Metrics for Transparency[*]

Dayana Spagnuelo[**], Cesare Bartolini, and Gabriele Lenzini

Interdisciplinary Centre for Security Reliability and Trust (SnT)
University of Luxembourg, Luxembourg, Luxembourg
{dayana.spagnuelo,cesare.bartolini,gabriele.lenzini}@uni.lu

Abstract. Transparency is a novel non-functional requirement for software systems. It is acclaimed to improve the quality of service since it gives users access to information concerning the system's processes, clarifying who is responsible if something goes wrong. Thus, it is believed to support people's right to a secure and private processing of their personal data. We define eight quality metrics for transparency and we demonstrate the usage and the effectiveness of the metrics by assessing transparency on the Microsoft HealthVault, an on-line platform for users to collect, store, and share medical records.

Keywords: Transparency, Metrics, Non-functional requirements, Requirements Engineering, Quality factors

1 Introduction

Transparency is defined as a quality that enables users to get informed of what will happen or what happened to their data [3,22]. When users of an IT system have an interest in being informed on how the system manages data, and in particular their personal data, transparency ensures an open policy about the system's functioning and processing information. Transparency is a non-functional requirement (NFR) that is believed to increase the quality of a service.

Transparency is also key in achieving privacy and personal data protection. It cannot *per se* guarantee confidentiality, but it can promote to have clear and transparent privacy policies or the availability of mechanisms that users can use to verify whether a system works as intended or as declared or to find who is accountable otherwise. This very perspective of having clear and transparent data protection policies is one of the founding principles of the new European General Data Protection Regulation (GDPR).

Transparency can be suitably expressed as a requirement in Requirements Engineering (RE). RE offers techniques and tools to specify, model, represent, implement, measure, and track functional and non-functional requirements of a

[*] The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-47072-6_1

[**] Supported by FNR/AFR project 7842804 TYPAMED

system. The so-called non-functional requirements (NFRs), rather than describing *what* a system does, specify *how* the system performs in terms of costs, performance, reliability, maintainability, portability, robustness, usability and the like [6]. Transparency falls into this category. As a NFR, it can be modelled and expressed in formal terms, but it has so many facets that modelling transparency requires approaches that differ somewhat from those already available for other NFRs. This is why we believe that resorting to RE practices may help model transparency and, in principle, it would be possible to introduce transparency as a requirement in the Systems Development Life Cycle (SDLC). Extending RE methodologies in order to encompass transparency as a new requirement is an activity that can be performed at several different levels, each requiring different processes [21]. Here, we focus on requirement *modelling* and *validation*.

Modelling transparency requires representing it in some formalism. A preliminary model of transparency has been presented in [22]. Validation means to measure whether and up to which degree a system provides transparency. This is usually the task of software metrics. In the control of software quality, they introduce a more formal and less subjective [15] assessment. We need metrics to describe and measure transparency.

We propose eight metrics that measure the degree of transparency of a system.

Outline §2 qualifies transparency in IT and surveys the related literature. §3 outlines the methodology that we use to classify and define metrics for the main factors of transparency. §4 defines and comments the mathematical functions that we can use to measure each of the selected factors. §5 applies the proposed metrics to an actual system. §6 concludes the paper and suggests future research.

2 Related Work

We have already determined in [22] that transparency means mainly to provide 1. *information* (e.g., data or evidence) on how a user's personal data will be handled or has been handled by the system; and 2. *mechanisms* (e.g., apps, plug-in) to assist the users in retrieving and presenting that information. In [22] we present 41 requirements that define transparency, but without suggesting how one can validate their implementation.

To this aim non-functional metrics [18] can help but not all NFRs can be expressed in terms that allows them to be easily measured [12,24]. The definition of metrics for validating software quality is an activity which has been the subject of attention. In particular, a standard [15] defines a methodology for defining metrics. We follow that methodology here.

What features metrics should have in order to provide useful results is clearly explained in [16], while the correctness of software metrics for specific NFRs can be validated using formal methods [20]: maintainability [7], re-usability [4] or reliability [1], safety and redundancy [11] have all been subject of formal analysis.

Unfortunately these metrics are not defining transparency. Metrics for transparency can be found in software design for control applications. Here an algorithm is considered transparent if “it is easy and clear to see what the controller does in the moment and what it will do in the next steps” [11]. However, these metrics are not applicable in our context, because they are intended to measure inputs, outputs, and the graphical representation of an algorithm.

A possible quest for metrics for transparency may look at the factors that have been proposed to qualify transparency. For instance [23], while studying requirements for trust and other trust-terms, describe transparency as an attribute that requires the *observability* of several types of *data* concerning the users. In our work [22], these trust-terms correspond to the attribute *instrument*. No metrics is discussed, but both the works indicate the terms to be considered when defining a criterion to measure transparency.

