

**Table S1.** Filtered terms from ISO/IEC/IEEE International Standard - Systems and software engineering – Vocabulary.

Term	Definition
<Viewpoint> language	<ol style="list-style-type: none"> <li>1. definitions of concepts and rules for the specification of an ODP system from the &lt;viewpoint&gt; viewpoint.</li> </ol> <p>NOTE: Thus, engineering language is defined as "definitions of concepts and rules for the specification of an ODP system from the engineering viewpoint".</p>
abstract data type	<ol style="list-style-type: none"> <li>1. a data type for which only the properties of the data and the operations to be performed on the data are specified, without concern for how the data will be represented or how the operations will be implemented</li> </ol>
abstraction	<ol style="list-style-type: none"> <li>1. a view of an object that focuses on the information relevant to a particular purpose and ignores the remainder of the information.</li> <li>2. the process of formulating a view cf. data abstraction</li> </ol>
acquire project team	<ol style="list-style-type: none"> <li>1. [Process] the process of confirming human resource availability and obtaining the team necessary to complete project assignments.</li> </ol>
acquirer	<ol style="list-style-type: none"> <li>1. stakeholder that acquires or procures a product or service from a supplier.</li> <li>2. the individual or organization that specifies requirements for and accepts delivery of a new or modified software product and its documentation.</li> </ol> <p>Syn: buyer, customer, owner, purchaser.</p> <p>NOTE: The acquirer may be internal or external to the supplier organization. Acquisition of a software product may involve, but does not necessarily require, a legal contract or a financial transaction between the acquirer and supplier.</p>
acquisition	<ol style="list-style-type: none"> <li>1. process of obtaining a system, software product or software service.</li> <li>2. process of obtaining a system product or service.</li> </ol> <p>Syn: outsourcing</p>
acquisition strategy	<ol style="list-style-type: none"> <li>1. specific approach to acquiring products and services that is based on considerations of supply sources, acquisition methods, requirements specification types, contract or agreement types, and related acquisition risks</li> </ol>
action	<ol style="list-style-type: none"> <li>1. element of a step that a user performs during a procedure.</li> <li>2. a description of an operation to be taken in the formulation of a solution.</li> </ol>
action entry	<ol style="list-style-type: none"> <li>1. an indication of the relevance of an action to a particular rule.</li> </ol>
action stub	<ol style="list-style-type: none"> <li>1. a list of all the actions to be taken in the solution of a problem.</li> </ol>
active digital artifact	<ol style="list-style-type: none"> <li>1. Active Digital Artefacts are running software applications, agents or Services that may access other Services or Resources.</li> </ol>
activity	<ol style="list-style-type: none"> <li>2. set of cohesive tasks of a process.</li> <li>3. a component of work performed during the course of a project.</li> <li>4. an order submitted to the system under test (SUT) by a user or an emulated user demanding the execution of a data processing operation according to a defined algorithm to produce specific output data from specific input data and (if requested) stored data.</li> <li>5. a defined body of work to be performed, including its required input information and output information.</li> <li>6. collection of related tasks.</li> <li>7. element of work performed during the implementation of a process.</li> </ol> <p>NOTE: An activity normally has an expected duration, cost, and resource requirements. Activities are often subdivided into tasks.</p>
activity group	<ol style="list-style-type: none"> <li>1. a set of related activities</li> </ol>
activity identifier	<ol style="list-style-type: none"> <li>1. a short unique numeric or text identification assigned to each schedule activity to differentiate that project activity from other activities. Typically unique within any one project schedule network diagram.</li> </ol>

activity list	1. [Output/Input] a documented tabulation of schedule activities that shows the activity description, activity identifier, and a sufficiently detailed scope of work description so project team members understand what work is to be performed.
activity type	1. a classification of activities defined by the execution of the same algorithm.
actor	<ol style="list-style-type: none"> <li>1. a role (with respect to that action) in which the enterprise object fulfilling the role participates in the action.</li> <li>2. organization or CASE tool that supplies and/or acquires SEE Services.</li> <li>3. in UML, someone or something outside the system that interacts with the system.</li> </ol> <p>NOTE: It may be of interest to specify which actor initiates that action.</p>
adoption process	1. set of activities by which an organization brings CASE tools into widespread use.
agent	<ol style="list-style-type: none"> <li>1. an enterprise object that has been delegated (authority, responsibility, a function, etc.) by and acts for another enterprise object (in exercising the authority, carrying out the responsibility, performing the function, etc.).</li> </ol> <p>NOTE: An agent may be a party or may be the ODP system or one of its components. Another system in the environment of the ODP system may also be an agent. The delegation may have been direct, by a party, or indirect, by an agent of the party having authorization from the party to so delegate.</p>
allocated requirement	1. requirement that levies all or part of the performance and functionality of a higher level requirement on a lower level architectural element or design component
allocation	<ol style="list-style-type: none"> <li>1. the process of distributing requirements, resources, or other entities among the components of a system or program.</li> <li>2. the result of the distribution of requirements, resources, or other entities among the components of a system or program.</li> <li>3. the decision to assign a function or decision to hardware, software, or humans.</li> </ol> <p>NOTE: Allocation may be made entirely to hardware, software, or humans, or to some combination to be resolved upon further functional decomposition.</p>
analysis	<ol style="list-style-type: none"> <li>1. the process of studying a system by partitioning the system into parts (functions, components, or objects) and determining how the parts relate to each other.</li> <li>2. investigation and collection phase of user documentation development that aims to specify types of users and their information needs.</li> </ol>
analysis model	1. algorithm or calculation combining one or more base and/or derived measures with associated decision criteria.
analyst	1. a member of the technical community (such as a systems engineer or business analyst, developing the system requirements) who is skilled and trained to define problems and to analyze, develop, and express algorithms
aperiodic task	1. a task activated on demand. Syn: asynchronous task
applicability to a functional domain	1. the ability of an FSM method to take into account the characteristics of functional user requirements (FUR) which are pertinent to FSM in a functional domain.
application	<ol style="list-style-type: none"> <li>1. a system for collecting, saving, processing, and presenting data by means of a computer.</li> <li>2. a coherent collection of automated procedures and data supporting a business objective</li> <li>3. a cohesive collection of automated procedures and data supporting a business objective.</li> </ol> <p>Syn: application system, information system cf. system</p> <p>NOTE: It consists of one or more components, modules, or subsystems. Frequently a synonym for "system", the word "application" is preferred to express more precisely the nature of the subject matter of functional size measurement.</p>

application administration function	1. functions performed by users which include installation, configuration, application backup, maintenance (patching and upgrading) and de-installation.
application area	<ol style="list-style-type: none"> <li>1. a general term for a grouping of applications that handle a specific business area.</li> <li>2. a category of projects that have common components significant in such projects, but are not needed or present in all projects. Application areas are usually defined in terms of either the product (i.e., by similar technologies or production methods) or the type of customer (i.e., internal versus external, government versus commercial) or industry sector (i.e., utilities, automotive, aerospace, information technologies, etc.) Application areas can overlap.</li> </ol> <p>NOTE: It corresponds to an administrative level for management purposes.</p>
application area level	1. the management level responsible for managing maintenance activities as well as new development or major enhancement projects for one or more applications.
application boundary	<ol style="list-style-type: none"> <li>1. the border between the application and its environment of other applications and users.</li> <li>2. the border between the software being measured and the user.</li> </ol>
application engineering	1. the process of constructing or refining application systems by reusing assets.
application frameworks	<ol style="list-style-type: none"> <li>1. a subsystem design made up of a collection of abstract and concrete classes and interfaces between them.</li> </ol> <p>NOTE: Frameworks are often instantiation of a number of patterns.</p>
application problem	1. a problem submitted by an end user and requiring information processing for its solution.
architect	1. the person, team, or organization responsible for systems architecture.
architecting	1. the activities of defining, documenting, maintaining, improving, and certifying proper implementation of an architecture.
architectural description (AD)	1. a collection of products to document an architecture.
architectural design	<ol style="list-style-type: none"> <li>1. the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.</li> <li>2. the result of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system</li> </ol> <p>cf. functional design</p>
architectural design phase	1. the life-cycle phase in which a system's general architecture is developed, thereby fulfilling the requirements laid down by the software requirements document and detailing the implementation plan in response to it
architectural structure	<ol style="list-style-type: none"> <li>1. a physical or logical layout of the components of a system design and their internal and external connections.</li> </ol> <p>EXAMPLE: function-oriented (structured) design, object-oriented design, and data structure-oriented design</p>
architectural style	<ol style="list-style-type: none"> <li>1. definition of a family of systems in terms of a pattern of structural organization.</li> <li>2. characterization of a family of systems that are related by sharing structural and semantic properties.</li> </ol> <p>EXAMPLE: pipes and filters, layers, rule-based systems, and blackboards</p>
architecture	<ol style="list-style-type: none"> <li>1. fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.</li> <li>2. the organizational structure of a system or component.</li> <li>3. the organizational structure of a system and its implementation guidelines.</li> </ol> <p>Syn: architectural structure cf. component, module, subprogram, routine</p>

