BS ISO/IEC IEEE 24765:2010

BSI Standards Publication

Systems and software engineering — vocabulary

bsi.

...making excellence a habit.™

Document Sharing Hub

Full document details: BS ISO/IEC IEEE 24765:2010

National foreword

This British Standard is the UK implementation of ISO/IEC IEEE 24765:2010.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software and systems engineering.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© BSI 2011

ISBN 978 0 580 62587 9

ICS 01.040.35; 35.080

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2011.

Amendments issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC IEEE 24765:2010

INTERNATIONAL STANDARD

ISO/IEC/ IEEE 24765

First edition
2010-12-15

Systems and software engineering — Vocabulary

Ingénierie des systèmes et du logiciel — Vocabulaire

Reference number
ISO/IEC/IEEE 24765:2010(E)

© ISO/IEC 2010
© IEEE 2010

Documents
Document Sharing Hub

Full document details: BS ISO/IEC IEEE 24765:2010

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2010
© IEEE 2010

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO or IEEE at the respective address below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York • NY 10016-5997, USA
E-mail stds.ipr@ieee.org
Web www.ieee.org

Published by ISO in 2011

Published in Switzerland

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC IEEE 24765:2010

Contents

Page

Foreword v

Introduction vi

1 Scope 1
1.1 Relationship of the print and internet-accessible versions 1
1.2 Vocabulary structure 1
1.3 PMI Glossary provisions 2

2 Conformance 2

3 Terms and definitions 2

Annex A (informative) **List of Source Standards** 404

Annex B (informative) **List of References** 409

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

List of Figures

Page

Figure 1 —Activity 9
Figure 2 —Block Diagram 35
Figure 3 —Box diagram 38
Figure 4 —Bubble chart 39
Figure 5 —Call graph 42
Figure 6 —Case construct 45
Figure 7 —Categorization of software 46
Figure 8 —Data flow diagram 90
Figure 9 —Data structure diagram 92
Figure 10 —Directed graph 110
Figure 11 —Documentation tree 113
Figure 12 —Flowchart 145
Figure 13 —Graph 158
Figure 14 —If-then-else construct 168
Figure 15 —Input-process-output chart 178
Figure 16 —Modification request 222
Figure 17 —Structure chart 349
Figure 18 —UNTIL 387
Figure 19 —Web site 399
Figure 20 —WHILE 400

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of ISO/IEC JTC 1 is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from ISO or the IEEE Standards Association.

ISO/IEC/IEEE 24765 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the Software & Systems Engineering Standards Committee of the IEEE Computer Society of the IEEE, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

Certain material contained in ISO/IEC/IEEE 24765 is reproduced, with permission, from *A Guide to the Project Management Body of Knowledge (PMBOK®) Guide — Fourth Edition* , copyright 2008, Project Management Institute.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

Introduction

The systems and software engineering disciplines are continuing to mature while information technology advances. New terms are being generated and new meanings are being adopted for existing terms. This International Standard was prepared to collect and standardize terminology. Its purpose is to identify terms currently in use in the field and standard definitions for these terms. It is intended to serve as a useful reference for those in the Information Technology field, and to encourage the use of systems and software engineering standards prepared by ISO and liaison organizations IEEE Computer Society and Project Management Institute (PMI). It provides definitions that are rigorous, uncomplicated, and understandable by all concerned.

While it is useful to find the meaning of a term, no word stands in isolation. This International Standard makes it possible to search for related concepts and to view how a term is used in definitions of other terms.

Every effort has been made to use definitions from established systems and software engineering standards of ISO JTC 1/SC 7 and its liaison organizations IEEE Computer Society and the PMI. When existing standards were found to be incomplete, unclear or inconsistent with other entries in the vocabulary, however, new, revised, or composite definitions have been developed. Some definitions have been recast in a systems, rather than software, context.

This International Standard replaces IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, which was contributed by the IEEE as a source document. The approach and lexical exactitude of IEEE Std 610.12-1990 served as a model for this International Standard. Nevertheless, approximately two-thirds of the definitions in this International Standard are new since IEEE Std 610.12 was last updated in 1990, a reflection of the continued evolution in the field.

The vocabulary is offered in both print and internet-accessible versions for ease of reference.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

Systems and software engineering —Vocabulary

1 Scope

Consistent with ISO vocabulary standards, each technical committee is responsible for standard terminology in its area of specialization. This International Standard provides a common vocabulary applicable to all systems and software engineering work falling within the scope of ISO JTC 1/SC 7.

The scope of each concept defined has been chosen to provide a definition that is suitable for general application. In those circumstances where a restricted application is concerned, a more specific definition might be needed.

Terms have been excluded if they were

- considered to be parochial to one group or organization;
- company proprietary or trademarked;
- multi-word terms whose meaning could be inferred from the definitions of the component words;
- terms whose meaning in the information technology (IT) field could be directly inferred from their common English meaning.

1.1 Relationship of the print and internet-accessible versions

The primary tool for maintaining this vocabulary is a database that is modified in a controlled fashion. Hosted by the IEEE Computer Society, the SEVOCAB (systems and software engineering vocabulary) database is publicly accessible at www.computer.org/sevocab. ISO/IEC 24765 is issued periodically as a formal, published International Standard reflecting a "snapshot" of the database.

The copyright notice provided with the database permits users to copy definitions from the database as long as the source of the definition is cited. Permitting public use of the definitions in the database is intended to encourage the use of other ISO/IEC JTC 1 and IEEE systems and software engineering standards.

1.2 Vocabulary structure

Entries in the vocabulary are arranged alphabetically. Blanks precede all other characters in alphabetizing. Hyphens and slashes (- and /) follow all other characters in alphabetizing.

An entry can consist of a single word, such as "software"; a phrase, such as "test case"; or an acronym, such as "CDR". Phrases are given in their natural order (test plan) rather than in reversed order (plan, test). Acronyms can be listed separately as well as in parentheses following the source term. Terms that are verbs are shown without the infinitive marker "to".

After each term, numbered definitions are listed in order of preference, or from the most general to the more specific usages. The different definitions can show the use of a term as a noun, verb and adjective.

This International Standard includes references to the active source standards for each definition, so that the use of the term can be further explored. The sources of most of the definitions are ISO JTC 1/SC 7 or IEEE Computer Society standards and the PMI Glossary, Fourth Edition. Sources are listed in Annex A. In some

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

cases, the same definition can also be found in other active or withdrawn standards. No source is shown if the original source standard has been withdrawn or archived and the definition has been retained in this vocabulary.

Notes (comments), Examples, and illustrations taken from the source standards have been included to clarify selected definitions.

The following cross-references are used to show a term's relationship to other terms in the dictionary.

- *Syn* refers to a synonym: a term with the same meaning. Synonyms are listed under the preferred term and can be located by searching.
- cf. refers to related terms that are not synonyms.

1.3 PMI Glossary provisions

The Project Management Institute (PMI) Glossary definitions have been included without alteration in accordance with the copyright agreement. Many of these definitions include explanatory material. For other terms and other definitions that have ISO/IEC and IEEE standards as their source, explanatory matter is shown in the Notes and Examples.

Many of the definitions from the PMI Glossary begin with a word or phrase in brackets, such as [Process], [Output/Input], [Technique]. These bracketed entries refer to the schema of the Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK®[1)] Guide)* – Fourth Edition, which provides further explanation.

2 Conformance

The definitions in this International Standard are drawn from normative standards and informative guidance documents, including ISO Technical Reports (TR). This International Standard may be used as a normative document for projects and organizations claiming conformance to the normative source standards. Where terms have multiple definitions, users should consult the source standards for further information on appropriate usage within a specific context. Annex B lists other references.

Terms, definitions, and notes use spelling preferred in the USA. The use of capital letters has been minimized and generally limited to proper names and acronyms. In some cases the source standard uses another correct spelling (such as *behaviour* rather than *behavior*, *on-line* rather than *online*). Other correct spellings and capitalization of the terms, according to a national standard, an authoritative general dictionary or accepted style guide may be used with the definitions.

3 Terms and definitions

3.1
<Viewpoint> language
1. definitions of concepts and rules for the specification of an ODP system from the <viewpoint> viewpoint. ISO/IEC 10746-3:1996 Information technology — Open Distributed Processing — Reference Model: Architecture.4.2.1.1

NOTE Thus, engineering language is defined as "definitions of concepts and rules for the specification of an ODP system from the engineering viewpoint".

---

1) PMBOK is a trademark of the Project Management Institute, Inc. which is registered in the United States and other nations.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.2
<X> federation
1. a community of <x> domains. *ISO/IEC 10746-3:1996, Information technology — Open Distributed Processing —Reference Model: Architecture.*5.1.2

3.3
<x> interceptor
1. an engineering object in a channel, placed at a boundary between <x> domains. *ISO/IEC 10746-3:1996, Information technology —Open Distributed Processing —Reference Model: Architecture.* 8.1.11

NOTE An <x> interceptor performs checks to enforce or monitor policies on permitted interactions between basic engineering objects in different domains; performs transformations to mask differences in interpretation of data by basic engineering objects in different domains. An inter-subnetwork relay is an example of an interceptor.

3.4
1GL
1. first-generation language

cf. machine language

3.5
2GL
1. second-generation language

cf. assembly language

3.6
3GL
1. third-generation language

cf. high order language

3.7
4GL
1. fourth-generation language

3.8
5GL
1. fifth-generation language

3.9
A-0 context diagram
1. the only context diagram that is a required for a valid IDEF0 model, the A-0 diagram contains one box, which represents the top-level function being modeled, the inputs, controls, outputs, and mechanisms attached to this box, the full model name, the model name abbreviation, the model's purpose statement, and the model's viewpoint statement. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0*

3.10
A4, A5
1. International Standard paper sizes. *ISO/IEC 15910:1999, Information technology — Software user documentation process.*4.1

NOTE A4 is 210 mm by 297 mm and A5 is 148 mm by 210 mm; see ISO 216:2007.

