

Requirements Types

Requirements Types



There is only ONE requirement type

REQ

A Requirement specifies a System Element's desired characteristic, quality, function aspect or behavior in defined circumstances under specific conditions

REQ

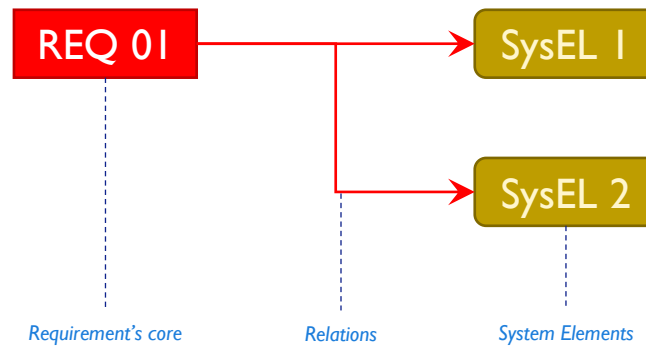
SysEL



“Ideal” situation



- Ideally, a Requirement specifies the desired characteristic, quality, aspect or behavior in an unambiguous, clear to understand manner, including all relevant information like circumstances, preconditions, constraints and more.
- This “ideal” situation is neither practical nor desirable in cases where the same requirement applies to several System Elements. We could for instance not include the names of the System Elements inside the Requirements body text (which one of the System Elements to choose?).



Direct Requirement's Context



- As said, “ideally” a Requirement would specify the desired aspect or behavior in an unambiguous and clear to understand manner, including all relevant information.

In practice most of this contextual information is implicit, often in the same document or data base that the requirement is located. The relation of the relevant information to the Requirement is often ambiguous, incomplete and implicit. In these cases, it becomes difficult to verify and validate these Requirements, based on their Specification Core.

- There is a clear need to make the direct Requirement's Context explicit and relate it directly to the Requirement.



Requirements Containers

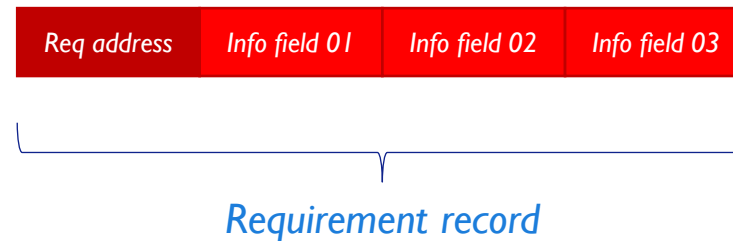


- Many Requirements nowadays are still documented by text-based tooling like MS Word or Excel. In this presentation we say that the Requirements are captured in a document.
- Without going into the details of revising Requirements and documents, let's say that the Requirements are managed in Baselines. A Requirement's revision can be captured in one or more Baselines.
- Baselines are linked to revisions of System Elements. In that way the Baselines become the Requirements Specifications of the System Elements.

Requirements' Informatics



- Let's consider the core information of a requirement as a standard data base record, a unique address label and several fields to store information (core data and meta data).



An example

Postal code	Current residents	Town name	Construction year
-------------	-------------------	-----------	-------------------

A requirement instance

BN14 OTA	Smith family	Findon	1969
----------	--------------	--------	------

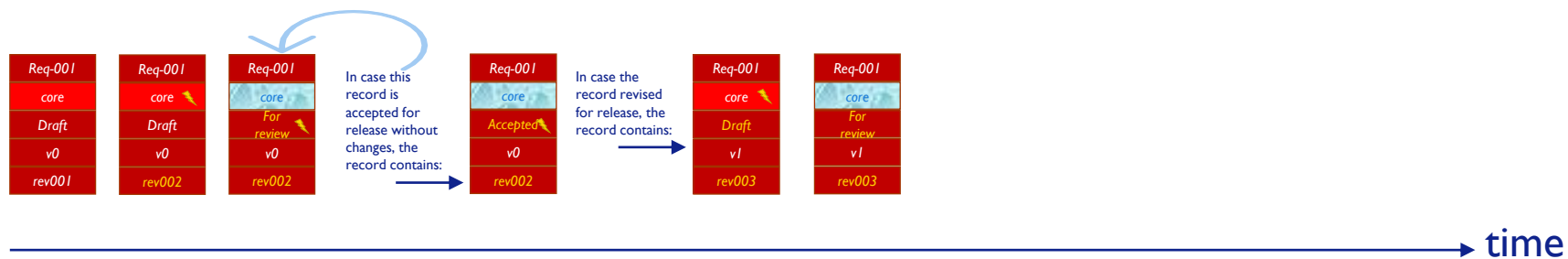
Requirement's versions and revisions



- Requirements (the record) can have versions and revisions. Versions are coupled to the status of a requirement (a meta data field) and revisions are coupled to changes to the requirement.

Req ID/address	Info field 01	Info field 02	Status field	Version field	Revision field
----------------	---------------	---------------	--------------	---------------	----------------

- When a requirement is created it automatically has version 0 and revision 1. Changes to the requirement core information are tracked through the revision number. When the status of a requirement changes from “draft” to “for review”, the core information is ‘frozen’.



Requirement Category 0



- The requirement is part of a text. Not distinguishable as a requirement by someone who is unfamiliar with the context.
- It is unclear where the requirement starts or ends.
- The requirement has no clear, unique identification.

Requirements Categories



	0	1	2	3	4
ID	No unique identification	No unique identification	Unique identification	Unique identification	Unique identification
Identified as requirement	Not clear if the text is intended as requirement	Text is marked as a requirement	Text is marked as a requirement	Text is marked as a requirement	Text is marked as a requirement
Context	Not clear if surrounding text is relevant	Not clear if surrounding text is relevant	Not clear if surrounding text is relevant	Not clear if surrounding text is relevant	Context for the requirement is identified, and linked to
Quantification	No quantification applied	No quantification applied	No quantification applied	Requirement is quantified	Requirement is quantified