

PMI Materijali

PMBok Rječnik

ability to perform - (See *common features*.)

acceptance criteria - The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. [IEEE-STD-610]

acceptance testing - Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. [IEEE-STD-610]

activity - Any step taken or function performed, both mental and physical, toward achieving some objective. Activities include all the work the managers and technical staff do to perform the tasks of the project and organization. (See *task* for contrast.)

activities performed - (See *common features*.)

action item- (1) A unit in a list that has been assigned to an individual or group for disposition. (2) An action proposal that has been accepted.

action proposal- A documented suggestion for change to a process or process-related item that will prevent the future occurrence of defects identified as a result of defect prevention activities. (See also *software process improvement proposal*.)

allocated requirements - (See *system requirements allocated to software*.)

application domain - A bounded set of related systems (i.e., systems that address a particular type of problem). Development and maintenance in an application domain usually requires special skills and/or resources. Examples include payroll and personnel systems, command and control systems, compilers, and expert systems.

assessment - (See *software process assessment*.)

audit - An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria. [IEEE-STD-610]

baseline - A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. [IEEE-STD-610]

baseline configuration management - The establishment of baselines that are formally reviewed and agreed on and serve as the basis for further development. Some software work products, e.g., the software design and the code, should have baselines established at predetermined points, and a rigorous change control process should be applied these items. These baselines provide control and stability when interacting with the customer. (See also *baseline management*.)

baseline management - In configuration management, the application of technical and administrative direction to designate the documents and changes to those documents that formally identify and establish baselines at specific times during the life cycle of a configuration item. [IEEE-STD-610]

benchmark - A standard against which measurements or comparisons can be made. [IEEE-STD-610]

bidder - An individual, partnership, corporation, or association that has submitted a proposal and is a candidate to be awarded a contract to design, develop, and/or manufacture one or more products.

capability maturity model - A description of the stages through which software organizations evolve as they define, implement, measure, control, and improve their software processes. This model provides a guide for selecting process improvement strategies by facilitating the determination of current process capabilities and the identification of the issues most critical to software quality and process improvement.

causal analysis - The analysis of defects to determine their underlying root cause.

causal analysis meeting - A meeting, conducted after completing a specific task, to analyze defects uncovered during the performance of that task.

CMM - Acronym for *capability maturity model*.

commitment - A pact that is freely assumed, visible, and expected to be kept by all parties.

commitment to perform - (See *common features*.)

common cause (of a defect) - A cause of a defect that is inherently part of a process or system. Common causes affect every outcome of the process and everyone working in the process. (See *special cause* for contrast.)

common features - The subdivision categories of the CMM key process areas. The common features are attributes that indicate whether the implementation and institutionalization of a key process area is effective, repeatable, and lasting. The CMM common features are the following:

- *commitment to perform* - The actions the organization must take to ensure that the process is established and will endure. Commitment to Perform typically involves establishing organizational policies and senior management sponsorship.
- *ability to perform* - The preconditions that must exist in the project or organization to implement the software process competently. Ability to Perform typically involves resources, organizational structures, and training.
- *activities performed* - A description of the roles and procedures necessary to implement a key process area. Activities Performed typically involve establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary.
- *measurement and analysis* - A description of the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed.

- *verifying implementation* - The steps to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance.

configuration - In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product. [IEEE-STD-610]

configuration control - An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. [IEEE-STD-610]

configuration identification - An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation. [IEEE-STD-610]

configuration item - An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process. [IEEE-STD-610]

configuration management - A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. [IEEE-STD-610]

configuration management library system - The tools and procedures to access the contents of the software baseline library.

configuration unit - The lowest level entity of a configuration item or component that can be placed into, and retrieved from, a configuration management library system.

consistency - The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of system or component. [IEEE-STD-610]

contingency factor - An adjustment (increase) of a size, cost, or schedule plan to account for likely underestimates of these parameters due to incomplete specification, inexperience in estimating the application domain, etc.

contract terms and conditions - The stated legal, financial, and administrative aspects of a contract.

critical computer resource - The parameters of the computing resources deemed to be a source of risk to the project because the potential need for those resources may exceed the amount that is available. Examples include target computer memory and host computer disk space.

