



Glossary

Application: The use of capabilities (services and facilities) provided by an information system to satisfy a set of user requirements, such as word processing.

Architecture: The underlying structure of a system.

Architecture baseline: The underlying structure of a system associated with a particular product or release.

Architecture framework: A description of a family of related architectures that allows an individual architecture to be created by selection from and modification of the components. An architectural framework describes an information system in terms of a model comprised of a set of conceptual building blocks. It shows how the building blocks fit together.

Artifact: A document representing the result of effort. Artifacts are often referred to as examples of work products needed to provide evidence in support of assessments.

A spec: Common name for the system specification or segment specification as defined by MIL-STD-961. See Martin, *Systems Engineering Guidebook*.

Attribute: A characteristic of a requirement that is useful in sorting, classifying, and/or managing requirements.

Bad fixes: Secondary defects injected as a by-product of defect repairs.

Baseline: A specification or product that has been formally reviewed and agreed on and thereafter serves as the basis for further development. It is changed only through formal change control procedures.

- Bottom up:** Putting lowest level system components together, one level at a time. Between each level's integration, the results are tested to make it work.
- Brown Bag:** An informal lunchtime meeting.
- B spec:** Common name for the development specification as defined by MIL-STD-961. This specification has several subtypes. B1 is for a prime item. B2 is for a critical item. B3 is for a noncomplex item. B4 is for a facility or ship modification. B5 is equivalent to a software requirements specification and its associated interface requirements specification. A functional specification is a special version of the B spec that lists the requirements for a particular functional definition. See Martin, *Systems Engineering Guidebook*.
- Business requirements:** The essential activities of an enterprise. Business requirements are derived from business goals (the objectives of the enterprise). Business scenarios may be used as a technique for understanding business requirements. A key factor in the success of a system is the extent to which the system supports the business requirements and facilitates an organization in achieving them.
- Business scenario:** A technique that can be used to understand an enterprise or organization. A business scenario describes the business process, application, or set of applications; the business and technology environment; the people who and the computing components that execute the scenario; and the desired outcome of proper execution.
- Business system:** Hardware, software, policy statements, procedures, and people that together implement a business function.
- Buyer:** An organization that procures a system from a supplier.
- Client:** An application component that requests services from a server.
- Commercial-off-the-shelf (COTS):** An item of hardware or software that has been produced by a contractor and is available for general purchase.
- Communications network:** A set of products, concepts, and services that enable the connection of computer systems for the purpose of transmitting data and other forms (for example, voice and video) between the systems.
- Complexity:** The degree to which a system has requirements, a design, and/or an implementation that is difficult.
- Concurrent engineering:** Physically collocated cross-functional teams of engineers from many disciplines working together. Some equate this simply to systems engineering. It seeks to address negative aspects of specialization.
- Configuration management:** A discipline that applies technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of a configuration item, (2) control changes to those

characteristics, and (3) record and report changes to processing and implementation status.

C spec: Common name for the product specification as defined by MIL-STD-961. This specification has several subtypes. C1 is for a prime item. C2 is for a critical item. C3 is for a noncomplex item fabrication. C4 is for an inventory item. C5 is equivalent to a software product specification. See Martin, *Systems Engineering Guidebook*.

Customer: The person with the funds to pay for the project or its end product. The customer is not necessarily the user.

Customer need: The set of requirements desired by a customer.

Database: A structured or organized collection of information that may be accessed by a computer.

Data dictionary: A repository of information describing the characteristics of data used to design, monitor, document, protect, and control data in information systems and databases.

Data warehouse: A collection of integrated subject-oriented databases designed to support the decision support system function.

Decomposition: Breaking apart the attributes of a customer need (the requirements of a system) so that they can be addressed.

Default: Command that is automatically executed if none is specifically indicated.

Defect: A variance from a desired product attribute.

Defect Prevention: Technologies and techniques (for example, statistical process control) that minimize the risk of making errors in deliverables.

Defect removal: Activities that find and correct defects in deliverables.

Defect removal efficiency: The ratio of development defects to customer defects.

Derived requirement: A requirement that is further refined from a primary source requirement or from a higher level derived requirement or a requirement that results from choosing a specific implementation or system element.

Design: The process of defining the architecture, components, interfaces, and other characteristics of a system.

Development: The process of transforming a design into hardware and software components.

Distributed database: A collection of information that is dispersed over a network of interconnected computers.

