication technologies · Metric

1 Introduction

Trust underlies almost every social and economic relation and is regarded as the glue that binds society together. Humans, processes and organizations, with different perceptions and goals, increasingly interact via the Internet. In such online settings, gaining and establishing trust relations within socio-economic systems becomes more difficult where interactions are mediated by technology rather than face-to-face communication making it more difficult to infer trust through social clues. The question this paper deals with is about the software system attributes that can foster trustworthiness in and within Socio-Technical Systems (STS) mediated through online networks.

STS are increasingly becoming part of our daily life in form of apps, Internet-based applications, cyber-physical systems, services, etc. The people involved in online businesses, though, have generally limited information about each other and about the STS supporting their online and offline transactions. There are several reports indicating an increasing number of victims of cyber-crime leading to massive deterioration of trustworthiness in current STS. Therefore, individuals and organizations are becoming more and more concerned about trusting and placing confidence on current STS and show interest in how to handle their business critical data. Consequently, trustworthiness of a software, app, service or platform becomes a key factor for their wider use and adoption by organizations and end-users.

There are limited contributions that approach the trust and trustworthiness issues described from angles other than security. However, security is not the only aspect of trustworthiness. Most existing approaches have assumed that one-dimensional properties of services lead to trustworthiness of such services, and even to trust in it by users, such as a certification (e.g., Common Criteria), the presence of certain technologies (encryption), or the use of certain methodologies (SSE-CMM) [1–3]. In this work, we relax the assumptions of such a one-dimensional approach and instead consider a multitude of attributes.

With a literature review, we attempt to identify and capture the attributes so far known as contributing to trustworthiness. These attributes have been classified to major quality categories. This paper provides a structured and comprehensive overview on SQA and their contribution to trustworthiness. In addition, we provide methods for deriving trustworthiness metrics, which is also considered an important extension to our previous work in [28].

The remainder of this paper is structured as follows: Sect. 2 provides a brief overview on the fundamentals on trust and trustworthiness of STS. Section 3 discusses related work. Section 4 describes the classification of SQA contributing to trustworthiness and capture them as trustworthiness attributes. In Sect. 4.13 we finalized the introduced trustworthiness attributes with some recommendations. Section 5 investigates the existing methods for deriving metrics and presents our proposed method for defining trustworthiness metrics for evaluation of trustworthiness attributes. Section 6 presents our conclusions and the future work.

2 Fundamentals

This section introduces the notion of trust from different perspectives and moves on to define the meaning of trustworthiness and its relation to trust. We then identify the relation between trust and trustworthiness. Finally, we discuss how they relate to STS.

2.1 Trust and Trustworthiness: A Discussion

From a sociological perspective two converging branches of sociology characterize the field of STS. The first branch focuses on the societal whole, its complex structures and social systems. The second branch focuses on societal members, individual actions and

relations between them. This second branch brought to attention trust as an element emerging from individual interactions and based on individual actions [4]. In this second branch, individuals rely on people engaged in representative activities [5], in other words, they rely on those who act on our behalf in matters of economy, politics, government and science. Such dependence implies high degrees of trust on part of the individual. Extending this view to information systems, we also rely on systems to run daily activities across large swaths of our society. They can be referred to as STS which are comprised of networks of individuals and IS organized around certain tasks. The delegation of tasks to such STS by individuals or organizations entails establishing some level of trust in such systems by the individuals. Consequently, it can be said that the trustworthiness of such systems is a key concern that needs to be fostered and even engineered in the fabric of these systems to maintain high levels of trust within society.

One of the problems occurring when studying a notion like trust is that everyone experiences trust. Hence, it is a personal view of what trust actually is [6]. This is the first intuitive explanation of why trust has multiple and varying definitions. A second explanation is the fact that there are multiple definitions of trust simply because there are many different types of trust [7, 8].

In [4] trust is defined as “a bet about the future contingent actions of others”. The components of this definition are belief and commitment. There is a belief that placing trust in a person or a system will lead to a good outcome and then a commitment to actually place trust and take an action to use this system based on this belief. E.g., when a user decides to use a system on the web, then he is confident that it will meet his expectations. In [9], a different outlook on trust is presented by Luhmann. He explains that “further increases in complexity call for new mechanisms for the reduction of complexity”. Luhmann suggests that trust is a more effective mechanism for this purpose. Given this view we can assert that increasing trust in STS has the effect of reducing uncertainty and complexity both online and offline in our society and this in turn has positive social and economic impacts.

In this paper, we stick to the earlier mentioned definition of trust in [4] while extending it to include STS: “a bet about the future contingent actions of others be they individuals or groups of individuals, or entire STS”.

Trustworthiness on the other hand has been used sometimes as a synonym for security and sometimes for dependability. Trustworthiness in general is a broad-spectrum term with notions including reliability, security, performance, and user experience as parts of trustworthiness [10].