Metrics for transparency exist in eGovernment [25], where transparency is discussed as a way of assessing accountability and qualified for efficiency, effectiveness and accessibility of the volume of information that public administrators provide to users. A potential metric for transparency is defined as “the percentage of processes on which there is information available for users”. A metric for efficiency is defined in terms of the time a user spends to use a service and in terms of the time spent by the organisation to produce the service. Effectiveness is regarded as “the closeness to user needs and expectations”, and the authors suggest that it can be measured considering the presence (or absence) of complaints. These are all valid suggestions for metrics, despite none of them is expressed formally.

3 Methodology

We adopt the methodology presented in the IEEE standard 1061 [15]. It consists of five steps that should be followed to define software metrics: 1. establish requirements, 2. identify metrics, 3. implement the metrics, 4. analyse them, and finally 5. validate them. Step 1 has been carried out in [22]. Steps 2 to 5 are considered here.

Transparency is a multi-faceted concept, and assigning direct metrics for it would end up in a very coarse assessment. Instead, following the suggestion of the IEEE standard, we first identify the quality *factors* and quality *sub-factors* that contribute to establishing transparency. Then, building on top of previous NFR literature, we search for suitable qualities that define the factors and the sub-factors identified. Eventually, we propose and assign metrics for those qualities. As an assistance for this task, we first build a questionnaire whose goal is to clarify how to decide when a quality is to be considered satisfied.

The search for quality factors that help define transparency does not present any difficulty. As stated in section 2, implementing transparency means to provide information and mechanisms. These are the *instruments* required to achieve transparency. The search for quality sub-factors required a review of the literature for software qualities and NFRs [5,19,6,25]. Four major sub-factors appear relevant: informativeness, understandability, accessibility, and validity. The first three refine the “providing information” factor, whereas the last two refine the “providing mechanisms” factor. Accessibility related to both factors.

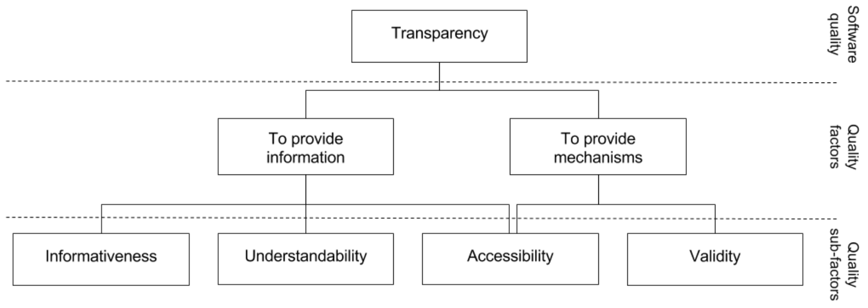


Fig. 1: Transparency and its factors and sub-factors

Informativeness concerns the ability of conveying a good quality of information, and helps understand the excellence of the information provided. *Understandability* represents the ability of “achieving a comprehensible meaning”. It is also connected to the provision of information once it explores the linguistic quality of an instrument. *Accessibility*, here in the sense of “being easy to obtain”, is a quality that refers to both categories of instruments. Since the instrument expresses the act of providing something, it must be easy for a user to obtain it, regardless of whether this something is information or mechanisms. *Validity*, here in the sense of “being precise and producing the correct result”, is linked with the provision of mechanisms, and defines how sound the mechanism is in doing its job. Figure 1 summarises the selected factors and sub-factors.

The questionnaire that we used to find out how to assess whether each quality is satisfied or not is reported in Table 1. We defined the questions on the basis of the definitions and descriptions found while exploring the literature. To maintain a high level of granularity, where necessary, questions are partitioned into sub-questions.

Not all of the questions correspond to some metric. Questions whose answer may vary depending on the user’s perceptions, such as question 3, have been disregarded. Instead, questions that admit objective answers (the grey boxes in Table 1) have been assigned metrics to measure the corresponding factors and sub-factors. The metrics are discussed in details in the next section.

