hardware. (1) physical equipment used to process, store, or transmit computer programs or data *(ISO/IEC 19770-1:2017 Information technology -- IT asset management -- Part 1: IT asset management systems--Requirements, 3.21)* **(2)** all or part of the physical components of an information system *(ISO/IEC 2382:2015 Information technology -- Vocabulary) Syn:* HW *See also:* software

firmware. (1) combination of a hardware device and computer instructions and data that reside as read-only software on that device *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)* **(2)** ordered set of instructions and associated data stored in a way that is functionally independent of main storage, usually in a ROM *(ISO/IEC 2382:2015 Information technology -- Vocabulary) Note:* The software cannot be readily modified under program control.

non-deliverable item. (1) hardware or software product that is not required to be delivered under the contract, but may be employed in the development of a product *(ISO/IEC/IEEE 24765g:2018) Syn:* nondeliverable item

qualification testing. (1) testing conducted to determine whether a system or component is suitable for operational use *(ISO/IEC/IEEE 24765:2017 Systems and software engineering--Vocabulary)* **(2)** testing conducted on a hardware element, software element, or system to evaluate conformance with specified requirements *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1) See also:* acceptance testing, development testing, operational testing

security. (1) protection against intentional subversion or forced failure *(ISO/IEC/IEEE 15288:2023 Systems and software engineering--System life cycle processes, 3.41)* **(2)** protection against intentional subversion or forced failure, containing a composite of four attributes: confidentiality, integrity, availability, and accountability, plus aspects of a fifth, usability, all of which have the related issue of their assurance *(ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.49)* **(3)** resistance to intentional, unauthorized acts designed to cause harm or damage to a system *(ISO/IEC 23643:2020, Software and systems engineering--Capabilities of software safety and security verification tools, 3.16)* **(4)** protection of computer hardware or software from accidental or malicious access, use, modification, destruction, or disclosure *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)* **(5)** protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)* **(6)** capability of a product to protect information and data so that persons or other products have the degree of data access appropriate to their types and levels of authorization, and to defend against attack patterns by malicious actors *(ISO/IEC 25010:2023, Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--Product quality model, 3.6)* **(7)** combination of people, process, and technology to protect data from unauthorized access and use *(IEEE 7005 2021, IEEE Standard for Transparent Employer Data Governance, 3.1)* **(8)** degree to which an IT service protects both users' assets and access to their information so that users have the degree of information access appropriate to their levels of authorization *(ISO/IEC TS 25011:2017 Information technology--Systems and software Quality Requirements and Evaluation (SQuaRE)--Service quality models, 3.2.3)* **(9)** ability of the system to withstand an attack, whether it is an intrusion, interference, or theft *(ISO/IEC/IEEE 24641:2023. Systems and Software engineering--Methods and tools for*

model-based systems and software engineering, 3.1.31) Note: Security includes authenticity, accountability, confidentiality, integrity, availability, nonrepudiation, and reliability, all of which have the related issue of their assurance. Security pertains to personnel, data, communications, and the physical protection of computer installations.

system. (1) combination of interacting elements organized to achieve one or more stated purposes *(ISO/IEC 25000:2014 Systems and software Engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 4.38) (ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.61)* **(2)** product of an acquisition process that is delivered to the user *(ISO/IEC/IEEE 24748-8:2019, Systems and software engineering Life cycle management Part 7: Application of systems engineering on defense programs, 3.1)* **(3)** something of interest as a whole or as comprised of parts *(ISO/IEC 10746-2:2009 Information technology -- Open Distributed Processing -- Reference Model: Foundations, 6.5)* **(4)** interacting combination of elements to accomplish a defined objective *(ISO/IEC TR 19759:2016 Software Engineering -- Guide to the Software Engineering Body of Knowledge (SWEBOK), 1.1.6)* **(5)** set of interrelated elements considered in a defined context as a whole and separated from their environment *(IEC/IEEE 82079-1:2019 Preparation of information for use (instructions for use) of products: Part 1: Principles and general requirements, 3.41)* **(6)** arrangement of parts or elements that together exhibit a stated behavior or meaning that the individual constituents do not *(ISO/IEC/IEEE 15288:2023 Systems and software engineering--System life cycle processes, 3.46)* **(7)** software or the combination of other components, including hardware and human managed processes *(IEEE 7002:2022, IEEE Standard for Data Privacy Process, 3.1) Note:* A system is sometimes 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' can be replaced by a context-dependent synonym, e.g., aircraft, though this obscures the system 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.