	NOTE: sometimes refers to the design of a system's hardware and software components
asset	<ol style="list-style-type: none"> <li>1. an item that has been designed for use in multiple contexts.</li> <li>2. an advantage or resource.</li> </ol>
association	<ol style="list-style-type: none"> <li>1. in UML, a relationship between an actor and a use case that indicates that the actor interacts with the system by means of the use case.</li> <li>2. a relationship (binding) between protocol objects (or between a protocol object and an interceptor) that is established independently of the protocol exchanges that support a particular computational interaction.</li> </ol>
base functional component (BFC)	<ol style="list-style-type: none"> <li>1. an elementary unit of functional user requirements defined by and used by an FSM method for measurement purposes.</li> </ol> <p>Syn: functional service</p> <p>EXAMPLE: a functional user requirement could be "Maintain Customers" which may consist of the following BFCs: "Add a new customer", "Report Customer Purchases", and "Change Customer Details". Another example might include a collection of logically related business data maintained by the software under study such as "Customer Details"</p>
block diagram	<ol style="list-style-type: none"> <li>1. a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks.</li> <li>2. a diagram of a system, computer, or device in which the principal parts are represented by suitably annotated geometrical figures to show both the functions of the parts and their functional relationships.</li> </ol> <p>Syn: configuration diagram, system resources chart</p> <p>cf. box diagram, bubble chart, flowchart, graph, input-process-output chart, structure chart</p>
bottom-up	<ol style="list-style-type: none"> <li>1. pertaining to an activity that starts with the lowest-level components of a hierarchy and proceeds through progressively higher-levels.</li> <li>2. pertaining to a method or procedure that starts at the lowest level of abstraction and proceeds towards the highest level.</li> </ol> <p>cf. topdown. critical piece first</p> <p>EXAMPLE: bottom-up design, bottom-up testing</p>
bottom-up design	<ol style="list-style-type: none"> <li>1. a design approach in which low-level pieces of a system are combined into an overall design.</li> <li>2. the process of designing a system by identifying low-level components, designing each component separately, and then designing a structure to integrate the low-level components into larger and larger subsystems until the design is finished</li> </ol>
box diagram	<ol style="list-style-type: none"> <li>1. a control flow diagram consisting of a rectangle that is subdivided to show sequential steps, if-then-else conditions, repetition, and case conditions.</li> </ol> <p>Syn: Chapin chart, Nassi-Shneiderman chart cf. block diagram, bubble chart, flowchart, graph, input-process-output chart, program structure diagram, structure chart</p>
bubble chart	<ol style="list-style-type: none"> <li>1. a data flow, data structure, or other diagram in which entities are depicted with circles (bubbles) and relationships are represented by links drawn between the circles cf. block diagram, box diagram, flowchart, graph, input-process-output chart, structure chart</li> </ol>
business objective	<ol style="list-style-type: none"> <li>1. strategy designed by senior management to ensure an organization's continued existence and enhance its profitability, market share, and other factors influencing the organization's success</li> </ol>
buyer	<ol style="list-style-type: none"> <li>1. the acquirer of products, services, or results for an organization.</li> <li>2. an individual or organization responsible for acquiring a product or service for use by themselves or other users.</li> <li>3. the person or organization that accepts the system and pays for the project.</li> </ol>
call graph	<ol style="list-style-type: none"> <li>1. a diagram that identifies the modules in a system or computer program and shows which modules call one another. Syn: call tree, tier chart</li> </ol>

case	<ol style="list-style-type: none"> <li>1. a single-entry, single-exit multiple-way branch that defines a control expression, specifies the processing to be performed for each value of the control expression, and returns control in all instances to the statement immediately following the overall construct.</li> <li>2. Computer Aided Software Engineering.</li> </ol> <p>Syn: multiple exclusive selective construct cf. go to, jump, if-then-else. multiple inclusive selective construct</p>
CASE needs	<ol style="list-style-type: none"> <li>1. organizational requirements which are met by CASE tool characteristics.</li> </ol> <p>NOTE: These characteristics are detailed in ISO/IEC 14102:1995. They include management process, development process, maintenance, documentation, configuration management, quality assurance, verification, validation, environment needs, CASE tool integrability, quality characteristics, acquisition needs, implementation needs, support indicators, and certification requirements.</p>
category	<ol style="list-style-type: none"> <li>1. a specifically defined division or grouping of software based upon one or more attributes or characteristics.</li> </ol>
classification	<ol style="list-style-type: none"> <li>1. a choice within a category.</li> <li>2. the manner in which the assets are organized for ease of search and extraction within a reuse library.</li> </ol>
classification process	<ol style="list-style-type: none"> <li>1. a series of activities, starting with the recognition of an anomaly through to its closure.</li> </ol> <p>NOTE: The process is divided into four sequential steps interspersed with three administrative activities. The sequential steps are as follows: a) Step 1: Recognition, b) Step 2: Investigation, c) Step 3: Action, d) Step 4: Disposition. The three administrative activities applied to each sequential step are as follows: a) Recording, b) Classifying, c) Identifying impact.</p>
cluster	<ol style="list-style-type: none"> <li>1. a configuration of basic engineering objects forming a single unit for the purposes of deactivation, checkpointing, reactivation, recovery and migration.</li> </ol>
commercial-off-the-shelf (COTS)	<ol style="list-style-type: none"> <li>1. software defined by a market-driven need, commercially available, and whose fitness for use has been demonstrated by a broad spectrum of commercial users.</li> <li>2. software product available for purchase and use without the need to conduct development activities.</li> <li>3. an item that a supplier offers to several acquirers for general use.</li> </ol> <p>cf. software product</p> <p>NOTE: COTS software product includes the product description (including all cover information, data sheet, web site information, etc.), the user documentation (necessary to install and use the software), the software contained on a computer sensible media (disk, CD-ROM, internet downloadable, etc.). Software is mainly composed of programs and data. This definition applies also to product descriptions, user documentation and software which are produced and supported as separate manufactured goods, but for which typical commercial fees and licensing considerations may not apply.</p>
communication interface	<ol style="list-style-type: none"> <li>1. an interface of a protocol object that can be bound to an interface of either an interceptor object or another protocol object at an interworking reference point.</li> </ol>
component	<ol style="list-style-type: none"> <li>1. an entity with discrete structure, such as an assembly or software module, within a system considered at a particular level of analysis.</li> <li>2. one of the parts that make up a system.</li> <li>3. set of functional services in the software, which, when implemented, represents a well-defined set of functions and is distinguishable by a unique name.</li> </ol> <p>NOTE: A component may be hardware or software and may be subdivided into other components. The terms "module," "component," and "unit" are often used interchangeably or defined to be subelements of one another in different ways depending upon the context. The relationship of these terms is not yet standardized. A component may or may not be independently managed from the end-user or administrator's point of view.</p>

component integration test	1. testing of groups of related components.
computer resource	1. an element of a data processing system needed to perform required operations. EXAMPLE: storage devices, input-output units, one or more processing units, data, files, and programs
concept of operations document	1. a user-oriented document that describes a system's operational characteristics from the end user's viewpoint.
concept phase	1. the period of time in the system life cycle during which the user needs are identified and system concepts are described and evaluated. NOTE: precedes the requirements phase
conceptual data model	1. a data model that illustrates the data groups as they are seen by the user.
conceptual model	1. a model of the concepts relevant to some endeavour.
conceptual system design	1. a system design activity concerned with specifying the logical aspects of the system organization, its processes, and the flow of information through the system.
constraint	1. a limitation or implied requirement that constrains the design solution or implementation of the systems engineering process and is not changeable by the enterprise. 2. a restriction on software life cycle process (SLCP) development. 3. a rule that specifies a valid condition of data. 4. a responsibility that is a statement of facts that are required to be true in order for the constraint to be met. 5. a restriction on the value of an attribute or the existence of any object based on the value or existence of one or more others. 6. an externally imposed limitation on system requirements, design, or implementation or on the process used to develop or modify a system. 7. [Input]. the state, quality, or sense of being restricted to a given course of action or inaction. An applicable restriction or limitation, either internal or external to a project, which will affect the performance of the project or a process. For example, a schedule constraint is any limitation or restraint placed on the project schedule that affects when a schedule activity can be scheduled and is usually in the form of fixed imposed dates. 8. a statement that expresses measurable bounds for an element or function of the system. cf. software life cycle process (SLCP) NOTE: That is, a constraint is a factor that is imposed on the solution by force or compulsion and may limit or modify the design changes.
consumer	1. the organization or person who buys the software package
context diagram	1. a diagram that presents the context of the top-level function of an IDEF0 model, whose diagram number is a-n, where 0#n#9.
continuous risk management	1. the process of analyzing the progress of a planned activity, project, or program on a periodic, ongoing basis and handling identified risk factors NOTE Includes developing options and fallback positions to permit alternative solutions to reduce the impact if a risk factor becomes a problem.
contract work breakdown structure (CWBS)	1. portion of the overall work breakdown structure applicable to a contract, developed and maintained by the supplier
contractual requirement	1. result of the analysis and refinement of customer requirements into a set of requirements suitable to be included in one or more solicitation packages, formal contracts, or supplier