4 Metrics

The eight metrics that we propose are: *accuracy* (questions 1 and 1.1); *currentness* (question 2 and 2.1); *conciseness* (question 5); *detailing* (question 6);

Table 1: Qualities questionnaire. The questions in grey cells have led to metrics

Factor	**Question**	**Sub-question**
Informativeness	1. Is the system providing accurate information?	1.1. Is the system providing consistent and flawless information?
	2. Is the system providing up-to-date information?	2.1. Is the system providing timely information?
	3. Is the information consistent to what the user experiences?	
	4. Is the system providing unbiased information?	
Understandability	5. Is the system providing the minimum possible information for the understanding of the matter?	
	6. Is the system providing enough details on the information for the understanding of the matter?	
	7. Is the system helping the user to understand the information provided?	
	8. Is the system providing clear and neat information?	8.1. Is the system providing information using the terminology appropriate to the area?
		8.2. Is the system providing information that does not use jargon?
Accessibility	9. Is the system making the instrument available?	9.1. Is the system providing an instrument that can be used whenever needed?
	10. Is the system providing portable information?	10.1. Is the system providing information that can be used in different environments?
		10.2. Is the system providing information that can be extracted in different formats?
		10.3. Is the system providing information that can be accessed through different means?
Validity	11. Is the system providing correct and precise mechanism?	11.1. Is the system providing ways to verify a mechanism?
		11.2. Is the system providing a mechanism that reaches the goal for which it has been provided?
		11.3. Is the system providing the source code of the mechanism?

Table 2: Metrics associated to quality sub-factors

Sub-factor	**Metric Name**
Informativeness	Accuracy
	Currentness
Understandability	Conciseness
	Detailing
	Readability
Accessibility	Availability
	Portability
Validity	Effectiveness

readability (question 7); *availability* (questions 9 and 9.1); *portability* (questions 10 and 10.1–10.3); and *effectiveness* (questions 11 and 11.2).

4.1 The eight metrics

Table 2 shows the metrics associated with the transparency sub-factors. All the metrics are normalised, ranging from 0 (worst value) to 1 (best value).

Accuracy This metric measures how much the information provided matches the real process of the system. The metric demands statements extracted from the information to be observed in the real system. To measure accuracy, we must first define what is considered a statement. Statements are going to depend on the nature of the information, but we suggest that at least claims and affirmations about what the system is or does should be considered. A representation of the system's process (for example, a model such as a business diagram) might also be considered, as it may help in the assessment of accuracy.

Each statement should be linked (mapped) to some part of the process. If it is not possible to link the statement, either because it is not present, or because it is dubious, then the information should not be considered accurate. The result is the proportion of accurate statements. If LS is the number of statements that can be linked to some parts of the process, and NLS is the number of statements which do not correspond to a specific part of the process, then accuracy $\mathcal{A}c$ can be expressed as shown in Equation (1).

$$\mathcal{A}c = \frac{LS}{LS + NLS} \tag{1}$$

Currentness Currentness depends on the time that passes between something happening in the system and the system providing information about it. More specifically, if Δt is the interval of time that the system has taken to inform about the change, and Δt_u is a time unit that measures the reasonable interval time (i.e., the ideal time) for updating that piece of information, the currentness is measured as shown in Equation (2).

$$\mathcal{C}u = \begin{cases} 1, & \text{if } \Delta t \leq \Delta t_u \\ 2^{-\left\lfloor \frac{\Delta t - \Delta t_u}{\Delta t_u} \right\rfloor}, & \text{if } \Delta t > \Delta t_u. \end{cases} \tag{2}$$

In other words, anything that takes less time than what would be deemed ideally reasonable for that information has $\mathcal{C}u = 1$.

It should be noted that while some pieces of information should be updated in a matter of minutes or hours (e.g., information on security breaches), for others a longer time would be acceptable (e.g., results of a research with patients). The time unit Δt_u is highly dependent on the nature of the system and of the type of information that must be updated, and must be carefully chosen for each case. A poorly chosen unit will result in inaccurate currentness values.

The floor function in the exponential simplifies the metric by providing discrete values (e.g., anything in the time range $\Delta t_u \leq \Delta t < 2\Delta t_u$ has the same currentness value). Let us consider an example in which an information is extremely relevant, for example because it concerns a security breach, and the time unit Δt_u is defined as one minute. If the system takes one hour ($\Delta t_u = 60$) for updating the information, then the currentness is $\mathcal{C}u = 2^{-60} \simeq 0$. On the other hand, if the acceptable range is 30 minutes, then this duration can be used as the time unit, and the currentness is $\mathcal{C}u = 2^{-\left\lfloor \frac{60-30}{30} \right\rfloor} = 2^{-1} = 0.5$.

Conciseness The conciseness metric measures how straightforward an information is. We measure the conciseness of an information in terms of the average number of words per sentence. The scales of this metric are based on recommendations for the English language. While [8] suggests that the average length of sentences should be between 15 and 20, it is stated in [13] that an average of 5 to 8 words per sentence can be read by people with moderate learning disabilities, and that by using common words it is possible to help all users to understand a sentence with around 25 words. For this reason, we use a Gaussian curve $N(\mu, \sigma^2)$, with a mean $\mu = 20$ and a standard deviation $\sigma = 5$, as expressed in Equation (3). However, we normalise this function so that its maximum value is one. The resulting formula for measuring the conciseness is shown in Equation (4). Here ASL denotes the average number of words per sentence, and it is calculated as N_W/N_S, where N_W is the total number of words, and N_S is the total number of sentences.

$$N(\mu, \sigma^2) = \frac{e^{-\frac{1}{2\sigma^2}(x-\mu)^2}}{\sigma\sqrt{2\pi}} \tag{3}$$

$$\mathcal{C}o = \sigma\sqrt{2\pi}N(\mu, \sigma^2) = e^{-\frac{1}{50}(ASL-20)^2} \tag{4}$$

We understand that conciseness is not only about short sentences, and that semantics analysis should be considered too. What is presented here, however, is an easy-to-calculate approximation for syntactic straightforwardness.

