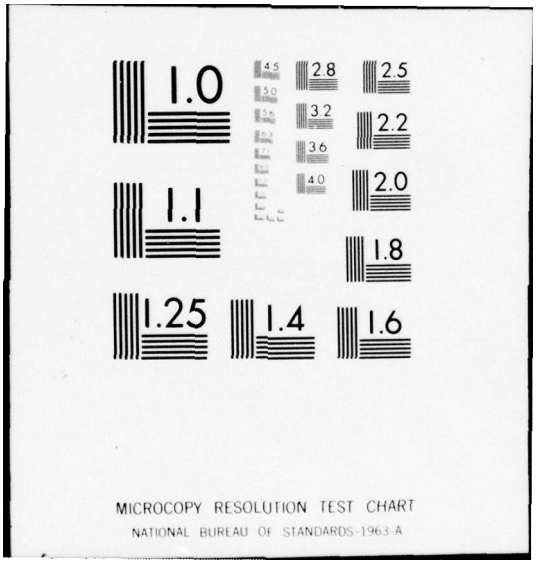
AD-A036 265 SYSTEM DEVELOPMENT CORP SANTA MONICA CALIF F/G 9/2
SOFTWARE DATA COLLECTION STUDY. VOLUME VIII. GLOSSARY.(U)
DEC 76 N E WILLMORTH F30602-75-C-0248
UNCLASSIFIED SDC-TM-5542/08/01 RADC-TR-76-329-VOL-8 NL

1 OF 1
ADA036265

END

DATE
FILMED
3-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RADC-TR-76-329, Volume VIII (of eight)
Final Technical Report
December 1976

SOFTWARE DATA COLLECTION STUDY
Glossary

System Development Corporation

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED: Richard T. Slavinski

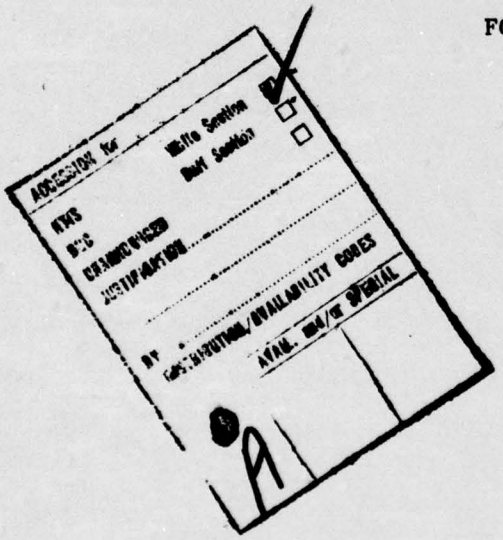
RICHARD T. SLAVINSKI
Project Engineer

APPROVED: Alan R Barnum

ALAN R. BARNUM
Assistant Chief
Information Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office



Do not return this copy. Retain or destroy.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AD036347

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-76-329-Vol ~~I~~ 8 ~~(eight)~~	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): SOFTWARE DATA COLLECTION STUDY, Volume VIII. Glossary.	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, Jun 75 — Jun 76
6. PERFORMING ORG. REPORT NUMBER: SDC-TM-5542/08/01	
7. AUTHOR(s): N. E. Willmorth	8. CONTRACT OR GRANT NUMBER(s): F30602-75-C-0248
9. PERFORMING ORGANIZATION NAME AND ADDRESS: System Development Corporation, 2500 Colorado Avenue, Santa Monica CA 90406	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63728F, 55500810
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (ISIS), Griffiss AFB New York 13441	12. REPORT DATE: Dec 76
	13. NUMBER OF PAGES: 37
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer
Richard Slavinski (ISIS)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Definitions
Dictionaries
Terminology

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This volume consists of a glossary of terms pertinent to the technical reports resulting from the RADC Software Data Collection Study. The glossary does not repeat definitions of most commonly used data processing terms.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

339 900

Glossary of Terminology for Data Collection Study

Abstraction

A conceptual simplification that makes what the system is doing clearer and more understandable. Some abstractions simplify the user's conception of the system while others simplify concepts of the designers. Examples of the former are the generation, retrieval, and updating of a data base. Examples of the latter tend to center around hardware resources and data resources.

Acceptance Test

All activities relating to certification that a software system meets the system specifications.

Access, Algorithmic

A technique that employs an operation, such as hashing, to transform some key of a record into an address for that record thereby permitting retrieval and storage.

Access, Chained

A means of storing in and retrieving from a file that has a linked list organization.

Access, Partitioned

A process that involves direct access to a segment of a serial data file and then sequential access of the segment.