However, given our chosen definition of trust we argue that while trust is a concern that emerges from the personal observation of an STS by individuals, trustworthiness is a characteristic of the system that has the potential to influence the trust this person has in the system in a positive or negative way.

2.2 Trustworthiness in Socio-Technical Systems

STS are systems that include humans, organization and their IS. There are interactions between these autonomous participants, between human and organizations as a social and software system as technical interactions [11]. These social and technical

components strongly influence each other. Our focus is on distributed applications that enable connection and communication of people via the Internet. Therefore, here, STS are applications, services, and platforms where technology and human behaviour are mutually dependent [12]. Thus, in STS people or organizations may communicate or collaborate with other people and organizations that emanate from interactions mediated by technology rather than face-to-face communication or collaboration [13].

STS are to be made trustworthy to merit the trust of their users. Trustworthiness has been defined as assurance that the system will perform as expected [14]. Furthermore, trustworthiness of software has been defined as worthy of being trusted to fulfil requirements which may be needed for a particular software component, application, system [15]. Trustworthiness is a potentially central aspect of distributed STS. We argue it as a multi-dimensional construct combining specific attributes, properties and characteristics.

The relation between trust and trustworthiness concepts always depends on reasoning processes which are performed by users of the system explicitly or implicitly considering the risk and possible consequences. There could be an imbalance between the level of trust in and the trustworthiness of the system with the possibility of two extreme cases. Typical situations are e.g., when too conservative users miss potential benefits of the system or when too optimistic users take too much risk by using the system (data misuse, etc.). Hence, there are major concerns about the trustworthiness of STS as the underestimation of side-effects of untrustworthy systems and mismanaging the vital and critical trust requirements has led to cyber-crime, e-frauds, cyber-terrorism, and sabotage. Reports [12] show an increased number of citizens that have fallen victim to these crimes, e.g., data loss. All of these issues occur because of either lack of trustworthiness or the awareness thereof.

Therefore, trustworthiness has recently gained increasing attention in public discussion. Figure 1 illustrates the identified gap in research in building a well-accepted STS for supporting socio-economic systems in the real world. The supporting applications lack expected (demonstrated) characteristics of such kinds of systems in the real world. The first step in closing this gap thus is the identification of trustworthiness attributes that may contribute to trust of socio-economic entities. Then, STS should be made capable to present these properties and characteristics.

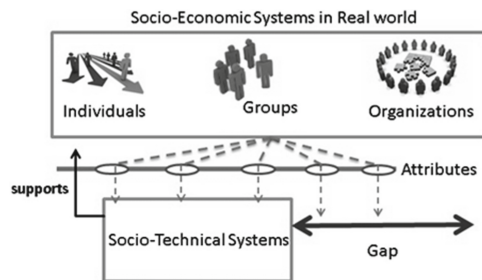


Fig. 1. The Socio-Technical gap inspired from [13].

There are, though, some inconsistencies between expected trust properties by the service consumer and promised trustworthiness from the service provider in general. To mitigate these deficiencies and to bridge the gap resulting from the asymmetry between trust and trustworthiness, we will investigate which trustworthiness attributes a system can hold (with which mechanism and/or technologies), and whether these attributes are capable of contributing to trustworthiness addressing the trust concerns of user.

3 Related Work

Trustworthiness in the literature has addressed the confidentiality of sensitive information, the integrity of valuable information, the prevention of unauthorized use of information, guaranteed QoS, the availability of critical data, reliability and integrity of infrastructure, the prevention of unauthorised use of infrastructure, etc. In order to prove being trustworthy, software applications could promise to cover a set of various quality attributes [10] depending on their domain and target users. Trustworthiness should promise a wide spectrum including reliability, security, performance, and user experience. But, Trustworthiness is domain and application dependent and a relative attribute, i.e. if a system was trustworthy in respect to some quality attribute like performance, it would not necessarily be successful in being secure. Trustworthiness and trust should not be regarded as a single construct with a single effect; rather it is strongly context dependent.

Related to this observation is the fact that the demonstration of trustworthiness attributes like Common Criteria certifications (ISO 15408) [16] or remote attestation procedures focus on security related attributes, whereas much more domains actually contribute to trustworthiness. E.g., a broad range of literature has argued and emphasized the relation between QoS and trustworthiness [17–22]. Therefore, trustworthiness is influenced by a number of quality attributes than just security-related. Trustworthiness of entities and individuals has been investigated in open, distributed systems (e.g., online marketplaces, multi agent systems, and peer-to-peer systems). Note that in this paper we strictly adhere to the perspective of a to-be-constructed system, and therefore will ignore potential trustworthiness (or trust) attributes like reputation or similar representing other users feedback, since they will only be available when the system is in use.