Detailing This metric describes a strategy for measuring whether an information provided is detailed enough for the general understanding of its subject. Detailing is measured by checking if the main crucial details are present in the instrument "information" that the system provides. The crucial details will vary from instrument to instrument, but we suggest that, at least, basic questions should be answered, such as: what? who? why? when? to whom? which? and so on. The information provided has to be cross-checked with the questions, and the result is a matrix of details provided versus important details. The metric $\mathcal{D}$ is the proportion of important details provided.

The detailing matrix should be constructed in such a way that only the questions pertinent to a given piece of information are counted towards the proportion. For example, assuming the system must inform the users on how their data are stored and who has access to them, questions like "why [is the data accessed]?" and "when [was the data stored]?" are not pertinent.

If n_I is the number of pieces of information provided, and m_Q is the total number of detailing questions, the detailing matrix has a size of $n_I \times m_Q$. For each piece of information $i = 1 \dots n_I$, there will be a number P_i^D of questions pertinent to the detailing metric, and a number $NP_i^D = m_Q - P_i^D$ of non-pertinent questions. The non-pertinent questions are not relevant and therefore do not count towards the metrics. On the other hand, the pertinent questions can be partitioned into a number d_i of questions for which the details are provided, and a number u_i of questions for which details are not provided, such that $d_i + u_i = P_i^D$. Under these premises, the detailing metrics $\mathcal{D}$ can be expressed as shown in Equation (5).

$$\mathcal{D} = \frac{\sum_{i=1}^{n_I} d_i}{\sum_{i=1}^{n_I} P_i^D} = 1 - \frac{\sum_{i=1}^{n_I} u_i}{\sum_{i=1}^{n_I} P_i^D} \qquad (5)$$

A highly-detailed system ($\mathcal{D} = 1$) will possibly answer all pertinent questions for each piece of information.

Readability This metric measures how easy it is for a user to read and understand a specific text. There are several well-established formulas available for this purpose. Each formula has its advantages and there are no general recommendations or standards stating which one should be used in each case. To select the formula, we searched the literature to understand how to measure readability in the medical domain (the domain used for our requirements). The most used formulas are the Flesch-Kincaid grade level (FKGL), the Simple Measure Of Gobbledygook (SMOG), and the Flesch Reading Ease (FRES) [26,17,9,14]. FKGL and FRES are variants of the same method, and both use the average sentence length and the average word length as an input. SMOG is calculated using the number of long words (three syllables or more). We chose to use FRES for being the only one that provides the results in easiness grades.

As already introduced in *conciseness* metric, the average sentence length is measured as $ASL = N_W / N_S$, where N_W is the total number of words and N_S is the total number of sentences. Similarly, the average number of syllables per word is $ASW = N_{SY} / N_W$, where N_{SY} is the total number of syllables. The $FRES$ can be expressed as shown in Equation (6). In theory, the higher boundary of the $FRES$ is 121.22, which is achieved by applying it to a sentence with one word of one syllable, like "yes" or "no". There is no theoretical lower boundary, but by applying the formula to long sentences with long words it is possible to reach huge negative scores. However, such extremes are non-realistic in the documentation of a system. The common interpretation of *FRES* considers scores from 0 to 100 only [10]. As a measure of the readability metric $\mathcal{R}$, we consider the bounded and normalised *FRES*, as shown in Equation (7).

$$FRES = 206.835 - (1.015 \times ASL) - (84.6 \times ASW) \tag{6}$$

$$\mathcal{R} = \begin{cases} 0, & \text{if } FRES < 0 \\ \frac{FRES}{100}, & \text{if } 0 \leq FRES \leq 100 \\ 1, & \text{if } FRES > 100 \end{cases} \tag{7}$$

