

GLOSSARY OF SOFTWARE OPERATIONS MAINTENANCE AND SUSTAINMENT TERMS AND MAINTENANCE MEASURES

Prepared by:

Donald J. Reifer
Reifer Consultants LLC
14820 N. Dragons Breath Lane
Prescott, AZ 86305
Phone: 928-237-9060
Email: don@reifer.com

GLOSSARY OF SOFTWARE OPERATIONS, MAINTENANCE AND SUSTAINMENT TERMS

1. PURPOSE AND SCOPE

This document defines terms used in our questionnaires which have been designed to collect data about software operations, maintenance and sustainment activities. Its purpose is to:

- Assist in the collection of software operations, maintenance and sustainment cost, productivity and quality data for use in benchmarking, model calibration and analysis.
- Capture data to develop a software inventory for Department of Defense (DOD) programs in the operations and maintenance phase of the weapons systems life cycle (after Milestone C).
- Define metrics to be reported and the information needs upon which they were based.
- Standardize the manner in which data are collected, normalized, analyzed and reported.

We assume that organizations performing software maintenance and sustainment tasks are separate from the one that initially developed the products.

Definitions for terms used within the volume are provided by this Glossary. Whenever possible, the terms that I use within this volume are based on industry standards like the following:

- IEEE Std. 610.12-90 (R2002), IEEE Standard Glossary of Software Engineering Terminology, Institute for Electrical and Electronics Engineers (IEEE), 1990.
- IEEE Std. 1219-1998, IEEE Standard for Software Maintenance, IEEE, 1998.
- ISO/IEC 14764-1999, Software Engineering – Software Maintenance, International Standards Organization (ISO) and International Electro-technical Commission (IEC), 1999.

2. ORGANIZATION OF DOCUMENT

This document is organized into the following two parts:

- Part A: Glossary of Software Maintenance, Operations and Sustainment Terms
- Part B: Software Maintenance Measures

The maintenance measures expand definitions. Definition of these measures is a work in progress.

3. ACKNOWLEDGEMENTS

This glossary was devised as a collaborative effort with the following people contributing to the definitions contained within:

- | | |
|-------------------------------------|---|
| ▪ Frances Anderson, Aerospace Corp. | ▪ Jack McGarry, U.S. Army |
| ▪ Daniel Beisel, Northrop Grumman | ▪ Michael McLendon, OSD/SEI |
| ▪ Richard Cline, Boeing | ▪ Celia Modell, Boeing |
| ▪ Susan Davis, U.S. Army | ▪ Shally Molhotra, SAIC |
| ▪ Brian Fersch, U.S. Air Force | ▪ Jeramia Poland, U.S. Army |
| ▪ Jennifer Fischer, KPMG | ▪ Donald Reifer, Reifer Consultants LLC |
| ▪ William Golaz, Lockheed Martin | ▪ Wilson Rosa, U.S. Air Force |
| ▪ Barbara Hitchings, SAIC | ▪ Denise Saltojan, U.S. Air Force |
| ▪ Cheryl Jones, U.S. Army | ▪ John Sautter, Northrop Grumman |
| ▪ James Judy, U.S. Army | ▪ Craig Stauffer, KPMG |
| ▪ Ilya Lipkin, U.S. Air Force | |

Part A: Glossary of Software Operations, Maintenance and Sustainment Terms

Acceptance testing – Black-box testing conducted by the customer on a software system aimed at confirming that it satisfies its specifications and performs as expected prior to accepting transfer of ownership.

Accreditation (security) – Refers to the process pertaining to ensuring that all the security requirements established for the system have been satisfied.

Acquisition – Refers to the process of acquiring something via a contract.

Acquisition support – Tasks performed by the life cycle maintenance organization in support of those in charge of acquiring software maintenance updates and support primarily from its contractor workforces.

Activity – Major unit of work to be completed. An activity has a precise starting and ending date, includes a set of tasks, consumes resources, and results in generation of work products.

Adaptation – In maintenance, adjusting software via data and parameters so that it will work in a given installation site or to given conditions in its operational environment.

Adaptive maintenance - Modification of a software product performed after delivery to keep a computer program usable in a changed or changing environment [ISO/IEC 14764].

Agile development process – Refers to a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.

Aging - The number of days a software trouble report has remained open counted from the day it was written until the day it was closed.

Agreement – The mutual acknowledgement of terms and conditions under which a working relationship will be conducted.

Annual change traffic – The number of source lines of code (SLOC) modified or added to a release or version on an annual basis.

Applications domain – Identifies the type of application that is being addressed by the maintenance and support activity. Examples of applications domains taken from benchmarks that we publish for the military include airborne, ground, information technology, medical, missile, space and trainers. Applications domains for commercial firms include automation, command and control, information technology [banking, general or insurance], medical, process control, scientific systems, software tools, telecommunications, test system, training/simulation, and web business. Applications domains for government shops include information technology, military, and sensitive systems.

Applications software – Refers to software that provides a set of services or solves some type of user problem.

Appraisal – Refers to the process of assessing the worth, significance, or status of something of interest.

Architecture – Structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time.

Asset – Something of value that a firm owns and can capitalize.

Audit – An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria.

Authentication – Refers to the process of determining whether someone or something (such as a computer or software process) is, in fact, who or what it is declared to be. Methods for human authentication include something you know (a password), something you have (a token), or something you are (fingerprint).

Auto-generated code – Refers to software code that is generated automatically by using a specialized tool from scripts written in some natural language.

Authority – In project management, refers to the right to give direction and allocate resources.

Availability – The degree to which the services of a system or component are operational and accessible when needed by their intended users.

Backlog – An accumulation of unfinished and/or deferred work that must be dealt with now or sometime in the future.

Baseline – A specification or product that has been reviewed and agreed upon, and thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

Benchmark – A standard against which measurements or comparisons can be made.