validation. (1) confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled *(ISO/IEC 25000:2014 Systems and software Engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 4.41) (ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.71)* **(2)** process of providing evidence that the system, software, or hardware and its associated products satisfy requirements allocated to it at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, and use the proper system assumptions), and satisfy intended use and user needs *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1.35)* **(3)** assurance that a product, service, or system meets the needs of the customer and other identified stakeholders *(A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) -- Seventh Edition)* **(4)** confirmation in a timely manner, through automated techniques where possible, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled *(IEEE 2675-2021, IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment, 3.1) Note:* Validation in a system life cycle context is the set of activities for gaining confidence that a system is able to accomplish its intended use, goals, and objectives (meet stakeholder requirements) in the intended operational

environment. The right system has been built or is operating to meet business objectives. Validation demonstrates that the system can be used by the users for their specific tasks. "Validated" is used to designate the corresponding status. *See also:* verification

verification. (1) confirmation, through the provision of objective evidence, that specified requirements have been fulfilled *(ISO/IEC 25000:2014 Systems and software Engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 4.43) (ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.72) (ISO/IEC/IEEE 15288:2023 Systems and software engineering--System life cycle processes, 3.55)* **(2)** evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition *(A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) -- Seventh Edition)* **(3)** process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1.36)* **(4)** process of providing objective evidence that the system, software, or hardware and its associated products conform to requirements (e.g., for correctness, completeness, consistency, and accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance), satisfy standards, practices, and conventions during life cycle processes, and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)* **(5)** confirmation in a timely manner, using automated techniques where possible, through the provision of objective evidence, that specified requirements have been fulfilled *(IEEE 2675-2021, IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment, 3.1) Note:* Verification is a set of activities that compares a system or system element to the required characteristics. This can include, but is not limited to, specified requirements, design description, and the system itself. The system has been built right. "Verified" is used to designate the corresponding status. Verification of interim work products is essential for proper understanding and assessment of the life cycle phase product(s). A system can be verified to meet the stated requirements, yet be unsuitable for operation by the actual users. *See also:* validation

anomaly. (1) anything observed in the documentation or operation of a system that deviates from expectations based on previously verified system, software, or hardware products or reference documents *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)* **(2)** judgment that a systems behavior or artifact deviates from expectations based on the systems high-level goals for outcomes, similar requirements, development or operational documentation; applicable standards, regulations, or laws; or an observers perceptions or experiences *(IEEE 982:2024, Standard for Measures of the Software Aspects of Dependability, 3.1) Note:* An anomaly can reveal an implicit requirement.

component testing. (1) testing of individual hardware or software components *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)*

integration testing. (1) testing in which software components, hardware components, or both are combined and tested to evaluate the interaction among them *(ISO/IEC 29110-5-1-2:2025 Systems and software engineering Life cycle profiles for very small entities (VSEs) Part 5-1-2: Software engineering guidelines for the generic Basic profile, 3.16)* **(2)** testing in which software components, hardware components, or both are combined and tested to evaluate the interaction

between them *(ISO/IEC TR 7052:2023, Software engineering--Controlling frequently occurring risks during development and maintenance of custom software, 3.28)*

integrity level. (1) value representing project-unique characteristic, such as complexity, criticality, risk, safety level, security level, desired performance, and reliability, that define the importance of the system, software, or hardware to the user *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1.15)* **(2)** claim of a system, product, or element that includes limitations on a property's values, the claim's scope of applicability, and the allowable uncertainty regarding the claim's achievement *(ISO/IEC/IEEE 15026-1:2019 Systems and software engineering--Systems and software assurance--Part 1: Concepts and vocabulary, 3.3.1)* **(3)** degree of confidence that the system-of-interest meets the associated integrity level claim *(ISO/IEC/IEEE 15026-3:2023, Systems and software engineering--Systems and software assurance--Part 3: System integrity levels, 3.1) Note:* Generally, the intention is that maintaining limitations on a property's values related to the relevant items results in maintaining system risks within limits. The words 'integrity level' form an indivisible label and do not depend on a concept of integrity by itself. An integrity level is different from the likelihood that the integrity level claim is met but they are closely related. The word 'confidence' implies that the definition of integrity levels can be a subjective concept. integrity levels are defined in terms of risk and hence, cover safety, security, financial and any other dimension of risk that is relevant to the system-of-interest.

life cycle processes. (1) set of interrelated or interacting activities that result in the development or assessment of system, software, or hardware products *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1) Note:* Each activity consists of tasks. The life cycle processes can overlap one another. For V&V purposes, no process is concluded until its development products are verified and validated according to the defined tasks in the validation and verification plan. *Syn:* life-cycle processes

minimum tasks. (1) those verification and validation tasks required for the integrity level assigned to the system, software, or hardware to be verified and validated *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)*

test design. (1) documentation specifying the details of the test approach for a system, software, or hardware feature or combination of features and identifying the associated tests *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1.32) Note:* commonly includes the organization of the tests into groups

fault. (1) manifestation of an error in software *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** defect in a system or a representation of a system that if executed/activated could potentially result in an error *(ISO/IEC/IEEE 15026-1:2019 Systems and software engineering--Systems and software assurance--Part 1: Concepts and vocabulary, 3.4.6)* **(3)** situation that can cause errors to occur in an object *(ISO/IEC 10746-2:2009 Information technology -- Open Distributed Processing -- Reference Model: Foundations, 13.6.3)* **(4)** defect in a hardware device or component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Note:* A fault, if encountered, can cause a failure. Faults can occur in specifications when they are not correct. *Syn:* bug

corrective maintenance. (1) modification of a software product performed after delivery to correct discovered problems *(ISO/IEC/IEEE 14764:2021, Software engineering -- Software life cycle processes -- Maintenance, 3.1.4)* **(2)** maintenance performed to correct faults in hardware or software *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Note:* The modification repairs the software product to satisfy defined system requirements.