Availability This metric measures how easy it is for a user to access the instrument, if accessible at all. To measure the availability $\mathcal{A}v$, we first define N_{int} as the number of interactions the user needs to perform to reach the desired instrument. An interaction is considered as any action the user must perform, such as typing, clicks, taps, slides, etc. Availability applies to any sort of information or mechanisms the system provides, and we define its metric as follows:

$$\mathcal{A}v = \begin{cases} 1 - \frac{(1-\omega)}{k} N_{\text{int}}, & \text{if } 0 \leq N_{\text{int}} \leq k \\ \omega e^{(1 - \frac{N_{\text{int}}}{k})}, & \text{if } N_{\text{int}} > k \end{cases} \tag{8}$$

Here k is the maximum number of interactions that are considered acceptable for reaching access, while $\omega \in [0, 1]$ is the grade that we give when accessing the instrument takes exactly k steps. Equation (8) degrades linearly from the maximum value 1, obtained when no steps are required to get access to the instrument, till value ω, obtained when k steps are required to get access to the instrument. From that point on, the degradation is exponential in the number of steps.

Portability This metric measures how easy it is for an information to be transferred and used in different systems. To measure portability, we reused the popular classification provided by the 5 star open data [2], which is a scheme for rating the degree of structuredness of data on the web. It is a model that uses an incremental scale from 1 to 5. To measure how portable an information is, we need to verify whether the properties described in each scale are implemented. We adapted the scale and normalised it to our context as shown in Equation (9).

$$\mathcal{P} = \begin{cases} 0, & \text{if no information available} \\ 0.2, & \text{if available in any open format} \\ 0.4, & \text{if available as a structured data} \\ 0.6, & \text{if available in a non-proprietary format} \\ 0.8, & \text{if uses URI} \\ 1, & \text{if based on linked data} \end{cases} \tag{9}$$

Effectiveness This metric measures how satisfactory the mechanism provided is. The strategy is very similar to the one presented in Equation (5). Effectiveness is measured by checking whether the goal of the mechanisms is being reached. The goal varies according to the requirements, but we suggest that the output of the mechanism addresses at least basic questions, such as: what? who? why? when?

If n_I is the number of pieces of information provided as the output, and m_Q is the total number of questions, the effectiveness matrix has a size of $n_I \times m_Q$. For each piece of information $i = 1 \dots n_I$, there will be a number P_i^E of questions pertinent to the effectiveness metric, and a number $NP_i^E = m_Q - P_i^E$ of non-pertinent questions. The pertinent questions can be partitioned into a number e_i of questions whose goal is reached, and a number v_i of questions whose goal is not reached, such that $e_i + v_i = P_i^E$. Under these premises, the efficiency metrics $\mathcal{E}$ can be expressed as shown in Equation (10).

$$\mathcal{E} = \frac{\sum_{i=1}^{n_I} e_i}{\sum_{i=1}^{n_I} P_i^E} = 1 - \frac{\sum_{i=1}^{n_I} v_i}{\sum_{i=1}^{n_I} P_i^E} \tag{10}$$

4.2 Synthesis

Although normalised and aligned on the same ranges, the metrics proposed are heterogeneous and cannot easily be combined into a mathematical expression that can clearly measure transparency as a whole. Instead, we adopt a benchmarking strategy, where each of the proposed metrics serves to assess the performance of one or more of the factors that determine the transparency quality

The benchmark can be represented as a radar chart (an example is shown in Figure 2). The blue area represents the best possible measurement for the factor “providing information”, while the orange area shows the best outcomes for the factor “providing mechanisms”. The metric “availability” appears twice in the chart because it is applicable to both factors.

The metrics we present are potentially applicable to any transparency requirement. In the context of our previous work [22], for example, our eight metrics apply to each of the 41 requirements according to what instrument the requirement is about. That means that for a complete assessment of transparency we may need to apply these metrics to each requirement. The interpretation of these results, regardless of the context in which transparency is desired, should provide insights on the factors and requirements that have room for improvement, and guide the way to a better transparency.

5 Use case

Microsoft HealthVault[1] is an integrated online platform that allows users to gather, store and share health information. The information in HealthVault can

[1] `https://www.healthvault.com/lu/en`.

be: provided by the user, in which case the system acts as the means for the user to fill in his/her personal and medical data, or to upload files with any kind of medical record in it; provided by compatible health applications, since the system can use information provided by external mobile or web applications whenever authorised by the user; or provided by compatible health devices, as the system can also use information collected by specific compatible health devices, such blood pressure monitors, weight scales, and others. To evaluate the applicability of our metrics in Microsoft HealthVault we choose two transparency requirements: 1. "S must provide P with disclosure of policies, regulations or terms concerning data sharing, processing and the use of data"; and 2. "S must provide P with accountability mechanisms" [22]; where S stands for "the system" and P for "the patient", or (in our example) "the user".