Access, Sequential

A process that involves reading or writing data serially.

Accuracy

The degree of freedom from error or the degree of conformity to the truth or the rule. Contrast with precision which deals with the degree of exactness or fineness of measure whether accurate or not.

Address

The number or name that uniquely identifies a register, memory location, or storage device.

Algorithm

A statement of the step-by-step procedure for solving complex problems by simple steps.

Annotation

Any comment or note included in a program, flowchart, or document to clarify a point. Commentary.

Automatic Theorem Provers

Automatic procedures for converting a program into a theorem and proving the properties of the theorem using the facilities of the first order predicate calculus. These procedures, not yet realized, are the object of much current research activity.

Availability

The degree to which a system or resource is ready when needed to process data.

Baselines

A configuration identification document or set of such documents, and also the computer program itself after the product baseline, formally designated and fixed at a specific time during the program's life cycle. Baselines, plus approved changes to those baselines, constitute the current configuration identification. Under baseline configuration management, there are usually three baselines.

Baseline, Allocation

The configuration identification established at end of requirements/ performance phase.

Baseline, Functional

The initial system configuration identification established at end of conceptual phase, normally existing prior to the start of a software development project.

Baseline, Product

The configuration identification established at end of test and acceptance phase.

Batch Processing

(1) Pertaining to the technique of executing a set of computer programs such that each completed before the next program of the set is started. (2) Pertaining to the sequential input of computer programs or data. (3) Loosely, the execution of computer program serially. (4) See also stacked job processing.

Block

A group of words, characters, or items of information considered or transported as a unit as, for instance, a block of code or data. In flow charting, a box or set of boxes representing a logical unit of programming. In data files, the unit of data that is retrieved from a device with one physical read.

Bottom-Up Development

The traditional procedure of software development where the lowest level processing programs are coded first, module tested and made ready for integration with additional programs. Work procedes in this manner up the hierarchy of the design. Additional code in the form of driver programs

is needed to perform the module testing and integration testing. In addition, internally formatted test data has to be prepared by hand or by test data generator programs.

Bucket

A place or unit of storage; for example, a section of storage or storage area, the storage location of a word. Thus, any place where a piece of data or item may be put.

Certification

The process of demonstrating the system meets its customer requirements and guaranteeing that compliance in writing. Also see Product Certification.

CDR

Critical Design Review.

Chain

A series of items linked together as a string of characters or words; a system of storing record sets so that each has a linking field for tracing the field.

Change Status Report

A listing or report of all proposed changes and corrections to a configuration and their current disposition and implementation status.

Characteristics, Environmental

Quantitative parameters describing the operating milieu of a program or project, such as external interfaces and interactions, operating conditions and levels of stress, and the relative efficiency or inefficiency of interfacing elements or agencies.

Characteristics, Functional

Quantitative performance and operating parameters and their respective tolerances.

Characteristics, Structural

Measurable properties of software that qualify the software size, interface descriptions, use of the data base, and use of various language elements.

Characteristics, Software

Quantitative measures of software describing its portability, utility (as described by its reliability, efficiency, and human engineering), and maintainability (as described by its testability, understandability and modifiability).

Chief Programmer Team

A management technique that organizes program production staff with a Chief Programmer as manager and 'super-programmer', being supported by a Backup Programmer, Programming Secretary and programmers as needed.

Commentary

In a computer program, an expression that explains or identifies a particular step in an operation, but that has no effect on the operation of the computer program. Annotation.

Complexity

The degree of complication of a software product, consisting of the combinatorial factor of such measures as the number of control paths, number of shared data references, number of nested loops, number of interactions between system components, user interfaces and hardware. There is presently no one accepted objective measure of software complexity. The subjective complexity measure may be characterized by the following definitions:

Easy - Programs that have few interactions with other system elements, such as a mathematical or logical problem.

Medium - Programs that have some interactions with other system elements, and/or are generalized in functional processing in order to handle multiple or variable inputs/outputs. For example these programs include utility programs, language compilers, schedulers, data management systems.

Hard - Programs that have many interactions with other system elements, such as monitor programs, operating systems, special purpose research programs.

Composite Design

A methodology used that defines and measures the quality of the design, strategies, and techniques of structuring a computer system in terms of its attributes of data, tasks, interfaces, and modules. It incorporates many of the concepts inherent in structured programming design and implementation, as well as giving criteria for accomplishing the design and implementation tasks.

Configuration

(1) The functional and/or physical characteristics of computer program or other configuration item. (2) The group of machines, devices and programs that make up a data processing system.

Configuration Accounting

See configuration status accounting.