commercial-off-the-shelf (COTS). (1) [software or hardware] product available for purchase and use without the need to conduct development activities *(ISO/IEC/IEEE 90003:2018 Software engineering -- Guidelines for the application of ISO 9001:2015 to computer software, 3.4) (IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3) 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 do not apply. *Syn:* commercial off the shelf, Commercial-Off-The-Shelf *See also:* software product

architectural design. (1) definition of a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* functional design

fault tolerance. (1) capability of a product to operate as intended despite the presence of hardware or software faults *(ISO/IEC 25010:2023, Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--Product quality model, 3.5.3) See also:* error tolerance, fail safe, fail soft, fault secure, robustness

hardware monitor. (1) device that measures or records specified events or characteristics of a computer system *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** software tool that records or analyzes hardware events during the execution of a computer program *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

implementation. (1) process of translating a design into hardware components, software components, or both *(ISO/IEC/IEEE 90003:2018 Software engineering -- Guidelines for the application of ISO 9001:2015 to computer software, 3.4)* **(2)** process of instantiation whose validity can be subject to test *(ISO/IEC 10746-3:2009 Information technology -- Open Distributed Processing -- Reference Model: Architecture, 9.1.2)* **(3)** construction *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(4)** system development phase at the end of which the hardware, software and procedures of the system become operational *(ISO/IEC 2382:2015 Information technology -- Vocabulary) See also:* coding

maintenance. (1) process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment *(ISO/IEC 25051:2014 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing, 4.1.9)* **(2)** process of retaining a hardware system or component in, or restoring it to, a state in which it can perform its required functions *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(3)** actions intended to retain a product in, or restore it to, a useful and safe condition, in which it can perform the intended use *(IEC/IEEE 82079-1:2019 Preparation of information for use (instructions for use) of products: Part 1: Principles and general requirements, 3.23) Note:* In the context of dependability, maintenance is a combination of all technical and management actions intended to retain an item in, or restore it to, a state in which it can perform as required. *See also:* adaptive maintenance, corrective maintenance, perfective maintenance, software maintenance

preliminary design review (PDR). (1) review conducted to evaluate the progress, technical adequacy, and risk resolution of the selected design approach for one or more configuration items; to determine each design's compatibility with the requirements for the configuration item; to evaluate the degree of definition and assess the technical risk associated with the selected manufacturing methods and processes; to establish the existence and compatibility of the physical and functional interfaces among the configuration items and other items of equipment, facilities, software and personnel; and, as applicable, to evaluate the preliminary operational and support documents *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** review as in (1) of any hardware or software component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* critical design review, system design review

requirements analysis. (1) process of studying user needs to arrive at a definition of system, hardware, or software requirements *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** process of studying and refining system, hardware, or software requirements *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(3)** systematic investigation of user requirements to arrive at a definition of a system *(ISO/IEC 2382:2015 Information technology -- Vocabulary)* **(4)** 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 *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

test readiness review (TRR). (1) review conducted to evaluate preliminary test results for one or more configuration items; to verify that the test procedures for each configuration item are complete, comply with test plans and descriptions, and satisfy test requirements; and to verify that a project is prepared to proceed to formal testing of the configuration items *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** review as in (1) for any hardware or software component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* code review, formal qualification review, design review, requirements review

trap. (1) conditional jump to an exception or interrupt handling routine, often automatically activated by hardware, with the location from which the jump occurred recorded *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** to perform the operation in (1) *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

big-bang testing. (1) type of integration testing in which software elements, hardware elements, or both are combined all at once into an overall system, rather than in stages *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

computer program. (1) combination of computer instructions and data definitions that enable computer hardware to perform computational or control functions *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

(2) syntactic unit that conforms to the rules of a particular programming language and that is composed of declarations and statements or instructions needed for a certain function, task, or problem solution *(ISO/IEC 2382:2015 Information technology -- Vocabulary) See also:* software

computer software configuration item (CSCI). (1) aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process *(ISO/IEC/IEEE 24765:2017*