4 Survey of Trustworthiness Attributes

In this work, we investigate the properties and attributes of a software system that contribute to trustworthiness. To this end, we built on the software quality reference model defined by S-Cube [23]. The S-Cube model is extensive and has considered several other models such as: presented by Boehm [24], Adrion, et al. [25], McCall, et al. [26], and ISO 9126-1 [27]. In this paper we have excluded two types of the S-Cube SQA from our analysis. Firstly, some of the attributes contributing to trustworthiness are not identified in our literature review. Hence they were excluded.

Secondly, some quality attributes, e.g., integrity, can be achieved, among other ways, through encryption. In this case we included the high level attribute (integrity) as a contributor to trustworthiness but did not include encryption on its own because it is encompassed by the higher level attribute. Both cases are further discussed in Sect. 4.13. Additionally, we have included attributes that have been studied in the literature in terms of trustworthiness. These attributes are marked with an asterisk (*) in Fig. 2. This study is an extensive literature review. We aimed on identifying, evaluating and interpreting all available research relevant to an SQA and their potential on contributing to trustworthiness. The diagram below (Fig. 3) shows the resulted papers and their distribution in classified quality categories. Contributing characteristics of software systems to trustworthiness are captured as trustworthiness attributes. These attributes and belonging quality category are discussed in the following sections. The detailed literature and surveyed papers can be found in [28] with an indication of the respective papers. Figure 2 outlines the result of this work.

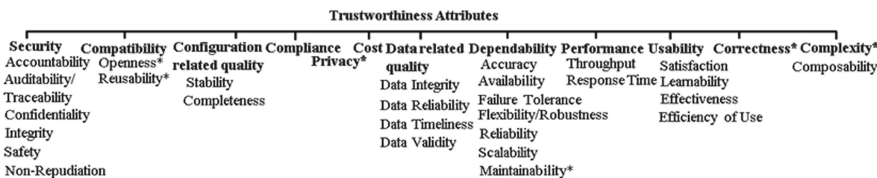


Fig. 2. Trustworthiness attributes.

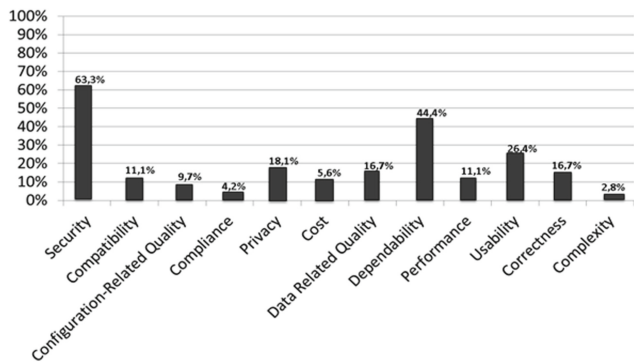


Fig. 3. Distribution of the SOTA in classified quality categories of trustworthiness attributes.

The definition of the trustworthiness attributes and their classified quality category is given in sub-sections bellow. Results of 72 relevant papers are presented in Fig. 3 as the distribution of the studies in different quality categories. A paper can obviously belong to multiple categories.

4.1 Security

Security covers the capability of a software system to protect entities against attacks and misuse despite certain vulnerabilities and to protect the access to resources. The sub-attributes of the security quality category are the following:

- **Accountability:** The state of being accountable, liable to be called on to render an account, the obligation to bear the consequences for failure to perform as expected.
- **Auditability/Traceability:** Capability of the service to be monitored and to generate in a reliable and secure way events producing an audit trail. Based on this audit a sequence of events can be reconstructed and examined. Security events could include authentication events, policy enforcement decisions, and others. The resulting audit trail may be used to detect attacks, confirm compliance with policy, deter abuse, or other purposes.
- **Confidentiality:** The ability to limit access to the system and its data only to authorised agents. It is defined as the absence of unauthorized disclosure of information.
- **Integrity:** The ability to ensure that the system and its data are not corrupted, improper system state alterations either accidental or malicious alternation or removal of information are prohibited.
- **Safety:** The ability to operate without risk of injury or harm to users and the system's environment. It can be achieved by absence of consequences on the users and the environment.
- **Non-Repudiation:** The ability to prove to the data sender that data have been delivered, and to prove the sender's identity to the recipient, so that neither the sender nor the recipient can deny operations of sending and receiving data.

4.2 Compatibility

Compatibility/Interoperability has been defined as the ability of diverse services to work constructively with each other. Actually, different services can coexist without side effects, without even knowing each other. Compatibility amounts to the necessity of two interacting parties to fulfil each other's constraints and, therefore, to correctly interact. The following sub-attributes belong to compatibility quality category:

- **Openness** means the system is designed in such a way that it is transparent how it works and how to connect to the system. This relates to other attributes like interoperability, transparency and extensibility [29, 30].
- **Reusability** can be defined on two levels, namely, syntactic level and operational. The former relies on type definition and type compatibility rules. The later is about operation signatures.