3.11
abend
1. abbreviation for abnormal end

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.12
abnormal end (abend)
1. termination of a process prior to completion

cf. abort, exception

3.13
abort
1. to terminate a process prior to completion

cf. abnormal end (abend), exception

3.14
absolute address
1. an address that is permanently assigned to a device or storage location and that identifies the device or location without the need for translation or calculation. *Syn:* explicit address, specific address

cf. relative address, relocatable address, symbolic address, absolute assembler, absolute code, absolute instruction

3.15
absolute assembler
1. an assembler that produces absolute code

cf. relocating assembler

3.16
absolute code
1. code in which all addresses are absolute addresses. *Syn:* specific code

cf. relocatable code

3.17
absolute instruction
1. a computer instruction in which all addresses are absolute addresses

cf. direct instruction, effective instruction, immediate instruction, indirect instruction

3.18
absolute loader
1. a loader that reads absolute machine code into main memory, beginning at the initial address assigned to the code by the assembler or compiler, and performs no address adjustments on the code

cf. relocating loader

3.19
abstract class
1. a class that cannot be instantiated independently. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject)*.3.1.1

NOTE That is, instantiation must be accomplished via a subclass. A class for which every instance must also be an instance of a subclass in the cluster (a total cluster) is called an abstract class with respect to that cluster.

3.20
abstract data type
1. a data type for which only the properties of the data and the operations to be performed on the data are specified, without concern for how the data will be represented or how the operations will be implemented

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.21
abstract design
1. a generic form that needs specialization (further design work) to produce concrete designs. ***2.*** design aimed at producing designs

3.22
abstraction
1. a view of an object that focuses on the information relevant to a particular purpose and ignores the remainder of the information. ***2.*** the process of formulating a view

cf. data abstraction

3.23
acceptability
1. the exposure to loss (financial or otherwise) that an organization is willing to tolerate from a risk

NOTE Risk acceptability may apply to an individual risk or to a collection of risks, such as the totality of risks confronting a project or enterprise. Acceptability may differ for different categories of risk and may depend on the cost of treatment or other factors.

3.24
acceptable
1. meeting stakeholder expectations that can be shown to be reasonable or merited. *ISO/IEC 38500:2008, Corporate governance of information technology.*1.3.1

3.25
acceptance criteria
1. the criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. ***2.*** those criteria, including performance requirements and essential conditions, which must be met before project deliverables are accepted. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

cf. requirement, test criteria

3.26
acceptance test
1. the test of a system or functional unit usually performed by the purchaser on his premises after installation with the participation of the vendor to ensure that the contractual requirements are met. *ISO/IEC 2382-20:1990, Information technology —Vocabulary —Part 20: System development.* 20.05.07.

cf. acceptance testing, validation testing

3.27
acceptance testing
1. testing conducted to determine whether a system satisfies its acceptance criteria and to enable the customer to determine whether to accept the system. *IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation.*3.1.1. *Syn:* validation testing, acceptance test. ***2.*** formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component. *IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation.*3.1.1.

cf. validation testing, acceptance test

3.28
access
1. to obtain the use of a resource. *ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.*01.01.04

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.29
access facility
1. a set of service primitives that allow a stub objects to negotiate the abstract and transfer syntax to be used for the operation data to be transmitted over the channel. *ISO/IEC 14752:2000, Information technology — Open Distributed Processing —Protocol support for computational interactions.* 3.3.1

3.30
access method
1. a technique to obtain the use of data, the use of storage in order to read or write data, or the use of an input-output channel to transfer data. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.*01.08.03

3.31
access routine
1. a routine that provides access to a data structure that is hidden, usually because it is a global variable or used in an abstract data type

3.32
access transparency
1. a distribution transparency which masks differences in data representation and invocation mechanisms to enable interworking between objects. *ISO/IEC 10746-3:1996, Information technology — Open Distributed Processing —Reference Model: Architecture.*4.4.1.1

3.33
accessibility
1. usability of a product, service, environment or facility by people with the widest range of capabilities. *ISO/IEC 25062:2006, Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability test reports.* 4.1; *ISO/IEC 26514, Systems and software engineering--requirements for designers and developers of user documentation.*4.1

NOTE Although "accessibility" typically addresses users who have disabilities, the concept is not limited to disability issues.

3.34
accident
1. an unplanned event or series of events that results in death, injury, illness, environmental damage, or damage to or loss of equipment or property. *IEEE Std 1228-1994 (R2002) IEEE Standard for Software Safety Plans.*3.1.1

3.35
accuracy
1. a qualitative assessment of correctness, or freedom from error. ***2.*** a quantitative measure of the magnitude of error

3.36
accuracy of measurement
1. the closeness of the agreement between the result of a measurement and the true value of the measurand. *ISO/IEC TR 14143-3:2003, Information technology — Software measurement — Functional size measurement —Part 3: Verification of functional size measurement methods.* 3.1

NOTE Accuracy is a qualitative concept. The term precision should not be used for "accuracy". [ISO/IEC Guide 99:2007 International vocabulary of metrology —Basic and general concepts and associated terms] A true value is a value consistent with the definition of a given particular quantity and this is a value that would be obtained by a perfect measurement. In contexts where perfect measurement is not practically feasible, a conventional true value is a value attributed to a particular quantity and accepted, sometimes by convention, as having an uncertainty appropriate for a given purpose. 'Conventional true value', in the same reference, is sometimes called assigned value, best estimate of the value, conventional value or reference value. The accuracy should be expressed in terms of the Mean magnitude of relative error.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.37
ACID
1. Atomicity Consistency Isolation Durability. *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing —Reference model: Overview*

3.38
acquire project team
1. [Process] the process of confirming human resource availability and obtaining the team necessary to complete project assignments. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fourth Edition*

3.39
acquirer
1. stakeholder that acquires or procures a product or service from a supplier. *ISO/IEC 12207:2008 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes.* 4.1. ***2.*** the individual or organization that specifies requirements for and accepts delivery of a new or modified software product and its documentation. *IEEE Std 1058-1998 IEEE Standard for Software Project Management Plans.*3.1; *ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes.*4.1. *Syn:* buyer, customer, owner, purchaser

NOTE The acquirer may be internal or external to the supplier organization. Acquisition of a software product may involve, but does not necessarily require, a legal contract or a financial transaction between the acquirer and supplier.

3.40
acquisition
1. process of obtaining a system, software product or software service. *ISO/IEC 12207:2008 (IEEE Std 12207-2008) Systems and software engineering —Software life cycle processes.* 4.2. ***2.*** process of obtaining a system product or service. *ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering —System life cycle processes.* 4.2. *Syn:* outsourcing

3.41
acquisition strategy
1. specific approach to acquiring products and services that is based on considerations of supply sources, acquisition methods, requirements specification types, contract or agreement types, and related acquisition risks

3.42
action
1. element of a step that a user performs during a procedure. *ISO/IEC 26514, Systems and software engineering--requirements for designers and developers of user documentation.*4.2. ***2.*** a description of an operation to be taken in the formulation of a solution. *ISO 5806:1984, Information processing —Specification of single-hit decision tables.*3.7

3.43
action entry
1. an indication of the relevance of an action to a particular rule. *ISO 5806:1984, Information processing — Specification of single-hit decision tables.*3.9

3.44
action of interest
1. an action in a transaction which leads to a state change of significance to the transaction. *ISO/IEC 10746-3:1996 Information technology — Open Distributed Processing — Reference Model: Architecture.*13.7.1.2

3.45
action stub
1. a list of all the actions to be taken in the solution of a problem. *ISO 5806:1984, Information processing — Specification of single-hit decision tables.*3.11

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.46
activation
1. one occurrence of a function's transformation of some subset of its inputs into some subset of its outputs. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.3

3.47
activation constraint
1. a function's requirement for the presence of a non-empty object set in a particular arrow role as a precondition for some activation of the function. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.* 2.1.4

3.48
active area
1. (on-screen documentation) area that responds to user input.4.3

EXAMPLE a window, icon or text field

3.49
active interconnection
1. a physical interaction mechanism allowing the action of one thing to cause a change or to stimulate an action in another thing. *IEEE Std 1175.2-2006 IEEE Recommended Practice for CASE Tool Interconnection — Characterization of Interconnections.*3.1

3.50
active redundancy
1. in fault tolerance, the use of redundant elements operating simultaneously to prevent, or permit recovery from, failures

cf. standby redundancy

3.51
active text
1. text displayed on the screen that responds to user input

3.52
active white space
1. area around textual or graphical elements, not including margins, which breaks up text, separates topic and subtopic groupings, indicates hierarchical and topical relationships, highlights information or makes text easier to read. *ISO/IEC 15910:1999, Information technology — Software user documentation process.* 4.54

3.53
activity
1. set of cohesive tasks of a process. *ISO/IEC 12207:2008 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes.* 4.3; *ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes.* 4.3. ***2.*** a component of work performed during the course of a project. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fourth Edition.* ***3.*** an order submitted to the system under test (SUT) by a user or an emulated user demanding the execution of a data processing operation according to a defined algorithm to produce specific output data from specific input data and (if requested) stored data. *ISO/IEC 14756:1999, Information technology — Measurement and rating of performance of computer-based software systems.*4.1. ***4.*** a defined body of work to be performed, including its required input information and output information. *IEEE Std 1074-2006 IEEE Standard for Developing a Software Project Life Cycle Process.*Annex E. ***5.*** collection of related tasks. *ISO/IEC 90003:2004, Software engineering — Guidelines for the application of ISO 9001:2000 to computer software.*3.1. ***6.*** element of work performed during the implementation of a process. *IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation.*3.1.2

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

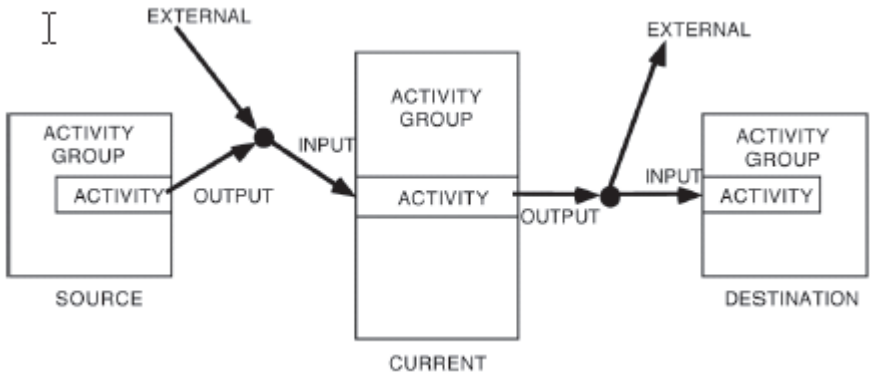

Figure 1 —Activity

NOTE An activity normally has an expected duration, cost, and resource requirements. Activities are often subdivided into tasks.