Systems and software engineering-Vocabulary) Syn: software configuration item (SWCI) *See also:* computer software component, hardware configuration item, configuration item, software configuration item, software item

computer system. (1) system containing one or more computers and associated software *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** system containing one or more components and elements such as computers (hardware), associated software, and data *(ISO/IEC 25024:2015 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of data, 4.3) Syn:* computing system *See also:* data processing system

computing center. (1) facility designed to provide computer services to a variety of users through the operation of computers and auxiliary hardware and through services provided by the facility's staff *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

configuration. (1) arrangement of a computer system or component as defined by the number, nature, and interconnections of its constituent parts *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** in configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(3)** arrangement of a system or network as defined by the nature, number, and chief characteristics of its functional units *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(4)** requirements, design, and implementation that define a particular version of a system or system component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(5)** manner in which the hardware and software of an information processing system are organized and interconnected *(ISO/IEC 2382:2015 Information technology -- Vocabulary)* **(6)** collection of objects able to interact at interfaces *(ISO/IEC 10746-2:2009 Information technology -- Open Distributed Processing -- Reference Model: Foundations, 10.2) See also:* configuration item; form, fit, and function; version

configuration item (CI). (1) item or aggregation of hardware, software, or both that is designated for configuration management and treated as a single entity in the configuration management process *(ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.15) (ISO/IEC/IEEE 15288:2023 Systems and software engineering--System life cycle processes, 3.11)* **(2)** component of an infrastructure or an item which is or will be under control of configuration management *(ISO/IEC 19770-5:2015 Information technology--IT asset management--Overview and vocabulary, 3.7)* **(3)** aggregation of work products that is designated for configuration management and treated as a single entity in the configuration management process *(IEEE 828-2012 IEEE Standard for Configuration Management in Systems and Software Engineering, 2.1)* **(4)** any system element or aggregation of system elements that satisfies an end use function and is designated by the acquirer for separate configuration control *(ISO/IEC/IEEE 24748-7:2019, Systems and software engineering Life cycle management Part 7: Application of systems engineering on defense programs, 3.1)* **(5)** item or aggregation of software that is designed to be managed as a single entity and its underlying components, such as documentation, data structures, scripts *(ISO/IEC 16350-2015 Information technology--Systems and software engineering--Application management, 4.12)* **(6)** system, system element, or artifact designated for configuration management *(INCOSE Systems Engineering Handbook, 5th ed.)* **(7)** unit or aggregation of hardware, software, or both that is designated for configuration management and treated as a single entity in the configuration management process *(IEEE 982:2024, Standard for Measures of the Software Aspects of Dependability, 3.1) Note:*

Configuration items can vary widely in complexity, size and type, ranging from an entire system including all hardware, software and documentation, to a single module or a minor hardware component. CIs have four common characteristics: defined functionality; replaceable as an entity; unique specification; formal control of form, fit, and function. *See also:* hardware configuration item, computer software configuration item, configuration identification, critical item

constant-failure period. (1) period of time in the life cycle of a system or component during which hardware failures occur at an approximately uniform rate *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* early-failure period, wearout-failure period, bathtub curve

design language. (1) specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document a hardware or software design *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** standardized notation, modeling technique, or other representation scheme and its usage conventions, shown to be effective in representing and communicating design information *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Note:* Types include hardware design language, program design language. *See also:* requirements specification language

design review. (1) formal, documented, comprehensive, and systematic examination of a design to determine if the design meets the applicable requirements, to identify problems, and to propose solutions *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** process or meeting during which a system, hardware, or software design is presented to project personnel, managers, users, customers, or other interested parties for comment or approval *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Note:* Types include critical design review, preliminary design review, system design review. *See also:* code review, formal qualification review, requirements review, test readiness review

downward compatible. (1) pertaining to hardware or software that is compatible with an earlier or less complex version of itself *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* upward compatible

early-failure period. (1) period of time in the life cycle of a system or component during which hardware failures occur at a decreasing rate as problems are detected and repaired *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Syn:* burn-in period *See also:* constant-failure period, wearout-failure period, bathtub curve

engineering change. (1) alteration in the configuration of a hardware/software configuration item or items, delivered, to be delivered, or under development, after formal establishment of their configuration identification *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** in configuration management, an alteration in the configuration of a configuration item or other designated item after formal establishment of its configuration identification *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* configuration control, engineering change proposal, deviation, waiver

HDL. (1) hardware description language *(ISO/IEC/IEEE 24765d:2015)* **(2)** hardware design language *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* design language

HWCI. (1) hardware configuration item *(ISO/IEC/IEEE 24748-8:2019, Systems and software engineering Life cycle management Part 7: Application of systems engineering on defense programs, 3.2)*

instrument. (1) in software and system testing, to install or insert devices or instructions into hardware or software to monitor the operation of a system or component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-*