Configuration Control

A methodology concerning with procedures for controlling the contents of a software system. A way of monitoring the status of system components, preserving the integrity of released and developing versions of a software system, and controlling the effects of changes throughout the system.

Configuration Control Board

An executive agency, sometimes composed of associated contractors, users and monitoring agencies, charged with determining the content and maintaining the integrity of a system.

Configuration Index

A listing of the contents and structure of a configuration.

Configuration Item

A computer program entity or element identified for configuration control.

Configuration Management

The application of technical and administrative management direction and surveillance to (a) the identification, authentication, and recording of the functional and physical characteristics of a system, (b) the control of changes to identified and authenticated characteristics, and (c) the maintenance of records and issuance of reports on configuration status.

Configuration Status

The current content, structure, and state of development or maintenance of a configuration.

Configuration Status Accounting

The recording and reporting of the approved configuration identification, the status of proposed changes to the approved configuration, and the implementation status of the approved configuration and of the changes to it.

Critical Design Review (CDR)

See Detailed Design Review.

Data

A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or automatic means. Any representations such as characters or analog quantities to which meaning is, or might be, assigned. See Information.

Data Base

A collection of data fundamental to an enterprise.

Data Base, Hierarchical

A logical view of a data base usually associated with a tree structure or data hierarchy.

Data Base, Network

A logical view of a data base usually associated with linked list physical organization. The data base is considered as a series of chains which may be one- or two-way and may intersect.

Data Base, Relational

A logical view of a data base as a table or array consisting of sets of n-tuples.

Data Compression

A technique that saves storage space by eliminating gaps, empty fields, redundancies, or unnecessary data to shorten the length of records or blocks.

Data Hierarchy

A data structure consisting of sets and subsets such that every subset of a set is of lower rank than the data of the set.

Data Item

The smallest logical unit of data. An occurrence of a data item is a representation of a value.

Data Management System

A system for handling and organizing large bodies of data. A data management system contains components for organizing data, storing data, retrieving data, and updating data and a language capability allowing user description of data base organization and user specification of data manipulations.

Data Organization

Any one of the data management conventions for physical and spatial arrangement of the physical records of a data set: The four data management organizations are:

Serial
Full Inverted
Direct
Linked List

These organizations differ in the arrangement of records on storage media, and also in the manner of accessing a particular record. The choice of organization will influence processing speed and efficiency, and also the density with which data can be packed (storage efficiency). In particular, the "HIT RATIO" (number of records accessed per run divided by number of records in data set) will often dictate the choice of organization.

Data Structure

The arrangement and interrelation of records in a file.

Deadlock

Unresolved contention for the use of a resource.

Debug

To detect, locate, and remove mistakes from a routine or malfunctions from a computer. Synonymous with troubleshoot.

Decision Table

A table of information consisting of all contingencies and actions that are to be considered for each possible set of conditions in the description of a problem. Decision tables may have multiple dimensions depending on the complexity of the arguments and actions inherent in the problem. Decision tables are sometimes used in place of flowcharts for problem description and documentation and may be used as a programming language.

Design Language

An artifical language used to state the design of a system or a program in a terse, but precise way, often a pidgin language based on some programming language.

Detailed Design Review (DDR)

A review that normally occurs at completion of the design of a system, subsystem, or level of abstraction. This review consists of a detailed examination of the design of a specific component of software, and may be performed numerous times during the development of the software project. Successful completion of this review established the Detailed Design Baseline. Coding of the component(s) subjected to the review begin upon completion of the review. This term is used interchangeably with Critical Design Review.

Deviation

Specific written authorization, granted prior to the development of an item, to depart from a particular performance, or design requirement, of a specification for a specific number of units, or a specific period of time. A deviation differs from an engineering change in that an approved

engineering change requires corresponding revision of the documentation defining the affected item, whereas a deviation does not contemplate revision of the application specification.

DDR

Detailed Design Review.

Direct Access

(1) Retrieval or storage of data by a reference to its location on a volume, rather than relative to the previously retrieved or stored data. (2) Pertaining to the process of obtaining data from, or placing data into, storage where the time required for such access is independent of the location of the data most recently obtained or placed in storage. (3) Pertaining to a storage device in which the access time is effectively independent of the location of the data. (4) Synonymous with random access.

Direct Access, Storage Device

A device in which the access time is effectively independent of the location of the data. Abbreviated DASD.

Directed Graph

A method of representing a program as a directed graph, whose arcs represent potential flow of control and whose nodes represent basic blocks of code. More, specifically, a basic block is a set of consecutive instructions with the property that whenever the first instruction of the sequence is executed every instruction of the sequence will be operated. The block, or nodes, of the graph are connected by arcs which have the property that each arc is an output of exactly one node and an input to exactly one node, and the output and input arc may be to the same node, i.e., a loop. Each node has at least one input arc, except the entry node, and one output arc, except the exit node. The representation of a program as a directed graph facilitates the analysis of control paths necessary for automatic execution analysis tools.

