

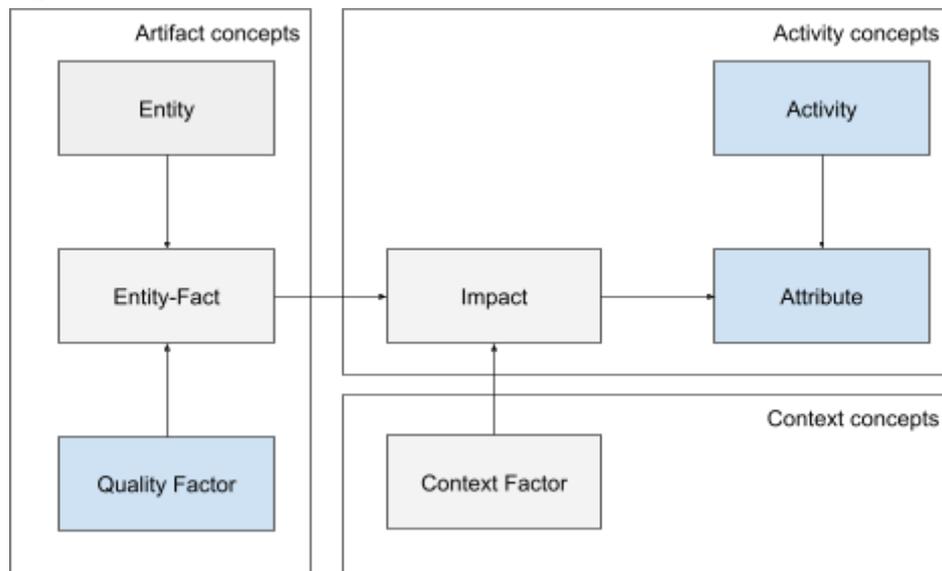
Coding Guideline

Reported relevant Factors of Requirements Quality

This document guides the coding process for the requirements quality issues. These issues report the consequences of problems introduced during the requirements engineering phase.

Concepts

The following concepts are relevant for the extraction.



Only the quality factor, activity, and attribute concept are relevant for coding. The entity concept is excluded since it contains sensitive information. The entity-fact, impact, and context factors are excluded since the issues did not contain sufficiently granular information about these concepts.

Concept	Description
Quality Factor	Property of the requirements specification
Activity	Impacted activity
Attribute	Affected property of the activity

Guideline

The following sections guide the coding process of the issue extractions.

Entity Codes

The entity codes describe a requirements entity (i.e., the requirements specification). A *Quality Factor Mention* can be coded by up to two *quality factor* codes.

Quality Factor

Definition: A *quality factor* is a normative metric that maps a textual requirement of a specific granularity to a scale that informs about the quality of this input [2]. It describes a property of the entity and should be decidable based on the entity alone - though it does not have to be decidable automatically (e.g., sentence length can be automatically decided, conciseness cannot be automatically decided).

Coding Rule: Assign 1-2 codes to the *Quality Factor Mention* that best represents the property of the requirement described by the interview participant.

Quality Factor	Definition
Completeness	The issue occurred because a requirement or details of a requirement were not present when committing to the requirement specification
Consistency	The issue occurred because a requirement contradicted or did not fully align with another requirement or other artifact (e.g., a design rule).
Ambiguity	The issue occurred because a requirement allowed multiple interpretations
Relevancy	The issue occurred because a requirements specification contained information that was not necessary and potentially even imposed on the solution-space
Correctness	The issue occurred because a requirement contained incorrect information, i.e., information that did not align with the actual intent of the customer
Feasibility	The issue occurred because a requirement was not possible to be implemented from the beginning.
(not a req problem)	The issue occurred because of a different issue (e.g., implementation shortcut, missing documentation). The requirements are not at fault.

Activity Codes

The activity codes describe the activity that is impacted by the requirements entity. An *Activity Mention* can be coded by up to two pairs of *activity* and *attribute* codes. The activity can be the same in both activity code sets.

Activity

Definition: A requirements-affected *activity* is impacted by an entity fact and context factors.

Coding Rule: The available activities are organized in a tree, where a parent activity represents a more abstract version (i.e., a “superclass”) of the child activity. The notation within the tree is “**Activity** (applicable attributes): description.”

- **Processing** (duration, feasibility, completeness): an abstract activity that considers a requirements specification as input and produces some output.
 - **Understanding** (uniqueness): comprehending a sentence on a general, lexical level
 - **Interpreting** (uniqueness): comprehending a requirements specification on a semantic level and relating it to its domain
 - **Translating** (stability, coherence): transforming a requirements specification into a different artifact
 - **Implementing**: developing code that exhibits the features described in the requirements specification
 - **Verifying**: deriving test cases that assess whether a piece of code exhibits the features described in the requirements specification
 - **Assessing** (precision): evaluating a requirement
 - **Assessing feasibility**: determining whether a requirement is realistic to be implemented
 - **Estimating effort**: predicting the effort to implement a requirement
 - **Planning** (stability): determining the life-cycle of a requirement
 - **Coordinating**: orchestrating the subsequent work involving a requirement with the owner of related requirements
 - **Reusing**: using an existing artifact (not the requirements specification) for a new activity, e.g., reusing existing code to fulfill a new requirement

Attribute

Definition: An *attribute* is the (measurable) property of the impacted activity.

Coding Rule: Once an activity has been determined, select the appropriate attribute that describes the impacted property of the activity. All attributes of a parent activity also apply to all available child nodes.

Attribute	Definition
Unspecific	No specific attribute of an activity is mentioned, just general "ease" or "success"
Uniqueness	Whether the output of an activity is always the same or can differ depending on other factors.
Duration	The amount of time that the completion of the activity takes
Completeness	The degree to which the output of the activity covers the implied content of the input (e.g., whether the derived test cases cover all functionality implied by a requirement)
Precision	Accuracy of a prediction
Stability	How stable the results of an activity are (e.g., how reliable the subsequent plan of a requirement is)
Feasibility	Whether it is realistic that an activity can be completed at all
Coherence	Whether the output of an activity remains coherent with the existing artifacts
Traceability	Whether the output of an activity can be traced back to the causing requirement, e.g., to understand a decision

The code "unspecific" is the default, fallback code. If the *Activity Mention* describes an attribute more specifically, then use this more specific attribute. Otherwise, fall back to "unspecific."

References

[1] Frattini, J., Montgomery, L., Fischbach, J., Mendez, D., Fucci, D., & Unterkalmsteiner, M. (2023). Requirements quality research: a harmonized theory, evaluation, and roadmap. *Requirements Engineering*, 1-14.

[2] Frattini, J., Montgomery, L., Fischbach, J., Unterkalmsteiner, M., Mendez, D., & Fucci, D. (2022, August). A live extensible ontology of quality factors for textual requirements. In *2022 IEEE 30th International Requirements Engineering Conference (RE)* (pp. 274-280). IEEE.