

Since the beginning of human evolution, humans have been building technical and organizational systems to support them in completing tasks or achieving objectives. With the rise of engineering, humans have also started to build systems that automate human tasks. Whenever humans decide to build a system to support or automate human tasks, they have to figure out what to build. This means that they have to learn about the desires and needs of the persons or organizations who will use the system, benefit from it, or be impacted by it. In other words, they need to know about the requirements for that system. Requirements form the basis for any development or evolution of systems or parts thereof. Requirements always exist, even when they are not explicitly captured and documented. The term requirement denotes three concepts [Glin2020]:

DEFINITION 1.1. REQUIREMENT:

1. A need perceived by a stakeholder.
2. A capability or property that a system shall have.
3. A documented representation of a need, capability, or property.

A systematically represented collection of requirements—typically for a system—that satisfies given criteria is called a requirements specification. We distinguish between three kinds of requirements: Functional requirements concern a result or behavior that shall be provided by a function of a system. This includes requirements for data or the interaction of a system with its environment. Quality requirements pertain to quality concerns that are not covered by functional requirements—for example, performance, availability, security, or reliability. Constraints are requirements that limit the solution space beyond what is necessary to meet the given functional requirements and quality requirements. Note that dealing with requirements for projects or development processes is outside the scope of this handbook. Distinguishing between functional requirements, quality requirements, and constraints is not always straightforward. One proven way to differentiate between them is to ask for the concern that a requirement addresses: if the concern is about required results, behavior, or interactions, we have a functional requirement. If it is a quality concern that is not covered by the functional requirements, we have a quality requirement. If the concern is about restricting the solution space but is neither a functional nor a quality requirement, we have a constraint. The popular rule “What the system shall do ☞ functional requirement vs. how the system shall do it ☞ quality requirement” frequently leads to misclassifications, particularly when requirements are specified in great detail or when quality requirements are very important.