	agreements between the acquirer and other appropriate organizations cf. acquirer, customer requirement
create WBS (work breakdown structure)	1. [Process] the process of subdividing project deliverables and project work into smaller, more manageable components.
customer	<ol style="list-style-type: none"> <li>1. organization or person that receives a product or service.</li> <li>2. the entity or entities for whom the requirements are to be satisfied in the system being defined and developed.</li> <li>3. an individual or organization who acts for the ultimate user of a new or modified hardware or software product to acquire the product and its documentation.</li> <li>4. the person, or persons, who pay for the product and usually (but not necessarily) decide the requirements.</li> </ol> <p>Syn: acquirer, buyer, beneficiary, purchaser cf. stakeholder</p> <p>EXAMPLE: an end-user of the completed system, an organization within the same company as the developing organization (e.g., System Management)</p> <p>NOTE: A customer can be internal or external to the organization. The customer may be a higher level project. This is the entity to whom the system developer must provide proof that the system developed satisfies the system requirements specified. Customers are a subset of stakeholders.</p>
customer requirement	1. the result of eliciting, consolidating, and resolving conflicts among the needs, expectations, constraints, and interfaces of the product's relevant stakeholders in a way that is acceptable to the customer
data	<ol style="list-style-type: none"> <li>1. a representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means.</li> <li>2. collection of values assigned to base measures, derived measures and/or indicators.</li> <li>3. the representation forms of information dealt with by information systems and users thereof.</li> <li>4. a reinterpretable representation of information in a formalized manner suitable for communication, interpretation, or communication, or processing.</li> </ol> <p>cf. data type</p>
data abstraction	<ol style="list-style-type: none"> <li>1. the process of extracting the essential characteristics of data by defining data types and their associated functional characteristics and disregarding representation details.</li> <li>2. the result of the process in (1) cf. encapsulation, information hiding.</li> </ol>
data analysis	1. a systematic investigation of the data and their flow in a real or planned system.
data communication	1. transfer of data among functional units according to sets of rules governing data transmission and the coordination of the exchange.
data flow diagram (DFD)	1. a diagram that depicts data sources, data sinks, data storage, and processes performed on data as nodes, and logical flow of data as links between the nodes. Syn: data flowchart, data flow graph cf. control flow diagram, data structure diagram
data input sheet	1. user documentation that describes, in a worksheet format, the required and optional input data for a system or component cf. user manual
data inventory	1. in an information processing system, all the data and their characteristics, including interdependencies.
data management	<ol style="list-style-type: none"> <li>1. in a data processing system, the functions that provide access to data, perform or monitor the storage of data, and control input-output operations.</li> <li>2. the disciplined processes and systems that plan for, acquire, and provide stewardship for business and technical data, consistent with data requirements, throughout the data lifecycle.</li> </ol>

data model	<ol style="list-style-type: none"> <li>1. a graphical and textual representation of analysis that identifies the data needed by an organization to achieve its mission, functions, goals, objectives, and strategies and to manage and rate the organization.</li> </ol> <p>NOTE: A data model is one that may be encoded and manipulated by a computer. A data model identifies the entities, domains (attributes), and relationships (associations) with other data and provides the conceptual view of the data and the relationships among data.</p>
data processing (DP)	<ol style="list-style-type: none"> <li>1. the systematic performance of operations upon data.</li> </ol> <p>Syn: automatic data processing (ADP)</p> <p>EXAMPLE: arithmetic or logic operations upon data, merging or sorting of data, assembling or compiling of programs, or operations on text, such as editing, sorting, merging, storing, retrieving, displaying, or printing.</p> <p>NOTE: The term data processing should not be used as a synonym for information processing.</p>
data processing system	<ol style="list-style-type: none"> <li>1. one or more computers, peripheral equipment, and software that perform data processing. Syn: computer system, computing system</li> </ol>
data protection	<ol style="list-style-type: none"> <li>1. the implementation of appropriate administrative, technical, or physical means to guard against unauthorized intentional or accidental disclosure, modification, or destruction of data.</li> </ol>
data repository	<ol style="list-style-type: none"> <li>1. an object providing the storage function.</li> </ol>
data store	<ol style="list-style-type: none"> <li>1. organized and persistent collection of data and information that allows for its retrieval.</li> </ol>
data structure	<ol style="list-style-type: none"> <li>1. a physical or logical relationship among data elements, designed to support specific data manipulation functions</li> </ol>
data structure diagram	<ol style="list-style-type: none"> <li>1. a diagram that depicts a set of data elements, their attributes, and the logical relationships among them cf. data flow diagram, entity-relationship diagram</li> </ol>
decision table	<ol style="list-style-type: none"> <li>1. a table of all contingencies that are to be considered in the description of a problem together with the action to be taken.</li> <li>2. a table used to show sets of conditions and the actions resulting from them.</li> <li>3. a table of conditions that are to be considered in the analysis of a problem, together with the action to be taken for each condition.</li> </ol>
decision tree analysis	<ol style="list-style-type: none"> <li>1. [Technique] the decision tree is a diagram that describes a decision under consideration and the implications of choosing one or another of the available alternatives. It is used when some future scenarios or outcomes of actions are uncertain. It incorporates probabilities and the costs or rewards of each logical path of events and future decisions, and uses expected monetary value analysis to help the organization identify the relative values of alternate actions.</li> </ol>
decomposition	<ol style="list-style-type: none"> <li>1. [Technique] a planning technique that subdivides the project scope and project deliverables into smaller, more manageable components, until the project work associated with accomplishing the project scope and providing the deliverables is defined in sufficient detail to support executing, monitoring, and controlling the work. 2. the partitioning of a modeled function into its component functions.</li> </ol>
decomposition diagram	<ol style="list-style-type: none"> <li>1. a diagram that details its parent box.</li> </ol>
deliverable	<ol style="list-style-type: none"> <li>1. [Output/Input] any unique and verifiable product, result, or capability to perform a service that must be produced to complete a process, phase, or project. Often used more narrowly in reference to an external deliverable, which is a deliverable that is subject to approval by the project sponsor or customer.</li> <li>2. item to be provided to an acquirer or other designated recipient as specified in an agreement cf. acquirer, product, result.</li> </ol> <p>NOTE: This item can be a document, hardware item, software item, service, or any type of work product.</p>



demodularization	1. in software design, the process of combining related software modules, usually to optimize system performance cf. downward compression, lateral compression, upward compression
deployment	1. phase of a project in which a system is put into operation and cutover issues are resolved cf. release
derived requirement	1. a lower-level requirement that is determined to be necessary for a top-level requirement to be met. 2. a requirement that is not explicitly stated in customer requirements, but is inferred from contextual requirements (such as applicable standards, laws, policies, common practices, and management decisions) or from requirements needed to specify a product or service component cf. product requirement. NOTE: Derived requirements can arise during analysis and design of components of the product or service.
design	1. the process of defining the architecture, components, interfaces, and other characteristics of a system or component. 2. the result of the process in (1). 3. the process of defining the software architecture, components, modules, interfaces, and data for a software system to satisfy specified requirements. 4. the process of conceiving, inventing, or contriving a scheme for turning a computer program specification into an operational program. 5. activity that links requirements analysis to coding and debugging. 6. stage of documentation development that is concerned with determining what documentation will be provided in a product and what the nature of the documentation will be.
design architecture	1. an arrangement of design elements that provides the design solution for a product or life cycle process intended to satisfy the functional architecture and the requirements baseline.
design description	1. a document that describes the design of a system or component. Syn: design document, design specification cf. product specification, requirements specification NOTE: Typical contents include system or component architecture, control logic, data structures, input/output formats, interface descriptions, and algorithm.
design element	1. a basic component or building block in a design
design entity	1. part of a design that is structurally, functionally, or otherwise distinct from other elements or that plays a different role relative to other design entities. 2. an element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced.
design level	1. the design decomposition of the software item EXAMPLE system, subsystem, program, or module
design methodology	1. a systematic approach to creating a design consisting of the ordered application of a specific collection of tools, techniques, and guidelines
design pattern	1. a description of the problem and the essence of its solution to enable the solution to be reused in different settings. NOTE: not a detailed specification, but a description of accumulated wisdom and experience
design phase	1. the period in the software life cycle during which definitions for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements. 2. the period in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements cf. detailed design, preliminary design