3.54
activity attributes
1. [Output/Input] multiple attributes associated with each schedule activity that can be included within the activity list. Activity attributes include activity codes, predecessor activities, successor activities, logical relationships, leads and lags, resource requirements, imposed dates, constraints, and assumptions. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.55
activity code
1. one or more numerical or text values that identify characteristics of the work or in some way categorize the schedule activity that allows filtering and ordering of activities within reports. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.56
activity duration
1. the time in calendar units between the start and finish of a schedule activity. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.57
activity group
1. a set of related activities. *IEEE Std 1074-2006 IEEE Standard for Developing a Software Project Life Cycle Process.* Annex E

3.58
activity identifier
1. a short unique numeric or text identification assigned to each schedule activity to differentiate that project activity from other activities. Typically unique within any one project schedule network diagram. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.59
activity list
1. [Output/Input] a documented tabulation of schedule activities that shows the activity description, activity identifier, and a sufficiently detailed scope of work description so project team members understand what work is to be performed. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fourth Edition*

3.60
activity type
1. a classification of activities defined by the execution of the same algorithm. *ISO/IEC 14756:1999, Information technology —Measurement and rating of performance of computer-based software systems.* 4.2

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.61
actor
1. a role (with respect to that action) in which the enterprise object fulfilling the role participates in the action. *ISO/IEC 15414:2006, Information technology — Open distributed processing — Reference model — Enterprise language*.6.3.1. ***2.*** organization or CASE tool that supplies and/or acquires SEE Services. *ISO/IEC 15940:2006, Information Technology — Software Engineering Environment Services.* 2.2.4. ***3.*** in UML, someone or something outside the system that interacts with the system

NOTE It may be of interest to specify which actor initiates that action.

3.62
actual cost (AC)
1. total costs actually incurred and recorded in accomplishing work performed during a given time period for a schedule activity or work breakdown structure component. Actual cost can sometimes be direct labor hours alone, direct costs alone, or all costs including indirect costs. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition. Syn:* actual cost of work performed (ACWP)

cf. earned value management, earned value technique

3.63
actual duration
1. the time in calendar units between the actual start date of the schedule activity and either the data date of the project schedule if the schedule activity is in progress or the actual finish date if the schedule activity is complete. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.64
ACWP
1. actual cost of work performed. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.65
adaptation data
1. data used to adapt a program to a given installation site or to given conditions in its operational environment

3.66
adaptation parameter
1. a variable that is given a specific value to adapt a program to a given installation site or to given conditions in its operational environment

EXAMPLE the variable Installation_Site_Latitude

3.67
adapter
1. an object adapter. *ISO/IEC 19500-2:2003, Information technology — Open Distributed Processing — Part 2: General Inter-ORB Protocol (GIOP)/Internet Inter-ORB Protocol (IIOP)*.3.2.1

3.68
adaptive maintenance
1. modification of a software product, performed after delivery, to keep a software product usable in a changed or changing environment. *ISO/IEC 14764:2006 (IEEE Std 14764-2006), Software Engineering — Software Life Cycle Processes —Maintenance*.3.1

EXAMPLE The operating system might be upgraded and some changes may be made to accommodate the new operating system.

NOTE Adaptive maintenance provides enhancements necessary to accommodate changes in the environment in which a software product must operate. These changes are those that must be made to keep pace with the changing environment.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.69
added source statements
1. the count of source statements that were created specifically for the software product

3.70
address
1. a number, character, or group of characters that identifies a given device or storage location. ***2.*** to refer to a device or storage location by an identifying number, character, or group of characters. ***3.*** to deal with, to take into consideration; (specifically) to decide whether and when a defined documentation topic is to be included, either directly or by reference to another document; to decide whether an item is to be recorded prior to the test execution (in a tool or not in a tool), recorded during the test execution, recorded post-test execution, not recorded (addressed by the process), or excluded. *IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation*.3.1.3

3.71
address field
1. a field of a computer instruction that contains addresses, information necessary to derive addresses, or values of operands. *Syn:* address part

cf. operation field

3.72
address format
1. the number and arrangement of address fields in a computer instruction. ***2.*** the number and arrangement of elements within an address, such as the elements needed to identify a particular channel, device, disk sector, and record in magnetic disk storage

cf. n-address instruction, n-plus-one address instruction

3.73
address modification
1. an arithmetic, logical, or syntactic operation performed on an address

cf. effective address, indexed address, relative address, relocatable address

3.74
address space
1. the addresses that a computer program can access. ***2.*** the number of memory locations that a central processing unit can address

NOTE In some systems, this may be the set of physical storage locations that a program can access, disjoint from other programs, together with the set of virtual addresses referring to those storage locations, which may be accessible by other programs.

3.75
addressing exception
1. an exception that occurs when a program calculates an address outside the bounds of the storage available to it

cf. data exception, operation exception, overflow exception, protection exception, underflow exception

3.76
adjusted function point count (AFP)
1. the unadjusted function point count multiplied by the value adjustment factor. *ISO/IEC 20926:2003; Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual*

NOTE calculated using a specific formula for development project, enhancement project, and application; commonly called the function point count

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.77
adjusted size
1. a size based on the functional size multiplied by the technical complexity adjustment. *ISO/IEC 20968:2002; Software engineering —Mk II Function Point Analysis —Counting Practices Manual.* 10

NOTE This measure does not represent functional size.

3.78
Administer Procurements
1. [Process] the process of managing procurement relationships, monitoring contract performance, and making changes and corrections as needed. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.79
adoption process
1. set of activities by which an organization brings CASE tools into widespread use. *ISO/IEC TR 14471:2007, Information technology —Software engineering —Guidelines for the adoption of CASE tools.*2.1.2

3.80
ADT
1. Abstract Data Type. *ISO/IEC 19500-2:2003, Information technology — Open Distributed Processing — Part 2: General Inter-ORB Protocol (GIOP)/Internet Inter-ORB Protocol (IIOP).*3.3

3.81
AE(I)
1. Application Entity (Invocation). *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing —Reference model: Overview*

3.82
afferent
1. pertaining to a flow of data or control from a subordinate module to a superordinate module in a software system

cf. efferent

3.83
agent
1. an enterprise object that has been delegated (authority, responsibility, a function, etc) by and acts for another enterprise object (in exercising the authority, carrying out the responsibility, performing the function, etc). *ISO/IEC 15414:2006, Information technology — Open distributed processing — Reference model — Enterprise language.*6.5.7

NOTE An agent may be a party or may be the ODP system or one of its components. Another system in the environment of the ODP system may also be an agent. The delegation may have been direct, by a party, or indirect, by an agent of the party having authorization from the party to so delegate.

3.84
aggregate responsibility
1. a broadly stated responsibility that is eventually refined as specific properties and constraints. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject).*3.1.3

3.85
agreement
1. mutual acknowledgement of terms and conditions under which a working relationship is conducted. *ISO/IEC 12207:2008 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes.*4.4; *ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes.*4.4

cf. contract

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.86
algebraic language
1. a programming language that permits the construction of statements resembling algebraic expressions, such as Y = X + 5

cf. algorithmic language, list processing language, logic programming language

EXAMPLE FORTRAN

3.87
algorithm
1. a finite set of well-defined rules for the solution of a problem in a finite number of steps. ***2.*** a sequence of operations for performing a specific task. ***3.*** a finite ordered set of well-defined rules for the solution of a problem. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.05.05

EXAMPLE a complete specification of a sequence of arithmetic operations for evaluating sine x to a given precision

3.88
algorithmic language
1. a programming language designed for expressing algorithms

cf. algebraic language, list processing language, logic programming language

EXAMPLE ALGOL

3.89
alias
1. an alternate name for an IDEF1X model construct (class, responsibility, entity, or domain). *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject)*.3.1.4

3.90
allocated baseline
1. in configuration management, the initial approved specifications governing the development of configuration items that are part of a higher-level configuration item

cf. developmental configuration, functional baseline, product baseline, allocated configuration identification

3.91
allocated configuration identification
1. in configuration management, the current approved specifications governing the development of configuration items that are part of a higher-level configuration item

cf. functional configuration identification, product configuration identification. allocated baseline

NOTE Each specification defines the functional characteristics that are allocated from those of the higher-level configuration item, establishes the tests required to demonstrate achievement of its allocated functional characteristics, delineates necessary interface requirements with other associated configuration items, and establishes design constraints, if any.

3.92
allocated requirement
1. requirement that levies all or part of the performance and functionality of a higher level requirement on a lower level architectural element or design component

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.93
allocation
1. the process of distributing requirements, resources, or other entities among the components of a system or program. ***2.*** the result of the distribution of requirements, resources, or other entities among the components of a system or program. ***3.*** the decision to assign a function or decision to hardware, software, or humans. *IEEE Std 1220-2005 IEEE Standard for the Application and Management of the Systems Engineering Process.*3.1.3

NOTE Allocation may be made entirely to hardware, software, or humans, or to some combination to be resolved upon further functional decomposition.

3.94
alpha testing
1. first stage of testing before a product is considered ready for commercial or operational use

cf. beta testing

NOTE often performed only by users within the organization developing the software

3.95
alphanumeric
1. pertaining to data that consists of letters, digits, and usually other characters, such as punctuation marks, as well as to processes and functional units that use the data. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.02.05

3.96
ALS
1. Application Layer Structure. *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing —Reference model: Overview*

3.97
alternate flow
1. the part of a use case that describes its alternative implementations. *Syn:* alternate path

NOTE It is also used to describe error conditions, since errors can be considered a kind of alternative.