Vocabulary)

instrumentation. (1) devices or instructions installed or inserted into hardware or software to monitor the operation of a system or component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

integration. (1) process of combining software components, hardware components, or both into an overall system *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** process of planning for and aggregating a progressively more complete set of physical, logical, or both, system elements and activating their interfaces to synthesize a system or part of a system whose properties can be verified and possibly validated *(ISO/IEC/IEEE 24748-6:2023 Systems and software engineering Life cycle management Part 6: System and software integration, 3.1.4) Note:* Integration can apply to the implemented system elements which compose a system and the necessary life-cycle related activities.

monitor. (1) software tool or hardware device that operates concurrently with a system or component and supervises, records, analyzes, or verifies the operation of the system or component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** collect project performance data, produce performance measures, and report and disseminate performance information *(A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) -- Seventh Edition) Syn:* execution monitor *See also:* hardware monitor, software monitor

operating system. (1) collection of software, firmware, and hardware elements that controls the execution of computer programs and provides such services as computer resource allocation, job control, input/output control, and file management in a computer system *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Syn:* OS

pipeline. (1) software or hardware design technique in which the output of one process serves as input to a second, the output of the second process serves as input to a third, and so on, often with simultaneity within a single cycle time *(IEEE 2675-2021, IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment, 3.1)*

portability. (1) ease with which a system or component can be transferred from one hardware or software environment to another *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** capability of a program to be executed on various types of data processing systems without converting the program to a different language and with little or no modification *(ISO/IEC 2382:2015 Information technology -- Vocabulary)* **(3)** degree to which a cloud service provides the ability to move data and migrate applications from one cloud service to another *(ISO/IEC TS 25052-1:2022, Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE): cloud services--Part 1: Quality model, 3.1.7)* **(4)** property that the reference points of an object allow it to be adapted to a variety of configurations *(ISO/IEC 10746-2:2009 Information technology -- Open Distributed Processing -- Reference Model: Foundations, 15.4.1) Syn:* transportability *See also:* machine-independent

program design language (PDL). (1) specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document a program design *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* hardware design language, pseudo code

prototyping. (1) hardware and software development technique in which a preliminary version of part or all of the hardware or software is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See*

also: rapid prototyping

requirements review. (1) process or meeting during which the requirements for a system, hardware item, or software item are presented to project personnel, managers, users, customers, or other interested parties for comment or approval *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Note:* Types include system requirements review, software requirements review. *See also:* code review, design review, formal qualification review, test readiness review

requirements specification language. (1) specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document hardware or software requirements *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* design language

software engineering environment (SEE). (1) environment that provides automated system context services and software-specific services for the engineering of software systems and related domains, such as project management and process management *(ISO/IEC 15940:2013 Systems and software engineering--Software Engineering Environment Services, 2.7)* **(2)** hardware, software, and firmware used to perform a software engineering effort *(IEEE 730-2014 IEEE Standard for Software Quality Assurance Processes, 3.2) Note:* It includes the platform, system software, utilities, and CASE tools installed. *See also:* infrastructure

software monitor. (1) software tool that executes concurrently with another program and provides detailed information about the execution of the other program *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* hardware monitor, monitor

stand-alone. (1) pertaining to hardware or software that is capable of performing its function without being connected to other components *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

HCI. (1) human computer interface *(ISO/IEC 10746-1:1998 Information technology -- Open Distributed Processing -- Reference model: Overview)* **(2)** hardware configuration item *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Syn:* HWCI

software test environment (STE). (1) facilities, hardware, software, firmware, procedures, and documentation needed to perform qualification or other testing of software *(ISO/IEC 25051:2014 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing, 4.1.15) Note:* Elements include simulators, code analyzers, test case generators, path analyzers, and elements used in the software engineering environment

server. (1) hardware system or software program which provides a service to clients *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

test bed. (1) environment containing the hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

turnkey. (1) pertaining to a hardware or software system delivered in a complete, operational state *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

upward compatible. (1) pertaining to hardware or software that is compatible with a later or more complex version of itself *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* downward compatible

user interface. (1) components of an interactive system (software or hardware) that provide information and controls

for the user to accomplish specific tasks with the interactive system *(ISO/IEC 25063:2014 Systems and software engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) Common Industry Format (CIF) for usability: Context of use description)* **(2)** ensemble of software and hardware that allows a user to interact with a computer system *(ISO/IEC/IEEE 26514:2022, Systems and software engineering -- Design and development of information for users, 3.1.55) (ISO/IEC/IEEE 26512:2018 Systems and software engineering--Requirements for acquirers and suppliers of information for users, 3.29)* **(3)** set of all the components of an interactive system that provide information and controls for the user to accomplish specific tasks with the interactive system *(ISO TR 25060:2023 Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--General Framework for Common Industry Format (CIF) for usability-related information, 3.2.2) Syn:* UI

wearout-failure period. (1) period in the life cycle of a system or component during which hardware failures occur at an increasing rate due to deterioration *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* constant-failure period, early-failure period, bathtub curve