Benchmark period – The period in which maintenance and support was supplied. Typically, this is measured in months from when the go-ahead was given to begin the release until it was delivered and fielded.

Best practice - Engineering or management activity that directly addresses the purpose of a particular process and contributes to the creation of its output (e.g., metrics provide insight using measurement data to create their results). Best practices in this context are activities that are established, based on general consensus, as the most effective means of delivering such output.

Budget – The resources (\$, people, equipment, facilities, etc.) allocated to perform software operations, maintenance and sustainment activities.

Build – An operational version of software that incorporates a specified subset of the capabilities that the final product will provide.

Business case – Refers to materials prepared for decision makers to show them that the business idea under consideration is a good one and that the numbers make financial as well as technical sense for the organization.

Capability Maturity Model (CMM) – A description of the stages through which organizations evolve as they define, implement, measure, control, and improve their maintenance processes. The 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 quality and process improvement.

Capability Maturity Model Integration (CMMI) – A process improvement approach for evaluating how efficiently an organization is able to deliver technology products to its customers. CMMI is often associated with integrating processes used within an organization by diverse engineering disciplines like systems engineering, software development and integrated product teams.

Certification – The process of confirming that software complies with specific requirements (interoperability, network operations, security, etc.) and is acceptable for operational use.

Certification and Accreditation (C&A) – The process of confirming that requirements associated with the Defense Information Assurance Certification and Accreditation Process (DIACAP) have been complied with per DOD Instruction 8510.01.

Champion – Refers to a high-level member of the senior management team who supports your ideas and acts as your proponent with other executives.

Change control – The process used to ensure that changes to a product or system are introduced in a controlled and coordinated manner.

Change Control Board (CCB) – A committee that makes decisions regarding whether or not proposed changes to a software project, release or version should be made. The board is constituted of project stakeholders or their representatives. The authority of the board may vary, but decisions reached are often accepted as final and binding.

Cloud computing – Refers to Internet-based computing, whereby shared resources, software, and information are provided to computers and other devices on demand, like either a time share system or the electricity grid.

Commercial Off-The-Shelf (COTS) – Hardware and software products that are ready made and available for sale to the general public on either a purchase or license basis. COTS products are used as-is with perhaps some tailoring via parameterization and data entry. The major advantages of COTS are that it is available immediately with known quality at an affordable and reasonable cost.

Commitment – Obligation to commit resources at some future time, such as a purchase order or travel authorization, which is charged against a budget even though it has not yet been paid.

Competency – Skills, knowledge and personal attributes that enable effective work performance.

Compliance – Refers to meeting the requirements of a standard established by a specification.

Component – Refers to an entity, with a prescribed structure and set of relationships within a system, which interacts with other components and performs those functions for which it was intended per a prescribed specification. A component can be subdivided into subcomponents.

Concession – Refers to authorization to use or release a product that does not totally conform to specified requirements.

Confidentiality – The extent to which the characteristics of a software component, product, or system – including its relationships with its execution environment and its users, its managed assets, and its content – are obscured or hidden from unauthorized entities.

Configuration – In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product.

Configuration index – A document used in configuration management that provides a listing of the components that make up a product (e.g., a bill of materials).

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.

Consistency – Refers to the degree of uniformity, standardization, and freedom from contradiction among the documents or parts or components of the system.

Contingency – In management, refers to the amount of design margin, time, or money used as a safety factor to accommodate future growth or uncertainty.

Contract - An agreement between two or more competent parties in which an offer is made and accepted, and each party benefits. The agreement can be formal, informal, written, oral or just plain understood. Some contracts are required to be in writing in order to be enforced.

Contract Data Requirements List (CDRL) – A specification of the documents to be delivered on a government contract including the data descriptions, delivery schedule, number of copies required, and associated reviewing and approving authority.

Contract terms and conditions – Refers to the legal, financial, administrative and other pertinent aspects of a contract.

Corrective maintenance - Reactive modification of a software product performed after delivery to correct discovered faults [ISO/IEC 14764].

Cost – The resources (dollars (\$), staff, staff-hours, etc.) required to develop an item, complete a task or activity, and/or deliver a product.

Costing – In management, refers to the process of developing a cost estimate for an item, task, or activity. Costing and pricing are separate but related activities typically done by different people at different times during the life cycle.

Critical success factor – Refers to those characteristics, conditions, or variables that have a direct influence on customers' satisfaction with the products and services that a firm offers.

Customer – The individual or organization that is in charge of acquiring new releases for the user during the software maintenance portion of the life cycle. This person or organization may or may not be the sponsor. However, they will represent them and the user as they manage the delivery budgets, schedules and pressures.

Database model or schema – A description of the structure or format of the database, defined in a formal language supported by the database management system. Schemas are generally stored in a data dictionary.

Database size – The size of the database maintained by application reported in Megabytes.

Data Item Description (DID) – A description of a document to be delivered. As a minimum, a DID specifies the format of the deliverable, its contents, and how many copies must be sent to whom, when and how (using what means – electronics, mail, etc.).

Defect - The difference between a computed, observed, or measured values or conditions and the true, specified, or theoretically correct value or conditions discovered via reviews by analysis

(defect found via code reading during a code walkthrough, etc.). Defects are often categorized based on their impact using the five level classification scheme that follows:

- **Category 1 Defects (Catastrophic)** – the number of catastrophic defects found and fixed in this release. Catastrophic defects are those that prevent the accomplishment of an operational or mission-essential capability and for which no work-around solution is known. In addition, catastrophic defects include all system/software lockups and those defects that jeopardize safety, security, or other requirement designated “critical.”
- **Category 2 Defects (Critical)** – the number of critical defects found and fixed in this release. Critical defects are those that adversely affect the accomplishment of an operational or mission-essential capability and for which a work-around solution is not known. In addition, such defects include those that adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system and for which no work-around solution is known.
- **Category 3 Defects (Serious)** – the number of serious defects found and fixed in this release. Serious defects are those that adversely affect the accomplishment of an operational or mission-essential capability, but for which a work-around solution is known.
- **Category 4 Defects (Annoyance)** – the number of annoyance defects found and fixed in this release. Annoyance defects are those that typically result in user/operator inconvenience, but do not affect any required operational or mission-essential capability.
- **Category 5 Defects (Minimal)** – the number of defects that both have minimal impacts and do not appear in any other category found and fixed in this release. They may be provided for informational purposes.