3.98
alternate key
1. a candidate key of an entity other than the primary key. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject).*3.1.5

NOTE [key style]

3.99
analog
1. pertaining to continuously variable physical quantities or to data presented in a continuous form, as well as to processes and functional units that use the data. *ISO/IEC 2382-1:1993, Information technology — Vocabulary —Part 1: Fundamental terms.* 01.02.06

3.100
analog computer
1. a computer whose operations are analogous to the behavior of another system and that accepts, processes, and produces analog data. *ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.*01.03.05

EXAMPLE an abacus

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.101
analogous estimating
1. [Technique] an estimating technique that uses the values of parameters, such as scope, cost, budget, and duration or measures of scale such as size, weight, and complexity from a previous, similar activity as the basis for estimating the same parameter or measure for a future activity. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.102
analysis
1. the process of studying a system by partitioning the system into parts (functions, components, or objects) and determining how the parts relate to each other. *IEEE Std 1362-1998 (R2007) IEEE Guide for Information Technology — System Definition — Concept of Operation Document.* 3.1. ***2.*** investigation and collection phase of user documentation development that aims to specify types of users and their information needs.4.4

3.103
analysis model
1. algorithm or calculation combining one or more base and/or derived measures with associated decision criteria. *ISO/IEC 25000:2005, Software Engineering — Software product Quality Requirements and Evaluation (SQuaRE) —Guide to SQuaRE.* 4.2

3.104
analyst
1. a member of the technical community (such as a systems engineer or business analyst, developing the system requirements) who is skilled and trained to define problems and to analyze, develop, and express algorithms. *IEEE Std 1233-1998 (R2002) IEEE Guide for Developing System Requirements Specifications.*3.1

3.105
ancestor (of a class)
1. a generic ancestor of the class or a parent of the class or an ancestor of a parent of the class. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject).*3.1.6

cf. generic ancestor, reflexive ancestor

3.106
ancestral box
1. a box related to a specific diagram by a hierarchically consecutive sequence of one or more parent/child relationships. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.6

3.107
ancestral diagram
1. a diagram that contains an ancestral box. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0.* 2.1.7

3.108
annotate
1. a command used for listing the latest version of each program's source code line, along with the date, the file version it was introduced, and the person who committed it

3.109
annotation
1. further documentation accompanying a requirement such as background information and/or descriptive material. *IEEE Std 1233-1998 (R2002) IEEE Guide for Developing System Requirements Specifications.*3.2

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.110
announcement
1. an interaction - the invocation - initiated by a client object resulting in the conveyance of information from that client object to a server object, requesting a function to be performed by that server object. *ISO/IEC 10746-3:1996, Information technology —Open Distributed Processing —Reference Model: Architecture.* 7.1.3

3.111
anomaly
1. condition that deviates from expectations, based on requirements specifications, design documents, user documents, or standards, or from someone's perceptions or experiences. *IEEE Std 1028-2008 IEEE Standard for Software Reviews and Audits*.3.1. ***2.*** anything observed in the documentation or operation of software or system that deviates from expectations based on previously verified software products, reference documents, or other sources of indicative behavior. *IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation*.3.1.4

3.112
ANSI
1. American National Standards Institute. *IEEE Std 829-2008 IEEE Standard for Software and System Test Documentation*.3.2

3.113
anticipatory buffering
1. a buffering technique in which data are stored in a buffer in anticipation of a need for the data

cf. dynamic buffering, simple buffering

3.114
anticipatory paging
1. a storage allocation technique in which pages are transferred from auxiliary storage to main storage in anticipation of a need for those pages

cf. demand paging

3.115
AP(I)
1. Application Process (Invocation). *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing —Reference model: Overview*

3.116
aperiodic task
1. a task activated on demand. *Syn:* asynchronous task

3.117
API
1. Application Program Interface. *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing —Reference model: Overview*

3.118
applicability to a functional domain
1. the ability of an FSM method to take into account the characteristics of functional user requirements (FUR) which are pertinent to FSM in a functional domain. *ISO/IEC TR 14143-3:2003, Information technology — Software measurement — Functional size measurement — Part 3: Verification of functional size measurement methods*.3.2

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.119
application
1. a system for collecting, saving, processing, and presenting data by means of a computer. *ISO/IEC 24570:2005, Software engineering —NESMA functional size measurement method version 2.1 —Definitions and counting guidelines for the application of Function Point Analysis* ***2.*** a coherent collection of automated procedures and data supporting a business objective. *ISO/IEC 20968:2002, Software engineering — Mk II Function Point Analysis —Counting Practices Manual.* 10. ***3.*** a cohesive collection of automated procedures and data supporting a business objective. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual. Syn:* application system, information system

cf. system

NOTE It consists of one or more components, modules, or subsystems. Frequently a synonym for "system", the word "application" is preferred to express more precisely the nature of the subject matter of functional size measurement.

3.120
application administration function
1. functions performed by users which include installation, configuration, application backup, maintenance (patching and upgrading) and de-installation. *ISO/IEC 25051:2006, Software engineering —Software product Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing.*4.1

3.121
application area
1. a general term for a grouping of applications that handle a specific business area. *ISO/IEC 20926:2003, Software engineering —IFPUG 4.1 Unadjusted functional size measurement method —Counting practices manual* ***2.*** a category of projects that have common components significant in such projects, but are not needed or present in all projects. Application areas are usually defined in terms of either the product (i.e., by similar technologies or production methods) or the type of customer (i.e., internal versus external, government versus commercial) or industry sector (i.e., utilities, automotive, aerospace, information technologies, etc.) Application areas can overlap. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fourth Edition*

NOTE It corresponds to an administrative level for management purposes.

3.122
application area level
1. the management level responsible for managing maintenance activities as well as new development or major enhancement projects for one or more applications. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method —Counting practices manual*

3.123
application boundary
1. the border between the application and its environment of other applications and users. *ISO/IEC 24570:2005, Software engineering —NESMA functional size measurement method version 2.1 —Definitions and counting guidelines for the application of Function Point Analysis* ***2.*** the border between the software being measured and the user. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method —Counting practices manual*

3.124
application engineering
1. the process of constructing or refining application systems by reusing assets. *IEEE Std 1517-1999 (R2004) IEEE Standard for Information Technology —Software Life Cycle Processes —Reuse Processes.* 3.1

3.125
application frameworks
1. a subsystem design made up of a collection of abstract and concrete classes and interfaces between them

NOTE Frameworks are often instantiation of a number of patterns.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.126
application function point count
1. a count that provides a measure of the functionality the application provides to the end-user. *ISO/IEC 20968:2002, Software engineering — Mk II Function Point Analysis — Counting Practices Manual.* 10. ***2.*** the size of an application expressed in function points. *ISO/IEC 24570:2005, Software engineering — NESMA functional size measurement method version 2.1 — Definitions and counting guidelines for the application of Function Point Analysis* ***3.*** a count that provides a measure of the current functionality the application provides to the user. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual*

NOTE i.e., the functionality already provided to the user or that is still to be provided. With it, the effort required to support the realized application can also be determined.

3.127
application generator
1. a code generator that produces programs to solve one or more problems in a particular application area

EXAMPLE a payroll generator

3.128
application manager
1. a person responsible for managing projects and support activities for one or more application areas. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual*

3.129
application problem
1. a problem submitted by an end user and requiring information processing for its solution. *ISO/IEC 2382-20:1990 Information technology — Vocabulary — Part 20: System development.* 20.01.13

3.130
application software
1. software designed to help users perform particular tasks or handle particular types of problems, as distinct from software that controls the computer itself.4.5. ***2.*** software or a program that is specific to the solution of an application problem. *ISO/IEC 2382-1:1993 Information technology — Vocabulary — Part 1: Fundamental terms.*01.04.01. ***3.*** software designed to fulfill specific needs of a user

3.131
application-oriented language
1. a computer language with facilities or notations applicable primarily to a single application area

cf. authoring language, query language, specification language

EXAMPLE a language for computer-assisted instruction or hardware design

3.132
apportioned effort
1. effort applied to project work in proportion to measurable discrete work efforts, but which is not readily divisible into discrete efforts

cf. discrete effort

3.133
appraisal findings
1. results of an appraisal that identify the most important issues, problems, or opportunities for process improvement within the appraisal scope

NOTE Appraisal findings are inferences drawn from corroborated objective evidence.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.134
appraisal participants
1. members of the organizational unit who participate in providing information during an appraisal

3.135
appraisal team leader
1. person who leads the activities of an appraisal and has satisfied qualification criteria for experience, knowledge, and skills defined by the appraisal method

3.136
approval
1. written notification, by an authorized representative, that an information item appears to satisfy requirements, is complete. *ISO/IEC 15289:2006, Systems and software engineering — Contents of systems and software life cycle information products (Documentation)* 5.1

NOTE Such approval does not shift responsibility from the supplier to meet requirements under a two-party situation.

3.137
approved change request
1. [Output/Input] a change request that has been processed through the integrated change control process and approved. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.138
approved modification
1. the disposition of one or more proposed changes authorizing change to any SCIs. *ISO/IEC TR 15846:1998, Information technology —Software life cycle processes —Configuration Management.* 4.1

NOTE There may be a many-to-many relationship of "proposed change" to "approved modification". A proposed change may cause modifications in several SCIs (even if only to the code and its test case). A modification may originate from several proposed changes, approved simultaneously or over a period of time while the modification is still in progress.

3.139
A-profile
1. Application profile. *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing — Reference model: Overview*

3.140
arc
1. a directed edge of a net which may connect a place to a transition or a transition to a place, normally represented by an arrow. *ISO/IEC 15909-1:2004, Software and system engineering —High-level Petri nets — Part 1: Concepts, definitions and graphical notation.*2.1.1

3.141
arc annotation
1. an expression that may involve constants, variables and operators used to annotate an arc of a net. *ISO/IEC 15909-1:2004, Software and system engineering — High-level Petri nets — Part 1: Concepts, definitions and graphical notation.*2.1.1.3. The expression must evaluate to a multiset over the type of the arc's associated place

3.142
architect
1. the person, team, or organization responsible for systems architecture. *IEEE Std 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.*3.2

3.143
architecting
1. the activities of defining, documenting, maintaining, improving, and certifying proper implementation of an architecture. *IEEE Std 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.*3.3

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.144
architectural description (AD)
1. a collection of products to document an architecture. *IEEE Std 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.*3.4

3.145
architectural design
1. the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system ***2.*** the result of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system

cf. functional design

3.146
architectural design phase
1. the life-cycle phase in which a system's general architecture is developed, thereby fulfilling the requirements laid down by the software requirements document and detailing the implementation plan in response to it

3.147
architectural design review

1. a joint acquirer-supplier review to evaluate the technical adequacies of the software architectural design as depicted in the software design descriptions

3.148
architectural structure
1. a physical or logical layout of the components of a system design and their internal and external connections

EXAMPLE function-oriented (structured) design, object-oriented design, and data structure- oriented design

3.149
architectural style
1. definition of a family of systems in terms of a pattern of structural organization. ***2.*** characterization of a family of systems that are related by sharing structural and semantic properties