critical path - A series of dependent tasks for a project that must be completed as planned to keep the entire project on schedule.

customer - The individual or organization that is responsible for accepting the product and authorizing payment to the developing organization.

defect - A flaw in a system or system component that causes the system or component to fail to perform its required function. A defect, if encountered during execution, may cause a failure of the system.

defect density - The number of defects identified in a product divided by the size of the product component (expressed in standard measurement terms for that product).

defect prevention - The activities involved in identifying defects or potential defects and preventing them from being introduced into a product.

defect root cause - The underlying reason (e.g., process deficiency) that allowed a defect to be introduced.

defined level - (See *maturity level*.)

defined software process - (See *project's defined software process*.)

dependency item - A product, action, piece of information, etc., that must be provided by one individual or group to a second individual or group so that the second individual or group can perform a planned task.

developmental configuration management - The application of technical and administrative direction to designate and control software and associated technical documentation that define the evolving configuration of a software work product during development. Developmental configuration management is under the direct control of the developer. Items under developmental configuration management are not baselines, although they may be baselined and placed under baseline configuration management at some point in their development.

deviation - A noticeable or marked departure from the appropriate norm, plan, standard, procedure, or variable being reviewed.

documented procedure - (See *procedure*.)

effective process - A process that can be characterized as practiced, documented, enforced, trained, measured, and able to improve. (See also *well-defined process*.)

end user - The individual or group who will use the system for its intended operational use when it is deployed in its environment.

end user representatives - A selected sample of end users who represent the total population of end users.

engineering group - A collection of individuals (both managers and technical staff) representing an engineering discipline. Examples of engineering disciplines include systems engineering, hardware engineering, system test, software engineering, software configuration management, and software quality assurance.

evaluation - (See *software capability evaluation*.)

event-driven review/activity - A review or activity that is performed based on the occurrence of an event within the project (e.g., a formal review or the completion of a life cycle stage). (See *periodic review/activity* for contrast.)

findings - The conclusions of an assessment, evaluation, audit, or review that identify the most important issues, problems, or opportunities within the area of investigation.

first-line software manager - A manager who has direct management responsibility (including providing technical direction and administering the personnel and salary functions) for the staffing and activities of a single organizational unit (e.g., a department or project team) of software engineers and other related staff.

formal review - A formal meeting at which a product is presented to the end user, customer, or other interested parties for comment and approval. It can also be a review of the management and technical activities and of the progress of the project.

function - A set of related actions, undertaken by individuals or tools that are specifically assigned or fitted for their roles, to accomplish a set purpose or end.

goals - A summary of the key practices of a key process area that can be used to determine whether an organization or project has effectively implemented the key process area. The goals signify the scope, boundaries, and intent of each key process area.

group - The collection of departments, managers, and individuals who have responsibility for a set of tasks or activities. A group could vary from a single individual assigned part time, to several part-time individuals assigned from different departments, to several individuals dedicated full time.

host computer - A computer used to develop software. (See *target computer* for contrast.)

initial level - (See *maturity level*.)

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

integrated software management - The unification and integration of the software engineering and management activities into a coherent defined software process based on the organization's standard software process and related process assets.

integration - (See *software integration*.)

key practices - The infrastructures and activities that contribute most to the effective implementation and institutionalization of a key process area.

key process area - A cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability. The key process areas have been defined to reside at a single maturity level. They are the areas identified by the SEI to be the principal building blocks to help determine the software process capability of an organization and understand the improvements needed to advance to higher maturity levels. The Level 2 key process areas in the CMM are Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management. The Level 3 key process areas in the CMM are Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, and Peer Reviews. The Level 4 key process areas are Quantitative Process Management and Software Quality Management. The Level 5 key process areas are Defect Prevention, Technology Change Management, and Process Change Management.

life cycle - (See *software life cycle*.)

maintenance - The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment. [IEEE-STD-610]

managed and controlled - The process of identifying and defining software work products that are not part of a baseline and, therefore, are not placed under configuration management but that must be controlled for the project to proceed in a disciplined manner. "Managed and controlled" implies that the version of the work product in use at a given time (past or present) is known (i.e., version control), and changes are incorporated in a controlled manner (i.e., change control).