design requirement	1. a requirement that specifies or constrains the design of a system or system component cf. functional requirement, implementation requirement, interface requirement, performance requirement, physical requirement
development project	1. a project in which a completely new application is realized. NOTE: It entails the specification, construction, testing, and delivery of a new application. During actualization, this project can be split up into a number of sub-projects. If these are carried out more or less in parallel, each being responsible for effectuating a certain sub-system of the total application, then each sub-project should be considered as an individual development project, if the sub-system itself is an application. Re-building an existing application, otherwise known as re-engineering, is considered as development.
device	1. a mechanism or piece of equipment designed to serve a purpose or perform a function.
diagram	1. an instantiation of the formal diagram structure that consists only of semantically and syntactically valid IDEF0 graphical statements. NOTE: Each diagram is a single unit of an IDEF0 model that presents the top-level function that is the subject of the model (the A-0 context diagram), presents the context of the subject function (other context diagrams), or presents the details of a box (decomposition diagrams).
digital entity	1. Any computational or data element of an ITC-based system.
documentation tree	2. a diagram that depicts all of the documents for a given system and shows their relationships to one another cf. specification tree
domain	1. a distinct scope, within which common characteristics are exhibited, common rules observed, and over which a distribution transparency is preserved. 2. a problem space.
domain analysis	1. the analysis of systems within a domain to discover commonalities and differences among them. 2. the process by which information used in developing software systems is identified, captured, and organized so that it can be reused to create new systems, within a domain. 3. the result of the domain analysis process.
domain architecture	1. a generic, organizational structure or design for software systems in a domain. NOTE: The domain architecture contains the designs that are intended to satisfy requirements specified in the domain model. The domain architecture documents design, whereas the domain model documents requirements. A domain architecture: 1) can be adapted to create designs for software systems within a domain, and 2) provides a framework for configuring assets within individual software systems. The term "architecture" has been deliberately redefined to more properly convey its meaning in the software reuse context.
domain definition	1. the process of determining the scope and boundaries of a domain.
domain engineer	1. a party that performs domain engineering activities, including domain analysis, domain design, asset construction, and asset maintenance.
domain engineering	1. a reuse-based approach to defining the scope (i.e., domain definition), specifying the structure (i.e., domain architecture), and building the assets for a class of systems, subsystems, or applications. NOTE: "Assets" may include requirements, designs, software code, or documentation. Domain engineering may include the following activities: domain definition, domain analysis, developing the domain architecture, and domain implementation.
domain expert	1. an individual who is intimately familiar with the domain and can provide detailed information to the domain engineers.
domain model	1. a product of domain analysis that provides a representation of the requirements of the domain. NOTE: The domain model identifies and describes the structure of data, flow of information, functions, constraints, and controls within the domain that are included in software systems

	in the domain. The domain model describes the commonalities and variabilities among requirements for software systems in the domain.
element	1. a component of a system; may include equipment, a computer program, or a human. EXAMPLE documents, requirements specifications, test cases, source code, installation information, and read-me files
element type	1. a category or class of elements
elementary process	1. the smallest unit of activity that is meaningful to the user(s).
end user	1. the person or persons who will ultimately be using the system for its intended purpose. 2. individual person who ultimately benefits from the outcomes of the system. 3. the person who uses the software package. 4. any person that communicates or interacts with the software at any time. Syn: end-user
entity	1. a fundamental thing of relevance to the user, about which information is kept. 2. in computer programming, any item that can be named or denoted in a program. 3. an object (i.e., thing, event or concept) that occurs in a model (i.e., transfer). 4. object that is to be characterized by measuring its attributes. 5. the representation of a set of real or abstract things that are recognized as the same type because they share the same characteristics and can participate in the same relationships. 6. a fundamental thing of relevance to the user, about which a collection of facts is kept. 7. object to be modeled. 8. logical component of the data store, representing fundamental things of relevance to the user, and about which persistent information is stored. EXAMPLE a data item, program statement, or subprogram
entity-relationship (E-R) diagram	1. a diagram that depicts a set of real-world entities and the logical relationships among them. Syn: entity-relationship map cf. data structure diagram
environment	1. anything affecting a subject system or affected by a subject system through interactions with it, or anything sharing an interpretation of interactions with a subject system. 2. the configuration(s) of hardware and software in which the software operates. 3. the circumstances, objects, and conditions that surround a system to be built. 4. the circumstances, objects, and conditions that will influence the completed system. 5. a concept space, i.e., an area in which a concept has an agreed-to meaning and one or more agreed-to names that are used for the concept.
error model	1. in software evaluation, a model used to estimate or predict the number of remaining faults, required test time, and similar characteristics of a system. Syn: error prediction model
error prediction	1. a quantitative statement about the expected number or nature of faults in a system or component cf. error model, error seeding
error processing	1. the process of detecting and responding to a program's errors
error seeding	1. the process of intentionally adding known faults to those already in a computer program for the purpose of monitoring the rate of detection and removal, and estimating the number of faults remaining in the program. Syn: bug seeding, fault seeding cf. indigenous error
error tolerance	1. the ability of a system or component to continue normal operation despite the presence of erroneous inputs cf. fault tolerance, robustness
evaluation checklist	1. list of questions, each of which is designed to check for conformity of a product, process or service to one or more provisions within a particular International Standard.
evaluation procedure	1. series of tasks and steps that, when completed, enable the evaluation team to determine if the product, process or service being evaluated is conformant to a particular standard.

event	<ol style="list-style-type: none"> <li>1. occurrence of a particular set of circumstances.</li> <li>2. an external or internal stimulus used for synchronization purposes</li> </ol> <p>NOTE: The event can be certain or uncertain. The event can be a single occurrence or a series of occurrences. The probability associated with the event can be estimated for a given period of time. An event can be an external interrupt, a timer expiration, an internal signal, or an internal message.</p>
event sequence analysis	<ol style="list-style-type: none"> <li>1. performance analysis of the sequence of tasks that must be executed to service a given external event</li> </ol>
event sequence diagram	<ol style="list-style-type: none"> <li>1. a diagram that identifies the sequence of tasks required to process an external event</li> </ol>
event sequencing logic	<ol style="list-style-type: none"> <li>1. a description of how a task responds to each of its message or event inputs NOTE in particular, what output is generated as a result of each input</li> </ol>
exclusive requirement	<ol style="list-style-type: none"> <li>1. requirement of a normative document that must necessarily be fulfilled in order to comply with that document. NOTE deprecated: mandatory requirement. [ISO/IEC Guide 2:2004]</li> </ol>
facility	<ol style="list-style-type: none"> <li>1. physical means or equipment for facilitating the performance of an action. EXAMPLE buildings, instruments, tools</li> </ol>
failure mode	<ol style="list-style-type: none"> <li>1. the physical or functional manifestation of a failure NOTE A system in failure mode may be characterized by slow operation, incorrect outputs, or complete termination of execution.</li> </ol>
failure mode and effect analysis (FMEA)	<ol style="list-style-type: none"> <li>1. [Technique] an analytical procedure in which each potential failure mode in every component of a product is analyzed to determine its effect on the reliability of that component and, by itself or in combination with other possible failure modes, on the reliability of the product or system and on the required function of the component; or the examination of a product (at the system and/or lower levels) for all ways that a failure may occur. For each potential failure, an estimate is made of its effect on the total system and of its impact. In addition, a review is undertaken of the action planned to minimize the probability of failure and to minimize its effects.</li> </ol>
flowchart	<ol style="list-style-type: none"> <li>1. a graphical representation of a process or the step-by-step solution of a problem, using suitably annotated geometric figures connected by flowlines for the purpose of designing or documenting a process or program.</li> <li>2. graphical representation of the definition, analysis, or method of solution of a problem in which symbols are used to represent operations, data, flow, equipment, etc.</li> <li>3. a control flow diagram in which suitably annotated geometrical figures are used to represent operations, data, or equipment, and arrows are used to indicate the sequential flow from one to another.</li> </ol> <p>Syn: flow chart, flow diagram cf. block diagram, box diagram, bubble chart, graph, input-process-output chart, structure chart</p>
flowcharting	<ol style="list-style-type: none"> <li>1. [Technique] the depiction in a diagram format of the inputs, process actions, and outputs of one or more processes within a system.</li> </ol>
framework	<ol style="list-style-type: none"> <li>1. a reusable design (models and/or code) that can be refined (specialized) and extended to provide some portion of the overall functionality of many applications.</li> <li>2. a partially completed software subsystem that can be extended by appropriately instantiating some specific plug-ins</li> </ol>
function	<ol style="list-style-type: none"> <li>1. an elementary unit of requirements and specifications defined and used for measurement purposes.</li> <li>2. a software module that performs a specific action, is invoked by the appearance of its name in an expression, may receive input values, and returns a single value.</li> </ol>