Delegation – In management, refers to the empowerment of another with the authority to act or represent someone else in the performance of responsibilities.

Deleted code – Refers to code that is removed from an application to improve performance, eliminate functionality or enhance packaging of the module.

Deliverable – Product, release or increment developed, tested, packaged and provided to satisfy customer documented needs and/or requirements.

Design recovery – Refers to techniques used to recreate design abstractions from a combination of code, documentation, personal experience, and general knowledge about the problem and the applications domain.

Development phase – Refers to one of a group of interrelated activities in a development life cycle (design, coding, testing, etc.).

Deviation - A specific written authorization, granted prior to the manufacture of an item, to depart from a particular requirement(s) of an item's current approved configuration documentation for a specific number of units or a specified period of time.

Direction – Management activities conducted to energize, motivate, and guide personal behavior for the purpose of achieving organizational goals.

Distribution management – Refers to managing the distribution of software patches or releases to field sites. The process involves ensuring that each site receives the correct configuration, i.e., that tailored and configured for their use.

Documentation – A list of the documents available to the maintenance team. This typically includes items such as requirements and design specifications, test plans/procedures, and manuals, both users and maintenance.

Domain - A sphere of activity, concern, or function; often refers to an application area like communications or human resources.

Earned value – Refers to a measure of budgetary performance used to relate actual expenditures to technical achievements as determined by milestone completions.

Effectiveness – Refers to a measure of the extent to which plans are realized and results achieved.

Efficiency – Refers to the relationship between the resources used and the results achieved.

Effort – The amount of work required to finish an activity or task expressed in staff-months or staff-hours of labor.

Emergency maintenance - Unscheduled corrective maintenance and repairs performed to keep the software operational [ISO/IEC 14764].

Engineering Change Proposal (ECP) - A proposed engineering change and the documentation in which the change is described, justified, and submitted to the Government for approval or disapproval.

Enhancement – Refers to added capabilities to an existing capability in a manner that preserves its previous functionality and performance.

Equivalent Source Lines of Code (SLOC) – The equivalent new source lines of code are determined by weighting counts of new, reused, modified, deleted and auto-generated code using the relative percentage effort expended in producing them. For example, reused lines are employed as-is with no code and design modifications. The only effort involved is testing them to ensure that they function and perform as expected. If the testing effort is thirty percent of a new effort, one can assume that rerunning regression tests might consume at most twenty-five percent or about seven and one half percent of the effort for new code.

Error – The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or conditions discovered during testing.

Escape – Defects originating in one phase that are not identified and repaired until a later phase of the life cycle (e.g., requirements defects found during testing). Escapes should be avoided because they lead to higher costs for defect repair (i.e., a defect caught in-phase is much cheaper to fix than one caught out-of-phase).

Estimate – Most knowledgeable forecast of the resources (\$, people, equipment, facilities, etc.) needed in the future to complete a maintenance task.

Exit criteria – Refers to specific accomplishments or conditions that must be achieved before an effort will be allowed to progress further in the life cycle.

Facility support – Refers to support provided for those software facilities used to operate, maintain and sustain operational software. Such facilities include those used to develop, test, distribute and field operational releases to the field.

Field support – Refers to support provided to those using software baselined products in the field where actual operational conditions govern how the system performs.

Field testing – Refers to testing the software under actual operational conditions in the field using deployed systems.

Framework – Semi-completed software system designed to be used to generate or create a new instance of itself from a template.

Full Time Equivalent (FTE) – The number of full time equivalent software personnel who work as members of the maintenance team.

Function points – A standard measure of software size and complexity as seen by the end-user determined from a specification by counting the number of unique inputs, outputs, inquiries, interfaces and logical internal files.

Government Off-the-Shelf (GOTS) – Like COTS packages, hardware and software packages provided by the government normally as-is without support using some form of an open source license.

Help desk calls – A count of the number of calls to the Help Desk for this application made during the maintenance period.

Host machine – The computer used to develop software intended to be operated on another computer.

Incremental development process – Refers to the manner in which software is development. Incremental development refers to software releases being developed in parallel in different builds after the requirements and architecture for it have been formulated.

Independent test and verification – Refers to testing performed by an independent organization typically aimed at incrementally verifying and validating that the work products developed by a software maintenance supplier will satisfy specified requirements and work as intended in their intended operational environment.

Information assurance – The technical and management measures designed to ensure the confidentiality, possession or control, integrity, authenticity, availability, and utility of information or information systems. The term, which has spread from government use to common parlance, is sometimes synonymous with information security.

Information Assurance Vulnerability Alerts/Assessments (IAVA) – Refers to those reviews conducted and repairs made to address evolving information assurance vulnerabilities and threat requirements.

Information technology – Refers to broad category of products and services based on digital technologies that are used for the creation, storage, and use of information. Computer hardware, software, communications media and content, and telecommunications equipment and services are manifestations of information technology.

Infrastructure – Underlying framework of an organization or system, including organizational structures, policies, standards, training, facilities, and tools, that supports its ongoing performance.

In-sourcing – Refers to acquiring maintenance capabilities by requiring external organizations and/or independent contractors/consultants who were hired to perform these tasks to become employees of the life cycle support organization.