managed level - (See *maturity level*.)

manager - A role that encompasses providing technical and administrative direction and control to individuals performing tasks or activities within the manager's area of responsibility. The traditional functions of a manager include planning, resourcing, organizing, directing, and controlling work within an area of responsibility.

maturity level - A well-defined evolutionary plateau toward achieving a mature software process. The five maturity levels in the SEI's Capability Maturity Model are:

- *initial* - The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.
- *repeatable* - Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
- *defined* - The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.
- *managed* - Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.
- *optimizing* - Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

maturity questionnaire - A set of questions about the software process that sample the key practices in each key process area of the CMM. The maturity questionnaire is used as a springboard to appraise the capability of an organization or project to execute a software process reliably.

measure - A unit of measurement (such as source lines of code or document pages of design).

measurement - The dimension, capacity, quantity, or amount of something (e.g., 300 source lines of code or 7 document pages of design).

method - A reasonably complete set of rules and criteria that establish a precise and repeatable way of performing a task and arriving at a desired result.

methodology - A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development of a product.

milestone - A scheduled event for which some individual is accountable and that is used to measure progress.

nontechnical requirements - Agreements, conditions and/or contractual terms that affect and determine the management activities of a software project.

operational software - The software that is intended to be used and operated in a system when it is delivered to its customer and deployed in its intended environment.

optimizing level - (See *maturity level*.)

organization - A unit within a company or other entity (e.g., government agency or branch of service) within which many projects are managed as a whole. All projects within an organization share a common top-level manager and common policies.

organization's measurement program - The set of related elements for addressing an organization's measurement needs. It includes the definition of organization-wide measurements, methods and practices for collecting organizational measurement data, methods and practices for analyzing organizational measurement data, and measurement goals for the organization.

organization's software process assets - A collection of entities, maintained by an organization, for use by projects in developing, tailoring, maintaining, and implementing their software processes. These software process assets typically include:

- the organization's standard software process,
- descriptions of the software life cycles approved for use,
- the guidelines and criteria for tailoring the organization's standard software process,
- the organization's software process database, and
- a library of software process-related documentation.

Any entity that the organization considers useful in performing the activities of process definition and maintenance could be included as a process asset.

organization's software process database - A database established to collect and make available data on the software processes and resulting software work products, particularly as they relate to the organization's standard software process. The database contains or references both the actual measurement data and the related information needed to understand the measurement data and assess it for reasonableness and applicability. Examples of process and work product data include estimates of software size, effort, and cost; actual data on software size, effort, and cost; productivity data; peer review coverage and efficiency; and number and severity of defects found in the software code.

organization's standard software process - The operational definition of the basic process that guides the establishment of a common software process across the software projects in an organization. It describes the fundamental software process elements that each software project is expected to incorporate into its defined software process. It also describes the relationships (e.g., ordering and interfaces) between these software process elements.

orientation - An overview or introduction to a topic for those overseeing or interfacing with the individuals responsible for performing in the topic area. (See *train* for contrast.)

Pareto analysis - The analysis of defects by ranking causes from most significant to least significant. Pareto analysis is based on the principle, named after the 19th-century economist Vilfredo Pareto, that most effects come from relatively few causes, i.e., 80% of the effects come from 20% of the possible causes.

peer review - A review of a software work product, following defined procedures, by peers of the producers of the product for the purpose of identifying defects and improvements.

peer review leader - An individual specifically trained and qualified to plan, organize, and lead a peer review.

periodic review/activity - A review or activity that occurs at specified regular time intervals. (See *event-driven review/activity* for contrast.)

policy - A guiding principle, typically established by senior management, which is adopted by an organization or project to influence and determine decisions.

prime contractor - An individual, partnership, corporation, or association that administers a subcontract to design, develop, and/or manufacture one or more products.

procedure - A written description of a course of action to be taken to perform a given task. [IEEE-STD-610]

process - A sequence of steps performed for a given purpose; for example, the software development process. [IEEE-STD-610]

process capability - The range of expected results that can be achieved by following a process. (See *process performance* for contrast.)