Discrepancy

Any difference between expected and actual results. Includes deviation, error, failure, fault, mistake, and malfunction.

Discrepancy Report Form (DRF)

A standard form to document reports of discrepancies.

DRF

Discrepancy Report Form.

ECP

Engineering Change Proposal.

Efficiency

The degree to which a task is performed with a minimum consumption of time and resources; in a computer, obtaining maximum throughput with minimum execution time, storage space, and peripheral device utilization.

Emulate

To imitate one system with another such that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated system. Contrast with simulate.

Emulation

The use of programming techniques and special machine features to permit a computing system to execute programs written for another system. Contrast with simulation.

Engineering Change

An alteration in the configuration of a computer program or other configuration item that is delivered, to be delivered, or under development, after formal establishment of its configuration identification.

Engineering Change Proposal (ECP)

Used to document and transmit a Class I change request to a customer for approval.

Entity

A real world thing (person, place, project, etc.) that has attributes associated with it (name, age, address, etc.).

Error

See software error.

Expression

One or more symbols to which meaning is assigned, as a mathematical expression or a logical expression.

Facility

(1) Ease of use. Together with system performance, a major factor on which the total productivity of an installation depends (2) A capability or tool. (3) An installation as computing facility or programming facility.

Fault

A physical condition that causes a device, a component, or an element to fail to perform in a required manner, for example, a short circuit, a broken wire, an intermittent connection. Contrast with Software Fault.

FCA

Functional Configuration Audit.

Field

A set of one or more characters which is treated as a whole. In punched cards, a set of one or more columns, fixed in number, into which some unit of information is regularly entered.

File

A collection of related records treated as a unit. For example, one line of an invoice may form an item, a complete invoice may form a record, the complete set of such records may form a file, the collection of inventory control files may form a library, and the libraries of files used by an organization are known as its data base.

Filter

(1) A device or program that separates data, signals, or material in accordance with specified criteria. (2) A mask.

Formal Program Proofs

Systematic, rigorous techniques of proving that a program produces correct results for all possible inputs. Validation of a program in the same way a mathematical theorem is proved correct, i.e., by mathematical analysis of its properties.

Formal Qualification Review (FQR)

A review that normally occurs at completion of software validation testing to certify that the test results correspond to preestablished acceptance criteria. Successful completion of the FQR establishes the Product Baseline.

Formal Qualification Test (FQT)

That portion of software testing which is conducted in accordance with approved test plans for the purpose of verifying that the software fulfills its requirements. The FQT is a complete and comprehensive test in a continuous test period prior to Functional Configuration Audit (FCA).

FQR

Formal Qualication Review.

FQT

Formal Qualification Test.

Function

A specific purpose of an entity or its charcteristics action. The function of a module is a description of what a program does - the transformation (input to output) that occurs when the module is called.

Functional Configuration Audit (FCA)

A formal examination of the test data for a configuration item's functional characteristics prior to acceptance, to verify that the item has achieved the performance specified by its functional or allocated configuration identification.

Granularity

Concerned with the depth of detail and sampling frequency, this metric describes the relative coarseness or fineness of control to be exercised over software development.

Hash

To transform a key field from its natural or logical form and length to a different representation. The transformed key usually becomes a hash address and relates to a file organization.

Hierarchical Input Process Output (HIPO)

A document technique for describing program logic. HIPO diagrams show data flow and logic flow.

HOL

Higher Order Language.

Higher Order Language (HOL)

Any one of the programming languages more powerful and more general than machine or assembly language (e.g., MOL) such as a POL (procedure oriented language), a design language, a simulation language, or a problem oriented language.

Immediate Access Store

A store whose access time is negligible in comparison with other operating times. Synonymous with zero-access storage, instantaneous storage, and immediate memory.

Immediate Memory

See Immediate Access Store.

Indentation

See Paragraphing.

Index

(1) An ordered reference list of the contents of a file or document, together with keys or reference notations for identification or location of those contents. (2) To prepare a list as in (1). (3) A symbol or a numeral used to identify a particular quantity in an array or similar quantities. For example, the terms of an array represent by X_1, X_2,..., X_{100} have the indexes 1, 2,... 100, respectively. (4) To move a machine part to a predetermined position, or by a predetermined amount on a quantized scale. (5) A table used to locate the records of an indexed data set.