Integration facility – A facility established to integrate and test the software prior to its delivery to the user and or customer for operational use. The facility integrates hardware and software typically with a man-in-the-loop to qualify the software for delivery to the field.

Integrity – The property of a system or component that reflects its logical correctness and reliability, completeness and consistency.

Intellectual property – Intangible output of the rational thought process that has some intellectual or informational value and is normally protected via using copyrights, patents, and/or trade secrets.

Interoperability testing – Refer to testing aimed at ensuring that two or more software programs will work together and with other systems to exchange information harmoniously.

Key performance indicator – Refer to quantifiable measurements, agreed to beforehand, that reflect critical success factors associated with organizational or project performance.

Key process area – Cluster of related activities that, when performed collectively, achieve a set of goals considered to be important for establishing process capability.

Latent defect – A hidden flaw, weakness or imperfection that may cause failure or malfunction that is not discoverable by reasonable inspection until after delivery.

Leadership - In management, the ability to influence the behavior of others and focus it toward the achievement of accepted goals.

Legacy – Software developed on one project that has the potential for reuse on another.

License management – Refers to the process the managing licenses across their life cycle and ensuring that they are kept up-to-date and current.

Life cycle – Refers to the series of stages that a software product goes through during its life time (i.e., product conception through retirement).

Life cycle support center – Refers to centers established by the customer.

Load balancing – Refers to the process of balancing the maintenance workload by varying what parts of the effort will be eliminated due to constraints.

Logistics support – Refers to the systematic acquisition and distribution of those facilities, equipment, software, spares, information and technical personnel needed to maintain and sustain the software through its required operational life.

Maintainability – The ease, in which the software can be maintained, enhanced, adapted, or corrected to satisfy specified requirements [IEEE 610.12-90].

Maintainer – The person or organization that performs the maintenance activity. The workforce may be made up of contractors, employees, or a mix.

Maintenance – Refers to the process of keeping a software product current after delivery. This involves updating the product to address new functionality and repairs and sustaining the facilities and infrastructure required to accomplish this task.

Maintenance action – The action or actions taken by the maintainer to add capability, fix a problem, optimize performance and/or keep the software operational.

Maintenance environment – The set of facilities, equipment, software and other resources used to maintain and sustain the system. Such an environment differs from the one used for development in that it uses actual operational equipment and external interfaces rather than simulated ones.

Maintenance percentage effort – The percentage of the effort devoted to performing maintenance and support work during the maintenance period.

Maintenance plan – Refers to a document that sets out the activities, schedules, practices and resources to be used to maintain and sustain a software product and generate a software release.

Maintenance process – The set of processes used to maintain and sustain software after it has been delivered to the field. These processes involve more than just software development. For example, they may address distribution control and adaptation to make the software work at different sites.

Maintenance schedule or duration - The calendar time spent to generate the version or release during maintenance from its start to its actual delivery date.

Management – Getting things done through work of other people.

Management percentage effort – The percentage of the effort devoted to performing management tasks during the maintenance period.

Management reserve – In project management, time and budget set aside for contingencies.

Maturity level – A well-defined evolutionary plateau that is reached as the organization moves toward achieving a mature process.

Measures – Variables that are quantified to provide insight into a project or organization's information needs.

Measurement – In management, the process of collecting, analyzing, and reporting metrics deemed useful in assessing status, progress, performance, and/or trends and providing insight into a project or organization's information needs.

Memorandum of Agreement (MOA) – Refers to the terms and conditions of an Agreement entered into by two or more organizations who decide to work together on some form of collaboration or common venture.

Metric – Quantitative measure of the degree to which a software system, process, or component possesses a given attribute. Error density, for example, provides an indicator of software reliability.

Middleware – Layer of software that sits between the operating system and applications that provides computing services through a single programming interface.

Milestone – Schedule event for which some person or organization is held accountable and that is used to gauge progress.

Modified code – Refers to code that is change to alter functionality, fix bugs or improve performance of the module.

Motivation – In management, refers to the act of influencing the behavior of others through the combined use of incentives and rewards.

Non-Disclosure Agreement – Refers to the terms and conditions of an Agreement entered into by two or more organizations aimed at protecting propriety information like intellectual property or classified materials.

Non-repudiation – For software entities that act as users (e.g., proxy agents, web services, peer processes), the ability to prevent the software-as-user from disproving or denying responsibility for actions it has performed.

Operations – Pertaining to that period of the life cycle when the software product is being employed operationally by its users to perform its intended function(s).

Operational baseline – Refers to the baselined configuration of the operational software release that is performing in the field along with all relative ancillary (User's Manual, as-built comments, etc.) and descriptive materials (Version Description Document, etc.).

Operational concept – For maintenance, refers to the maintenance strategy that will be followed for the release (in-house/contracted, in-sourced/out-sourced, etc.).

Operational life – Refers to the anticipated time period in which the software will be maintained and sustained in the field.

Operational support – Refers to hands-on support provided to those using and maintaining the operational software. The activity includes those staffing the help desk and handling requests for information from operators and users in the field.

Operational testing – Testing that is conducted in the field to demonstrate that the software satisfies its requirements, performs as expected, and works in its actual operational environment.

Organizational type – The type of organization tasked to perform the maintenance effort; commercial, government or captive support.

Out-sourcing – Acquiring maintenance capabilities by hiring an external organization or independent contractors/consultants to perform tasks at a price typically much lower than it would cost to do the work yourself.

Patch release – Refers to software being released to the field to repair minor problems or make fixes that do not require a new version.

Patching – Performing emergency repairs by installing temporary fixes in the software's object code without reassembling or recompiling from the source program.

Paradigm – Modeling approach used for the software development and maintenance process (waterfall, incremental development, etc.).