	<ol style="list-style-type: none"> <li>3. a task, action, or activity that must be accomplished to achieve a desired outcome.</li> <li>4. the features or capabilities of an application as seen by the user.</li> <li>5. a defined objective or characteristic action of a system or component.</li> <li>6. a transformation of inputs to outputs, by means of some mechanisms, and subject to certain controls, that is identified by a function name and modeled by a box.</li> <li>7. an aspect of the intended behavior of the system.</li> <li>8. a single-valued mapping.</li> <li>9. part of an application that provides facilities for users to carry out their tasks.</li> </ol>
function type	<ol style="list-style-type: none"> <li>1. the five basic information services provided to the user of an application and identified in function point analysis: external input, external output, external inquiry, internal logical file, and external interface file.</li> <li>2. the five types of components of which an application consists, seen from the perspective of FPA.</li> </ol>
functional analysis	<ol style="list-style-type: none"> <li>1. a systematic investigation of the functions of a real or planned system.</li> <li>2. examination of a defined function to identify all the subfunctions necessary to accomplish that function, to identify functional relationships and interfaces (internal and external) and capture these in a functional architecture, to flow down upper-level performance requirements and to assign these requirements to lower-level subfunctions</li> </ol>
functional architecture	<ol style="list-style-type: none"> <li>1. an arrangement of functions and their subfunctions and interfaces (internal and external) that defines the execution sequencing, conditions for control or data flow, and the performance requirements to satisfy the requirements baseline.</li> <li>2. hierarchical arrangement of functions, their internal and external (external to the aggregation itself) functional interfaces and external physical interfaces, their respective functional and performance requirements, and their design constraints</li> </ol>
functional configuration identification	<ol style="list-style-type: none"> <li>1. in configuration management, the current approved technical documentation for a configuration item cf. allocated configuration identification, product configuration identification functional baseline</li> </ol> <p>NOTE: It prescribes all necessary functional characteristics, the tests required to demonstrate achievement of specified functional characteristics, the necessary interface characteristics with associated configuration items, the configuration item's key functional characteristics and its key lower-level configuration items, if any, and design constraints.</p>
functional decomposition	<ol style="list-style-type: none"> <li>1. a type of modular decomposition in which a system is broken down into components that correspond to system functions and subfunctions cf. hierarchical decomposition, stepwise refinement</li> </ol>
functional design	<ol style="list-style-type: none"> <li>1. the process of defining the working relationships among the components of a system.</li> <li>2. the result of the process in (1).</li> <li>3. the specification of the functions of the components of a system and of the working relationships among them.</li> </ol>
functional domain	<ol style="list-style-type: none"> <li>1. a class of software based on the characteristics of functional user requirements which are pertinent to FSM.</li> </ol>
functional domain categorization (FDC)	<ol style="list-style-type: none"> <li>1. a process for identifying functional domains that conforms to the requirements of ISO/IEC TR 14143-5:2004, clause 53.</li> </ol>
functional manager	<ol style="list-style-type: none"> <li>1. someone with management authority over an organizational unit within a functional organization. The manager of any group that actually makes a product or performs a service.</li> </ol>
functional organization	<ol style="list-style-type: none"> <li>1. a hierarchical organization where each employee has one clear superior, and staff are grouped by areas of specialization and managed by a person with expertise in that area.</li> </ol>
functional process	<ol style="list-style-type: none"> <li>1. an elementary component of a set of functional user requirements, comprising a unique, cohesive and independently executable set of data movements.</li> </ol>

	<p>2. an elementary component of a set of Functional User Requirements, comprising a unique, cohesive and independently executable set of data or data movements (functional services).</p> <p>NOTE: It is triggered by one or more Triggering events either directly, or indirectly via an 'actor'. It is complete when it has executed all that is required to be done in response to the Triggering event (-type).</p>
functional product	<p>1. a product capable of performing computations.</p>
functional requirement	<p>1. a statement that identifies what a product or process must accomplish to produce required behavior and/or results.</p> <p>2. a requirement that specifies a function that a system or system component must be able to perform</p>
functional service	<p>1. service that must be implemented in the piece of software in order to fulfill functional user requirements.</p> <p>2. base functional component (BFC).</p>
functional specification	<p>1. a document that specifies the functions that a system or component must perform NOTE often part of a requirements specification</p>
functional unit	<p>1. an entity of hardware or software, or both, capable of accomplishing a specified purpose.</p>
functional user requirements (FUR)	<p>1. a subset of the user requirements describing what the software does, in terms of tasks and services.</p> <p>NOTE: Functional User Requirements include but are not limited to: data transfer (for example, Input customer data, Send control signal); data transformation (for example, Calculate bank interest, Derive average temperature); data storage (for example, Store customer order, Record ambient temperature over time); data retrieval (for example, List current employees, Retrieve aircraft position). User Requirements that are not Functional User Requirements include but are not limited to: quality constraints (for example, usability, reliability, efficiency and portability); organizational constraints (for example, locations for operation, target hardware and compliance to standards); environmental constraints (for example, interoperability, security, privacy and safety); implementation constraints (for example, development language, delivery schedule).</p>
functionality	<p>1. the capabilities of the various computational, user interface, input, output, data management, and other features provided by a product. NOTE This characteristic is concerned with what the software does to fulfil needs.</p>
function-oriented design	<p>1. the partitioning of a design into subsystems and modules, with each one handling one or more functions cf. object-oriented design, data-structure-oriented design</p>
Gantt chart	<p>1. [Tool] a graphic display of schedule-related information. In the typical bar chart, schedule activities or work breakdown structure components are listed down the left side of the chart, dates are shown across the top, and activity durations are shown as date-placed horizontal bars.</p>
generation	<p>1. act of defining and describing a methodology from a particular metamodel.</p> <p>NOTE: Generating a methodology includes explaining the structural position and semantics of each methodology element using the selected metamodel. Thus, what methodology elements are possible, and how they relate to each other, are constrained by such a metamodel. Usually, method engineers perform generation, yielding a complete and usable methodology.</p>
glossary	<p>1. the collection of the names and narrative descriptions of all terms that may be used for defined concepts within an environment.</p>
high-level design	<p>1. the process of defining the high-level concepts that guide low-level design and implementation cf. architecture.</p> <p>NOTE: High-level design typically involves organizing a system into subprograms and specifying the interfaces between them.</p>

high-level Petri Net graph	1. a net graph and its associated annotations comprising place types, arc annotations and transition conditions, and their corresponding definitions in a set of declarations, and an initial marking of the net.
human resource planning	1. identification and documentation of project roles, responsibilities and reporting relationships, as well as estimation of required staff by time period and creation of a staffing management plan
ICT	1. information and communication technology.
IDEF0 model	1. abstractly, a hierarchical set of IDEF0 diagrams that depict, for a specific purpose and from a specific viewpoint, the functions of a system or subject area, along with supporting glossary, text, and For Exposition Only (FEO) information. NOTE: Concretely, a set of model pages that include at least an A-0 context diagram and an A0 decomposition diagram, a glossary or specific glossary pages, one or more text pages to accompany each diagram, and FEO pages and model pages of other types as needed.
identify risks	1. [Process] the process of determining which risks may affect the project and documenting their characteristics.
identify stakeholders	1. [Process] the process of identifying all people or organizations impacted by the project, and documenting relevant information regarding their interests, involvement, and impact on project success.
identifying relationship	1. a specific (not many-to-many) relationship in which every attribute in the primary key of the parent entity is contained in the primary key of the child entity. cf. nonidentifying relationship [key style]
impact analysis	1. identification of all system and software products that a change request affects and development of an estimate of the resources needed to accomplish the change NOTE This includes determining the scope of the changes to plan and implement work, accurately estimating the resources needed to perform the work, and analyzing the requested changes' cost and benefits.
implementation	1. the process of translating a design into hardware components, software components, or both. 2. the result of the process in (1). 3. a definition that provides the information needed to create an object and allow the object to participate in providing an appropriate set of services. 4. the installation and customization of packaged software. 5. construction. 6. the system development phase at the end of which the hardware, software and procedures of the system considered become operational. 7. a process of instantiation whose validity can be subject to test. 8. phase of development during which user documentation is created according to the design, tested, and revised.
implementation phase	1. period of time in the software life cycle during which a software product is created from design documentation and debugged
implementation requirement	1. a requirement that specifies or constrains the coding or construction of a system or system component cf. design requirement, functional requirement, interface requirement, performance requirement, physical requirement
implied needs	1. needs that may not have been stated but are actual needs. NOTE Implied needs are real needs which may not have been documented.
influence diagram	1. [Tool] a graphical representation of situations showing causal influences, time ordering of events, and other relationships among variables and outcomes.
information	1. knowledge that is exchangeable amongst users about things, facts, concepts, and so on, in a universe of discourse.