4.3 Configuration-Related Quality

This quality category contains quality attributes that influence the way a service is configured to function or characterize if the promised functional and quality level has

been actually delivered during the service's lifetime period e.g., completeness, stability. The following sub-attributes belong to configuration-related quality category:

- **Change Cycle/Stability:** Change related to the service in terms of its interface and/or implementation/recomposition.
- **Completeness:** A measure of the difference between the specified set of features (e.g., functions) and the implemented set of features.

4.4 Compliance

The service should comply with standards (e.g., industry specific standards) and/or regulations. This can affect a number of other attributes, such as e.g., the security, portability and interoperability of the service. Behaviour of a service should always comply with the user's expectation (specifications).

4.5 Privacy

In internet connected systems, privacy from a system perspective is viewed as the system's ability and functionality that allows users to take control of the usage of their private information. From this system perspective privacy is a strong contributor to trustworthiness of the system. Consequently, when designing systems the designers must ensure through their design process that the way in which the system will handle private information is in compliance with the local and international laws in order to render these systems as trustworthy.

4.6 Cost

Cost is a (composite) quality attribute consisting of three (atomic) service attributes: cost model, fixed costs and variable costs. Actually, cost can be computed either from all atomic cost attributes or only from the fixed costs attribute.

4.7 Data Related Quality

Data related quality (information and data quality) characterize input/output data by quality attributes that traditionally have been used in the information and data quality domains, e.g., accuracy and timeliness. The following sub-attributes belong to data related quality category:

- **Data Integrity:** It can be compromised by human errors, malicious attacks, intentional data modification, transmission errors, system/software bugs or viruses, or hardware malfunctions.
- **Data Reliability:** Correctness of the data used by the system. It depends on the sub-systems used as well as on the provenance of the data.
- **Data Timeliness:** The property of information being able to arrive early or at the right time.

- **Data Validity:** The data values satisfy acceptance requirements of the validation criteria or fall within the respective domain of acceptable values. Validity criteria are often based on “expert opinion” and are generally viewed as “rules of thumb” although some validity criteria may be based on established theory or scientific fact.

4.8 Dependability

Dependability of a computing system is the property/ability that reliance can justifiably be placed on the service it delivers. It also has been defined as a correct and predictable execution and ensured that, when executed, it functions as intended. In [14], dependability and trustworthiness are considered to have same goals while both suffering the same threats (faults, errors, and failures). The attributes belong to this quality category are as below:

- **Accuracy:** Definition of the error rate produced by the service calculated on the basis of the expected results.
- **Availability:** The ability to deliver services whenever it is required.
- **Failure Tolerance:** The ability of a service to provide its functionality to clients in case of failures. In general, it is the capability of a service to handle failures. The circumstances of service failures and how a service will react to failures are described. Compensation is its sub-attribute. It is the ability to undo the effects of a service invocation when using stateful services.
- **Flexibility/Robustness:** It refers to the capability of the service to behave in an acceptable way in anomalous or unexpected situations or when the context changes. Adaptability, reparability, self-healability, recoverability, predictability and survivability are grouped under this attribute.
- **Reliability:** The ability of a service to perform its required functions under stated conditions for a specified period of time (failure-free operation capability in specified circumstances and for a specified period of time).
- **Scalability:** The capability of increasing the computing capacity of the SP’s computer system and the ability of the system to process more operations or transactions in a given period.
- **Maintainability** is the ability of a system to undergo evolution with the corollary that the system should be designed so that evolution is not likely to introduce new faults into the system [31]. Maintainability has been defined as the process of making engineering changes to the system by involving the system designers and installers. Therefore, it is in contrast to adaptability, which is the process of changing a system to configure it for its environment of use.

4.9 Performance

This quality category contains quality attributes that characterize how well a service performs. The following attributes belong to performance quality category:

- **Transaction Time:** Time elapsed while a service is processing a transaction.
- **Throughput:** It refers to the number of event responses handled during an interval. It can be further distinguished into input-data-throughput (arrival rate of user data in

the input channel), communication throughput (user data output to a channel) and processing throughput (amount of data processed).

- **Response Time:** The time that passes while the service is completing one complete transaction. Latency as sub-attribute of response time is the time passed from the arrival of the service request until the end of its execution/service. Latency itself has been constructed with Execution time and delay time in queue. The former is the time taken by a service to process its sequence of activities. The latter is the time it takes for a service request to actually be executed.