Information

The meaning that a human assigns to data by means of the known conventions used in their representation.

Information Probe

(1) a mechanism - either hardware or software - to recognize or measure the occurrence of an incident in the system in operation. (2) The means by which information is gathered in an instrumentation effort. An information probe can be a milestone product, progress report in a process control effort, or a sequence of code in a program instrumentation effort.

Instrumentation

(1) The controlled selection of points in a process for the insertion of information probes with proper arrangements for gathering the information. (2) Code inserted at strategic points of a program under test designed to collect data during program execution in order to demonstrate the thoroughness of the test effort.

Integration Test

A test to determine the compatibility of two or more systems or system elements, usually of a different kind (i.e., programs and hardware, men and machines and program, etc.), or to ensure the compatibility and proper operation of a new or changed element when it is integrated into an existing system.

Interactive Processing

Pertaining to a mode of computer operation or specific application in which a user entry elicits a response, as in an inquiry system or an airline reservation system. An interactive system may also be conversational, implying a continuous dialog between the user and the operating system.

Interface

A shared boundary between system components. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

Key

(1) One or more characters within an item of data that are used to identify it or control its use. (2) The attribute selected to order records in a file.

Key, Prime

The key that determines the location of a record in a file. For a file of direct organization the prime key is the value hashed. Otherwise, it is the key that is used to index all records in the file.

Key, Secondary

A key, in addition to the prime key, that can also be used to access records of a file. Identification of secondary keys must be supported by construction of an associated index and on-going maintenance during update operations.

Label

One or more characters used to identify a statement or an item of data in a computer program.

Labeling Conventions

Rules in force that control the assignment, generation, composition, structure and use of names, such as labels within a computer program. An element of programming style.

Levels of Abstraction, Hierarchical

A design methodology used to divide simple or complex systems into logical, effective and easily comprehended sets of functionally related modules. Each level of the design are organized in a strict hierarchy, and each level consists of modules that support a functional abstraction or share common resources. The top level is closest to the user and reflects the major interface of the user to the problem.

Machine Oriented Language (MOL)

A programming language that is more like a machine language than a human language.

Maintainability

The extent to which a software product facilitates updating to correct errors and to satisfy new requirements. A maintainable software product is one which is understandable and testable and can be easily modified to rectify a deficiency and/or add new capabilities.

Malfunction

The effect of a fault. Contrast with error, mistake.

Managerial Resistance

Non-cooperative behavior of project personnel in complying with the requirements of a data collection or management control system, such as expressing reluctance, failure to comply, evasion, half truths and falsifications, deliberate attempts to sabotage the system, and other instances of passive and active resistance, often seen as counter-aggressive response to a perceived threat.

Mass Storage Device

A device having a very large storage capacity, 10^{10} bytes or more.

Measurement

Involves collection of quantitative data that reflects computer system resource consumption and performance.

Mensuration

Determining what information to gather, when to gather it, and the weight to be assigned to each measurement.

Milestones

A specific point in the incremental development of software that establishes a baseline product or signifies the completion of a task, and generally involves or requires a management decision of some type.

Milestone Review

A management meeting which examines whether all predefined tasks of a milestone are met satisfactorily.

Mistake

A human action that produces an unintended result. Contrast with error, fault, malfunction.

Modification Transmittal Memorandum

A closure report giving the disposition of a problem opened by a SPR and used to transmit changes and corrections to system elements.

Modify

To alter a part of an instruction, statement, or module.

Modifiability

The extend to which a software product facilitates the incorporation of changes once the nature of the desired change has been determined.

Modular Programming

Division of software into blocks of code called modules. Each module is described by a single function to be performed, with an associated set of inputs and outputs.

Modularization

The decomposition of a software system into smaller components, or modules, with each module generally representing some unique function or abstraction.

Module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, an assembler, compiler, linkage editor, or executive routine. The term module is used interchangeably with program.

Module Testing

The testing of successfully compiled modules independent of other modules to verify the coded logic agrees with the module's specifications.

MOL

Machine Oriented Language.

MTM

Modification Transmitted Memorandum.

Operability

The ease of using and operational availability of a program or system, including its execution resistance to the ill effects of operator errors, overloads and system failures.

Paragraphing

An aspect of programming style used to show the logical structure of a program by indenting sequences of code to identify levels of control nesting.

PCA

Physical Configuration Audit.

PDR

Preliminary Design Review.

Performance

Together with facility, one of the two major factors on which the total productivity of a system depends. Performance is largely determined by a combination of such other factors as throughput, response time, availability, operability, maintainability, efficiency, portability, and reliability.