EXAMPLE pipes and filters, layers, rule-based systems, and blackboards

3.150
architecture
1. fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution. *ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering —System life cycle processes.*4.5. ***2.*** the organizational structure of a system or component. ***3.*** the organizational structure of a system and its implementation guidelines. *Syn:* architectural structure

cf. component, module, subprogram, routine

NOTE sometimes refers to the design of a system's hardware and software components

3.151
archival pages
1. on-line data that is 1) not expected to change, and 2) no longer maintained. *IEEE Std 2001-2002 IEEE Recommended Practice for the Internet —Web Site Engineering, Web Site Management, and Web Site Life Cycle.*3.1.1

NOTE This data also may not be readily renderable by future tools.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.152
argument
1. an independent variable. ***2.*** a specific value of an independent variable. ***3.*** a constant, variable, or expression used in a call to a software module to specify data or program elements to be passed to that module

EXAMPLE the variable *m* in the equation $E = mc^2$

3.153
argument sort
1. the sort of an argument of an operator. *ISO/IEC 15909-1:2004, Software and system engineering —High-level Petri nets —Part 1: Concepts, definitions and graphical notation.* 2.1.23.1. *Syn:* input sort

3.154
arity
1. the number of roles that participate in a relationship. *ISO/IEC 15474-1:2002, Information technology — CDIF framework — Part 1: Overview.* 4.2. ***2.*** the input sorts and output sort for an operator. *ISO/IEC 15909-1:2004 Software and system engineering —High-leve l Petri nets —Part 1: Concepts, definitions and graphical notation.*2.1.2

NOTE A binary relationship has an arity of two. An n-ary relationship has an arity of n. (n>2) sometimes known as the ""degree"" of a relationship.

3.155
array
1. an n-dimensional ordered set of data items identified by a single name and one or more indices, so that each element of the set is individually addressable

EXAMPLE a matrix, table, or vector

3.156
arrow
1. a directed line, composed of one or more connected arrow segments in a single diagram from a single source (box or diagram boundary) to a single use (box or diagram boundary). *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.* 2.1.8. ***2.*** graphic presentation of a logical relationship between schedule activities in the precedence diagramming method

cf. arrow segment, boundary arrow, internal arrow

3.157
arrow label
1. a noun or noun phrase associated with an arrow segment to signify the arrow meaning of the arrow segment. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.9

NOTE Specifically, an arrow label identifies the object type set that is represented by an arrow segment.

3.158
arrow meaning
1. the object types of an object type set, regardless of how these object types may be collected, aggregated, grouped, bundled, or otherwise joined within the object type set. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0.* 2.1.10

EXAMPLE a physical thing, a data element

3.159
arrow role
1. the relationship between an object type set represented by an arrow segment and the activity represented by the box to which the arrow segment is attached. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0.* 2.1.12

NOTE There are four arrow roles: input, control, output, and mechanism.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.160
arrow segment
1. a directed line that originates at a box side, arrow junction (branch or join), or diagram boundary and terminates at the next box side, arrow junction (branch or join), or diagram boundary that occurs in the path of the line. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.13

3.161
artificial intelligence (AI)
1. the branch of computer science devoted to developing data processing systems that perform functions normally associated with human intelligence, such as reasoning, learning, and self-improvement. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.06.12

3.162
artificial language
1. a language whose rules are explicitly established prior to its use. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.05.09. *Syn:* formal language

3.163
ask
1. the combination of a specific activity; a demanded execution time, defined by a specific timeliness function; a specific task mode. *ISO/IEC 14756:1999, Information technology — Measurement and rating of performance of computer-based software systems.*4.19

3.164
ASO
1. Application Service Object. *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing —Reference model: Overview*

3.165
assemble
1. to translate a computer program expressed in an assembly language into its machine language equivalent.
2. the process of constructing from parts one or more identified pieces of software. *IEEE Std 1517-1999 (R2004) IEEE Standard for Information Technology — Software Life Cycle Processes — Reuse Processes.*3.2

cf. compile, disassemble, interpret

3.166
assemble-and-go
1. an operating technique in which there are no stops between the assembling, linking, loading, and execution of a computer program

3.167
assembled origin
1. the address of the initial storage location assigned to a computer program by an assembler, a compiler, or a linkage editor

cf. loaded origin, offset (1), starting address

3.168
assembler
1. a computer program that translates programs expressed in assembly language into their machine language equivalents

cf. absolute assembler, compiler, cross-assembler, interpreter, relocating assembler

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.169
assembly code
1. computer instructions and data definitions expressed in a form that can be recognized and processed by an assembler. *Syn:* assembler code

cf. compiler code, interpretive code, machine code

3.170
assembly language
1. a programming language that corresponds closely to the instruction set of a given computer, allows symbolic naming of operations and addresses, and usually results in a one-to-one translation of program instructions into machine instructions. *Syn:* assembler language, low-level language, second-generation language

cf. fifth-generation language, fourth-generation language, high order language, machine language

3.171
assertion
1. a logical expression specifying a program state that must exist or a set of conditions that program variables must satisfy at a particular point during program execution. ***2.*** a function or macro that complains loudly if a design assumption on which the code is based is not true

cf. invariant, proof of correctness

NOTE Types include input assertion, loop assertion, output assertion.

3.172
assessed capability
1. the output of one or more relevant process assessments conducted in accordance with the provisions of ISO/IEC 15504. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.*3.2

3.173
assessment
1. an action of applying specific documented criteria to a specific software module, package or product for the purpose of determining acceptance or release of the software module, package or product. *ISO/IEC 14102:2008, Information technology —Guideline for the evaluation and selection of CASE tools.*3.1

3.174
assessment constraints
1. restrictions placed on the use of the assessment outputs and on the assessment team's freedom of choice regarding the conduct of the assessment. *ISO/IEC 15504-1:2004, Information technology — Process assessment —Part 1: Concepts and vocabulary.* 3.3

3.175
assessment indicator
1. sources of objective evidence used to support the assessors' judgment in rating process attributes. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.4

EXAMPLE work products, practice, or resource

3.176
assessment input
1. information required before a process assessment can commence. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.5

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.177
assessment instrument
1. a tool or set of tools that is used throughout an assessment to assist the assessor in evaluating the performance or capability of processes, in handling assessment data and in recording the assessment results. *ISO/IEC 15504-1:2004, Information technology — Process assessment — Part 1: Concepts and vocabulary.*3.6

3.178
assessment output
1. the tangible results from an assessment. *ISO/IEC 15504-1:2004, Information technology — Process assessment —Part 1: Concepts and vocabulary.* 3.7

cf. assessment record

3.179
assessment participant
1. an individual who has responsibilities within the scope of the assessment. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.8

EXAMPLE the assessment sponsor, assessors, and organizational unit members

3.180
assessment process
1. a determination of the extent to which the organization's standard processes contribute to the achievement of its business goals and to help the organization focus on the need for continuous process improvement [. *ISO/IEC 15504-1:2004, Information technology — Process assessment — Part 1: Concepts and vocabulary.*3.9

3.181
assessment purpose
1. a statement, provided as part of the assessment input, which defines the reasons for performing the assessment. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.*3.10

3.182
assessment record
1. an orderly, documented collection of information which is pertinent to the assessment and adds to the understanding and verification of the process profiles generated by the assessment. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.11

3.183
assessment scope
1. a definition of the boundaries of the assessment, provided as part of the assessment input, encompassing the organizational limits of the assessment, the processes to be included, and the context within which the processes operate. *ISO/IEC 15504-1:2004, Information technology — Process assessment — Part 1: Concepts and vocabulary.*3.12

cf. process context

3.184
assessment sponsor
1. the individual or entity, internal or external to the organizational unit being assessed, who requires the assessment to be performed, and provides financial or other resources to carry it out. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.13

3.185
assessment team
1. one or more individuals who jointly perform a process assessment. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.14

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.186
assessor
1. an individual who participates in the rating of process attributes. *ISO/IEC 15504-1:2004, Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.15

NOTE An assessor is either a competent assessor or a provisional assessor.

3.187
asset
1. an item that has been designed for use in multiple contexts. *IEEE Std 1012-2004 IEEE Standard for Software Verification and Validation.*3.1.3. ***2.*** an advantage or resource. *ISO/IEC 20926:2003, Software engineering —IFPUG 4.1 Unadjusted functional size measurement method —Counting practices manual* ***3.*** a capital asset of the enterprise. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method —Counting practices manual*

3.188
assignment
1. for a set of variables, the association of a value (of correct type) to each variable. *ISO/IEC 15909-1:2004, Software and system engineering — High-level Petri nets — Part 1: Concepts, definitions and graphical notation.*2.1.3. *Syn:* binding

3.189
assignment statement
1. a computer program statement that assigns a value to a variable

cf. control statement, declaration, clear, initialize, reset

EXAMPLE Y = X - 5

3.190
assist
1. tester intervention in the form of direct procedural help provided by the test administrator to the test participants in order to allow the test to continue when the participants could not complete the tasks on their own. *ISO/IEC 25062:2006, Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) —Common Industry Format (CIF) for usability test reports.* 4.12

3.191
assistive technologies
1. hardware or software that is added to or incorporated within a system that increases accessibility for an individual EXAMPLES Braille displays, screen readers, screen magnification software and eye tracking devices are assistive technologies. *ISO/IEC 25062:2006, Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) —Common Industry Format (CIF) for usability test reports.* 4.11

3.192
association
1. in UML, a relationship between an actor and a use case that indicates that the actor interacts with the system by means of the use case ***2.*** a relationship (binding) between protocol objects (or between a protocol object and an interceptor) that is established independently of the protocol exchanges that support a particular computational interaction. *ISO/IEC 14752:2000 Information technology — Open Distributed Processing — Protocol support for computational interactions.*3.3.2

3.193
association management facility
1. a set of service primitives which support the management of an association between protocol objects. *ISO/IEC 14752:2000, Information technology — Open Distributed Processing — Protocol support for computational interactions.*3.3.3

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.194
associative class
1. a class introduced to resolve a many-to-many relationship. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject).*3.1.7

3.195
associative entity
1. an entity used to represent a relationship between other entities An associative entity is used when a relationship does not otherwise provide sufficient mechanisms. *ISO/IEC 15474-1:2002, Information technology —CDIF framework —Part 1: Overview.*4.2

3.196
associative entity type
1. an entity type that contains attributes which further describe a many-to-many relationship between two other entity types. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method —Counting practices manual*

cf. entity type

3.197
associative literal
1. a literal that denotes an instance in terms of its value. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject).*3.1.8

NOTE The form of expression used to state an associative literal is className with propertyName: propertyValue.