Domain expert: An individual who has been working in a particular field for an extensive period of time and who is trained in that area. A domain expert is often referred to as a subject matter expert.

Enterprise: The highest level in an organization.

Event: A change in a system's environment that creates a response/set of actions.

Feasibility study: An analysis that provides an initial understanding of the cost, viability, high-level technical architecture, and requirements of a capability or system

Framework: A basic structure of ideas or a frame of reference.

Function: A useful capability provided by one or more components of a system.

Functional architecture: The framework for developing applications and defining their interrelationships in support of an organization's information structure. It identifies the major functions or processes an organization performs and their operational interrelationships.

Functional document: A comprehensive collection of the characteristics of a system and the capabilities it will make available to the users. It provides a detailed analysis of the data the system will be expected to manipulate. It may include a detailed definition of the user interfaces of the system.

Function point: A measure of the complexity of software development.

Gold plating: Adding features and capabilities to systems when not required by the system specification or the real requirements.

Graphical user interface: A computer program designed to allow a user to interact easily with a computer, typically by using a mouse or a pointing device to make choices from menus or groups of icons.

Hardware: Physical equipment.

Heuristic: Involving or serving as an aid to learning, discovery, or problem solving by experimental and especially trial-and-error methods (assumptions).

Hot swap: To replace a board, component, servlet, or server application with a new one while the system is running, without shutting down the system.

Information system: The computer-based portion of a business system.

Information technology (IT): Applied science utilizing hardware and software to support transfer of ideas.

Institutionalization: The building of an infrastructure and corporate culture that support methods, practices, and procedures so that they are the ongoing way of doing business, even after those who originally defined them are gone.

Integrated product team (IPT): A group that includes customers and developers and that blends perspectives into a functioning or unified whole. The joint team recommended in this book is an example of an IPT.

Integration and test: The activity in which modules of a system are combined according to the technical specification, and the interfaces between the modules

Glossary

are examined critically to ensure that expected results are achieved. The results of testing provide the basis for acceptance or rejection of the system.

Interface: The interaction or communication between independent systems or components of systems.

Interoperability: The ability of two or more systems or components to exchange and use information.

Iterate: To repeat a sequence of operations to yield results that are successively closer to a desired result.

Life cycle: The period of time that begins when a system is conceived and ends when the system is no longer available.

Life cycle model: A framework of processes and activities concerned with evolving a system that also acts as a common reference for communication and understanding among the participants in the effort.

Major defect: A problem that precludes effective use of a work product, such as a design deficiency or discovery of conflicting requirements.

Manifest: A term used in Canada instead of *project charter* or *project vision document*.

Measures of effectiveness (MOEs): High-level indicators of how well the system performs its functions, defined in the terms and with the same dimensionality of the requirements document. For example, if we are dealing with a city's metro system, we may specify that a typical user during rush hour should not wait more than some period of time, on the average, for the next train.

Mechanism: A way to get something done or to achieve a result.

Method: A way, technique, process, plan, mechanism, body of skills or techniques, discipline, practice, system, model, framework, capability, or procedure for doing something.

Methodology: A body of methods, rules, and postulates employed by a discipline; a particular procedure or set of procedures.

Middle out: Working upward to integrate major elements (such as government-furnished equipment) and downward to decompose into component subsystems simultaneously.

Minor defect: A problem that doesn't preclude effective use of a work product, such as a formatting issue, spelling error, language usage problem, or acronym or definition not provided or explained.

Objectory: The object factory for software development; an object-oriented method developed by Ivar Jacobson at Objective Systems in Sweden. See Jacobson, *Object-Oriented Software Engineering*.

