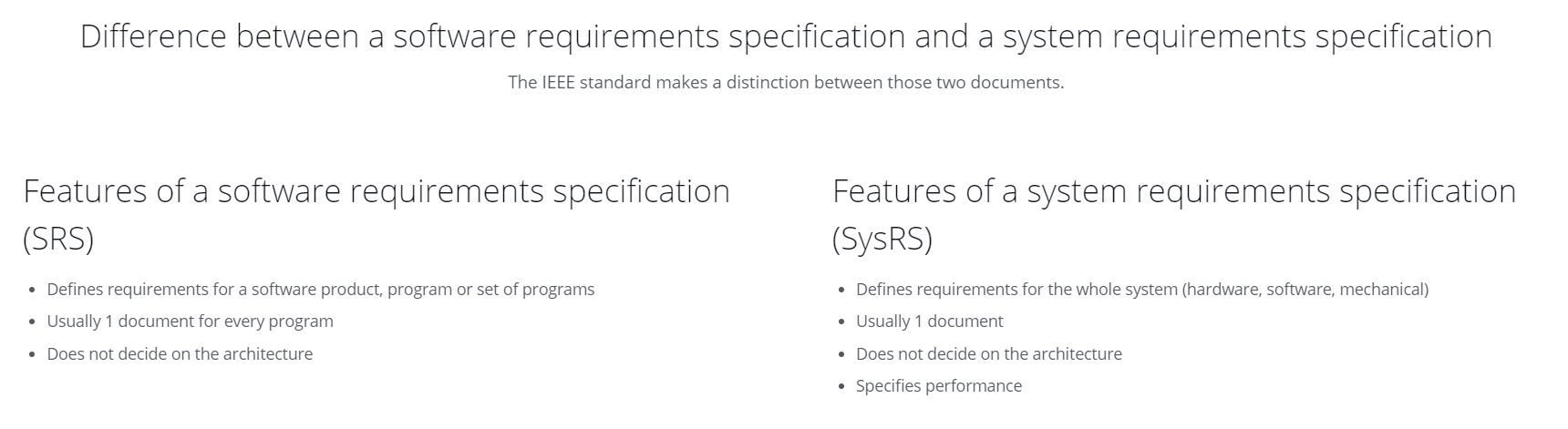
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ISO/IEC/IEEE 29148:2018(E) | IIBA – BABOK  (Business Analysis Body Of Knowledge) | PMI – PMBOK  (Project Management Body of Knowledge) | SEBOK  (Systems Engineering Body of Knowledge) | SWEBOK  (Software Engineering Body of Knowledge) |
| Functional / Performance requirements | Business requirements | Business requirements | Functional Requirements | System Requirements |
| Interface requirements | Stakeholder requirements | Stakeholder requirements | Performance Requirements |  |
| Process requirements | Solution requirements   * Functional * Nonfunctional | Solution requirements   * Functional * Nonfunctional | Usability Requirements | Software Requirements   * Product and Process Requirements * Functional and Nonfunctional Requirements |
| Quality (Non-Functional) requirements | Transition requirements | Transition requirements | Interface Requirements |  |
| Usability / Quality-in-Use requirements |  | Project requirements | Operational Requirements |  |
| Human Factors requirements |  | Quality requirements | Modes and/or States Requirements |  |
|  |  |  | Adaptability Requirements |  |
|  |  |  | Physical Constraints |  |
|  |  |  | Design Constraints |  |
|  |  |  | Environmental Conditions |  |
|  |  |  | Logistical Requirements |  |
|  |  |  | Policies and Regulations |  |
|  |  |  | Cost and Schedule Constraints |  |

BABOK: <https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/1538/Describe-the-basic-classification-of-requirements-as-defined-by-the-BABOK-v20.aspx>



According to Ian Sommerville (1997) the software requirements types are:

* User requirements: are written for customer in natural language and simplified diagrams which describe system features and services.
* System requirements: are written as a contract between client and the contractor, and it is a structured document setting out detailed descriptions of the system services.
* Software specification: A detailed software description which can serve as a basis for a design or implementation and are written for developers.

According to Karl E. Wiegers (2003) the software requirements levels or types are:

* Business requirements: are written to represent the high level objectives and goal of the organization or stakeholders who want to build the system.
* User requirements: describe the user goal which the user desires from the new system.
* Functional requirements: specify the software functionality that the developers must build into the product to enable users to accomplish their tasks.

According to Suzanne Robertson and James Robertson (2006) the software requirements level or types are:

* High-level requirements: are written to specify the business the product is intended to support
* Business requirements: are written in a technologically neutral manner they specify what the product is to do for the work, not which technology is used to do it.
* Product's technological requirements: are written by designer who adds those requirements needed to make the product work in its technological environment.

According to the Rational Unified Process the software requirements level or types are:

* Stakeholder requests: are written in User language to capture all user requests.
* Features: are written to capture all feature of the system under development.
* Use cases: are written to capture functional requirements.
* Supplementary requirements: are written to capture Non-functional requirements (Leffingwell 1999).

**References:**

*Business Analysis Body of Knowledge* (version 2.0). 2008. International Institute of Business Analysis. pp 8-9.

Leffingwell, Dean & Don Widrig. 1999. *Managing Software Requirements*(First Edition). Addison Wesley.

Robertson, James and Suzanne Robertson. 2006. *Mastering the Requirements Process* (Second Edition). Addison-Wesley Professional.

Sawyer, Pete and Ian Sommerville. 1997.*Requirements Engineering: A Good Practice Guide* (First Edition). Wiley.