3.198
assumptions
1. factors that, for planning purposes, are considered to be true, real, or certain without proof or demonstration. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.199
assumptions analysis
1. [Technique] a technique that explores the accuracy of assumptions and identifies risks to the project from inaccuracy, inconsistency, or incompleteness of assumptions. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.200
asynchronous
1. pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing. *ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.*01.01.29

3.201
asynchronous I/O device
1. an I/O device that generates an interrupt after producing some input or generating some output

3.202
asynchronous I/O device interface task
1. a task that interfaces to an I/O device and is activated by interrupts from that device

3.203
asynchronous message communication
1. communication in which a producer task sends a message to a consumer task and does not wait for a response. *Syn:* loosely coupled message communication

NOTE A message queue could build up between the tasks.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.204
atomic type
1. a data type, each of whose members consists of a single, nondecomposable data item. *Syn:* primitive type

cf. composite type

3.205
attribute
1. a property associated with a set of real or abstract things that is some characteristic of interest. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject)*.3.1.9. ***2.*** inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means. *ISO/IEC 15939:2007, Systems and software engineering--Measurement process*.3.2; *ISO/IEC 25000:2005, Software Engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE.* 4.3. ***3.*** a measurable physical or abstract property of an entity. *IEEE Std 1061-1998 (R2004) IEEE Standard for Software Quality Metrics Methodology*.2.1. ***4.*** an identifiable association between an object and a value. *ISO/IEC 19500-2:2003, Information technology —Open Distributed Processing —Part 2: General Inter-ORB Protocol (GIOP)/Internet Inter-ORB Protocol (IIOP)*.3.2.2. ***5.*** a function from the instances of a class to the instances of the value class of the attribute. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject)*.3.1.9. ***6.*** a unique item of information about an entity. *ISO/IEC 20968:2002, Software engineering —Mk II Function Point Analysis —Counting Practices Manual.* 10. ***7.*** a single-valued characteristic of an entity or relationship. *ISO/IEC 15474-1:2002, Information technology — CDIF framework —Part 1: Overview.* 4.2

EXAMPLE people, objects, places, events, ideas, combinations of things

NOTE can refer either to general characteristics such as reliability, maintainability, and usability or to specific features of a software product. ISO 9000 distinguishes two types of attributes: a permanent characteristic existing inherently in something; and an assigned characteristic of a product, process or system (e.g. the price of a product, the owner of a product). The assigned characteristic is not an inherent quality characteristic of that product, process or system. An attribute expresses some characteristic that is generally common to the instances of a class. The name of the attribute is the name of the role that the value class plays in describing the class, which may simply be the name of the value class (as long as using the value class name does not cause ambiguity).

3.206
attribute for quality measure
1. attribute that relates to software product itself, to the use of the software product or to its development process. *ISO/IEC 25000:2005, Software Engineering — Software product Quality Requirements and Evaluation (SQuaRE) —Guide to SQuaRE.* 4.4

NOTE Attributes for quality measure are used in order to obtain quality measure elements.

3.207
attribute indicator
1. an assessment indicator that supports the judgment of the extent of achievement of a specific process attribute. *ISO/IEC 15504-1:2004, Information technology — Process assessment — Part 1: Concepts and vocabulary*.3.16

3.208
attribute name
1. a role name for the value class of the attribute. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject)*.3.1.10

3.209
attributed relationship
1. a relationship that has attributes. *ISO/IEC 15474-1:2002, Information technology — CDIF framework — Part 1: Overview*.4.2

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.210
attributive entity type
1. an entity type that further describes one or more attributes of another entity type. *ISO/IEC 20926:2003, Software engineering —IFPUG 4.1 Unadjusted functional size measurement method —Counting practices manual*

cf. entity

3.211
audience
1. category of users sharing the same or similar characteristics and needs (for example, reason for using the documentation, tasks, education level, abilities, training, experience). *ISO/IEC 26514, Systems and software engineering —Requirements for designers and developers of user documentation.* 4.6

NOTE There may be different audiences for documentation (for example, management, data entry, maintenance) that determine the content, structure, and use of the intended documentation.

3.212
audience research
1. planned process of interview, and of the analysis of interview records and personnel records. *ISO/IEC 15910:1999, Information technology —Software user documentation process.* 4.4

NOTE The purpose of audience research is to determine the abilities, training, experience, limitations, prejudices and preferences of the intended readers of a document.

3.213
audit
1. independent assessment of products and processes, conducted by an authorized person to assess compliance with requirements. *ISO/IEC 12207:2008 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes.* 4.5. ***2.*** an independent examination of a software product, software process, or set of software processes to assess compliance with specifications, standards, contractual agreements, or other criteria. *IEEE Std 1028-2008 IEEE Standard for Software Reviews and Audits.*3.2. ***3.*** an independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria ***4.*** systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which audit criteria are fulfilled. *ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes.*4.6

NOTE An audit should result in a clear indication of whether the audit criteria have been met.

3.214
authoring language
1. a high-level programming language used to develop courseware for computer-assisted instruction

cf. authoring system

3.215
authoring system
1. a programming system that incorporates an authoring language

3.216
authority
1. the right to apply project resources, expend funds, make decisions, or give approvals. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.217
authorization
1. a prescription that a particular behavior shall not be prevented. *ISO/IEC 15414:2006, Information technology —Open distributed processing —Reference model —Enterprise language.* 6.4.2

NOTE Unlike a permission, an authorization is an empowerment.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.218
automate
1. to make a process or equipment automatic. *ISO/IEC 2382-1:1993, Information technology —Vocabulary — Part 1: Fundamental terms.*01.01.13

3.219
automated or assisted software process
1. software process that is performed either fully or partially supported by CASE tools. *ISO/IEC 15940:2006, Information Technology —Software Engineering Environment Services.* 2.2.3

3.220
automated verification system
1. a software tool that accepts as input a computer program and a representation of its specification and produces, possibly with human help, a proof or disproof of the correctness of the program ***2.*** a software tool that automates part or all of the verification process

3.221
automatic
1. pertaining to a process or equipment that, under specified conditions, functions without human intervention. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.01.12

3.222
automation
1. the conversion of processes or equipment to automatic operation, or the results of the conversion. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.01.14

3.223
availability
1. the degree to which a system or component is operational and accessible when required for use. ***2.*** ability of a component or service to perform its required function at a stated instant or over a stated period of time. *ISO/IEC 20000-1:2005, Information technology —Service management —Part 1: Specification.* 2.1

cf. error tolerance, fault tolerance, robustness

NOTE Often expressed as a probability. Availability is usually expressed as a ratio of the time that the service is actually available for use by the business to the agreed service hours.

3.224
B5
1. International Standard paper size, 176 mm by 250 mm. *ISO/IEC 15910:1999, Information technology — Software user documentation process.*4.5

NOTE See ISO 216:2007.

3.225
BAC
1. budget at completion. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.226
back matter
1. material that appears at the end of a book or manual, such as an index. *ISO/IEC 15910:1999, Information technology —Software user documentation process.* 4.6

3.227
background
1. in job scheduling, the computing environment in which low-priority processes or those not requiring user interaction are executed

cf. foreground. background processing

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.228
background processing
1. the execution of a low-priority process while higher priority processes are not using computer resources, or the execution of processes that do not require user interaction

cf. foreground processing

3.229
backout
1. to undo the effects of a commit

NOTE Often by introducing a new commit that restores things to their previous state.

3.230
back-to-back testing
1. testing in which two or more variants of a program are executed with the same inputs, the outputs are compared, and errors are analyzed in case of discrepancies

cf. mutation testing

3.231
backup
1. a system, component, file, procedure, or person available to replace or help restore a primary item in the event of a failure or externally caused disaster ***2.*** to create or designate a system, component, file, procedure, or person as a replacement. *Syn:* back-up

3.232
backup programmer
1. the assistant leader of a chief programmer team

cf. chief programmer

NOTE Responsibilities include contributing significant portions of the software being developed by the team, aiding the chief programmer in reviewing the work of other team members, substituting for the chief programmer when necessary, and having an overall technical understanding of the software being developed.

3.233
backward pass
1. the calculation of late finish dates and late start dates for the uncompleted portions of all schedule activities. Determined by working backwards through the schedule network logic from the project's end date. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

cf. schedule network analysis

3.234
backward recovery
1. the reconstruction of a file to a given state by reversing all changes made to the file since it was in that state. ***2.*** a type of recovery in which a system, program, database, or other system resource is restored to a previous state in which it can perform required functions

cf. forward recovery

3.235
bag
1. a collection class whose members are unordered but in which duplicates are meaningful. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject)*.3.1.11

cf. list, set

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.236
base address
1. an address used as a reference point to which a relative address is added to determine the address of the storage location to be accessed

cf. indexed address, relative address, self-relative address

3.237
base functional component (BFC)
1. an elementary unit of functional user requirements defined by and used by an FSM method for measurement purposes. *ISO/IEC 14143-1:2007, Information technology — Software measurement — Functional size measurement; Part 1: Definition of concepts.*3.1. *Syn:* functional service

EXAMPLE a functional user requirement could be "Maintain Customers" which may consist of the following BFCs: "Add a new customer", "Report Customer Purchases", and "Change Customer Details". Another example might include a collection of logically related business data maintained by the software under study such as "Customer Details"

3.238
base measure
1. measure defined in terms of an attribute and the method for quantifying it. *ISO/IEC 15939:2007, Systems and software engineering — Measurement process.*3.3; *ISO/IEC 25000:2005, Software Engineering — Software product Quality Requirements and Evaluation (SQuaRE) —Guide to SQuaRE.* 4.5

NOTE A base measure is functionally independent of other measures.