process capability baseline - A documented characterization of the range of expected results that would normally be achieved by following a specific process under typical circumstances. A process capability baseline is typically established at an organizational level. (See *process performance baseline* for contrast.)

process database - (See *organization's software process database*.)

process description - The operational definition of the major components of a process. Documentation that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a process. It may also include the procedures for determining whether these provisions have been satisfied. Process descriptions may be found at the task, project, or organizational level.

process development - The act of defining and describing a process. It may include planning, architecture, design, implementation, and validation.

process measurement - The set of definitions, methods, and activities used to take measurements of a process and its resulting products for the purpose of characterizing and understanding the process.

process performance - A measure of the actual results achieved by following a process. (See *process capability* for contrast.)

process performance baseline - A documented characterization of the actual results achieved by following a process, which is used as a benchmark for comparing actual process performance against expected process performance. A process performance baseline is typically established at the project level, although the initial process performance baseline will usually be derived from the process capability baseline. (See *process capability baseline* for contrast.)

process tailoring - The activity of creating a process description by elaborating, adapting, and/or completing the details of process elements or other incomplete specifications of a process. Specific business needs for a project will usually be addressed during process tailoring.

product - (See *software product* and *software work product*.)

profile - A comparison, usually in graphical form, of plans or projections versus actuals, typically over time.

project - An undertaking requiring concerted effort, which is focused on developing and/or maintaining a specific product. The product may include hardware, software, and other components. Typically a project has its own funding, cost accounting, and delivery schedule.

project's defined software process - The operational definition of the software process used by a project. The project's defined software process is a well-characterized and understood software process, described in terms of software standards, procedures, tools, and methods. It is developed by tailoring the organization's standard software process to fit the specific characteristics of the project. (See also *organization's standard software process*, *effective process*, and *well-defined process*.)

project manager - The role with total business responsibility for an entire project; the individual who directs, controls, administers, and regulates a project building a software or hardware/software system. The project manager is the individual ultimately responsible to the customer.

project software manager - The role with total responsibility for all the software activities for a project. The project software manager is the individual the project manager deals with in terms of software commitments and who controls all the software resources for a project.

quality - (1) The degree to which a system, component, or process meets specified requirements. (2) The degree to which a system, component, or process meets customer or user needs or expectations. [IEEE-STD-610]

quality assurance - (See *software quality assurance*.)

quantitative control - Any quantitative or statistically-based technique appropriate to analyze a software process, identify special causes of variations in the performance of the software process, and bring the performance of the software process within well-defined limits.

repeatable level - (See *maturity level*.)

required training - Training designated by an organization to be required to perform a specific role.

risk - Possibility of suffering loss.

risk management - An approach to problem analysis which weighs risk in a situation by using risk probabilities to give a more accurate understanding of the risks involved. Risk management includes risk identification, analysis, prioritization, and control.

risk management plan - The collection of plans that describe the risk management activities to be performed on a project.

role - A unit of defined responsibilities that may be assumed by one or more individuals.

SCE - Acronym for *software capability evaluation*.

SCM - Acronym for *software configuration management*.

senior manager - A management role at a high enough level in an organization that the primary focus is the long-term vitality of the organization, rather than short-term project and contractual concerns and pressures. In general, a senior manager for engineering would have responsibility for multiple projects.

software architecture - The organizational structure of the software or module. [IEEE-STD-610]

software baseline audit - An examination of the structure, contents, and facilities of the software baseline library to verify that baselines conform to the documentation that describes the baselines.