4.10 Usability

Usability/Representation collects all those quality attributes that can be measured subjectively according to user feedback. It refers to the ease with which a user can learn to operate, prepare input for, and interpret the output of the service. The attributes belong to usability quality category are described below:

- **Satisfaction:** Freedom from discomfort and positive attitudes towards the use of the service. Attractiveness as a sub-attribute is the capability of the service to attract the user and their trust (e.g., having contact information and pictures of staff).
- **Learnability:** Capability of the service to enable the user to learn how to apply/use it. Comprehensibility (sub-attribute) is the capability of the service to enable the user to understand whether its functionality is suitable, and how it can be used for particular tasks and under particular conditions of use. Perceivable content (sub-attribute) makes the service useable and understandable to users, unambiguous or difficult.
- **Effectiveness:** Accuracy and completeness with which users achieve specified goals.
- **Efficiency of Use:** Resources expended in relation to the accuracy and completeness with which users achieve their goals.

4.11 Correctness

Correctness deals with the system behaviour conformed to the formal specification (accordance to expected behaviour and the absence of improper system states).

4.12 Complexity

Complexity deals with highly fragmented composite services which in most cases would be considered less trustworthy than a more atomic one.

- **Composability** has been defined as the ability to create systems and applications with predictably satisfactory behaviour from components, subsystems, and other systems.

4.13 Further Trustworthiness Attributes: A Discussion

In the previous sub-sections we have analysed the trustworthiness attributes found in the literature. In this sub-section we complement the presented collection of trustworthiness attributes with the further attributes from realistic use-cases. We discuss two domains, context and application dependence of trustworthiness by looking at the realistic use-cases as following:

- **Ambient Assisted Living:** These systems are in health care domain and application. The set of attributes which have primarily been considered consists of availability, confidentiality, integrity, maintainability, reliability and safety, but also performance and timeliness.
- **Cyber Crisis Management:** These systems deal with critical infrastructures, thus, the major trustworthiness attributes to be considered are integrity, timeliness, correctness, failure tolerance, and availability.

Below are the attributes, which potentially contribute to trustworthiness but have not been addressed in literature:

- **Provability:** The service performs provably as expected, resp. as defined. This is more a property of the engineering process rather than of the service delivered, but should be taken into account as well.
- **Predictability:** In general, the service performs in such a way that the user can predict its behaviour, either according to past experience (=best practices), or just due to logic inference of activities.
- **Flexible Continuity:** In case the service does not perform as expected, or fails, then there is a process to not only fix the issue in adequate time, but also to inform the user, give them the chance to be involved, and to re-use the service as soon as possible. This relates to recoverability and flexibility, but specifically applies to situations with failure potential.
- **Level of Service** is defined as the type of QoS commitment given to the application or user. It is often part of contractual agreements and therefore is often expressed in measurable terms. Although less well treated in literature related to trustworthiness, it constitutes an important trustworthiness component in most business applications. This attribute should be part of the “performance” group of attributes.
- **Accessibility** defines whether the service is capable of serving requests, specifically to clients with limited capabilities. While many services are ready to use, they might not be accessible to specific clients. For instance, the connection between the service and the client is problematic or the service requests the clients to be able to read. This attribute should be part of the “usability” group of attributes.
- **Content Accessibility** is ensuring that the content of the service can be navigated and read by everyone, regardless of location, experience, or the type of computer technology used. It is also part of the “usability” group of attributes.
- **Data Accuracy** is defined as correctness of a data value or set of values as source in view of an expected level of exact computing. It should be part of the “data related qualities” set of attributes.

- **Data Completeness** is defined as the availability of all required data. Completeness can refer to both the temporal and spatial aspect of data quality.
- **Data Consistency** means that when a service fails and then restarts, or is evoked to different points in time, the data returned by the service should be still valid, respectively responding with the same result.
- **Resolution** denotes the granularity of information treated, and although being of good value for decision making, it does not reflect an attribute of the system in general.
- **Operability** is the capability of the service to enable the user to operate on it.

5 Deriving Metrics for Identified Trustworthiness Attributes

Identifying the components of trustworthiness and aiming to build them into software does not necessarily ensure the trustworthiness of the designed system. Therefore, there is a need for trustworthiness evaluation, i.e., measure and make the trustworthiness of the system evident [32]. As mentioned above, trustworthiness can be interpreted differently by users and organizations. Hence, trustworthiness may be evaluated with respect to different targets like: the confidentiality of sensitive information, the integrity of valuable information, the availability of critical data, the response time or the accuracy of outputs. This makes the evaluation of trustworthiness challenging.

We defined trustworthiness attribute as a property of the system that indicates its ability to prevent potential threats from becoming active, i.e., a resilience assurance that it will not produce an unacceptable outcome. Therefore, a trustworthiness attribute is defined as a characteristic of a system that encourages or discourages trust in the system, e.g., availability, reliability. Based on the state-of-the-art analysis, trustworthiness attributes (around 40) are identified that can potentially contribute to the trustworthiness of the system. This collection of attributes can be used as indicators of the overall trustworthiness. Trustworthiness attributes influence different phases of the software life-cycle. For instance, some attributes can be measured only at software execution time, while others belong specifically to the development process. There is a need for the systematic treatment of trustworthiness metrics throughout the entire software life-cycle.