3.239
base practice
1. an activity that, when consistently performed, contributes to achieving a specific process purpose. *ISO/IEC 15504-1:2004 Information technology —Process assessment —Part 1: Concepts and vocabulary.* 3.17

3.240
baseline
1. specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. *ISO/IEC 12207:2008 (IEEE Std 12207-2008), Systems and software engineering — Software life cycle processes.*4.6, *ISO/IEC 15288:2008 (IEEE Std 15288-2008), Systems and software engineering — System life cycle processes.*4.7. ***2.*** formally approved version of a configuration item, regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle. *ISO/IEC 19770-1:2006, Information technology — Software asset management — Part 1: Processes.* 3.1. ***3.*** agreement or result designated and fixed at a given time, from which changes require justification and approval. ***4.*** document or a set of such documents formally designated and fixed at a specific time during the life cycle of a configuration item ***5.*** work product that has been placed under formal configuration management. ***6.*** snapshot of the state of a service or individual configuration items at a point in time. *ISO/IEC 20000-1:2005, Information technology — Service management — Part 1: Specification.* 2.2. ***7.*** description of a system and its components (configuration items) at a particular period including any approved updates. ***8.*** an approved plan (for a project), plus or minus approved changes. It is compared to actual performance to determine if performance is within acceptable variance thresholds. Generally refers to the current baseline, but may refer to the original or some other baseline. Usually used with a modifier (e.g., cost performance baseline, schedule baseline, performance measurement baseline, technical baseline). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

NOTE A baseline should be changed only through formal configuration management procedures. Some baselines may be project deliverables while others provide the basis for further work. Baselines, plus approved changes from those baselines, constitute the current configuration identification.

3.241
baseline design
1. a system design that has been agreed on by all stakeholders interested in the system development

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.242
baseline document
1. a system/software document that defines a work product that has been placed under configuration management

EXAMPLE system specifications, requirements specifications, and design specifications

3.243
baseline function point count
1. application function point count taken of the functionality at a point in time, from which changes can be measured. *ISO/IEC 20968:2002, Software engineering —Mk II Function Point Analysis —Counting Practices Manual.*10

3.244
baseline management
1. in configuration management, the application of technical and administrative direction to designate the documents and changes to those documents that formally identify and establish baselines at specific times during the life cycle of a configuration item

3.245
basic engineering object
1. an engineering object that requires the support of a distributed infrastructure. *ISO/IEC 10746-3:1996, Information technology —Open Distributed Processing —Reference Model: Architecture.* 8.1.1

3.246
basic flow
1. the part of a use case that describes its most common implementation. *Syn:* basic path, happy day scenario

NOTE The basic flow is written assuming that no errors or alternatives exist.

3.247
basic interworking facility
1. a set of service primitives which have a direct correspondence with computational signals which model computational operations. *ISO/IEC 14752:2000, Information technology — Open Distributed Processing — Protocol support for computational interactions.*3.3.4

3.248
basic symbol
1. symbol used when the precise nature or form of, for example, the process or data media is not known or when it is not necessary to depict the actual medium. *ISO 5807:1985, Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.*3.1

3.249
basis set
1. the set of objects used to create a multiset. *ISO/IEC 15909-1:2004, Software and system engineering — High-level Petri nets —Part 1: Concepts, definitions and graphical notation.*2.1.4

3.250
batch
1. pertaining to a system or mode of operation in which inputs are collected and processed all at one time, rather than being processed as they arrive, and a job, once started, proceeds to completion without additional input or user interaction

cf. conversational, interactive, online, real time

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.251
bathtub curve
1. a graph of the number of failures in a system or component as a function of time

NOTE The name is derived from the usual shape of the graph: a period of decreasing failures (the early-failure period), followed by a relatively steady period (the constant-failure period), followed by a period of increasing failures (the wearout-failure period).

3.252
BCWP
1. budgeted cost of work performed. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.253
BCWS
1. budgeted cost of work scheduled. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.254
behavior
1. observable activity of a system, measurable in terms of quantifiable effects on the environment whether arising from internal or external stimulus. *IEEE Std 1175.1-2002 (R2007) IEEE Guide for CASE Tool Interconnections-Classification and Description.*3.1. ***2.*** the peculiar reaction of a thing under given circumstances. *IEEE Std 1175.1-2002 (R2007) IEEE Guide for CASE Tool Interconnections-Classification and Description.*3.1. ***3.*** the aspect of an instance's specification that is determined by the state-changing operations it can perform. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject).*3.1.12

3.255
behavior specification
1. a structured collection of data that describes the potential variety of behavior possible from a system. *IEEE Std 1175.1-2002 (R2007) IEEE Guide for CASE Tool Interconnections —Classification and Description.* 3.2

3.256
benchmark
1. a standard against which measurements or comparisons can be made. ***2.*** a procedure, problem, or test that can be used to compare systems or components to each other or to a standard. ***3.*** a recovery file

3.257
BEO
1. Basic Engineering Object. *ISO/IEC 10746-1:1998, Information technology — Open Distributed Processing —Reference model: Overview*

3.258
beta testing
1. second stage of testing when a product is in limited production use

cf. alpha testing

NOTE often performed by a client or customer

3.259
BFC
1. base functional component. *ISO/IEC 14143-1:2007, Information technology — Software measurement — Functional size measurement; Part 1: Definition of concepts.*4

3.260
BFC class
1. defined group of BFC types. *ISO/IEC 29881:2008, Information technology — Software and systems engineering —FiSMA 1.1 functional size measurement method.* 3.1

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.261
BFC Type
1. a defined category of BFCs. *ISO/IEC 14143-1:2007, Information technology —Software measurement — Functional size measurement; Part 1: Definition of concepts.*3.2

EXAMPLE 'External Inputs', 'External Outputs' and 'Logical Transactions', and data stores such as 'Internal Logical Files'

3.262
bidirectional traceability
1. association among two or more logical entities that is discernable in either direction (to and from an entity)

cf. requirements traceability

3.263
big-bang testing
1. a type of integration testing in which software elements, hardware elements, or both are combined all at once into an overall system, rather than in stages

3.264
bill of materials (BOM)
1. documented formal hierarchical tabulation of the physical assemblies, subassemblies, and components needed to fabricate a product

3.265
binary digit (bit)
1. a unit of information that can be represented by either a zero or a one. ***2.*** an element of computer storage that can hold a unit of information as in (1). ***3.*** a numeral used to represent one of the two digits in the binary numeration system; zero (0) or one (1)

3.266
bind
1. to assign a value to an identifier

cf. dynamic binding, static binding

EXAMPLE to assign a value to a parameter or to assign an absolute address to a symbolic address in a computer program

3.267
binder
1. an engineering object in a channel, which maintains a distributed binding between interacting basic engineering objects. *ISO/IEC 10746-3:1996, Information technology — Open Distributed Processing — Reference Model: Architecture.*8.1.10

3.268
binding endpoint identifier
1. an identifier, in the naming context of a capsule, used by a basic engineering object to select one of the bindings in which it is involved, for the purpose of interaction. *ISO/IEC 10746-3:1996, Information technology —Open Distributed Processing —Reference Model: Architecture.* 8.1.15

EXAMPLE a memory address (for a data structure representing an engineering interface)

NOTE The same form of binding endpoint identifier can be used, whether the binding involved is either local or distributed.

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.269
binding object
1. a computational object which supports a binding between a set of other computational objects. *ISO/IEC 10746-3:1996, Information technology — Open Distributed Processing — Reference Model: Architecture.*7.1.14

3.270
bit
1. either of the digits 0 or 1 when used in the binary numeration system. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.02.08. ***2.*** binary digit

3.271
bit steering
1. a microprogramming technique in which the meaning of a field in a microinstruction is dependent on the value of another field in the microinstruction. *Syn:* immediate control

cf. residual control, two-level encoding

3.272
black box
1. a system or component whose inputs, outputs, and general function are known but whose contents or implementation are unknown or irrelevant ***2.*** pertaining to an approach that treats a system or component whose inputs, outputs, and general function are known but whose contents or implementation are unknown or irrelevant

cf. glass box

3.273
block
1. a group of contiguous storage locations, computer program statements, records, words, characters, or bits that are treated as a unit ***2.*** to form a group

cf. block-structured language, delimiter

3.274
block diagram
1. a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. *ISO/IEC 2382-1:1993, Information technology —Vocabulary — Part 1: Fundamental terms.*01.01.27. ***2.*** a diagram of a system, computer, or device in which the principal parts are represented by suitably annotated geometrical figures to show both the functions of the parts and their functional relationships. *Syn:* configuration diagram, system resources chart

cf. box diagram, bubble chart, flowchart, graph, input-process-output chart, structure chart

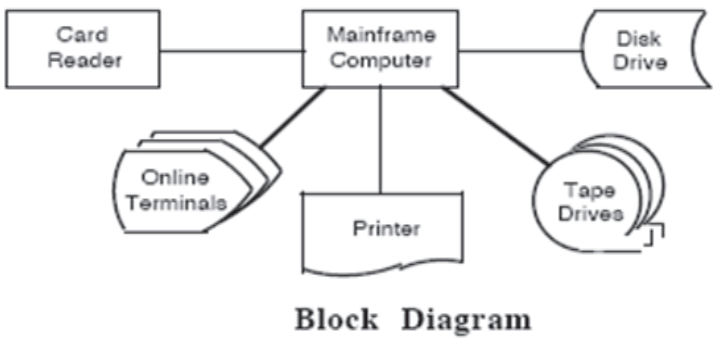

Figure 2 —Block Diagram

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.275
blocking factor
1. the number of records, words, characters, or bits in a block

3.276
block-structured language
1. a design or programming language in which sequences of statements, called blocks, are defined, usually with begin and end delimiters, and variables or labels defined in one block are not recognized outside that block

cf. structured programming language

EXAMPLE Ada, ALGOL, PL/I

3.277
BMT
1. Bench Mark Test. *ISO/IEC 14102:2008, Information technology — Guideline for the evaluation and selection of CASE tools.*8.2

3.278
body metadata
1. elements in the body of an HTML document providing administrative and/or navigational facilities for the user or administrator. *IEEE Std 2001-2002 IEEE Recommended Practice for the Internet — Web Site Engineering, Web Site Management, and Web Site Life Cycle.*3.1.2

3.279
BOM
1. bill of materials

3.280
Boolean expression
1. an expression that evaluates to true or false. *ISO/IEC 15909-1:2004, Software and system engineering — High-level Petri nets —Part 1: Concepts, definitions and graphical notation.*2.1.26.1

3.281
Boolean signature
1. a signature where one of the sorts is Bool, corresponding to the carrier Boolean in any associated algebra, and one of the constants is true sub Bool, corresponding to the value true in the algebra. *ISO/IEC 15909-1:2004, Software and system engineering —High-level Petri nets —Part 1: Concepts, definitions and graphical notation.*2.1.22.1