To implement the first functionality, HealthVault provides a dedicated section called "Microsoft Privacy Statement" concerning the personal and medical data. To test for *accuracy* we selected only the section of the privacy statement that directly addresses the peculiarities of HealthVault that are not common to other Microsoft products. That section contains information on signing in, on the account and records, on sharing health data, on reporting to health care providers in the US, on access and control, and on email communications. We chose the main statement for each of those topics: 1. "You can use more than one credential with HealthVault to help ensure continued access"; 2. "You can add or remove data to a health record you manage at any time"; 3. "As a custodian, you can share data in a health record with another person by sending an email invitation through HealthVault. You can specify what type of access they have (including custodian access), how long they have access, and whether they can modify the data in the record"; 4. "In the United States, we enable participating providers to obtain reports about whether the information they send to a record is used"; 5. "You can review, edit or delete your HealthVault account data, or close your HealthVault account at any time"; and 6. "You can unsubscribe from these emails [communications] at any time".

Statements 2, 3, 5 and 6 could be easily verified, as for each of those the system contains areas available to the users. There is also a specific area for managing the credentials, but the only option offered is to use the Microsoft credentials (at least in the version of the system available in Europe), which invalidates statement 1. Statement 4 could not be verified as it is only valid in the United States, and so it is not considered in the calculation. As a result we have $\mathcal{A}c = 0.8$.

Microsoft HealthVault provides no information on how long they take to update the privacy statement once something has changed in the policy. Although they inform when was the last time the statements changed and what exactly has changed, we do not have enough information to calculate *currentness* metric. Thus currentness metric is not measurable without access to the internal system.

The privacy statements from HealthVault score very high in *conciseness*. In average, sentences are 17.71 words long, slightly less than the mean considered in the metric. This value, applied to Equation (4), provides a conciseness value

Table 3: Detailing matrix: desirable details compared with the delivered details. Greyed-out cells represent the non-pertinent questions.

	Delivered Details					
Desired Details	DWC	UPD	SPD	CST	IPI	MHS
Is data shared? With whom? For what purpose?			✓			
Is data processed? For what purpose?	✓	✓				
How is data used? For what purpose?				✓	✓	✓

of $\mathcal{C}o \simeq 0.90$. Although HealthVault has a good score for conciseness, the *FRES* formula only results in 36.02 when applied to the privacy statements. This value indicates that the text is reasonably difficult to understand; applied to Equation (7), it provides a readability $\mathcal{R} \simeq 0.36$.

The *detailing* metric can be calculated considering the purpose for which the information has been made available. In this case, the users must be informed of the policies and regulations for data sharing, processing and usage of the data. So the privacy statement should ideally help the users understand: whether the data is shared, with whom, and for what purpose; whether the data is processed and for what purpose; how is the data used and for what purpose. Relevantly for this requirement, the privacy statement provides information separated into the following categories: Personal Data We Collect (DWC); How We Use Personal Data (UPD); Reasons We Share Personal Data (SPD); Cookies and Similar Technologies (CST); Other Important Privacy Information (IPI); and Microsoft Health Services (MHS). We use a three letters identifier to simplify Table 3. The detailing metric reaches the maximum score $\mathcal{D} = 1$, as all the desired details are provided by the privacy statement.

To measure *availability*, we first need to define the maximum number of acceptable interactions k, and the grade ω we attribute for k. For this example, we chose $k = 3$ and set its grade $\omega = 0.7$. To access these data, users simply need to access the "Privacy & Cookies" section available through the main page of the system. As the user needs only one interaction to reach the desired content, the availability metric reaches the score: $\mathcal{A}v = 0.9$.

Regarding the *portability* of the privacy statement section, HealthVault scores the value $\mathcal{P} = 0.8$. Applying Equation (9), we have the following: the information is provided in HTML, an open format; since it is presented as HTML, it is also structured, and available in a non-proprietary format; the information is available on the web and can be accessed through a Uniform Resource Locator (URL), which is a subset of a Uniform Resource Identifier (URI). Although the statement contains several links to other data that provide a better understanding, these do not provide access to external data sources and cannot be considered linked data.

The second requirement "S must provide P with accountability mechanisms" is implemented by Microsoft HealthVault by providing a way for users to consult the history of accesses and changes made on their data up to one year ago. They can see the changes made by one specific person or application, or even see

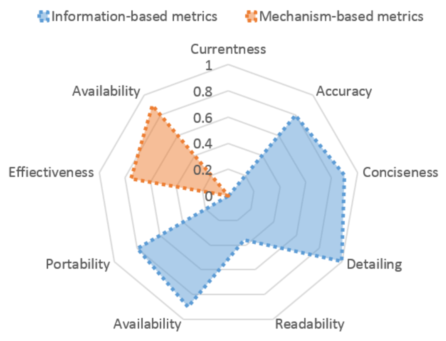
Table 4: Effectiveness matrix: desired goals compared with the real outputs. Grey cells represent the non pertinent questions.