Performance

A description of how well the program performs its functions, measured in such terms as execution speed, storage size, resource usage, and mean-time-to-failure.

Performance Characteristics

A quantitative description of actual performance attributes of a product.

Performance Level

A measure of how well performance characteristics compare to performance requirements.

Performance Specification

A description of all the operational and functional requirements of a program in sufficient detail to support the design, test, and maintenance of the program. It provides the logical, detailed descriptions of the performance requirements of a program. It serves as a controlling document by which a customer is able to procure software and assess developer compliance.

Performance Requirement

A statement of a performance condition that must be satisfied. For example, a requirement that a compiler be able to process n statements per second.

Physical Configuration Audit (PCA)

The formal examination of the coded configuration of a program element against its technical documentation in order to establish the element's initial configuration identification.

POL

Procedure Oriented Language.

Portability

The extend to which a software product can be readily converted to operate in computing environments other than its current one.

PR

See SPR.

Precision

The degree of exactness with which measure is taken or a quantity is stated as the number of significant digits in a number. Contrast to accuracy, the degree of freedom from error, regardless of precision.

Preliminary Design Review (PDR)

A review that normally occurs a completion of the System Design Phase. Successful completion of this review establishes the preliminary or system-level computer program development specifications, interface specifications, and data requirements specifications in the System Design Baseline. This term is used interchangeably with the System Design Review.

Procedure Oriented Language (POL)

A programming language designed for the convenient expression of procedures used in the solution of a wide class of problems.

Product Certification

Those actions taken by the producer or procurer of a system or by an independent certification agency to establish that the system will perform as required with a reasonable degree of assurance and as the basis for the issuance of warranties and guarantees.

Productivity

The number of units of work, or product units, processed per unit of time or other resource unit. A work unit for data processing is normally expressed as lines of source or object code, pages of documentation, data records compiled, cards punched, computer jobs run, and/or some combination of the above. For comparative evaluations between projects, productivity measures need to be adjusted for differing system size and complexity and differing product quality goals. For example, efficient, tightly written program is smaller than a less efficient, loosely coded program, but the efficient code may represent more "work units" then the loosely written program.

Program Maintenance

The correction of detected errors and/or installation of modifications in a program that has been declared operational. A record of all program errors, corrections, and modifications is usually included in program documentation during program maintenance.

Programming

The design, the writing, and the testing of a program.

Programming, Application

Any programming activity for, or by, a user that directly supports the specific needs of that particular user. This activity does not include the elements of system programming.

Programming, Maintenance

Any programming activity intended to eliminate software faults or to keep programs in satisfactory working conditions, including tests, measurements, replacements, adjustments, improvements, and repairs.

Programming, System

Any programming activity concerned with planning, generating, maintaining, extending and controlling programming support systems to support a variety of users. Especially, such activities connected with an operating system.

Programming Standards

Adopted conventions that define acceptable programming style.

Programming Style

The custom or plan to be followed in writing computer programs. It includes choice of language forms, commenting, labeling conventions, and paragraphing.

Programming Style, Elements of

A handbook providing guidance to program writers to promote compliance to the requirements of programming standards.

Quality Control

The application of measurement, statistical evalation, and the taking of corrective measures in case of deviation from the norm or standard with regard to a product or process; the systematic control of quality of a product or process.

Readability

The extent to which a program is easily read and understood. The concepts of programming style and techniques of structured programming enhance readability.

Record

A collection of related items of data, treated as a unit. For example one line of an invoice may form a record; a complete set of such records may form a file.

Recovery

The ability to restore data and programs to a given execution point after a system failure. Contrast with restart.

Redundancy

In the transmission of information, that fraction of the gross information content of a message that can be eliminated without loss of essential information; in a system, the inclusion of duplicate or alternate system elements to ensure or improve reliability of operation.

Release

Transfer of physical custody and control of software products or documentation from the originator to another organization in a controlled environment, such as a software control center; issuance of a new version of a program.

Reliability, Software

The probability that the instructions to the computer are sufficient to perform the functions as stated in the requirement specification without error within the required data range for a given length of operational time.

Reliability Modeling

Mathematical methods used to predict the reliabilty of software; specifically, the probability density of time between failures or the failure rate.

Repeating Data

A group of data items that can occur more than once when logically associated with other data items that can occur only once per entity.

Resources

Any commodity or service normally expended in a software production effort including people time, calendar time, computer time, memory, auxilliary storage, computer peripherals, money, and materials.

Resource Utilization

Referring to a plan for the assignment of resources (men, machines, money, time, and materials) to project elements (tasks, products, accounts and organizational units), and the accounting for the actual expenditures of these resources in the accomplishment of the project mission.