The system design processes will benefit from trustworthiness evaluation. As observed above, by identifying which software characteristics measurably contribute to trustworthiness. Related literature mostly studies trustworthiness from a security perspective while assuming that single properties (certification, certain technologies or methodologies) of services lead to trustworthiness. Compared to this, such a one-dimensional approach is insufficient to capture all the factors that contribute to a system's trustworthiness. Instead, a multitude of attributes need to be taken into account. Multifaceted views and concepts of trustworthiness bring complexity in evaluation of trustworthiness.

Since each stakeholder can have different ideas on trustworthiness, a structured and acknowledged model for describing trustworthiness attributes is needed. This model ideally should include a set of acknowledged methods, mechanisms and means for measuring the previously defined trustworthiness attributes. Hereto, their evaluation can produce tangible results of the trustworthiness characteristics of software.

This section focuses on measures or metrics for identified trustworthiness attributes of the to-be-developed software, service or application. A set of attributes describes generic trustworthiness requirements. Each of these generic trustworthiness requirements need to be detailed during the requirements engineering activities to specify the exact level and expected quality of the attribute. One way of detailing such a requirement engineering step is to define a measure for each trustworthiness attribute for the software in question. The attributes can be further complemented with properties to allow a more fine-grained goal definition. We use the term metric as a standard for measuring attributes. It is a function that takes one or more property values and produces a measure related to an attribute, e.g., reliability of a system, overall trustworthiness of a system, average response time. Trustworthiness attributes presented in Sect. 4 describes generic non-functional requirements for any software to fulfill certain trustworthiness expectations. The development of metrics for different attributes leads to various levels of complexity. In some cases like integrity a common (universal) set of metrics that apply to any software is impossible. Therefore, it is necessary to define a process for the identification of metrics in the requirements engineering phase.

There are a number of different methods available for systematically defining metrics, among others [32]: Goal-Question-Metric (GQM), artificial intelligence assessment techniques, assembling and mapping practical, concrete, “bottom-up” measurement methods, intrinsically measurable components and formal modelling techniques. In this paper, we employ GQM in deriving metrics for trustworthiness attributes, because of its universality and simplicity. Furthermore, this approach is widely adopted in software engineering field.

5.1 The GQM Method

GQM is a generic way of developing metrics. It has been introduced by Basili and Rombach [32]. For each Goal (e.g., the different trustworthiness attributes), a set of questions is identified that helps in identifying what supports achieving the goal and subsequently metrics that measure the gradual fulfillment of the goal (or sub-goals thereof). GQM follows two main processes: (1) definition process as a top down approach (left hand side of Fig. 4) and (2) analysis process as a bottom-up approach (right hand side of Fig. 4). It handles the problem of how to decide, and what to measure to reach your goals by defining measurable goals [33, 34].

An important success factor for the metrics to be helpful in due course of the development process (independently of using agile or more traditional process models) is that the metric must be able to be applied during the development process itself, at different stages of the software development (potentially, using slight variants of the same metric). Yet it seems very unlikely that requirements engineers will develop individual metrics for each trustworthiness attribute. Consequently, it is our aim to simplify this process by preparing a set of questions with corresponding potential metrics that only need to be specified in more detail. We aim at providing plug-ins for existing requirements engineering tools. Table 1 shows the derived metrics for the integrity attribute from the security category as an example.

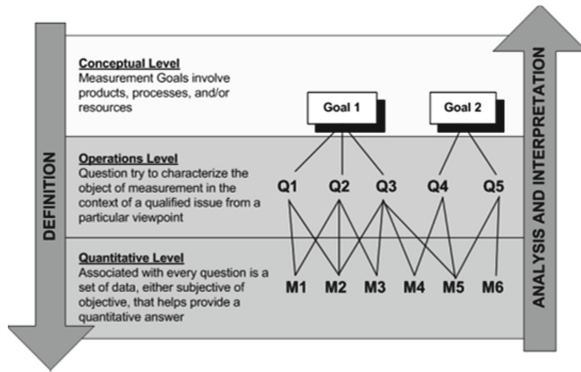


Fig. 4. The GQM Paradigm as a hierarchical structure [35].

Table 1. Example of GQM application for integrity metrics from security category.

Goal	Questions	Metric	Computation	Description
The data that is considered confidential used must be kept confidential at any time and at any place.	What percentage of interfaces needs authentication?	required authentication: % of interfaces which require authentication	$x = (\text{interfaces which require authentication} / \text{all interfaces}) * 100\%$	This metric measure the interfaces which require authentication.
	Does the authorization follow the least privilege principle?	Least privilege's: % of authorization (objects) that are only used for authorizing one function/class etc.	$x = \text{number of authorization (objects) that are only used for authorizing one function/class etc.} / \text{number of authorization (objects)} * 100\%$	This metric aims to reflect how granular the authorization required for executing a function, a class, etc. is (design principle: least privilege).