- Pareto analysis** – Refers to the process of ranking causes of defects from the most to the least significance.
- Parts** – Refers to the acquisition of those parts needed to kept the system operational during the maintenance phase of the life cycle.
- Perfective maintenance (or change)** - Modification of a software product performed after delivery to improve either its performance or maintainability [ISO/IEC 14764].
- Performance** – The degree to which the software accomplishes its designated functions within given constraints, such as speed, accuracy, or memory utilization.
- Performance evaluation** – Refers to a formal appraisal of the computer system performance.
- Planning** - Management activities conducted to establish future courses of action at all levels of an organization. At the top, plans tend to be strategic. At lower levels, plans tend to be tactical. At all levels, plan set the standards against which progress is measured.
- Platform** – The hardware/software configuration on which the application runs (PC/Windows, Mainframe/Posix, etc.).
- Post-delivery** – The point in the life cycle after the software has been delivered and when it is operational in the field. This period includes retirement.
- Power** – In management, refers to the perceived ability of one person to influence the action of others.
- Preventive maintenance** – Refers to the modification after delivery of the software application to correct latent defects in the product before they become problems. This often includes redesign, restructuring and/or upgrade of the software to make it easier to maintain.
- Pricing** – In management, the process of determining how much to charge a customer or user for products and services. Costing and pricing are separate activities. Organizations can price services for less than their cost and still make a profit because of economies of scale.
- Problem report** – A document used to identify and describes failures detected in a product. Typically, problems are ranked by impact and actions taken accordingly.
- Process** – Sequence of steps performed for a given purpose, for example, the software repair process.
- Process maturity** – Refers to the extent to which a process is explicitly documented, managed, measured, controlled, and continually improved.
- Product** – Software and all of its associated work products (configuration index, readme files, documentation, etc.) that represent the current configuration of the operational baseline.
- Product support** – Refers to product support provided during maintenance by configuration management, quality assurance, safety, reliability and other specialized organizations.
- Project** – Organized undertaking that uses human and physical resources in one effort to achieve a specific goal.
- Project management** – Refers to a form of organization in which all of the people on the project report to the project manager, either directly or indirectly.

Project management oversight support percentage effort – The percentage of the effort devoted to providing oversight and supporting other project office tasks. In this arrangement, this effort is accomplished as a service in support of the customer.

Prototype – Refers to a model of a product often built to address/assess risks and demonstrate feasibility or critical features.

Qualification testing – Testing conducted by the maintainer and witnessed by the acquirer (as appropriate), to demonstrate that a software product satisfies its requirements, performs as expected, and is ready for operational use in its target environment.

Quality assurance – A set of activities conducted to provide adequate confidence that the adopted software processes are being used and the product generated conform to established technical requirements.

Quantitative control - Refers to quantitative or statistically-based techniques used to analyze a software process, identify the causes of variations in its performance, and bring its performance within well-defined limits.

Reengineering – Refers to the process of examining and updating a system with the purpose of reconstructing it in some new and better form.

Regression test – Refers to the selective testing of software to verify that modifications made to it have not caused unintended effects and that the software system and/or its components still comply with its specified requirements and performs as intended.

Regression test baseline – The test baseline that identifies the tests required to revalidate requirements once changes are made to the code. Such tests, mappings and the related test scripts, cases, procedures and results are kept under configuration control to ensure changes to them are controlled during maintenance.

Release – A software version that is currently under configuration control and is made available to potential users for a specific purpose (replace previous version to fix problems and add functionality, beta testing, etc.).

Release Description Document (RDD) A document that accompanies and identifies a given software release. Typical contents include an inventory of parts and a description of the steps that need to be taken to build the program.

Repair – Actions taken on a non-conforming product to find and fix problems so that it will perform as intended.

Repair time – The average number of hours elapsed from the time the defect was detected until it was repaired.

Replace – Actions taken on a non-conforming product to make it acceptable for the intended use during maintenance using a functionally-equivalent substitute in its place.

Re-qualification testing – Refers to those tests run to re-qualify a release after changes have been made to it.

Requirement – A condition or capability needed by a user to solve a problem or achieve an objective.

Resources – Intellectual and financial possessions available to an organization.

Restructuring – Refers to transforming from one representation to another; i.e., like the move from structured to object-oriented design.

Retirement – Permanent removal of support or of the product from operations.

Retirement phase – Refers to the period of time in the software life cycle during which support for a software product is terminated.

Retrofit - The incorporation of new design parts resulting from an approved engineering change to an item's current approved product configuration documentation into already accepted and/or operational items.

Reusable software – Software designed and implemented to be reused without modification.

Reused code – Refers to code that is instantiated and used within a module to provide needed features and functionality.

Reverse engineering – Refers to the process of analyzing a system in order to understand its architecture so that you can transform it into another representation or abstraction while preserving its salient functionality.

Rework – To redo again because the original objective established for the process had not been properly accomplished.

Risk – In financial circles, refers to exposure to loss.

Risk management – Process of identifying, analyzing, quantifying, and developing plans to eliminate or mitigate risk before it harms a project.

Safety analysis and test – Refers to those activities conducted to both ensure and demonstrate that the safety considerations spelled out in the requirements specification are integral to the software's design and construction.

Schedule – Actual calendar time budgeted for accomplishing goals established for activities or tasks associated with a maintenance action.

Scheduling – In management, refers to the process of allocating and interrelating tasks within the schedule. The activity is like figuring out a jigsaw puzzle, especially when many of the tasks must be done in parallel.

Security – Refers to protection of information so that unauthorized persons cannot read or modify it and unauthorized persons are denied access to it.

Security assurance – The basis for gaining justifiable confidence that the software will consistently exhibit all properties required to ensure that the software, in operation, will continue to operate dependably despite the presence of sponsored (intentional) faults.

Service-oriented architecture – Refers to an architecture that provides its users with a loosely-integrated suite of services (authentication, database, etc.). Underlying and enabling all of orchestration involved is metadata defined in sufficient detail to describe not only the characteristics of these services, but also the data that drives them.