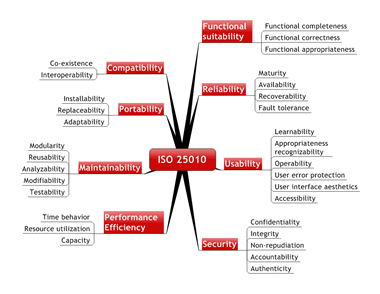
Wiegers, Karl. 2003. *Software Requirements* (Second Edition). Microsoft Press

<https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/343/If-you-were-to-group-software-requirements-into-levels-how-might-you-group-them.aspx>

SEBOK in detail (<https://www.sebokwiki.org/wiki/System_Requirements>):

|  |  |
| --- | --- |
| **Functional Requirements** | Describe qualitatively the system functions or tasks to be performed in operation. |
| **Performance Requirements** | Define quantitatively the extent, or how well, and under what conditions a function or task is to be performed (e.g. rates, velocities). These are quantitative requirements of system performance and are verifiable individually. Note that there may be more than one performance requirement associated with a single function, functional requirement, or task. |
| **Usability Requirements** | Define the quality of system use (e.g. measurable effectiveness, efficiency, and satisfaction criteria). |
| **Interface Requirements** | Define how the system is required to interact or to exchange material, energy, or information with external systems (external interface), or how system elements within the system, including human elements, interact with each other (internal interface). Interface requirements include physical connections (physical interfaces) with external systems or internal system elements supporting interactions or exchanges. |
| **Operational Requirements** | Define the operational conditions or properties that are required for the system to operate or exist. This type of requirement includes: human factors, ergonomics, availability, maintainability, reliability, and security. |
| **Modes and/or States Requirements** | Define the various operational modes of the system in use and events conducting to transitions of modes. |
| **Adaptability Requirements** | Define potential extension, growth, or scalability during the life of the system. |
| **Physical Constraints** | Define constraints on weight, volume, and dimension applicable to the system elements that compose the system. |
| **Design Constraints** | Define the limits on the options that are available to a designer of a solution by imposing immovable boundaries and limits (e.g., the system shall incorporate a legacy or provided system element, or certain data shall be maintained in an online repository). |
| **Environmental Conditions** | Define the environmental conditions to be encountered by the system in its different operational modes. This should address the natural environment (e.g. wind, rain, temperature, fauna, salt, dust, radiation, etc.), induced and/or self-induced environmental effects (e.g. motion, shock, noise, electromagnetism, thermal, etc.), and threats to societal environment (e.g. legal, political, economic, social, business, etc.). |
| **Logistical Requirements** | Define the logistical conditions needed by the continuous utilization of the system. These requirements include sustainment (provision of facilities, level support, support personnel, spare parts, training, technical documentation, etc.), packaging, handling, shipping, transportation. |
| **Policies and Regulations** | Define relevant and applicable organizational policies or regulatory requirements that could affect the operation or performance of the system (e.g. labor policies, reports to regulatory agony, health or safety criteria, etc.). |
| **Cost and Schedule Constraints** | Define, for example, the cost of a single exemplar of the system, the expected delivery date of the first exemplar, etc. |

Softwarequalität ISO 25010 als Nachfolger von ISO 9126



SWEBOK:

A product requirement is a need or constraint on the software to be developed (for example, “The software shall verify that a student meets all prerequisites before he or she registers for a course”).

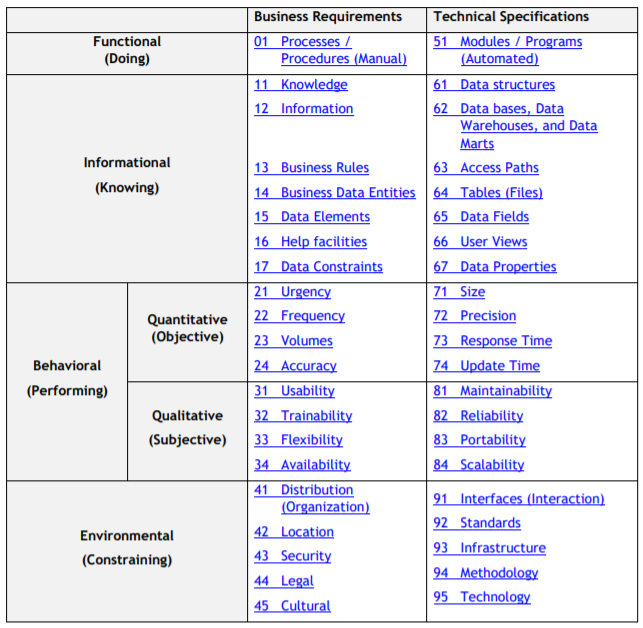
A process requirement is essentially a constraint on the development of the software (for example, “The software shall be developed using a RUP process”).

Some software requirements generate implicit process requirements. The choice of verification technique is one example. Another might be the use of particularly rigorous analysis techniques (such as formal specification methods) to reduce faults that can lead to inadequate reliability. Process requirements may also be imposed directly by the development organization, their customer, or a third party such as a safety regulator.

<http://swebokwiki.org/Chapter_1:_Software_Requirements#Product_and_Process_Requirements>

<https://businessanalysisexperts.com/Job_Aids_RSG/RequirementsTaxonomy.pdf>

A Requirements Taxonomy



<https://www.peterjohann-consulting.de/requirements-engineering/>

IREB – International Requirements Engineering Board

<https://www.ireb.org/content/downloads/11-cpre-advanced-level-requirements-management-handbook/ireb-cpre-handbook-for-requirements-management-advanced-level-en-v1.1.pdf>