	<p>2. in information processing, knowledge concerning objects, such as facts, events, things, processes, or ideas, including concepts, that within a certain context has a particular meaning.</p> <p>NOTE: Although information will necessarily have a representation form to make it communicable, it is the interpretation of this representation (the meaning) that is relevant in the first place.</p>
information analysis	<p>1. a systematic investigation of information and its flow in a real or planned system.</p>
input-process-output	<p>1. a software design technique that consists of identifying the steps involved in each process to be performed and identifying the inputs to and outputs from each step.</p> <p>cf. data structure-centered design, input-process-output chart, modular decomposition, object-oriented design, rapid prototyping</p>
interface	<p>1. a shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical signal exchanges, and other characteristics.</p> <p>2. a hardware or software component that connects two or more other components for the purpose of passing information from one to the other.</p> <p>3. to connect two or more components for the purpose of passing information from one to the other.</p> <p>4. to serve as a connecting or connected component as in (2).</p> <p>5. the declaration of the meaning and the signature for a property or constraint.</p> <p>6. a shared boundary across which information is passed.</p> <p>7. a task's external specification.</p>
interface testing	<p>1. testing conducted to evaluate whether systems or components pass data and control correctly to one another cf. component testing, integration testing, system testing, unit test</p>
interface requirements specification (IRS)	<p>1. documentation that specifies requirements for interfaces between or among systems and components. cf. interface specification, interface design document NOTE These requirements include constraints on formats and timing</p>
interface specification	<p>1. the description of essential functional, performance, and design requirements and constraints at a common boundary between two or more system elements.</p> <p>2. a document that specifies the interface characteristics of an existing or planned system or component cf. interface requirements specification</p> <p>NOTE: This includes interfaces between humans and hardware or software, as well as interfaces between humans themselves.</p>
layer	<p>1. the result of the functional partitioning of the software environment such that all included functional processes perform at the same level of abstraction.</p> <p>NOTE: In a multi-layer software environment, software in one layer exchanges data with software in another layer through their respective functional processes. These interactions are hierarchical in nature; when considered in pairs, one layer is a "client" to the other. A "client" layer uses the functional services provided by other subordinate layers. Software items in the same layer can also exchange data. This type of data exchange is usually called "peer-to-peer" data exchange.</p>
list	<p>1. a set of data items, each of which has the same data definition.</p> <p>2. to print or otherwise display a set of data items.</p> <p>3. a collection class that contains no duplicates and whose members are ordered.</p>
metamodel	<p>1. a logical information model that specifies the modeling elements used within another (or the same) modeling notation.</p> <p>2. a metamodel Vm for a subset of IDEFObject is a view of the constructs in the subset that is expressed using those constructs such that there exists a valid instance of Vm that is a description of Vm itself.</p>



	<ol style="list-style-type: none"> <li>3. a model containing detailed definitions of the meta-entities, meta-relationships and meta-attributes whose instances appear in the model section of a CDIF transfer.</li> <li>4. specification of the concepts, relationships and rules that are used to define a methodology.</li> </ol>
methodology	<ol style="list-style-type: none"> <li>1. a system of practices, techniques, procedures, and rules used by those who work in a discipline.</li> <li>2. specification of the process to follow together with the work products to be used and generated, plus the consideration of the people and tools involved, during an Information-Based Domain development effort.</li> </ol>
methodology element	<ol style="list-style-type: none"> <li>1. simple component of a methodology.</li> </ol> <p>NOTE: Usually, methodology elements include the specification of what tasks, activities, techniques, models, documents, languages and/or notations can or must be used when applying the methodology. Methodology elements are related to each other, comprising a network of abstract concepts. Typical methodology elements are Capture Requirements, Write Code for Methods (kinds of tasks), Requirements Engineering, High-Level Modeling (kinds of activities), Pseudo-code, Dependency Graphs (notations), Class, Attribute (kinds of model building blocks), Class Model, Class Diagram, Requirements Specification (kind of work products).</p>
model	<ol style="list-style-type: none"> <li>1. a representation of a real world process, device, or concept.</li> <li>2. a representation of something that suppresses certain aspects of the modeled subject.</li> <li>3. an interpretation of a theory for which all the axioms of the theory are true.</li> <li>4. a related collection of instances of meta-objects, representing (describing or prescribing) an information system, or parts thereof, such as a software product.</li> <li>5. a semantically closed abstraction of a system or a complete description of a system from a particular perspective</li> </ol>
model glossary	<ol style="list-style-type: none"> <li>1. the collection of the names and definitions of all defined concepts that appear within the views of a model.</li> </ol>
node	<ol style="list-style-type: none"> <li>1. in a diagram, a point, circle, or other geometric figure used to represent a state, event, or other item of interest.</li> <li>2. a configuration of engineering objects forming a single unit for the purpose of location in space, and which embodies a set of processing, storage and communication functions.</li> <li>3. a modeled function located within the hierarchical graph structure of an IDEF0 model by its designated node number.</li> <li>4. one of the defining points of a schedule network; a junction point joined to some or all of the other dependency lines.</li> <li>5. a vertex of a net graph, i.e., a place or a transition. cf. graph (2)</li> </ol> <p>EXAMPLE: a computer and the software it supports (operating system and applications); a parallel computer the control of a single operating system</p> <p>NOTE: A node may have internal structure which is not of concern in an engineering specification.</p>
nonfunctional requirement	<ol style="list-style-type: none"> <li>1. a software requirement that describes not what the software will do but how the software will do it.</li> </ol> <p>Syn: design constraints, non-functional requirement cf. functional requirement</p> <p>EXAMPLE: software performance requirements, software external interface requirements, software design constraints, and software quality attributes. Nonfunctional requirements are sometimes difficult to test, so they are usually evaluated subjectively</p>
nontechnical requirement	<ol style="list-style-type: none"> <li>1. requirement affecting product and service acquisition or development that is not a property of the product or service EXAMPLE numbers of products or services to be delivered; data rights for delivered COTS non-developmental items; delivery dates; milestones with exit criteria; work constraints associated with training, site provisions, and deployment schedules.</li> </ol>
OBS	<ol style="list-style-type: none"> <li>1. organizational breakdown structure.</li> </ol>

ontology	1. a logical structure of the terms used to describe a domain of knowledge, including both the definitions of the applicable terms and their relationships.
operation	<ol style="list-style-type: none"> <li>1. in computer mathematics, the action specified by an operator on one or more operands.</li> <li>2. in programming, a defined action that can be performed by a computer system.</li> <li>3. the process of running a computer system in its intended environment to perform its intended functions.</li> <li>4. an interaction between a client object and a server object which is either an interrogation or an announcement.</li> <li>5. a property that is a mapping from the (cross product of the) instances of the class and the input argument types to the (cross product of the) instances of the other (output) argument types.</li> <li>6. an action needed to perform an activity.</li> <li>7. arithmetic or logical operation performed in an algorithmic and manipulation BFC.</li> </ol> <p>EXAMPLE: in the expression <math>A = B + 3</math>, the process of adding B to 3 to obtain A</p>
organizational breakdown structure (OBS)	1. [Tool] a hierarchically organized depiction of the project organization arranged so as to relate the work packages to the performing organizational units. (Sometimes OBS is written as Organization Breakdown Structure with the same definition.).
output product	1. the physical form that information can take and that an application distributes. EXAMPLE a report, an output file, or a message to a different application
passive digital artifact	1. Passive Digital Artefact Digital Artefacts are passive software elements such as data-base entries or other digital representations of the Physical Entity.
perform quality control	2. [Process] the process of monitoring and recording results of executing the quality activities to assess performance and recommend necessary changes.
physical entity	1. A Physical Entity is a discrete, identifiable part of the physical environment that is of interest to the user for the completion of her goal. Physical Entities can be almost any object or environment; from humans or animals to cars; from store or logistics chain items to computers; from electronic appliances to closed or open environments.
plan risk management	2. [Process] the process of defining how to conduct risk management activities for a project.
plan risk responses	1. [Process] the process of developing options and actions to enhance opportunities and to reduce threats to project objectives.
previously developed software	1. software that has been produced prior to or independent of the project for which the plan is prepared, including software that is obtained or purchased from outside sources.
priority	<ol style="list-style-type: none"> <li>1. the level of importance assigned to an item.</li> <li>2. a rank order of status, activities, or tasks.</li> </ol>
probability and impact matrix	1. [Tool] a common way to determine whether a risk is considered low, moderate, or high by combining the two dimensions of a risk: its probability of occurrence, and its impact on objectives if it occurs.
problem	<ol style="list-style-type: none"> <li>1. unknown underlying cause of one or more incidents.</li> <li>2. a negative situation to overcome.</li> </ol> <p>NOTE: risk factor becomes a problem when a risk metric (an objective measure) crosses a pre-determined threshold (the problem trigger).</p>
problem definition	1. a statement of a problem, which may include a description of the data, the method, the procedures, and algorithms used to solve it.
problem domain	1. a set of similar problems that occur in an environment and lend themselves to common solutions.