	Delivered Outputs					
Desired Goals	Date	Action	Type	Changed by	App	Summary
What action?		✓	✓		✓	✓
Who did it?				✓	✓	
When did it happen?	✓					
For what purpose?						

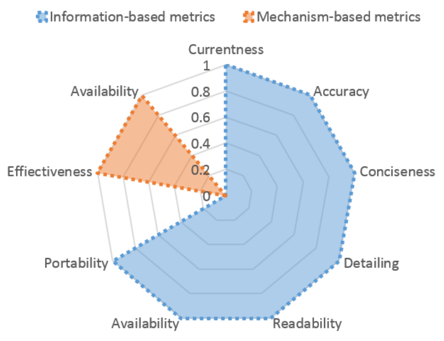
the history of granted access rights. These functions are centralised in a section called “Record History” that can be accessed with one click from the main page, provided the user is already logged in the system. Considering the same parameters $k = 3$ and $\omega = 0.7$, Microsoft HealthVault reaches again the score $\mathcal{A}v = 0.9$ in the *availability* metric with regard to this requirement.

Finally, in our example, we claim that HealthVault provides accountability mechanisms by making a “Record History” available to their users. For a mechanism to be effective in helping a user hold a person accountable for an action, it requires some means to check what actions happened in the system with regard to the user’s data; who did the action; when the action happened; and the purpose of the action. As seen in Table 4, HealthVault reaches three out of the four desired goals in accountability tools. Thus, the *effectiveness* metric scores $\mathcal{E} = 0.75$.

A summary of the results is presented in Figure 2. The results for the first requirement (information-based) are shown in blue together with those for the second one (mechanism-based), which are in orange. The assessment of Microsoft HealthVault is presented in Figure 2a, whereas Figure 2b displays what the ideal scenario would be. Currentness is the only metric that is not applicable, and therefore it is presented with no value in the chart.



(a) Microsoft HealthVault.



(b) The ideal scenario.

Fig. 2: Synthesis of the transparency measurement.

6 Discussion and Conclusion

Non-functional requirements are a useful instrument to compare systems that offer similar functionalities. They help assess which systems perform better, or which ones more faithfully embed specific user requirements. For this reason, modern SDLC methodologies tend to integrate NFRs in the system design.

Transparency is a new NFR that is recently becoming crucial as a promoter of the quality of a service and as a guarantee of respect of users' rights. Since we live in a digitalised world where mobile devices are ubiquitous and cloud computing is in our public and private daily activities, end users have the right to know whether the personal information they entrust to their devices and online services are managed securely and privately. Providing such information to end users is of paramount importance: what a device, an application, a service actually do, what they access, and for what purpose. Transparency comes into play by enabling users to endow devices and services (and their manufacturers and providers) with a motivated trust. Besides, as it can be used to express commitment to users and clarify accountability, transparency may also become a significant competition factor.

Designing for transparency, however, can be problematic. On the one side, the relevant information should be provided without exposing the system's security to wanton risk. On the other side, users might lack the technical skills to understand the content of the information, or to isolate meaningful material from an informative flooding. Thus, the information should be carefully selected and presented in a concise and intelligible form. Alternatively, users can be assisted by tools that convert a completely inscrutable streams of bytes into a human-friendly fashion.

Transparency is not a monolithic concept. It is rather a complex quality partitioned into several requirements. However, there are a few factors that all those requirements have in common. They all have to provide information (e.g., about a policy, a process) or the tools to get that information. These factors offer different perspectives under which transparency can be viewed.

In this work, we prove that *transparency* of a system is not just a high-level concept but a quality that can be measured. We introduced a few metrics to separately assess some of the most significant factors of transparency. This provides a meaningful way of benchmarking transparency and comparing systems. Our set of metrics is not complete, and each metric may not be the most accurate possible. But we demonstrate that the metrics are applicable to obtain a reasonable estimation of a system's transparency with respect to a specific desired requirement.

Further research directions are possible. An interesting work for the future is to apply the proposed metrics to systems in different domains, and analyse the differences in the results. In this way, it would be possible to classify the various sub-factors of transparency according to their importance in specific domains. Another planned research direction is to evaluate the transparency metrics to a new use case, but having access to the internal documentation and SDLC (i.e., with the assistance of the provider). Such an analysis could unveil some details (which could be measured on their own) about the asymmetry of information between the provider and the user. The problem of asymmetry of information is well-known but, to the best of our knowledge, has never been explored from an analytic perspective. Finally, another possible evolution would be to adjust the model presented in our research to allow its integration into a SDLC, for example by modifying the software development workflow to address the transparency requirement. In order to extend a software design methodology (and tools) in such a way, it is necessary to analyse the interaction and possible collisions between transparency and other NFRs.