Software Change Request (SCR) - A mechanism used to track requests for change, their implementation and satisfactory delivery as part of a software release. These requests communicate requirements that must be accommodated by the current release.

Software integration – Refers to the process of putting selective software components together to provide a set of the capabilities that the final software product will possess.

Software inventory – refers to the total number, size and characteristics of all of the software programs that an organization owns or licenses.

Software license – Revocable rights to use software in particular places or identified platforms in specified ways.

Software life cycle – Refers to the period of time that begins when a software product is conceived and ends when the product is retired from use. The operations and maintenance portion of the life cycle commences after delivery to the user and ends when the product is retired from use.

Software maintenance - Software maintenance is the process of modifying a software system or its components after delivery to correct faults, improve performances or other attributes, or adapt to a changed environment [IEEE 1219-98].

Software reliability – Refers to the probability of failure-free operation of software for a specified period of time in a specified environment.

Software requirements – The number of functional, performance and/or interface requirements established for a release typically in either a specification or in the applicable software change requests.

Software size – The logical size of the software measured in non-blank, non-comment equivalent source lines of code (SLOC). Equivalent SLOC represents a normalized measure of the equivalent new source lines of code that will be generated during the maintenance cycle [see measures which is Part B of this report]. Normalization is normally accomplished by looking at the relative work performed when developing the code.

Software Trouble Report (STR) – A mechanism (sometimes called a trouble ticket) used to track the detection, reporting and resolution of some type of problem as part of a software release.

Spares – Refers to the acquisition of those spares needed to keep the system operational during the maintenance phase of the life cycle. Spares are included in this software glossary because the software team is often tasked to fix hardware during software maintenance.

Specification Change Notice (SCN) - The incorporation of new design parts resulting from an approved engineering change to an item's current approved product configuration documentation into already accepted and/or operational items.

Sponsor – An individual or organization that acts as a proponent for change and for providing the resources for accomplishing it per the agreed-upon schedule and budget.

Staff – Persons assigned to an organization to do the work.

Staffing – Refers to those management activities conducted to acquire, develop, and retain staff in an organization.

Stakeholder – An individual or organization that has a vested interest in the maintenance of the product and who collaborates with others to identify requirements, prioritize repairs and increase the product's fitness for use.

Subject Matter Expert (SME) – An individual who is recognized as an expert in particular area or topic.

Supplier – Those third parties that contribute directly to the success of an organization or project. They include, but are not limited to, contractors, subcontractor, vendors and independent contractors and consultants who supply services on a time and materials basis.

Support equipment – Refers to equipment required to maintain, test, and/or operate an item, or facility, in its intended environment.

Sustainability – Refers to the software product attribute of being maintainable in an operational environment, whether or not the product is in use.

Sustaining engineering percentage effort – The percentage of the effort devoted to performing sustaining engineering tasks during the maintenance period.

Sustaining engineering – Refers to that continuing engineering and technical support that is needed to sustain maintenance operations. This activity includes, but is not limited to, such activities as user training and support, staffing a help desk, keeping facilities and equipment up-to-date, performing configuration management, managing software licenses, performing quality assurance, managing networks and administering security.

Sustainment – The maintenance and repair activities necessary to keep software operational.

Tailoring - The process by which individual requirements (sections, paragraphs, or sentences) of a specification, standard, or data requirement are evaluated to determine the extent to which they are most suitable for a specific system, and the deletion of some requirements to ensure that each achieves an optimal balance between operational needs and cost.

Task – Smallest unit of work subject to management accountability. A task contains a well-defined work assignment for one or more team members. The specification for the work to be performed is documented in a work package. Related tasks form activities.

Team size – The average size of the team that is assigned to perform the maintenance and support function reported in FTE (reported in whole numbers), independent of whether or not they are available.

Technology transfer – Process used to prove, transfer, and put technology into widespread use within an organization.

Test effectiveness – A measure of the relative coverage that the test scripts provide as a function of the total number of requirements incorporated and defects fixed within a release.

Test environment – The totality of equipment, facilities and software tools dedicated to support the update, repair and testing of a new release prior to its being sent to the field.

Test facilities – Refers to those facilities that have been put into place to integration and test the software to ensure that it operates as expected under realistic operational conditions.

Test percentage effort – The percentage of the effort devoted to performing independent testing tasks during the maintenance period. In this arrangement, the test team performs integration and testing tasks across projects as a service.

Test procedure – Standardized and documented process for performing a test.

Test script – Detailed instructions for the set-up, execution, and evaluation of the results of testing.

Threat (security) – Any entity, circumstance, or event that can cause potential harm or damage a software system or component via unauthorized access, destruction, modification, and/or denial of service.

Tool – Refers to software used to assist in automating tasks or processes associated with the analysis, synthesis, test and evaluation, maintenance and/or performance improvement of models, designs, documentation and software that are part of the product.

Tracking – In management, refers to the process of identifying the cost and schedule variances by comparing actual expenditures of effort and time with projections.

Training – Planned and orderly development of skills and abilities needed by personnel to perform their jobs.

Transition - The process employed to transfer the software and responsibility for its maintenance and support from the developer to the maintainer.

Turnover – The point in the life cycle when the software and responsibility for its maintenance and support is transferred from the developer to the maintainer. For this transfer to occur, all terms and conditions spelled out in the transfer agreement must be satisfied (or waived).

Type of money – Refers to the kinds of money that the government uses to fund programs. The types of money used for maintenance budgets include R&D money for new requirements, Operations & Maintenance (O&M) money for repairs, production money for sustaining activities, and others. Each type of money has a purpose. Because of this, each type also has restrictions on what it can and cannot purchase.

Uncertainty – In management, refers to the degree of entropy associated with the information used to make decisions.

Useful life – Refers to the period over which an asset like software when capitalized can be reasonably used in a trade or business.