- Open architecture:** Construction of the underlying structure of a system in a way that allows additional capabilities to be added with little or no adjustment.
- Open system:** People, machines, and methods organized to accomplish a set of specific functions that implement sufficient specifications for interfaces, services, and supporting formats to enable properly engineered applications software to (1) be ported with minimal changes across a wide range of systems, (2) interoperate with other applications on local and remote systems, and (3) interact with users in a style that facilitates user portability. Open systems are vendor independent.
- Portability:** The ease with which a system or component can be transferred from one hardware or software environment to another.
- Practice:** The performance of work activities repeatedly so as to become proficient; the usual way of doing something to produce a good result.
- Prioritized requirements:** Categorization of the real requirements into subsets according to criticality of need for a system or capability.
- Process:** A set of activities that results in the accomplishment of a task or the achievement of an outcome.
- Process capability:** The range of expected results that can be achieved by following a process.
- Process description:** A document that describes a process, including (for example) its purpose, customers, customer requirements, entrance criteria, inputs, outputs, exit criteria, tasks involved, and who is responsible for each, measurement indicators, resources needed, and version.
- Process flowchart:** A diagram that shows a step-by-step series of actions through a procedure using connecting lines and a set of standard symbols adopted by an organization.
- Process model:** A framework for identifying, defining, and organizing the functional strategies, rules, and processes needed to manage and support the way an organization does or wants to do business. The process model provides a graphical and textual framework for organizing the data and processes into manageable groups to facilitate their shared use and control throughout the organization.
- Project:** An undertaking focused on developing or maintaining a product. Typically a project has its own funding, accounting, and delivery schedule.
- Project champion:** An advocate who is very familiar with the set of real customer needs for a system and who provides an active role in the development effort, facilitating the tasks of the development team.
- Project or program manager (PM):** The person who has total business responsibility for a project and is ultimately responsible to a customer.

Protocol: A set of rules governing network functionality.

Prototyping: A technique for building a quick and rough version of a desired system or parts of that system. The prototype illustrates the capabilities of the system to users and designers. It serves as a communications mechanism to allow reviewers to understand interactions with the system. It enables them to identify problems and to consider ways to improve a system. It sometimes gives an impression that developers are further along than is actually the case, giving users an overly optimistic impression of completion possibilities!

Quality: Meeting real customer needs.

Quality culture: The presence of an attitude of continuous improvement and customer satisfaction throughout an organization.

Quality function deployment (QFD): A methodology originally conceived in Japan in the 1970s that provides an opportunity for the user and the developer of a system to understand requirements more fully and to prioritize them.

Rational Unified Process: A methodology advocated by Rational Software, Inc.

Real requirements: The subset of stated requirements that reflects the verified needs for a particular system or capability.

REQ FLOW: How a requirement flows through an organization or department.

REQ MULTIPLEXING: The assignment of business requirements to the right project or projects. One requirement manager can multiplex requirements between many projects.

Requirement: A necessary attribute in a system; a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a user.

Requirements allocation: Assignment of requirements to architectural components of a system (for example, a hardware or software configuration item, training, or documentation); sometimes referred to as *flowdown*.

Requirements analysis: A structured (organized) method to understand the attributes that will satisfy a customer need.

Requirements baseline: The set of requirements associated with a particular release of a product or system.

Requirements definition: A detailed description, in general rather than functional terms, of the attributes needed in a system.

Requirements derivation: To obtain requirements for a system from sources provided by the customer.

Requirements document: A repository of the attributes in a system.

- Requirements elicitation:** The process of emerging requirements based on information provided by the customer.
- Requirements engineering:** An area within the broader field of systems and software engineering that focuses on the requirements process.
- Requirements leakage:** The addition or leaking in of unofficial requirements to the requirements specification when the requirements are not really needed.
- Requirements management:** Tracking requirements status and change activity and tracing requirements to the various phases and products of the development effort.
- Requirements process:** A full system life cycle set of actions concerning the necessary attributes of systems. The requirements process involves understanding customer needs and expectations (requirements elicitation), requirements analysis and specification, requirements prioritization, requirements derivation, partitioning and allocation, requirements tracing, requirements management, requirements verification, and requirements validation.
- Requirements traceability:** The ability to map the customer need to the requirement (connectivity) or to map a parent requirement to a child and vice versa. The ability to trace a requirement throughout the system development process, from requirements specification to design, to system component development, through testing and system documentation. This is absolutely critical for all systems.
- Requirements verification:** Independent assurance that requirements are addressed and met in a system.
- Requirements verification matrix:** An analysis that shows the verification method for each requirement.
- Risk:** The possibility of suffering loss.
- Robust architecture:** An underlying structure of a system that can readily meet and adapt to real requirements.
- Role:** A set of defined responsibilities that may be assumed by one or more individuals
- Rules of engagement:** A term used in Canada to describe the roles and responsibilities for project decision makers, including requirement prioritization and escalation procedures.
- Scalability:** The capability to grow to accommodate increased workloads.
- Security:** Services that protect data, ensuring its confidentiality, availability, and integrity.
- Senior management:** A role sufficiently high in the organization that the primary focus is on the long-term vitality of the organization.
- Software quality:** Software that combines the characteristics of low defect rates and high user satisfaction.