6 Conclusion and Future Work

STS lie at the intersection of the social aspects of people, society and organizations with the technical aspects and IS used by and underlying such social structures. A premise of the STS theory is that optimization of the socio-elements or the technical-elements of a system independently of each other will increase the unpredictable relationships inside the system, particularly the relationships that may be harmful to the system.

Trust can be viewed as a mechanism to reduce complexity in society and trustworthiness can be viewed as a driver for building trusting relationships. Hence, determining the system attributes that foster trustworthiness contributes to building and optimizing STS such that higher trust can be achieved in such systems.

To identify the attributes that foster trustworthiness we explored an extensive literature survey guided by earlier work in the S-Cube project [23] to identify software attributes related to trustworthiness. While passing through this survey, we also identified some software attributes that either have ambiguous definitions or their

relationships to trust have not been well studied. The paper highlights several interesting issues about the subject of trustworthiness with respect to STS:

- The concept of trustworthiness needs rigorous specification and definition in the context of STS before we are able to build grounded trustworthiness measures.
- To be able to work operationally with trustworthiness attributes, metrics are necessary to set targets, measure progress, and identify the best possible investment by using ROI calculations. While this paper identifies software attributes that foster trustworthiness, it falls short of identifying software trustworthiness metrics that could be universally applied. Such metrics require further analysis and study.
- Trustworthiness in the context of STS includes some subjective component, and always will to some extent. To limit the subjective nature of any trustworthiness metric, a restriction of the context in which the metric is used will be essential.

Our future research will focus on some important questions:

- It is important to understand how the attributes identified in this paper actually influence trust by the users of the system. Empirical research is necessary, and needs to be carried out. Just as for the identification of the attributes, existing literature will only look at individual aspects.
- We need to understand how to identify interdependencies between different attributes, and how consequently to define a “profile” (=set of trustworthiness attributes) for a certain application area.
- Substantial work is needed to investigate existing development methodologies, and to show how they can be enhanced to enable taking trustworthiness attributes into account, in a measurable and comparable way.
- Current certification and attestation programs need to be investigated how they could benefit from taking a wider range of attributes into account than just those related to security, as it is mostly the case today.
- In view of this last research target, metrics shall be developed that express a quantitative view on the assurance of trustworthiness attributes respected/covered by the development practices.

Acknowledgements. The research leading to these results has received funding from the European Union’s 7th Framework Programme FP7/2007-2013 under grant agreement 317631 (OPTET).

References

1. Pazos-Revilla, M., Siraj, A.: Tools and techniques for SSE-CMM implementation. In: 12th World Multi-Conference on Systemics, Cybernetics and Informatics, (2008)
2. Huang, L., Bai, X., Nair, S.: Developing a SSE-CMM-based security risk assessment process for patient-centered healthcare systems. In: 6th International Workshop on Software Quality, pp. 11–16. ACM, New York (2008)
3. Capability Maturity Model[®] Integration, Software Engineering Institute, Carnegie Mellon University Version 1.1