Response Time

(1) The time between the submission of a item of work to a computing system and the return of results. (2) In systems with time sharing, the time between the end of a block of user input and the display of the first character of system response information at the terminal.

Restart

The ability to continue operations after a system failure, usually from the beginning of a job. Constrast with recovery.

Robustness

The ability of a software system or component to resist the ill effects of changes, including transporting to a new environment, computer, or application. The general strength or health of a software system in continuing to operate correctly under all conditions.

Sampling

(1) obtaining the values of a function for regularly or irregularly spaced, discrete values of the independent variable. (2) In statistics, obtaining a sample from a population.

Sampling Frequency

How often samples, such as milestone products and progress reports, will be taken to control a software development process.

SCN

Specification Change Notice.

SDR

System Design Review.

Security

The degree of protection of data and/or programs from illegal or unauthorized access or use. Generally referring to the classification level of the data or program in question.

Segment

A part or portion of a larger system entity, such as a system segment, program segment or data segment. Segmentation may be somewhat arbitrary, such as to fit a large program or data set into core loads, or into pages of secondary storage, or it may represent logical subunits, as a structured program segment would represent a unitary function.

Schema

A model of a data base which includes definitions of the types of records, the data items that the records contain, and the sets into which they may be grouped. It also includes the mapping to storage.

Simulate

(1) To represent certain features of the behavior of a physical or abstract system by the behavior of another system. (2) To represent the functioniong of a device, system, or computer program by another. For example, to represent the functioning of one computer by another, to represent the behavior of a physical system by the execution of a computer program, to represent a biological system by a mathematical model. Contrast with emulate.

Software

Computer programs, procedures, data, and associated documentation concerned with the operation of a data processing system. The term software includes (a) computer program products in the form of card decks, magnetic tapes, or disks and (b) all documentation associated with computer programs, including specifications, listings, manuals, flow charts, version description documents, test plans, and test procedures.

Software Data Collection

The acquisition of information about a software development project and its products, performance, resource utilization, and environment.

Software Data Repository

A data base in which software development data may be deposited for use in cross project comparisons of software methodologies, in development of further understanding of the software development process, and in various analyses and studies concerning the development and use of software systems.

Software Development Data

Information about the development and performance of a computer program or system, including project characteristics, project performance, product characteristics and product quality.

Software Development Life Cycle

The process by which user requirements are translated, via software, into a functioning system. The actual steps involved may differ according to the size, purpose, and end use of the software. For the formal development of a large program the following steps are involved: system requirements definition, software requirements definition, preliminary design, analysis and detailed design, coding and checkout, development and system testing, delivery, and maintenance.

Software Error

Any discrepancy between a computed, observed or measured quantity and its true, specified, or theoretically correct value. Errors are introduced into software by human mistakes, that is; (1) deficiences or misinterpretations of design criteria, (2) logical mistakes, (3) a syntatical mistakes and (4) mistakes made in transcribing program statements into the input data. Contrast with fault, malfunction, and mistake.

Software Failure

The result of a computer executing a section of code containing an error. (Some examples include program aborts, system crashes, etc.).

Software Fault

A departure from correct, specified program execution which is due to error(s) occuring during the translation of the original specification of an algorithm to the program being executed.

Software Problem Report (SPR)

A form used to report a suspected or existing discrepancy or deficiency in an existing computer program, its operational documentation, or interfacing hardware.

Software Products

Computer programs, data bases, and documentation output at each step of the software development life cycle.

Software Reliability Measures

Quantitative measures used to predict the quality of software during the development and test stages. A typical measure is the mean time between software failures, the reliability function, and the instantaneous failure rate.

Software Testing

The process of exercising software in an attempt to detect errors that exist in the code. Software testing does not prove that a program is correct.

Software System Dependability

The probability that the application program together with its supervisory program, data base and hardware will perform in its intended environment. The environment may include anomalies and failures, such as:

- Deficiencies in requirement
- Software design errors (incorrect algorithms, word length problems, timing problems, etc.)
- Software failures
- Processor errors
- Memory errors
- Failures in the communication network
- Failures in peripherical devices
- Operator mistakes
- Power failures
- Environmental failuares
- Gradual erosion of the data base
- Hardware saturation (CPU memory, I/O channels)

Specification

Detailed descriptions or statements of the characteristics of a system, or system component, which are to be followed in producing the software.

Specification Change Notice (SCN)

A document used to propose, transmit, and record changes to a specification. In the proposal, or preliminary form, prior to approval, the SCN supplies copies of the pages containing the proposed changes.

SPR

Software Problem Report.