- Specification:** A document that describes technical requirements and verification procedures for items, materials, and services; an output of the requirements analysis process.
- Stakeholder:** Anyone who has an interest in a system or in its possessing qualities that meet particular needs.
- Stated requirements:** Requirements provided by a customer at the beginning of a system or software development effort; should be distinguished from real requirements.
- Subject matter expert (SME):** An individual who has been working in a particular field for an extensive period of time and who is trained in that area; often referred to as a *domain expert*.
- Supplier:** An organization that contracts with a buyer to provide a system.
- SWAT team:** A group tasked to provide a fast-action cure for a problem.
- System:** An integrated set of people, products, and processes that provides a capability to satisfy a customer need.
- System life cycle:** The set of activities involved in understanding a customer need; defining and analyzing requirements; preparing a design; developing a system; testing, implementing, operating, and maintaining it; and ending in its retirement.
- Systems engineering:** (1) A technical and management discipline that translates a customer need into a system that meets the customer need. Another source states system engineering is the iterative but controlled process during which user needs are understood and evolved into an operational system. The role of systems engineering is technical authority on a project; single interface to customer and project; architecture and system design; requirements derivation, allocation, and interpretation; and others (check SE-CMM key process area list).
- Systems engineering:** (2) (According to the CMM) The selective application of scientific and engineering efforts to transform an operational need into a description of the system configuration that best satisfies the operational need according to the measures of effectiveness; to integrate related technical parameters and to ensure compatibility of all physical, functional, and technical program interfaces in a manner that optimizes the total system definition and design; and to integrate efforts of all engineering disciplines and specialties into a total engineering effort.
- Tailoring:** The activity of modifying, elaborating, or adapting a process or document for another use. Reuse of tailored artifacts saves time and money and is an advantage of a process-oriented approach.
- TCP/IP gateway:** Transmission Control Protocol/Internet Protocol gateway. A device, or pair of devices, that interconnects two or more networks or subnetworks, enabling the passage of data from one (sub)network to another.

- Teamwork:** Proactive support of one another; necessary for success of any significant undertaking. Physical colocation facilitates teamwork and may be a prerequisite to success.
- Technical performance measures:** Indicators of how well the system works and how well the requirements are met; estimates or measures of the values of essential performance parameters. TPMs are used to evaluate the impact on cost, schedule, and technical effort.
- Technical reference model (TRM):** A structure that allows the components of an information system to be described in a consistent manner.
- Technical specification:** A comprehensive collection of the details of how a system will be implemented, including the technical architecture (hardware and software), decomposition of the system into subsystems, identification of common modules that will be developed, and other details requiring definition to allow development of the system.
- Technique:** A set of rules to follow to accomplish a task, a treatment of technical details, a body of technical methods, or a method of accomplishing a desired aim.
- Technology insertion:** Adding new technology to a system throughout the system life cycle.
- Time-boxed:** A technique that forces trade-offs in the context of short delivery cycles. See Highsmith, *Adaptive Software Development*.
- Tool:** Something used to facilitate performing an operation or practicing a process or activity.
- Top down:** Breaking down a system (decomposing) into component subsystems; often referred to as a *structured approach*.
- Trade study:** An analysis of alternative courses of action in which a balancing of factors, all of which are not obtainable at the same time, is performed.
- Transaction:** Interaction between a user and a computer in which the user inputs a command to receive a specific result from the computer.
- Unified Modeling Language (UML):** A general-purpose notation (a way to document) that describes the static and dynamic behavior of a system; not a design method or a development process.
- Use case:** A picture of actions a system performs, depicting the actors.
- Use case driven:** Describing the behavior of a system based on how the users interact with the system.
- Use case model:** A description of the functional behavior of a system that includes all the actors and all the use cases through which the actors interact with the system.

Glossary

User: The individual or group that uses a system in its environment.

User friendly: Easy to use.

User perspective: The view the user wants, needs, prefers, is happy with, and can use.

User satisfaction: Clients who are pleased with a vendor's products, quality levels, ease of use, and support.

Validation: A process for confirming that the real requirements are implemented in the delivered system.

Venn diagram: An illustration that employs circles to represent the relationships between and among sets (groups of objects that share something in common).

Verification: A process for ensuring that the design solution satisfies the requirements.

Verification methods: The approaches used to perform verification: test, inspection, demonstration, and analysis.

View: A perspective of a system such as the functional, implementation, or physical view.

Workbench: A suite of development tools.

Work product: Something produced or created as a result of systems or software development activity.