context of use. (1) users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a system, product or service is used *(ISO/IEC 25063:2014 Systems and software engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) Common Industry Format (CIF) for usability: Context of use description, 3.2)* **(2)** users, tasks, equipment (hardware, software, and materials), and the physical and social environments in which a product is used *(ISO/IEC 25000:2014 Systems and software Engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 4.2)* **(3)** conditions and constraints under which ICT products are used by specific users in a specific environment to achieve specific goals as part of the larger information system *(ISO/IEC 25030:2019 Systems and software engineering--Systems and software quality requirements and evaluation (SQuaRE)--Quality requirements framework, 3.2)* **(4)** intended operational environment for a system *(IEEE 7000:2021, IEEE Standard Model Process for Addressing Ethical Concerns during System Design, 3.1)* **(5)** combination of users, goals and tasks, resources, and environment *(ISO TR 25060:2023 Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--General Framework for Common Industry Format (CIF) for usability-related information, 3.1.10) (ISO/IEC 25019:2023, Systems and software engineering Systems and software Quality Requirements and Evaluation (SQuaRE) Quality-in-use model, 3.1.4) Note:* Context of use includes direct use or use supported by assistive technologies. Environment includes physical aspects such as equipment and resources as well as social aspects such as demographics and culture.

external I/O device. (1) hardware input and/or output device that is outside the software system and part of the external environment *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

hardware configuration item (HCI). (1) aggregation of hardware that is designated for configuration management and treated as a single entity in the configuration management process *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Note:* An HCI exists where functional allocations have been made that clearly distinguish equipment functions from software functions and where the hardware has been established as a configuration item. *See also:* software configuration item

integrate. (1) to combine software components, hardware components, or both into an overall system *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** to pull in the changes from one child branch into its

parent *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

performance analysis. (1) quantitative analysis of a real-time system (or software design) executing on a given hardware configuration with a given external workload applied to it *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

software configuration item (SCI). (1) software entity that has been established as a configuration item *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) Note:* The SCI exists where functional allocations have been made that clearly distinguish equipment functions from software functions and where the software has been established as a configurable item. *See also:* computer software component, computer software configuration item, hardware configuration item, software item

software requirements analysis. (1) process of studying user needs to arrive at a definition of system, hardware, or software requirements *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

configuration management database (CMDB). (1) specific type of repository for CM information, usually a data store, used to record attributes of configuration items, and the relationships between configuration items, throughout their lifecycle *(IEEE 828-2012 IEEE Standard for Configuration Management in Systems and Software Engineering, 2.1)*

(2) database containing all the relevant details of each configuration item and details of the important relationships between them *(ISO/IEC 19770-5:2015 Information technology--IT asset management--Overview and vocabulary, 3.8)* **(3)** database that is used by an organization to store information about the hardware and software in use *(ISO/IEC TR 7052:2023, Software engineering--Controlling frequently occurring risks during development and maintenance of custom software, 3.10) See also:* configuration management system

test environment. (1) environment containing facilities, hardware, software, firmware, and procedures needed to conduct a test *(ISO/IEC/IEEE 29119-2:2021, Software and systems engineering--Software testing--Part 2: Test processes, 3.34) Note:* A test environment can contain multiple environments to accommodate specific test levels or types, e.g., a unit test environment, a performance test environment. A test environment can comprise several interconnected systems or virtual environments.

assistive technologies. (1) hardware or software that is added to or incorporated within a system that increases accessibility for an individual *(ISO/IEC/IEEE 24765n:2025)*

disaster recovery. (1) in computer system operations, the return to normal operation after a hardware or software failure *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** ability of the information and communications technology elements of an organization to support its critical business functions to an acceptable level within a predetermined period following a disaster *(ISO/IEC/IEEE 26511:2018 Systems and software engineering--Requirements for managers of information for users of systems, software, and services, 3.1.11)*

computer center. (1) facility that includes personnel, hardware, and software, organized to provide information processing services *(ISO/IEC 2382:2015 Information technology -- Vocabulary) Syn:* data processing center

information system. (1) information processing system, together with associated organizational resources such as human, technical, and financial resources, which provides and distributes information *(ISO/IEC 2382:2015 Information technology -- Vocabulary)* **(2)** all of the functions (input, output, transport, processing, and storage) of an application, databases, technical facilities, and manual procedures which support business processes *(ISO/IEC 16350-2015*