User – An individual or group of individuals that use the software operationally in the field.

User support – The process of providing user support including training via mentoring, the staffing of a help desk and development and use of a web site.

Validation- The process of evaluating software to determine whether or not it satisfies specified requirements.

Vendor Coordination – Refers to those activities conducted to coordinate the delivery and update of vendor supplied products that are essential ingredients of a new software release.

Verification – The process of evaluating software as it is developed to determine whether or not it satisfies the conditions imposed on it at the start of each phase of development.

Version – A release of the software that has been placed under configuration control and to which changes are permitted only when approved by the change control authority, typically a Change Control Board (CCB).

Version Description Document (VDD) – A document that accompanies and identifies a given version of the software. Typical contents include an inventory of parts, identification of changes incorporated into the version, and installation and operating information unique to the version described.

Vulnerability (security) – A development fault or weakness in deployed software that can be exploited with malicious intent by a threat with an objective or subverting (violation of integrity) or sabotaging information handled by the software. Vulnerabilities can originate from weaknesses in the software's design, faults in its implementation (bugs), or problems in its operations or use.

Waiver - Written authorization to accept an item, which during manufacture, or after having been submitted for Government inspection or acceptance, is found to depart from specified requirements, but nevertheless is considered suitable for use "as is" or after repair by an approved method.

Waterfall development process – Refers to the manner software is developed. Software requirements are developed first and then design, coding and testing is carried out in sequence with feedback being incorporated along the way.

Weakness (security) – Refers to a flaw, defect, or anomaly in the software that can be potentially vulnerability to be exploited when the software is placed into operations. A weakness may originate from a flaw in the software's security requirements or design, a defect in its implementation, or an inadequacy in its operational and security procedures and controls.

Work Breakdown Structure (WBS) – Family tree that organizes, defines, and graphically illustrates the products, services, and tasks necessary to achieve project objectives.

Work package – Specification of the work to be accomplished in completing a function, activity, or task. A work package defines the work product(s), the staffing needs, the expected duration, the resources planned to be used, the acceptance criteria, the responsible individual(s), and any special considerations for the work.

Work product – Artifact associated with the execution of a practice (document, code, etc.).

Part B: Software Maintenance Measures

B.1 Introduction

This Appendix discusses potential measures that have been developed for addressing management's software operations, maintenance and sustainment information needs. Along these lines, Table B.1, List of Potential Maintenance Measures, suggests a number of such measures that can be employed to gain insight into overall effectiveness of these efforts. This Table was developed by Practical Systems and Software Measurement (PSM) workshop participants during the past two years. It is a work in progress that will be updated as we attempt to collect data from target projects.

B.2 Proposed Maintenance Measures

The following maintenance measures are proposed for use in our data collection efforts:

1. **Equivalent Source Lines of Code (SLOC_{EQUIV})** – the approximate size of the software release in equivalent new logical source lines of code (SLOC) computed using the following formula:

$$\text{SLOC}_{\text{EQUIV}} = \text{SLOC}_{\text{NEW}} + (0.07) \text{SLOC}_{\text{RUSE}} + (0.50) \text{SLOC}_{\text{MOD}} + (0.65) \text{SLOC}_{\text{DEL}} + (0.10) \text{SLOC}_{\text{AGEN}}$$

Where:

- **SLOC_{NEW} - New (added) lines** – The number of new source lines of code added to the new version or release.
 - **SLOC_{RUSE} - Reused (old) lines** – The number of existing source lines of code that were included in the new version or release. These lines are not changed in any way.
 - **SLOC_{MOD} - Modified (changed) lines** – The number of existing source lines of code that were changed and included in the new version or release. These lines can include design modified, code modified and/or integration modified elements.
 - **SLOC_{DEL} - Deleted lines** – The number of existing source lines of code that were deleted from the previous version or release.
 - **SLOC_{AGEN} - Auto-generated lines** – The number of auto-generated source lines of code added to the new version or release. Auto-generated code is produced using specialized tools at a pace far exceeding manual development.
2. **Number of Escapes** – the number of defects originating in one maintenance release that are not detected and repaired until a later release. The total number of escapes (Escapes_{TOT}) can be computed as follows:

$$\text{Escapes}_{\text{TOT}} = \text{Defects}_{\text{OIPR}} + \text{Defects}_{\text{OICR}}$$

Where:

- **Defects_{OICR}** – Defects originating in Current Release – the number of defects whose origins occurred in the current release and were caught.
- **Defects_{OIPR}** – Defects originating in Previous Releases – the number of defects whose origins occurred in previous releases that were not caught.
- **Defects_{TOT}** – Total defects in this release (sum of Defects_{OICR} and Defects_{OIPR})

3. **Number of Systems Requirements (REQT_{TOTAL})** – the number of systems requirements for a software release expressed in a complete sentence containing both a subject and predicate. Such requirements can be tabulated based on the number and compositions of Software Change Requests (SCRs). These sentences shall consistently use the verb “shall” or “will” or “must” to show the requirement’s mandatory nature. The whole requirement specifies a desired end goal or result and contains success criterion or other measurable indication of quality. The number of requirements is determined using the following formula:

$$\text{REQT}_{\text{TOTAL}} = \text{REQT}_{\text{ADD}} + \text{REQT}_{\text{DEL}} + \text{REQT}_{\text{CHG}} + \text{REQT}_{\text{DEF}}$$

Where:

- **REQT_{ADD} – Added requirements** – The number of new requirements added to the current version or release.
- **REQT_{DEL} – Deleted requirements** – The number of existing requirements deleted from the previous version or release.
- **REQT_{CHG} – Changed requirements** – The number of existing requirements modified for the current version or release.
- **REQT_{DEF} – Deferred requirements** – The number of new and/or deleted requirements deferred from the new version or release solely due to funding constraints.