1. software baseline library - The contents of a repository for storing configuration items and the associated records.

software build - An operational version of a software system or component that incorporates a specified subset of the capabilities the final software system or component will provide. [IEEE-STD-610]

software capability evaluation - An appraisal by a trained team of professionals to identify contractors who are qualified to perform the software work or to monitor the state of the software process used on an existing software effort.

software configuration control board - A group responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes.

software development plan - The collection of plans that describe the activities to be performed for the software project. It governs the management of the activities performed by the software engineering group for a software project. It is not limited to the scope of any particular planning standard, such as DOD-STD-2167A and IEEE-STD-1058, which may use similar terminology.

software engineering group - The collection of individuals (both managers and technical staff) who have responsibility for software development and maintenance activities (i.e., requirements analysis, design, code, and test) for a project. Groups performing software-related work, such as the software quality assurance group, the software configuration management group, and the software engineering process group, are not included in the software engineering group.

software engineering process group - A group of specialists who facilitate the definition, maintenance, and improvement of the software process used by the organization. In the key practices, this group is generically referred to as "the group responsible for the organization's software process activities."

software engineering staff - The software technical people (e.g., analysts, programmers, and engineers), including software task leaders, who perform the software development and maintenance activities for the project, but who are not managers.

software integration - A process of putting together selected software components to provide the set or specified subset of the capabilities the final software system will provide.

software life cycle - The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. [IEEE-STD-610]

software manager - Any manager, at a project or organizational level, who has direct responsibility for software development and/or maintenance.

software plans - The collection of plans, both formal and informal, used to express how software development and/or maintenance activities will be performed. Examples of plans that could be included: software development plan, software quality assurance plan, software configuration management plan, software test plan, risk management plan, and process improvement plan.

software process - A set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products (e.g., project plans, design documents, code, test cases, and user manuals).

software process assessment - An appraisal by a trained team of software professionals to determine the state of an organization's current software process, to determine the high-priority software process-related issues facing an organization, and to obtain the organizational support for software process improvement.

software process assets - (See *organization's software process assets*.)

software process capability - (See *process capability*.)

software process description - The operational definition of a major software process component identified in the project's defined software process or the organization's standard software process. It documents, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a software process. (See also *process description*.)

software process element - A constituent element of a software process description. Each process element covers a well-defined, bounded, closely related set of tasks (e.g., software estimating element, software design element, coding element, and peer review element). The descriptions of the process elements may be templates to be filled in, fragments to be completed, abstractions to be refined, or complete descriptions to be modified or used unmodified.

software process improvement plan - A plan, derived from the recommendations of a software process assessment, that identifies the specific actions that will be taken to improve the software process and outlines the plans for implementing those actions. Sometimes referred to as an action plan.

software process improvement proposal - A documented suggestion for change to a process or process-related item that will improve software process capability and performance. (See also *action proposal*.)

software process maturity - The extent to which a specific process is explicitly defined, managed, measured, controlled, and effective. Maturity implies a potential for growth in capability and indicates both the richness of an organization's software process and the consistency with which it is applied in projects throughout the organization.

software process performance - (See *process performance*.)

software process-related documentation - Example documents and document fragments, which are expected to be of use to future projects when they are tailoring the organization's standard software process. The examples may cover subjects such as a project's defined software process, standards, procedures, software development plans, measurement plans, and process training materials.

software product - The complete set, or any of the individual items of the set, of computer programs, procedures, and associated documentation and data designated for delivery to a customer or end user. [IEEE-STD-610] (See *software work product* for contrast.)

software project - An undertaking requiring concerted effort, which is focused on analyzing, specifying, designing, developing, testing, and/or maintaining the software components and associated documentation of a system. A software project may be part of a project building a hardware/software system.

software quality assurance - (1) A planned and systematic pattern of all actions necessary to provide adequate confidence that a software work product conforms to established technical requirements. (2) A set of activities designed to evaluate the process by which software work products are developed and/or maintained.

software quality goal - Quantitative quality objectives defined for a software work product.

software quality management - The process of defining quality goals for a software product, establishing plans to achieve these goals, and monitoring and adjusting the software plans, software work products, activities, and quality goals to satisfy the needs and desires of the customer and end users.

software-related group - A collection of individuals (both managers and technical staff) representing a software engineering discipline that supports, but is not directly responsible for, software development and/or maintenance. Examples of software engineering disciplines include software quality assurance and software configuration management.

software requirement - A condition or capability that must be met by software needed by a user to solve a problem or achieve an objective. [IEEE-STD-610]

software work product - Any artifact created as part of defining, maintaining, or using a software process, including process descriptions, plans, procedures, computer programs, and associated documentation, which may or may not be intended for delivery to a customer or end user. (See *software product* for contrast.)