problem report (PR)	1. a document used to identify and describe problems detected in a product. NOTE PRs are either submitted directly to denote faults or established after impact analysis is performed on Modification Requests and faults are found.
procedure	1. ordered series of steps that specify how to perform a task. 2. a written description of a course of action as in (1). 3. a portion of a computer program that is named and that performs a specific action. 4. specified way to carry out an activity or process. 5. a routine that does not return a value
process	1. set of interrelated or interacting activities which transforms inputs into outputs. 2. a predetermined course of events defined by its purpose or by its effect, achieved under given conditions. 3. to perform operations on data. 4. a collection of steps taking place in a prescribed manner and leading to an objective. 5. in data processing, the predetermined course of events that occur during the execution of all or part of a program. 6. an executable unit managed by an operating system scheduler. 7. system of activities, which use resources to transform inputs into outputs. NOTE: [ISO 9000:2005] The term "activities" covers use of resources. A process may have multiple starting points and multiple end points. The prescribed manner may be a partially ordered sequence. A process specification can be a workflow specification. An enterprise specification may define types of processes and may define process templates.
process action plan	1. plan, usually resulting from appraisals, that documents how specific improvements targeting the weaknesses uncovered by an appraisal will be implemented
process action team	1. team that has the responsibility to develop and implement process improvement activities for an organization as documented in a process action plan
process architect	1. the person or group that has primary responsibility for creating and maintaining the software life cycle process (SLCP).
process architecture	1. ordering, interfaces, interdependencies, and other relationships among the process elements in a standard process. NOTE: Process architecture also describes the interfaces, interdependencies, and other relationships between process elements and external processes, such as contract management
process capability	1. a characterization of the ability of a process to meet current or projected business goals. 2. range of expected results that can be achieved by following a process
process description	1. documented expression of a set of activities performed to achieve a given purpose NOTE: A process description provides an operational definition of the major components of a process. The description specifies, in a complete, precise, and verifiable manner, the requirements, design, behavior, or other characteristics of a process. It also may include procedures for determining whether these provisions have been satisfied. Process descriptions can be found at the activity, project, or organizational level.
product	1. an artifact that is produced, is quantifiable, and can be either an end item in itself or a component item. 2. complete set of software and documentation. 3. output of the software development activities (e.g., document, code, or model). 4. result of a process. Syn: material, goods cf. activity, deliverable, result. NOTE: [ISO 9000:2005] There are four agreed generic product categories: hardware (e.g., engine mechanical part); software (e.g., computer program); services (e.g., transport); and processed materials (e.g., lubricant). Hardware and processed materials are generally tangible products, while software or services are generally intangible. Most products comprise elements belonging to different generic product categories. Whether the product is then called hardware, processed material, software, or service depends on the dominant element.

product requirement	1. refinement of customer requirements into the developers' language, making implicit requirements into explicit derived requirements
project	<ol style="list-style-type: none"> <li>endeavor with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements.</li> <li>an undertaking with pre-specified objectives, magnitude and duration.</li> <li>a temporary endeavor undertaken to create a unique product, service, or result.</li> <li>a collection of work tasks with a time frame and a work product to be delivered.</li> <li>set of activities for developing a new product or enhancing an existing product.</li> </ol> <p>NOTE: A project may be viewed as a unique process comprising coordinated and controlled activities and may be composed of activities from the Project Processes and Technical Processes.</p>
project management (PM)	<ol style="list-style-type: none"> <li>the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.</li> <li>the activities concerned with project planning and project control.</li> </ol>
project risk management	1. [Knowledge Area] project risk management includes the processes concerned with conducting risk management planning, identification, analysis, responses, and monitoring and control on a project.
relationship	<ol style="list-style-type: none"> <li>a real-world association among one or more entities.</li> <li>an association between two (not necessarily distinct) classes that is deemed relevant within a particular scope and purpose.</li> <li>a semantic connection between model elements.</li> <li>an association of interest between two entities.</li> <li>a predicate involving two or more roles with assigned values.</li> </ol> <p>NOTE: The association is named for the sense in which the instances are related. A relationship can be represented as a time-varying binary relation between the instances of the current extents of two state classes.</p>
relevant stakeholder	1. stakeholder that is identified for involvement in specified activities and is included in a plan
requirement	<ol style="list-style-type: none"> <li>a condition or capability needed by a user to solve a problem or achieve an objective.</li> <li>a condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents.</li> <li>a documented representation of a condition or capability as in (1) or (2).</li> <li>a condition or capability that must be met or possessed by a system, product, service, result, or component to satisfy a contract, standard, specification, or other formally imposed document. Requirements include the quantified and documented needs, wants, and expectations of the sponsor, customer, and other stakeholders.</li> </ol> <p>cf. design requirement, functional requirement, implementation requirement, interface requirement, performance requirement, physical requirement</p>
requirements analysis	<ol style="list-style-type: none"> <li>the process of studying user needs to arrive at a definition of system, hardware, or software requirements.</li> <li>the process of studying and refining system, hardware, or software requirements.</li> <li>a systematic investigation of user requirements to arrive at a definition of a system.</li> <li>determination of product- or service-specific performance and functional characteristics based on analyses of customer needs, expectations, and constraints; operational concept; projected utilization environments for people, products, services, and processes; and measures of effectiveness</li> </ol>
requirements elicitation	1. the process through which the acquirers (customers or users) and the suppliers (contractor) of a system discover, review, articulate, understand, and document the users' needs and the constraints on the system and the development activity.

	2. the use of systematic techniques, such as prototypes and structured surveys, to proactively identify and document customer and end-user needs
requirements specification	1. a document that specifies the requirements for a system or component cf. design description, functional specification, performance specification. NOTE: Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.
requirements traceability	1. the identification and documentation of the derivation path (upward) and allocation/flow-down path (downward) of requirements in the requirements hierarchy. 2. discernable association between a requirement and related requirements, implementations, and verifications
requirements traceability matrix (RTM)	1. a table that links requirements to their origin and traces them throughout the project life cycle.
resource	1. skilled human resources (specific disciplines either individually or in crews or teams), equipment, services, supplies, commodities, materiel, budgets, or funds. 2. asset that is utilized or consumed during the execution of a process. 3. a role (with respect to that action) in which the enterprise object fulfilling the role is essential to the action, requires allocation, or may become unavailable. 4. an enterprise object which is essential to some behavior and which requires allocation or may become unavailable. 5. people, procedure, software, information, equipment, consumable, infrastructure, capital and operating funds, and time. EXAMPLE: diverse entities such as funding, personnel, facilities, capital equipment, tools, and utilities such as power, water, fuel and communication infrastructures. NOTE: Allocation of a resource may constrain other behaviors for which that resource is essential. Resources may be reusable, renewable or consumable. A consumable resource may become unavailable after some amount of use or after some amount of time (in case a duration or expiry has been specified for the resource).
resource breakdown structure (RBS)	1. a hierarchical structure of resources by resource category and resource type used in resource leveling schedules and to develop resource-limited schedules, and which may be used to identify and analyze project human resource assignments.
responsibility assignment matrix (RAM)	1. [Tool] a structure that relates the project organizational breakdown structure to the work breakdown structure to help ensure that each component of the project's scope of work is assigned to a person or team.
risk analysis	1. the process of examining identified risk factors for probability of occurrence, potential loss, and potential risk-handling strategies risk category.
risk category	1. a class or type of risk. 2. a group of potential causes of risk. Risk causes may be grouped into categories such as technical, external, organizational, environmental, or project management. A category may include subcategories such as technical maturity, weather, or aggressive estimating. EXAMPLE: technical, legal, organizational, safety, economic, engineering, cost, schedule NOTE: A risk category is a characterization of a source of risk.
risk criteria	1. the terms of reference by which the significance of risk is assessed. NOTE Risk criteria can include associated cost and benefits, legal and statutory requirements, socio-economic and environmental aspects, the concerns of stakeholders, priorities and other inputs to the assessment.
risk dimension	1. a perspective from which risk assessment is being made for the system. EXAMPLE safety, economic, security

risk handling	1. a course of action taken in response to a risk factor NOTE includes risk acceptance, risk avoidance, risk transfer, and risk mitigation
risk identification	1. an organized, systematic approach to determining the risk factors associated with a planned activity, project, or program cf. identify risks
risk management	<p>1. an organized process for identifying and handling risk factors.</p> <p>2. organized, analytic process to identify what might cause harm or loss (identify risks); to assess and quantify the identified risks; and to develop and, if needed, implement an appropriate approach to prevent or handle causes of risk that could result in significant harm or loss</p> <p>3. coordinated activities to direct and control an organization with regard to risk.</p> <p>NOTE The primary goal of risk management is to identify and respond to potential problems with sufficient lead-time to avoid a crisis situation. Includes initial identification and handling of risk factors as well as continuous risk management.</p>
risk management plan	<p>1. a description of how the elements and resources of the risk management process will be implemented within an organization or project.</p> <p>2. [Output/Input]. The document describing how project risk management will be structured and performed on the project. It is contained in or is a subsidiary plan of the project management plan. Information in the risk management plan varies by application area and project size. The risk management plan is different from the risk register that contains the list of project risks, the results of risk analysis, and the risk responses.</p>
risk management process	1. a continuous process for systematically identifying, analyzing, treating, and monitoring risk throughout the life cycle of a product or service.
risk management system	<p>1. set of elements of an organization's management system concerned with managing risk.</p> <p>NOTE: Management system elements can include strategic planning, decision making, and other processes for dealing with risk. The culture of an organization is reflected in its risk management system. [ISO Guide 73:2002]</p>
risk metric	1. an objective measure associated with a risk factor to be mitigated
risk mitigation	<p>1. a course of action taken to reduce the probability of and potential loss from a risk factor.</p> <p>2. [Technique] a risk response planning technique associated with threats that seeks to reduce the probability of occurrence or impact of a risk to below an acceptable threshold.</p> <p>NOTE includes executing contingency plans when a risk metric crosses a predetermined threshold (when a risk factor becomes a problem).</p>
risk register	1. [Output/Input] the document containing the results of the qualitative risk analysis, quantitative risk analysis, and risk response planning. The risk register details all identified risks, including description, category, cause, probability of occurring, impact(s) on objectives, proposed responses, owners, and current status.
role	<p>1. the participation of an entity in a relationship.</p> <p>2. a defined function to be performed by a project team member, such as testing, filing, inspecting, coding.</p> <p>3. the expression of an object playing a part in a relationship.</p> <p>NOTE: Each instance of a role has a minimum and maximum cardinality, and may be attributed. The direction of the role indicates how to read the name of the role.</p>
scenario	<p>1. a step-by-step description of a series of events that may occur concurrently or sequentially.</p> <p>2. an account or synopsis of a projected course of events or actions.</p> <p>3. a description of a specific sequence of actions. Syn: script, set, suite cf. use case</p>
software design	1. the use of scientific principles, technical information, and imagination in the definition of a software system to perform pre-specified functions with maximum economy and efficiency
software design audit	1. a review of a software product to determine compliance with requirements, standards, and contractual documents
software design concept	1. a fundamental idea (such as information hiding) that can be applied to designing a system