Information technology--Systems and software engineering--Application management, 4.21) **(3)** one or more computer systems and communication systems together with associated organizational resources such as human, technical, and financial resources that provide and distribute information *(ISO/IEC 25024:2015 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of data, 4.24)* **(4)** system that is comprised of software, hardware, communication facility, data, and the people who use it in a given environment to satisfy their information processing needs *(ISO/IEC 25030:2019 Systems and software engineering--Systems and software quality requirements and evaluation (SQuaRE)--Quality requirements framework, 3.10)* **(5)** system which is designated to collect, organize, store, communicate, and process data *(IEEE 7005 2021, IEEE Standard for Transparent Employer Data Governance, 3.1)* **(6)** system that comprises ICT products, ICT environment, and the people who use them or are impacted by them, which become a combination of interacting elements organized to achieve one or more stated purposes *(ISO/IEC 25019:2023, Systems and software engineering Systems and software Quality Requirements and Evaluation (SQuaRE) Quality-in-use model, 3.1.9) See also:* application

functional unit. (1) entity of hardware or software, or both, capable of accomplishing a specified purpose *(ISO/IEC 2382:2015 Information technology -- Vocabulary)*

computer crime. (1) crime committed through the use, modification, or destruction of hardware, software, or data *(ISO/IEC 2382:2015 Information technology -- Vocabulary)*

system design. (1) process of defining the hardware and software architecture, components, modules, interfaces and data for a system to satisfy specified requirements *(ISO/IEC 2382:2015 Information technology -- Vocabulary)*

system description. (1) documentation that results from system design defining the organization, essential characteristics and the hardware and software requirements of the system *(ISO/IEC 2382:2015 Information technology -- Vocabulary)*

platform. (1) type of computer or hardware device and/or associated operating system, or a virtual environment, on which software can be installed or run **(2)** combination of an operating system and hardware that makes up the operating environment in which a program runs *(ISO/IEC/IEEE 26513:2017 Systems and software engineering--Requirements for testers and reviewers of information for users, 3.30)* **(3)** type of computer or hardware device and/or associated operating system, or a virtual environment on which software can be installed or run *(ISO/IEC 19770-5:2015 Information technology--IT asset management--Overview and vocabulary, 3.23) Note:* A platform is distinct from the unique instances of that platform, which are typically referred to as devices or instances. *See also:* device

critical design review (CDR). (1) review conducted to verify that the detailed design of one or more configuration items satisfy specified requirements; to establish the compatibility among the configuration items and other items of equipment, facilities, software, and personnel; to assess risk areas for each configuration item; and, as applicable, to assess the results of producibility analyses, review preliminary hardware product specifications, evaluate preliminary test planning, and evaluate the adequacy of preliminary operation and support documents *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)* **(2)** review as in (1) of any hardware or software component *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

external interface requirement. (1) system or software requirement that specifies a hardware, software, or database element with which a system/software system or system/software component must interface, or that sets forth

constraints on formats, timing, or other factors caused by such an interface *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary)*

hardware engineering. (1) application of a systematic, disciplined, and quantifiable approach to design, implement, and maintain a tangible product by transforming a set of requirements that represent the collection of stakeholder needs, expectations, and constraints; using documented techniques and technology *(ISO/IEC/IEEE 24765:2017 Systems and software engineering-Vocabulary) See also:* software engineering, systems engineering

reusable product. (1) system, software, or hardware product developed for one use but having other uses, or one developed specifically to be usable on multiple projects or in multiple roles on one project *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1) Note:* Each use can include all or part of the product and can involve its modification. This term can be applied to any software or system product (for example, requirements or architectures), not just to software or system itself.

adaptability. (1) ability of a system to react to changes in its environment in order to continue meeting both functional and non-functional requirements *(ISO/IEC TR 29119-11:2020, Software and systems engineering--Software testing--Part 11: Guidelines on the testing of AI-based systems, 3.1.5)* **(2)** capability of a product to be effectively and efficiently adapted for or transferred to different hardware, software or other operational or usage environments *(ISO/IEC 25010:2023, Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--Product quality model, 3.8.1) Note:* Adaptability can enable a system to function in limited operation mode with reduced energy supplies or no communication network connectivity due to an accident or a disaster. *See also:* flexibility, functional adaptability

constituent configuration item. (1) individual item to be controlled that is a constituent (part) of a larger configuration item, such as a reference model, hardware prototype or software build *(IEEE 828-2012 IEEE Standard for Configuration Management in Systems and Software Engineering, 2.1)*

runtime platform. (1) set of hardware and software components that implement the services utilized by the application software *(ISO/IEC/IEEE 24765l:2024)*

portability testing. (1) type of testing conducted to evaluate the ease with which a test item can be transferred from one hardware or software environment to another, including the level of modification needed for it to be executed in various types of environments *(ISO/IEC/IEEE 29119-2:2021, Software and systems engineering--Software testing--Part 2: Test processes, 3.59)*

HDD. (1) hardware design description *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.2)*

HRS. (1) hardware requirements specification *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1)*

HW. (1) hardware *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.2)*

development environment. (1) hardware, software, platform and tools for designers and developers of computer solutions *(ISO/IEC/IEEE 24765c:2014)* **(2)** workplace facility in which a system is developed and maintained *(IEEE 982:2024, Standard for Measures of the Software Aspects of Dependability, 3.1) Note:* It has staffing, resources, interfaces, capabilities, and tooling not found in that systems target environments.