Table B.1 – List of Recommended Maintenance Measures

Information Categories	Measureable Concepts	Prospective Measures	Description
Project (by release)	Cost	Estimated/actual cost (\$)	Estimated and actual cost in \$ by project, release or version separated by year into the following categories: (1) labor costs, (2) license costs, (3) travel costs, (4) facility costs (including equipment, maintenance, etc.), (4) Other Direct Costs, (5) indirect costs, and (6) profit, as applicable.
	Effort	Estimated/actual hours (directly chargeable labor hours)	Estimated and actual number of directly chargeable labor hours by project, release or version separated by year into the following categories: (1) requirements changes, (2) defect repairs, (3) sustaining engineering including user, help desk, training and facility support, (4) test preparation, (5) test dry runs, (6) regression test conduct, (7) FQT conduct, (8) interoperability test conduct, (9) Quality Assurance (QA), (10) Configuration Management (CM), (11) systems administration, and (12) information assurance.
	Schedule	Planned/actual schedule (days)	Schedule including planned and actual start and delivery dates in calendar days by project, release or version.
	Product size	Estimated/actual size (SLOC)	Estimated and actual size in ESLOC by project, release or version separated into the following categories: (1) new SLOC (added), (2) reused SLOC (old), (3) modified SLOC (changed), (4) deleted SLOC, and (5) auto-generated SLOC.
	Team size	Planned/actual team size (FTE)	Planned and actual average and peak staff count in FTE by project, release or version.
	Contractors used	Percent utilization	Planned and actual contractor percentage utilization by project, release or version measured in terms of labor-hours expended separated into (1) government % and (2) contractor %.
	Languages used	Percent utilization	Planned and actual language percentage utilization by project, release or version by programming language including assembly and machine language.
Enhancements	Requirements volume	Estimated/actual requirements changes	Estimated and actual number of requirements as a function of Software Change Requests (SCRs) by project, release or version separated into the following categories: (1) requirements added, (2) requirements deleted, (3) requirements changed, (4) requirements deferred, and (5) total number of requirements including those that currently exist.
	Enhancement size	Estimated/actual size (SLOC)	Estimated and actual size in ESLOC as a function of SCRs by project, release or version separated into the following categories: (1) new SLOC (added), (2) reused SLOC (old), (3) modified SLOC (changed), (4) deleted SLOC, and (5) auto-generated SLOC.
	Team size	Planned/actual team	Planned and actual average and peak staff count in FTE by project, release or version.

		size (FTE)	
	Requirements effort	Estimated/actual hours per change	Estimated and actual number of directly chargeable labor hours needed to implement the requirements change requests by project, release or version.
	Requirements backlog	Planned/actual number of requirements deferred/dropped	Planned and actual number of requirements changes at the start and end of the project, release or version separated into the following two categories: (1) requirements deferred and (2) requirements dropped.
	Test volume	Planned/actual number of tests	Planned and actual number of tests executed to validate changes made by project, release or version separated into the following categories: (1) regression tests, (2) FQT (Formal Qualification Tests) or acceptance tests, and (3) interoperability tests.
	Test effectiveness	Planned/actual number of tests automated	Planned and actual number of tests needed to validate changes automated by project, release or version by percentage of total separated into the following categories: (1) regression tests, (2) FQT or acceptance tests, and (3) interoperability tests.
Repairs	Repair volume	Estimated/actual number of defects by age	Estimated and actual number of defects as a function of Software Trouble Reports (STRs) by project, release or version classified by age in calendar days into the following five categories: (1) Category 1 (catastrophic), (2) Category 2 (critical), (3) Category 3 (serious), (4) Category 4 (annoyance), and (5) Category 5 (minimal).
	Repair size	Estimated/actual size (SLOC)	Estimated and actual size in ESLOC as a function of STRs by project, release or version separated into the following categories: (1) new SLOC (added), (2) reused SLOC (old), (3) modified SLOC (changed), (4) deleted SLOC, and (5) auto-generated SLOC.
	Team size	Planned/actual team size (FTE)	Planned and actual average and peak staff count in FTE by project, release or version.
	Repair effort	Estimated/actual hours per repair	Estimated and actual number of directly chargeable labor hours needed to implement the approved repairs by project, release or version.
	Latent defect volume	Planned/actual number of latent defects	Planned and actual number of latent defects at the start and end of the project, release or version separated into the following two categories: (1) latent defects found and fixed and (2) latent defects found and deferred.
	Escape rate	Planned/actual number of escapes	Planned and actual number of defects at the start and end of the project, release or version that are identified in-phase and out-of-phase.
	Defect backlog	Planned/actual number of defects remaining	Planned and actual number of defects at the start and end of a project, release or version classified by the following categories: (1) Category 1 (catastrophic), (2) Category 2 (critical), (3) Category 3 (serious), (4) Category 4 (annoyance), and (5) Category 5 (minimal).

	Test volume	Planned/actual number of tests	Planned and actual number of tests needed to validate the repairs implemented by project, release or version separated into the following categories: (1) regression tests, (2) FQT (Formal Qualification Tests) or acceptance tests, and (3) interoperability tests.
	Test effectiveness	Planned/actual number of tests automated	Planned and actual number of tests needed to validate repairs automated by project, release or version by percentage of total separated into the following categories: (1) regression tests, (2) FQT or acceptance tests, and (3) interoperability tests.
Information assurance (IA)	IA effort	Estimated/actual hours per IA action	Estimated and actual number of directly chargeable labor hours by project, release or version separated by year into the following categories: (1) anti-tamper, (2) cyber protection, (3) DITSCAP, and (4) sensitive programs.
	Team size	Planned/actual team size (FTE)	Planned and actual average and peak staff count in FTE by project, release or version.
	IA effectiveness	Planned/actual IA action automated	Estimated and actual number of cyber attacks mitigated as a function of vulnerabilities identified via means of protection selected.