

CLEAR REQUIREMENT



A good requirement is a clear requirement, and no one will spontaneously question such a key feature. But things come to be fuzzier when one tries to pragmatically master what “clarity” operationally entails when it comes to drafting requirements.

When looking at IncoSE requirement guide, (INCOSE-TP-2010-006-01), one immediately first remarks that only textual requirements are addressed. We remain mired in the contextual and qualitative domain of the natural language. Whatever the definition, it is therefore incompatible by nature with the scientific ideal of a standalone formalization, pointing to some reality domain, understandable and as precise as relevant through its grounding in the universally shared reference of human scientific knowledge.

This admitted, it is then no surprise that “clarity” itself be not even a general admitted feature of what a “good (textual) requirement” is. If clarity is a key notion for IT Professionals such as the [Irish Computer Society¹](#) (ICS), or a reference book such as [Mastering the Requirements Process^{\(2\)}](#), it is not even part of IncoSE selected characteristics to specify what a “good” requirement is.

When going deeper into details of mentioned references (interesting exercise, because they are obviously quite different), one may nevertheless strive to find allusive resemblances in the notions manipulated. Adopting IncoSE characteristics as reference, we may for example think of the hereafter far-fetched correspondences, in connection with the “clarity” notion:

¹ See <https://www.ics.ie/news/view/1552>

² Mastering the Requirements Process: Getting Requirements Right, 3rd Edition Aug 6 2012, Suzanne Robertson, James Robertson, Addison-Wesley Professional

INCOSE characteristics intuitively related to “clarity” notion		“Clarity” according to	
Characteristic	Meaning	ICS	Mastering Req. Process
C3 - Unambiguous	<i>Lends to a single interpretation</i>	<ul style="list-style-type: none"> • Avoid combining requirement • Provide context • Use present active 	<ul style="list-style-type: none"> • Concise (<i>minimalist</i>) • Precise • Simple (<i>finality driven, without any interference with other considerations</i>) • Terse
C4 - Complete	<i>Standalone from design to test</i>	Provide context	Precise
C5 - Singular	<i>Addresses a single thought</i>	Avoid combining requirement	Simple
C8 - Correct	<i>Correct expression of the stakeholder expectation</i>	Provide context	<ul style="list-style-type: none"> • Concise • Precise
C9 - Conforming	Conform to organization standard	Consistent terminology	

For sure, anyone else scrutinizing mentioned references to make up his mind as to what “Clarity” entails, would come up with an analysis of his own... and any attempt to set up general, accepted and applicable rules would get bogged down in endless discussions in rough verbiage.

Let’s just remark that the sparse operational guidelines, such as “use active voice”, only refer to lexical rules while anything trying to point to some reality outside common language float in the clouds.

This is precisely this kind of pitfalls that has moved us up to develop a System Engineering method, scientifically grounded and physically meaningful. It is not here the place to unfold the whole stuff, rooted in the most fundamental science, developing the unconventional breakthrough of [Pr. M. Mugür-Schächter](#)³, De Broglie’s last student, but let’s stress some features that straightforwardly point to the “clarity” notion:

- The “requirement” concept is given the more general status of a factual **anticipation** -whose mathematically basis is defined in [Relativized Systemic](#)⁴ and that is developed in [Relativized System Engineering](#)⁵ - that is endowed with of scientific character (refutability, repeatability, ...). Such a requirement formally points to expected correlations between observables, while the possibility of “explaining” these correlations through scientific rules roots this expectation in the corpus of human scientific knowledge. Thus is a requirement a **standalone concept**, as **precise** as relevant, with regards to the way factuality is captured and described.
- The **minimalist** and **sufficient** character are formalized through the concept of **description cell** that poses that nothing relatively exists outside the potentialities associated with the context, processes, and qualification grid, defined in the associated **epistemic referential**. This clearly delineates the context out of pre-defined concept and specify what is supposed, in that context, interfere or not with the expected results.

³ See https://fr.wikipedia.org/wiki/Mioara_Mugur-Sch%C3%A4chter

⁴ See <https://www.amazon.fr/Syst%C3%A9mique-Relativis%C3%A9e-Essences-conceptualisations-R%C3%A9el/dp/6138478835>

⁵ See https://mersyse.com/en/isr_features.php



- Each description cell comes with a **single finality**, that makes it possible to ensure, by construction, the requirement **correctness** relative to the stakeholder's expectation and avoid the uncontrolled combination of multiple stakes. This feature is made possible through the concept of **relative existence** of any entity, an existence relative to the way an entity is **generated or selected** in a given state and to the way it is then **examined and qualified**. The introduction of **conceptualization levels** makes it possible to bridge the gap between factuality, such as it is described and the expression of the finality. It makes it also possible to endow facts with distinct meanings, depending on the standpoint, thus stressing the independence, the correlation or the incompatibility of different stakes.
- Let's just still mention that requirement **conformance** is ensured by construction, since any requirement is set-up according to a mathematized framework that integrates it seamlessly in the wider scope of Product design construction.

For sure, all that is just a flavor, and must be considered within the comprehensive methodological framework. We would have so much to say, for example about the formal definition of what a system is, or how **Relativized System Engineering** guarantees the autonomy and verifiable character of any standpoint all a Project development long, or still how we master complex system development through a twofold approach based on system engineering, finality driven, and modular architecture, resource optimization driven, ...

That's somewhat frustrating... But we are resilient and used to mastering frustration, when we consider the powerless verbiage, the widespread illusions, compromises, and interests that comes with the still undominated System Engineering discipline: looks rather like far west!

Now is holyday time, and we do want to take this opportunity of having good time in this unpredictable and troubled period! Enjoy!