development tool. (1) hardware and software for developing or modifying applications *(ISO/IEC/IEEE 24765c:2014)*

infrastructure. (1) hardware and software environment to support computer system and software design, development, and modification *(ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.25)* **(2)** facilities such as power, cooling, and physical security of the data center, networking, hardware, and software needed to support the systems life cycle and maintain information technology (IT) services *(IEEE 2675-2021, IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment, 3.1) Note:* Does not include the associated people or processes. In DevOps, software-defined infrastructure enables elasticity. *Syn:* ecosystem *See also:* IT infrastructure, software engineering environment

complex programmable logic device (CPLD). (1) hardware component with a fully programmable AND/OR gate array *(ISO/IEC/IEEE 24765d:2015)*

in-circuit emulator (ICE). (1) hardware device used to debug the software of an embedded system *(ISO/IEC/IEEE 24765d:2015)*

software testing environment. (1) facilities, hardware, software, firmware, procedures, and documentation needed to perform testing of software *(IEEE 730-2014 IEEE Standard for Software Quality Assurance Processes)*

interactive system. (1) combination of hardware, software and/or services that receives input from and communicates output to users *(ISO/IEC 25063:2014 Systems and software engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) Common Industry Format (CIF) for usability: Context of use description)* **(2)** combination of hardware or software or services or people that users interact with to achieve specific goals *(ISO TR 25060:2023 Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--General Framework for Common Industry Format (CIF) for usability-related information, 3.1.2) Note:* This includes, where appropriate, packaging, user documentation, online and human help, support and training.

hardware description language (HDL). (1) software programming language used to design and model hardware, especially digital logic circuits *(ISO/IEC/IEEE 24765d:2015) Syn:* hardware design language

Verilog hardware description language (VHDL). (1) hardware description language used to design and verify digital circuits *(ISO/IEC/IEEE 24765e:2015) Note:* standardized in IEEE 1394.

software development environment. (1) facilities, hardware, software, procedures, and documentation needed to perform software development *(ISO/IEC/IEEE 24748-5:2017 Systems and software engineering--Life cycle management--Part 5: Software development planning, 3.14) Note:* Plans for software development environments can include where the specified environment is to be constructed, when sites provide different environments or facilities. For example, different testing environments can be requested to be constructed at the acquirer's site and the supplier's site. *See also:* enabling system

nth of a kind. (1) re-manufacturing or re-installation of a previously verified and validated hardware or software design *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.1) Note:* The nth of a kind component or system is equivalent to the first
application in all relevant aspects, including functional and performance requirements, design documentation, environment, and regulatory constraints.

dynamic attribute. (1) element of a hardware identification (HWID) tag that can change over the life of the product or

be defined after creation *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.1.1)*

hardware asset management (HAM). (1) coordinated activity of an organization to realize value from hardware assets *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.1.2) Note:* specialization of IT asset management

HAM. (1) hardware asset management *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.1.2)*

hardware identification creator. (1) entity that initially creates a HWID record *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.1.3) Note:* This entity can be part of the organization that manufactured the component to which the record relates, in which case the HWID creator and component manufacturer are the same. The HWID creator can also be a separate organization or third party unrelated to the manufacturer (such as in the case where HWID records are created for existing hardware components by an operating system or a tool deployed by the device owner). *Syn:* HWID creator

HWID. (1) hardware identification *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.2)*

hardware identification schema. (1) information structure containing a digital description of a hardware component and its associated information *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.1.4) Syn:* HWID schema

legacy hardware. (1) hardware originally created without native information structures *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.1.5)*

static attribute. (1) element of a hardware identification (HWID) that does not change over the life of the product or is defined at creation *(ISO/IEC 19770-6:2024, Information technology IT asset management Part 6: Hardware identification tag, 3.1.7)*

runtime environment. (1) specific instance of a configuration of hardware and software, as provisioned for a specific operational environment, on which an installed system of interest runs *(IEEE 982:2024, Standard for Measures of the Software Aspects of Dependability, 3.1) Note:* A runtime environment includes executable application software, data stores, utility programs, locally hosted services, operating systems, network interfaces, and physical computer systems necessary for system of interest (SOI) execution, but not the SOI itself.

HILS. (1) hardware in the loop simulation *(IEEE 1012-2024 IEEE Standard for System, Software, and Hardware Verification and Validation, 3.2)*

assistive technology. (1) equipment, product, system, hardware, software, or service that is used to increase, maintain or improve capabilities of individuals *(ISO/IEC 29110-5-1-2:2025 Systems and software engineering Life cycle profiles for very small entities (VSEs) Part 5-1-2: Software engineering guidelines for the generic Basic profile, 3.6) Note:* Assistive technology can include assistive services, and professional services needed for assessment, recommendation and provision.