SPA - Acronym for *software process assessment*.

special cause (of a defect) - A cause of a defect that is specific to some transient circumstance and not an inherent part of a process. Special causes provide random variation (noise) in process performance. (See *common cause* for contrast.)

SQA - Acronym for *software quality assurance*.

staff - The individuals, including task leaders, who are responsible for accomplishing an assigned function, such as software development or software configuration management, but who are not managers.

stage - A partition of the software effort that is of a manageable size and that represents a meaningful and measurable set of related tasks which are performed by the project. A stage is usually considered a subdivision of a software life cycle and is often ended with a formal review prior to the onset of the following stage.

standard - Mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development.

standard software process - (See *organization's standard software process*.)

statement of work - A description of all the work required to complete a project, which is provided by the customer.

subcontract manager - A manager in the prime contractor's organization who has direct responsibility for administering and managing one or more subcontracts.

subcontractor - An individual, partnership, corporation, or association that contracts with an organization (i.e., the prime contractor) to design, develop, and/or manufacture one or more products.

system - A collection of components organized to accomplish a specific function or set of functions. [IEEE-STD-610]

system engineering group - The collection of individuals (both managers and technical staff) who have responsibility for specifying the system requirements; allocating the system requirements to the hardware, software, and other components; specifying the interfaces between the hardware, software, and other components; and monitoring the design and development of these components to ensure conformance with their specifications.

system requirement - A condition or capability that must be met or possessed by a system or system

component to satisfy a condition or capability needed by a user to solve a problem. [IEEE-STD-610]

system requirements allocated to software - The subset of the system requirements that are to be implemented in the software components of the system. The allocated requirements are a primary input to the software development plan. Software requirements analysis elaborates and refines the allocated requirements and results in software requirements which are documented.

tailor - To modify a process, standard, or procedure to better match process or product requirements.

target computer - The computer on which delivered software is intended to operate. (See *host computer* for contrast.)

task - (1) A sequence of instructions treated as a basic unit of work. [IEEE-STD-610] (2) A well-defined unit of work in the software process that provides management with a visible checkpoint into the status of the project. Tasks have readiness criteria (preconditions) and completion criteria (postconditions). (See *activity* for contrast.)

task kick-off meeting - A meeting held at the beginning of a task of a project for the purpose of preparing the individuals involved to perform the activities of that task effectively.

task leader - The leader of a technical team for a specific task, who has technical responsibility and provides technical direction to the staff working on the task.

team - A collection of people, often drawn from diverse but related groups, assigned to perform a well-defined function for an organization or a project. Team members may be part-time participants of the team and have other primary responsibilities.

testability - (1) The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met. (2) The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met. [IEEE-STD-610]

technical requirements - Those requirements that describe what the software must do and its operational constraints. Examples of technical requirements include functional, performance, interface, and quality requirements.

technology - The application of science and/or engineering in accomplishing some particular result.

traceability - The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another. [IEEE-STD-610]

train - To make proficient with specialized instruction and practice. (See also *orientation*.)

training group - The collection of individuals (both managers and staff) who are responsible for coordinating and arranging the training activities for an organization. This group typically prepares and conducts most of the training courses and coordinates use of other training vehicles.

training program - The set of related elements that focus on addressing an organization's training needs. It includes an organization's training plan, training materials, development of training, conduct of training, training facilities, evaluation of training, and maintenance of training records.

training waiver - A written approval exempting an individual from training that has been designated as required for a specific role. The exemption is granted because it has been objectively determined that the individual already possesses the needed skills to perform the role.

unit - (1) A separately testable element specified in the design of a computer software component. (2) A logically separable part of a computer program. (3) A software component that is not subdivided into other components. [IEEE-STD-610]

user- (See *end user*.)

validation- The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. [IEEE-STD-610]

verification- The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]

verifying implementation - (See *common features*.)

waiver - (See *training waiver*).

well-defined process - A process that includes readiness criteria, inputs, standards and procedures for performing the work, verification mechanisms (such as peer reviews), outputs, and completion criteria. (See also *effective process*.)