References

1. Bauer, E., Adams, R.: Reliability and Availability of Cloud Computing. Wiley-IEEE Press, 1^{st} edn. (2012)
2. Berners-Lee, T.: Linked data, `https://www.w3.org/DesignIssues/LinkedData.html`, last accessed in May 2016
3. Berthold, S., Fischer-Hübner, S., Martucci, L., Pulls, T.: Crime and punishment in the cloud - accountability, transparency, and privacy. Pre-Proc. of Int. Workshop on Trustworthiness, Accountability and Forensics in the Cloud in conjunction with the 7th IFIP WG 11.11 Int. Conf. on Trust Management (2013)
4. Caldiera, G., Basili, V.R.: Identifying and qualifying reusable software components. Computer 24(2), 61–70 (1991)
5. Cappelli, C.: Uma abordagem para transparência em processos organizacionais utilizando aspectos. Ph.D. thesis, PUC-Rio (2009)
6. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering, Int. Series in Software Engineering, vol. 5. Springer US (2000)
7. Coleman, D., Ash, D., Lowther, B., Oman, P.: Using metrics to evaluate software system maintainability. Computer 27(8), 44–49 (1994)
8. Cutts, M.: Oxford Guide to Plain English. Oxford University Press (2007)
9. Eloy, J.A., Li, S., Kasabwala, K., Agarwal, N., Hansberry, D.R., Baredes, S., Setzen, M.: Readability assessment of patient education materials on major otolaryngology association websites. Otolaryngology–Head and Neck Surgery 147(5), 848–854 (2012)
10. Flesch, R.F.: How to Write Plain English. Barnes & Noble (1981)
11. Frey, G., Litz, L.: A measure for transparency in net based control algorithms. In: IEEE Int. Conf. on Systems, Man, and Cybernetics. vol. 3, pp. 887–892 vol.3 (1999)
12. Glinz, M.: On non-functional requirements. In: Proc. of the 15^{th} Int. Requirements Engineering Conf. (RE). pp. 21–26. IEEE (2007)
13. GOV.UK: UK Government Digital Service Style Guide, `https://www.gov.uk/guidance/content-design/writing-for-gov-uk#short-sentences`, last accessed in May 2016
14. Greywoode, J., Bluman, E., Spiegel, J., Boon, M.: Readability analysis of patient information on the american academy of otolaryngology–head and neck surgery website. Otolaryngology-Head and Neck Surgery 141(5), 555–558 (2009)
15. IEEE Computer Society: IEEE standard for a software quality metrics methodology. IEEE Standard 1061-1998, IEEE Computer Society (1998)

16. Kaner, C., Bond, W.P.: Software engineering metrics: What do they measure and how do we know? In: Proc. of the 10th Int. Symposium on Software Metrics. IEEE (2004), `http://kaner.com/pdfs/metrics2004.pdf`
17. Kasabwala, K., Agarwal, N., Hansberry, D.R., Baredes, S., Eloy, J.A.: Readability assessment of patient education materials from the american academy of otolaryngologyhead and neck surgery foundation. Otolaryngology–Head and Neck Surgery 147(3), 466–471 (2012)
18. Keller, S., Kahn, L., Panara, R.: Specifying software quality requirements with metrics. IEEE Computer Society Press (1990)
19. Leite, J.C.S.d.P., Cappelli, C.: Software Transparency. Business and Information Systems Engineering 2, 127–139 (2010)
20. Schneidewind, N.F.: Methodology for validating software metrics. IEEE Transactions on Software 18(5), 410–422 (1992)
21. Sommerville, I.: Software Engineering. Addison-Wesley Longman, Inc., 10th edn. (2016)
22. Spagnuelo, D., Lenzini, G.: Patient-centred transparency requirements for medical data sharing systems. In: New Advances in Information Systems and Technologies, pp. 1073–1083. Springer (2016)
23. Sullivan, K., Clarke, J., Mulcahy, B.P.: Trust-terms ontology for defining security requirements and metrics. In: Proc. of the 4th Eur. Conf. on Software Architecture: Companion Volume. pp. 175–180. ECSA '10, ACM, New York, NY, USA (2010)
24. Svensson, R.B., Gorschek, T., Regnell, B.: Quality Requirements in Practice: An Interview Study in Requirements Engineering for Embedded Systems, LNCS, vol. 5512, pp. 218–232. Springer-Verlag (2009)
25. Viscusi, G., Batini, C., Mecella, M.: Information systems for eGovernment: A quality-of-service perspective, chap. 7, pp. 127–144. Springer-Verlag (2010)
26. Zarcadoolas, C.: The simplicity complex: exploring simplified health messages in a complex world. Health promotion international 26(3), 338–350 (2011)