4. Sztopka, P.: *Trust: A Sociological Theory*. Cambridge University Press, Cambridge (1999)
5. Dahrendorf, R.: *Reflections on the Revolution in Europe*. Transaction Publishers, New Brunswick (2005)
6. Golembiewski, R., McConkie, M.: The centrality of interpersonal trust in group processes. In: Cooper, C.L. (ed.) *Theories of Group Processes*, pp. 131–185. Wiley, London (1975)
7. Deutsch, M.: Cooperation and trust: some theoretical notes. In: Jones, M.R. (ed.) *Nebraska Symposium on Motivation*, pp. 275–319. University of Nebraska Press, Lincoln (1962)
8. Shapiro, S.P.: The social control of impersonal trust. *The Am. J. Sociol.* **93**(3), 623–658 (1987)
9. Luhmann, N.: *Trust and Power*. Wiley, Chichester (1979)
10. Mei, H., Huang, G., Xie, T.: Internetwork: a software paradigm for internet computing. *Computer* **45**(6), 26–31 (2012)
11. Sommerville, I.: *Software Engineering*. Pearson, London (2011)
12. OPTET Consortium: Project 317631 OPERational Trustworthiness Enabling Technologies, Annex I – Description of Work, Technical report, (2012)
13. Whitworth, B.: A Brief Introduction to Sociotechnical Systems. In: Khosrow-Pour, M. (ed.) *Encyclopedia of Information Science and Technology*, 2nd edn, pp. 394–400. IGI Global, CITY (2009)
14. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secure Comput.* **1**(1), 11–33 (2004)
15. Li, M., Li, J., Song, H., Wu, D.: Risk management in the trustworthy software process: a novel risk and trustworthiness measurement model framework. In: 5th International Joint Conference on INC, IMS and IDC, pp. 214–219. IEEE Computer Society Press, Los Alamitos (2009)
16. ISO 15408-1, Common Criteria, 2009. Information technology – Security techniques – Evaluation criteria for IT security. Geneva, Switzerland
17. San-Martín, S., Camarero, C.: A cross-national study on online consumer perceptions, trust, and loyalty. *J. Organ. Comput. Electron. Commer.* **22**, 64–86 (2012)
18. Chen, C., Wang, K., Liao, S., Zhang, Q., Dai, Y.: A Novel server-based application execution architecture. In: International Conference on Computational Science and Engineering, 12th IEEE International Conference on Computational Science and Engineering, pp. 678–683, IEEE Computer Society Press, Los Alamitos (2009)
19. Harris, L.C., Goode, M.M.: The four levels of loyalty and the pivotal role of trust: a study of online service dynamics. *J. Retail.* **80**, 139–158 (2004)
20. Gómez, M., Carbó, J., Benac-Earle, C.: An anticipatory trust model for open distributed systems. In: Butz, M.V., Sigaud, O., Pezzulo, G., Baldassarre, G. (eds.) *ABIALS 2006*. LNCS (LNAI), vol. 4520, pp. 307–324. Springer, Heidelberg (2007)
21. Yolum, P., Singh, M.P.: Engineering self-organizing referral networks for trustworthy service selection. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **35**(3), 396–407 (2005)
22. Yan, Z., Goel, G.: An adaptive trust control model for a trustworthy component software platform. In: Xiao, B., Yang, L.T., Ma, J., Muller-Schloer, C., Hua, Yu. (eds.) *ATC 2007*. LNCS, vol. 4610, pp. 226–238. Springer, Heidelberg (2007)
23. S-Cube: Quality Reference Model for SBA. Technical report, S-Cube European Network of Excellence (2008)
24. Boehm, B.W., Brown, J.R., Lipow, M.: Quantitative evaluation of software quality. In: 2nd International Conference on Software Engineering, pp. 592–605. IEEE Computer Society Press, Los Alamitos (1976)
25. Adrion, W.R., Branstad, M.A., Cherniavsky, J.C.: Validation, verification, and testing of computer software. *ACM J. Comput. Surv.* **14**(2), 159–192 (1982)

26. McCall, J.A., Richards, P.K., Walters, G.F.: Factors in Software Quality: US Department of Commerce, National Technical Information Service (1977)
27. ISO/IEC: ISO 9126-1: 2001, Software Engineering – Product quality – Part 1: Quality Model. Standard, International Organization of Standardization (2001)
28. Gol Mohammadi, N., Paulus, S., Bishr, M., Metzger, A., Koennecke, H., Hartenstein, S., Pohl, K.: An analysis of software quality attributes and their contribution to trustworthiness, In: 3rd International Conference on Cloud Computing and Services Science, Special Session on Security Governance and SLAs in Cloud Computing, (2013)
29. McKnight, D.H., Choudhury, V., Kacmar, C.: Developing and validating trust measures for e-Commerce: An integrative typology. *J. Inf. Syst. Res.* **13**(3), 334–359 (2002)
30. Patil, V., Shyamasundar, R.K.: Trust management for e-Transactions. *Sadhana* **30**(2–3), 141–158 (2005)
31. Sommerville, I., Dewsbury, G.: Dependable domestic systems design: a socio-technical approach. *J. Interact. Comput.* **19**(4), 438–456 (2007)
32. Paulus, S., Mohammadi, N.G., Weyer, T.: Trustworthy software development. In: De Decker, B., Dittmann, J., Kraetzer, C., Vielhauer, C. (eds.) CMS 2013. LNCS, vol. 8099, pp. 233–247. Springer, Heidelberg (2013)
33. Basili, V.R., Rombach, H.D.: The TAME Project: Towards improvement oriented software environments. *IEEE Trans. Softw. Eng.* **14**(6), 758–773 (1988)
34. Li, M., Li, J., Song, H., Wu, D.: Risk management in the trustworthy software process: a novel risk and trustworthiness measurement model framework. In: 5th International Joint Conference on INC, IMS and IDC, pp. 214–219. IEEE Computer Society Press, Los Alamitos (2009)
35. Herrmann, D.S.: Complete Guide to Security and Privacy Metrics: Measuring Regulatory Compliance, Operational Resilience, and ROI. Auerbach Publications, Boca Raton (2007)

Cloud Computing and Services Science

Third International Conference, CLOSER 2013, Aachen,

Germany, May 8-10, 2013, Revised Selected Papers

Helfert, M.; Desprez, F.; Ferguson, D.; Leymann, F.

(Eds.)

2014, XI, 129 p. 41 illus., Softcover

ISBN: 978-3-319-11560-3