software design description (SDD)	<ol style="list-style-type: none"> <li>1. a representation of software created to facilitate analysis, planning, implementation, and decision-making.</li> </ol> <p>NOTE: The software design description is used as a medium for communicating software design information and may be thought of as a blueprint or model of the system.</p>
stakeholder	<ol style="list-style-type: none"> <li>1. individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations.</li> <li>2. individual, group or organization that can affect, be affected by, or perceive itself to be affected by, a risk.</li> <li>3. individual, group, or organization who may affect, be affected by, or perceive itself to be affected by a decision or activity.</li> <li>4. person or organization (e.g., customer, sponsor, performing organization, or the public) that is actively involved in the project, or whose interests may be positively or negatively affected by execution or completion of the project. A stakeholder may also exert influence over the project and its deliverables.</li> </ol> <p>EXAMPLE: end users, end user organizations, supporters, developers, producers, trainers, maintainers, disposers, acquirers, supplier organizations and regulatory bodies.</p> <p>NOTE: The decision-maker is also a stakeholder</p>
state diagram	<ol style="list-style-type: none"> <li>1. a diagram that depicts the states that a system or component can assume, and shows the events or circumstances that cause or result from a change from one state to another</li> </ol>
system	<ol style="list-style-type: none"> <li>1. combination of interacting elements organized to achieve one or more stated purposes.</li> <li>2. an interdependent group of people, objects, and procedures constituted to achieve defined objectives or some operational role by performing specified functions.</li> <li>3. a collection of interacting components organized to accomplish a specific function or set of functions within a specific environment.</li> <li>4. an interacting combination of elements to accomplish a defined objective.</li> <li>5. a set or arrangement of elements that are related, and whose behavior satisfies operational needs and provides for the life cycle sustainment of the products.</li> <li>6. a conceptual entity defined by its boundaries.</li> <li>7. set of interrelated or interacting elements.</li> </ol> <p>NOTE: A system may be considered as a product or as the services it provides. In practice, the interpretation of its meaning is frequently clarified by the use of an associative noun, e.g., aircraft system. Alternatively, the word 'system' may be substituted simply by a context-dependent synonym, e.g., aircraft, though this may then obscure a system principles perspective. A complete system includes all of the associated equipment, facilities, material, computer programs, firmware, technical documentation, services, and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment.</p>
system analysis	<ol style="list-style-type: none"> <li>1. a systematic investigation of a real or planned system to determine the information requirements and processes of the system and how these relate to each other and to any other system.</li> </ol> <p>Syn: systems analysis.</p>
system architecture	<ol style="list-style-type: none"> <li>1. the composite of the design architectures for products and their life cycle processes.</li> </ol>
system breakdown structure (SBS)	<ol style="list-style-type: none"> <li>1. a hierarchy of elements, related life cycle processes, and personnel used to assign development teams, conduct technical reviews, and to partition out the assigned work and associated resource allocations to each of the tasks necessary to accomplish the objectives of the project.</li> </ol> <p>NOTE: It also provides the basis for cost tracking and control.</p>
system description	<ol style="list-style-type: none"> <li>1. documentation that results from system design defining the organization, essential characteristics and the hardware and software requirements of the system.</li> </ol>
system design	<ol style="list-style-type: none"> <li>1. a process of defining the hardware and Software architecture, components, modules, interfaces and data for a system to satisfy specified requirements.</li> </ol>

system development	1. a process that usually includes requirements analysis, system design, implementation, documentation and quality assurance.
system development cycle	1. the period of time that begins with the decision to develop a system and ends when the system is delivered to its end user cf. system life cycle software development cycle. NOTE: This term is sometimes used to mean a longer period of time, either the period that ends when the system is no longer being enhanced, or the entire system life cycle.
system documentation	1. the collection of documents that describe the requirements, capabilities, limitations, design, operation, and maintenance of an information processing system.
systems engineering	1. interdisciplinary approach governing the total technical and managerial effort required to transform a set of customer needs, expectations, and constraints into a solution and to support that solution throughout its life. cf. hardware engineering, software engineering NOTE includes the definition of technical performance measures; the integration of engineering specialties toward the establishment of an architecture; and the definition of supporting lifecycle processes that balance cost, performance, and schedule objectives.
task	1. required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process. 2. in software design, a software component that can operate in parallel with other software components. 3. the activities required to achieve a goal. 4. a concurrent object with its own thread of control. 5. a sequence of instructions treated as a basic unit of work by the supervisory program of an operating system. 6. smallest unit of work subject to management accountability; a well-defined work assignment for one or more project members. NOTE: Related tasks are usually grouped to form activities.
task behavior specification	1. a specification describing a concurrent task's interface, structure, timing characteristics, relative priority, errors detected, and task event sequencing logic.
task completion	1. timely event when for a specific task the total output string or, in case of a set of output strings, all parts are completely received by to the emulated user or another instance. NOTE: The time of task completion defines the end time of the preceding preparation time and the begin time of the execution time of the following task.
technique	1. a defined systematic procedure employed by a human resource to perform an activity to produce a product or result or deliver a service, and that may employ one or more tools. 2. methods and skills required to carry out a specific activity. 3. technical or managerial procedure that aids in the evaluation and improvement of the software development process
technology viewpoint	1. a viewpoint on an ODP system and its environment that focuses on the choice of technology in that system.
Thing (as in IoT)	1. Generally speaking, any physical object in combination with its digital representation. In other words, it denotes the same concept as an Augmented Entity.
timing diagram	1. a diagram showing the time-ordered execution sequence of a group of tasks.
Unified Modeling Language (UML)	1. a graphical language for visualizing, specifying, constructing, and documenting an object-oriented software-intensive system's artifacts
use case	1. in UML, a complete task of a system that provides a measurable result of value for an actor NOTE More formally, a use case defines a set of use case instances or scenarios.
use case diagram	1. a UML diagram that shows actors, use cases, and their relationships



use case model	1. a model that describes a system's functional requirements in terms of use cases
use case specification	1. a document that describes a use case. NOTE: A use case specification's fundamental parts are the use case name, brief description, precondition, basic flow, post-condition, and alternate flow.
user	1. person who performs one or more tasks with software; a member of a specific audience. 2. person who interacts with the product. 3. individual or organization who uses a software-intensive system in daily work activities or recreational pursuits. 4. individual or group that benefits from a system during its utilization. 5. any person or thing that communicates or interacts with the software at any time. 6. a person (or instance) who uses the functions of a CBSS via a terminal (or an equivalent machine-user-interface) by submitting tasks and receiving the computed results. 7. the person who derives engineering value through interaction with a CASE tool. cf. developer NOTE: The user may perform other roles such as acquirer or maintainer. The role of user and the role of operator may be vested, simultaneously or sequentially, in the same individual or organization.
user view	1. a formal representation of the user's business needs in the user's language. 2. the application as seen through the eyes of the user. NOTE: Developers translate the user information into information technology language in order to provide a solution.
verb phrase	1. a part of the label of a relationship that names the relationship in a way that a sentence can be formed by combining the first class name, the verb phrase, the cardinality expression, and the second class name or role name. 2. a phrase used to name a relationship, which consists of a verb and words that constitute the object of the phrase. EXAMPLE: The statement "each project funds one or more tasks" could be derived from a relationship showing "project" as the first class, "task" as the second class with a "one or more" cardinality, and "funds" as the verb phrase. NOTE: A verb phrase is ideally stated in active voice.
view	1. a developer's copy of a branch. 2. a collection of subject domains, classes, relationships, responsibilities, properties, constraints, and notes assembled or created for a certain purpose and covering a certain scope. 3. a collection of entities and assigned attributes (domains) assembled for some purpose. 4. a set of related categories. 5. a representation of a whole system from the perspective of a related set of concerns. NOTE: A view may cover the entire area being modeled or only a part of that area.
view diagram	1. a graphic representation of the underlying semantics of a view.
viewpoint	1. a specification of the conventions for constructing and using a view. NOTE: A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.
viewpoint (on a system)	1. a form of abstraction achieved using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within a system.
viewpoint statement	1. a brief statement of the perspective of an IDEF0 model that is presented in the a-0 context diagram of the model.
virtual entity	1. Computational or data element representing a Physical Entity. Virtual Entities can be either Active or Passive Digital Artefacts