

Quality of individual requirements artefacts

Several sources (see e.g. [IEEE Std 830-1998; Robertson and Robertson 2006]) suggest similar quality criteria for requirements artefacts. These criteria apply to the documentation of requirements in natural language as well as to the model-based documentation of requirements. To ensure that all defined quality criteria are satisfied, typically several elicitation, documentation, validation, and management activities are performed, each of them focussing on one or few of the defined criteria.

In the following, we sketch out important, common quality criteria for requirements artefacts suggested in the literature. A requirements artefact should be:

- No missing information* ☐ **Complete:** A requirements artefact is complete if it adheres to the rules and guidelines defined for this type of requirements artefact and does not omit any piece of information that is relevant for some stakeholder (user, client, architect, tester, etc.). For example, a template for requirements artefacts of a specific type (see Section 18.4) can facilitate the validation of whether a requirements artefact of this type is complete (in the sense that no attribute is missing). For this purpose, the template defines the attributes to be documented for each requirements artefact of the specific type. In general, we differentiate between two kinds of completeness: the completeness of a single requirements artefact (“Has the requirement been captured completely?”) as well as the completeness of a requirement document (“Does the requirements document include all necessary requirements?”). The completeness of requirements documents is discussed in Section 17.3.
- Source, evolution, impact, and use* ☐ **Traceable:** A requirements artefact is traceable if its source, its evolution as well as its impact and use in later development phases are traceable. Requirements traceability is discussed in detail in Chapter 31.
- Confirmed by stakeholders* ☐ **Correct³:** A requirements artefact is correct, if the relevant stakeholders confirm its correctness and demand that the system must realise the documented requirement completely. Hence, a documented requirement is incorrect, for instance, if it unnecessarily adds system functionality or quality properties (so-called gold plating).
- Single valid interpretation* ☐ **Unambiguous:** A requirements artefact is unambiguous, if its documentation permits only one valid interpretation. In contrast, an ambiguous requirement allows for different interpretations. The consequence of an ambiguous requirement may be that the requirement is not realised in the system as intended. The problem of ambiguity in natural language specification is discussed in detail in Section 17.4.2.

E

Example 16-1: Ambiguous requirement

R14-1: In order to authenticate himself, the driver enters an electronic card and a PIN. If invalid, the engine does not start.

Ambiguity: Does the “if invalid” refer to the card, the PIN, or both?

Easy to understand for the relevant stakeholders

- ☐ **Comprehensible:** A requirement is comprehensible if its content is easy to comprehend. The comprehensibility of a requirements artefact depends, among other

³ The term “correctness” for requirements artefacts is somewhat problematic, since correctness in software development normally denotes the compliance of a program with its specification. However, in requirements engineering there is generally no documented reference against which the correctness of a requirement can be checked. Nevertheless, using validation techniques one can check whether the stakeholders agree with and demand the realisation of the requirements artefact.

things, on the documentation format chosen and the stakeholder(s) involved. Comprehensibility and unambiguousness are independent quality criteria. A requirement can be comprehensible but at the same time ambiguous (e.g. interpreted differently by different stakeholders). Vice versa, a requirement can be unambiguous but still incomprehensible. For instance, a requirements artefact documented in a formal language might not be understandable to many stakeholders, who can thus not detect a potential ambiguity.

- **Consistent:** A requirements artefact is consistent, if the statements within the artefact do not contradict each other. Besides the consistency of each single requirements artefact, especially the consistency of the entire set of requirements artefacts has to be taken into account during requirements documentation (see Section 17.3).
- **Verifiable:** A requirements artefact is verifiable, if the stakeholders can check whether the implemented system fulfils the documented requirement or not. For example, if a requirement is underspecified, it is not possible to decide objectively whether the requirement is realised as defined in the requirements artefact or not (see Example 16-2). To facilitate verifiability, typically acceptance criteria are defined (see Section 16.4).

No contradictions

Clear and objective acceptance criteria

Example 16-2: Verifiability of underspecified requirements

Verifiable: R1: The system must respond to event ES-2 in at least 80% of the cases within 2 s, and in all cases within 3 s at the latest for a system load between 80% and 90% of the maximum load as specified by constraint C14 (System load profile).

Unverifiable: R2: The normal response time of the system shall be less than 2 s.

R2 cannot be verified since it is unclear what the expression "normal response time" refers to, e.g. what type of response, which system load, and under what conditions the response time may be higher than 2 s.

- **Rated:** A requirements artefact is rated, if its relevance and/or its stability have been determined and documented.
- **Up to date:** A requirements artefact is up to date, if it reflects the current status of the system and the system context, such as the current stakeholder wishes or current legal regulations.
- **Atomic:** A requirements artefact is atomic, if it describes a single, coherent fact. A requirement is not atomic, if it describes multiple isolated or merely loosely coupled facts which can easily be divided into several requirements artefacts (see Example 16-3).

Known relevance and stability

Reflects current status of the system and its context

Single coherent fact

Example 16-3: Non-atomic requirement

R23: The user must log on to the system in order to be able to search for a particular order by order ID or by full-text search in the customer database.

Non-atomicity: The requirement R23 is not atomic because it describes user authentication as well as different ways of searching for orders.

E