Storage

Capacity for retaining data temporarily or permanently. Hence, pertaining to a device into which data can be entered, in which they can be held, and from which they can be retrieved at a later time. Loosely, a device that can store data. Capacity for retaining data; hence, any device which data can be entered into, held in and retrieved from.

Storage, Auxiliary

(1) Data storage other than main storage; for example, storage on magnetic tape or direct access devices. Synonymous with external storage, secondary storage. (2) A storage that supplements another storage. Constrast with storage, main.

Storage, Main

(1) The general purpose storage of a computer. Usually, main storage can be accessed directly by the operating registers. (2) All program-addressable storage from which instructions may be executed, and from which data can be loaded directly into registers. Constrast with auxiliary storage.

Storage, Primary

See Storage, Main.

Storage, Secondary

See Storage, Auxiliary.

Stacked Job Processing

A technique that permits multiple job definitions to be grouped (stacked) for presentation to the system, which automatically recognizes the jobs, one after the other. More advanced systems allow job definitions to be added to the group (stack) at any time and from any source, while honoring priorities. See also batch processing.

Structure

(1) A description of the construction of a program, in terms of coding structure, module structure, task (parallel-process) structure, memory layout, and module interfaces. See also data structure.

Structured Programming

A programming discipline providing a means of expressing a system design that ensures a testable and understandable implementation, and that enforces simple and well-defined connections between program modules. The programming discipline uses the repeated application of a small number of basic control statements to form simple program constructs that represent large and complex programs. The process of creating the program modules includes: making local or tactical programming decisions within the designed module; writing program segments that represent these decisions; and integrating program segments into a unit corresponding to a system module.

Stub

A minimal piece of code to simulate the operation of a program. In top-down program development, stubs are the first realization of a program's functional specification. They are expanded into programs when the development effort moves to the level of abstraction to which the stubs belongs.

Subschema

A consistent and logical subset of a schema which is one user's view of a data base.

Subsystem

A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system.

Summary Data

A condensed set of data that represents information about project products and/or progress derived through totalling, averaging, or other reduction process.

System

(1) An assembly of methods, procedures, programs, or techniques united by regulated interaction to form an organized whole. (2) An organized collection of men, machines, and methods required to accomplish a set of specific functions. (3) A structured combination of interacting parts satisfying a set of functional objectives and performance objectives.

System Design Review (SDR)

A review that normally occurs on completion of the System Design Phase. Successful completion of this review establishes the preliminary or system-level computer program development specifications, interface specifications, and data requirements specifications in the System Design Baseline. This term is used interchangeably with the Preliminary Design Review.

System Productivity

A measure of the work performed by a system. System productivity depends on a combination of many factors, such as the facility (ease of use) of the system, and the overall system performance, including throughput, response time, efficiency availability.

System Representation

A definition of a system at some point in the software development life cycle. For example, the system can be represented at various times as sets of functional requirements, program designs, coded program modules, or integrated, tested, and certified program modules and using procedures.

System Test

The operation of a system under test conditions with a set of known inputs which should result in a set of predicted outputs. Although often cast as an 'evaluation' most system tests are a demonstration that the system will operate satisfactorily as a unit within the allowable ranges of input and output data.

Throughput

The total volume of work performed by a computing system over a given period of time.

Top-Down Development

The process of designing a software system through a sequence of step-wise refinement through successive levels of abstraction; and then implementing the design by giving development priority to control structures and module interfaces. Application routines (i.e., bottom-level modules) are first represented by program stubs simulating the control and interaction interfaces. The actual program are subsequently integrated into the control structure such that system level testing for the control structure and interfaces are performed first and continually verified as application routines are substituted for their stub representations.

Turnaround Time

The total amount of time between submission of a job, and the return of the job to the user. This time may include delays waiting to get in the computer for execution and waiting for the delivery of the output. To a computer, it is time-off minus time-on.

Unit Test

The testing of successfully compiled modules independent of other modules to verify the coded logic agrees with the module's specifications. See Module Testing.

Usability

The extend to which a software product is convenient and practicable to use. This includes its scope of applicability and how easily it is comprehended by potential human users.

Validation

The process of ensuring that specific program functions meet requirement specifications.

Validation, Formal

Mathematical techniques for proving program correctness.

Validity

(1) Correctness, especially the degree of closeness by which iterated results approach the correct results. (2) The degree to which a measuring or testing device measures what it is intended or designed to measure.

Verification

(1) The process of testing designed to ensure that the system and/or system components are logically correct. (2) The process by which the contents of a computer program's physical medium (tape, deck, etc.) are authenticated.

☆U.S. GOVERNMENT PRINTING OFFICE: 1977-714-025/90