3.282
boot
1. to initialize a computer system by clearing memory and reloading the operating system

NOTE derived from bootstrap

3.283
bootstrap
1. a short computer program that is permanently resident or easily loaded into a computer and whose execution brings a larger program, such as an operating system or its loader, into memory ***2.*** to use a program to bring up a larger program, such as an operating system

3.284
bootstrap loader
1. a short computer program used to load a bootstrap

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.285
bottom-up
1. pertaining to an activity that starts with the lowest-level components of a hierarchy and proceeds through progressively higher-levels ***2.*** pertaining to a method or procedure that starts at the lowest level of abstraction and proceeds towards the highest level. *ISO/IEC 2382-20:1990, Information technology — Vocabulary — Part 20: System development.*20.01.11

cf. topdown. critical piece first

EXAMPLE bottom-up design, bottom-up testing

3.286
bottom-up design
1. a design approach in which low-level pieces of a system are combined into an overall design. ***2.*** the process of designing a system by identifying low-level components, designing each component separately, and then designing a structure to integrate the low-level components into larger and larger subsystems until the design is finished

3.287
bottom-up estimating
1. [Technique] a method of estimating a component of work. The work is decomposed into more detail. An estimate is prepared of what is needed to meet the requirements of each of the lower, more detailed pieces of work, and these estimates are then aggregated into a total quantity for the component of work. The accuracy of bottom-up estimating is driven by the size and complexity of the work identified at the lower levels. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.288
boundary
1. conceptual interface between the software under study and its users. *ISO/IEC 29881:2008, Information technology —Software and systems engineering —FiSMA 1.1 functional size measurement method.* 3.2

NOTE The boundary provides the measurement analyst(s) with a solid delimiter to distinguish, without ambiguity, what is included inside the measured software from what is part of the measured software's operating environment.

3.289
boundary arrow
1. an arrow with one end (source or use) not connected to any box in a diagram. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0.* 2.1.14

cf. internal arrow

3.290
boundary ICOM code
1. an ICOM code that maps an untunneled boundary arrow in a child diagram to an arrow attached to the parent box that is detailed by that diagram. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0.* 2.1.15

3.291
boundary value
1. a data value that corresponds to a minimum or maximum input, internal, or output value specified for a system or component

cf. stress testing

3.292
box
1. a rectangle containing a box name, a box number, and possibly a box detail reference and representing a function in a diagram. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.16

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.293
box detail reference
1. a square enclosure encompassing a box number, which indicates that the box is decomposed or detailed by a child diagram. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.17

3.294
box diagram
1. a control flow diagram consisting of a rectangle that is subdivided to show sequential steps, if-then-else conditions, repetition, and case conditions. *Syn:* Chapin chart, Nassi-Shneiderman chart

cf. block diagram, bubble chart, flowchart, graph, input-process-output chart, program structure diagram, structure chart

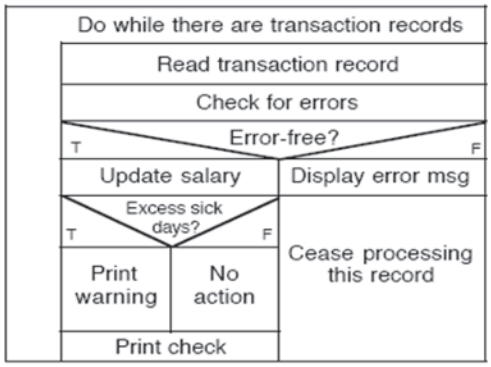

Figure 3 —Box diagram

3.295
box ICOM code
1. an ICOM code that maps a tunneled boundary arrow to an arrow attached to some ancestral box. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.18

3.296
box name
1. the verb or verb phrase placed inside a box that names the modeled function. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0.* 2.1.19

cf. function name

NOTE A box takes as its box name the function name of the function represented by the box.

3.297
box number
1. a single digit (0, 1, 2, ..., 9) placed in the lower right corner of a box to uniquely identify that box in a diagram. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.20

NOTE The only box that may be numbered 0 is the box that represents the A0 function in A-0 and A-1 context diagrams.

3.298
brainstorming
1. [Technique] a general data gathering and creativity technique that can be used to identify risks, ideas, or solutions to issues by using a group of team members or subject-matter experts. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.299
branch
1. a computer program construct in which one of two or more alternative sets of program statements is selected for execution. ***2.*** a point in a computer program at which one of two or more alternative sets of program statements is selected for execution. ***3.*** a junction at which a root arrow segment (going from source to use) divides into two or more arrow segments. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language - Syntax and Semantics for IDEF0.*2.1.21. ***4.*** to perform the selection in (1). ***5.*** any of the alternative sets of program statements in (1). ***6.*** a set of evolving source file versions

cf. case, jump, go to, if-then-else

NOTE Every branch is identified by a tag. Often, a branch identifies the file versions that have been or will be released as a product release. May denote unbundling of arrow meaning, i.e., the separation of object types from an object type set. Also refers to an arrow segment into which a root arrow segment has been divided.

3.300
branch testing
1. testing designed to execute each outcome of each decision point in a computer program

cf. path testing, statement testing

3.301
breakpoint
1. a point in a computer program at which execution can be suspended to permit manual or automated monitoring of program performance or results

NOTE Types include code breakpoint, data breakpoint, dynamic breakpoint, epilog breakpoint, programmable breakpoint, prolog breakpoint, static breakpoint. A breakpoint is said to be set when both a point in the program and an event that will cause suspension of execution at that point are defined; it is said to be initiated when program execution is suspended.

3.302
bubble chart
1. a data flow, data structure, or other diagram in which entities are depicted with circles (bubbles) and relationships are represented by links drawn between the circles

cf. block diagram, box diagram, flowchart, graph, input-process-output chart, structure chart

Figure 4 —Bubble chart

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.303
budget
1. a planned sequence of expenditures over time with monetary costs assigned to specific tasks or jobs. *ISO/IEC 20926:2003, Software engineering —IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual* ***2.*** the approved estimate for the project or any work breakdown structure component or any schedule activity. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

cf. estimate

NOTE often used also to refer to work effort as well as, or instead of, money

3.304
budget at completion (BAC)
1. the sum of all the budgets established for the work to be performed on a project or a work breakdown structure component or a schedule activity. The total planned value for the project. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.305
buffer
1. a device or storage area used to store data temporarily to compensate for differences in rates of data flow, time of occurrence of events, or amounts of data that can be handled by the devices or processes involved in the transfer or use of the data ***2.*** a routine that accomplishes the objectives in (1). ***3.*** to allocate, schedule, or use devices or storage areas as in (1)

3.306
build
1. an operational version of a system or component that incorporates a specified subset of the capabilities that the final product will provide

3.307
built-in class
1. a class that is a primitive in the IDEF1X metamodel. *IEEE Std 1320.2-1998 (R2004) IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X97 (IDEFobject)*.3.1.13

3.308
bundle
1. an arrow segment that collects multiple meanings into a single construct or abstraction, i.e., an arrow segment that represents an object type set that includes more than one object type. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.* 2.1.22. ***2.*** to combine separate arrow meanings into a composite arrow meaning, expressed by joining arrow segments, i.e., the inclusion of multiple object types into an object type set. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language —Syntax and Semantics for IDEF0.* 2.1.22

3.309
business objective
1. strategy designed by senior management to ensure an organization's continued existence and enhance its profitability, market share, and other factors influencing the organization's success

3.310
busy
1. pertaining to a system or component that is operational, in service, and in use

cf. down, idle, up

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.311
busy time
1. in computer performance engineering, the period of time during which a system or component is

cf. down time, idle time, set-up time, up time

NOTE operational, in service, and in use

3.312
buyer
1. the acquirer of products, services, or results for an organization. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition.* ***2.*** an individual or organization responsible for acquiring a product or service for use by themselves or other users. *IEEE Std 1362-1998 (R2007) IEEE Guide for Information Technology — System Definition — Concept of Operation Document.* 3.2. ***3.*** the person or organization that accepts the system and pays for the project. *IEEE Std 1362-1998 (R2007) IEEE Guide for Information Technology —System Definition —Concept of Operation Document.* 3.2

cf. customer

NOTE A software system could be such a product or service.

3.313
byte
1. a group of adjacent binary digits operated upon as a unit and usually shorter than a computer word (frequently connotes a group of eight bits) ***2.*** an element of computer storage that can hold a group of bits as in (1). ***3.*** a string that consists of a number of bits, treated as a unit, and usually representing a character or a part of a character. *ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms.*01.02.09

3.314
CAD
1. Computer Aided Design. *ISO/IEC 10746-1:1998, Information technology —Open Distributed Processing — Reference model: Overview*

3.315
calculator
1. a device that is suitable for performing arithmetic operations, but that requires human intervention to alter its stored program, if any, and to initiate each operation or sequence of operations. *ISO/IEC 2382-1:1993, Information technology —Vocabulary —Part 1: Fundamental terms.* 01.03.17

NOTE A calculator performs some of the functions of a computer, but usually operates only with frequent human intervention.

3.316
calendar unit
1. the smallest unit of time used in scheduling a project. Calendar units are generally in hours, days, or weeks, but can also be in quarter years, months, shifts, or even in minutes. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) —Fourth Edition*

3.317
call
1. a transfer of control from one software module to another, usually with the implication that control will be returned to the calling module ***2.*** a computer instruction that transfers control from one software module to another as in (1) and, often, specifies the parameters to be passed to and from the module ***3.*** to transfer control from one software module to another as in (1) and, often, to pass parameters to the other module

cf. go to

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010

3.318
call arrow
1. an arrow that enables the sharing of detail between IDEF0 models (linking them together) or within an IDEF0 model. *IEEE Std 1320.1-1998 (R2004) IEEE Standard for Functional Modeling Language — Syntax and Semantics for IDEF0.*2.1.23

NOTE The tail of a call arrow is attached to the bottom side of a box. One or more page references are attached to a call arrow.

3.319
call by name
1. a method for passing parameters, in which the calling module provides to the called module a symbolic expression representing the parameter to be passed, and a service routine evaluates the expression and provides the resulting value to the called module

cf. call by reference, call by value

NOTE Because the expression is evaluated each time its corresponding formal parameter is used in the called module, the value of the parameter may change during the execution of the called module.

3.320
call by reference
1. a method for passing parameters, in which the calling module provides to the called module the address of the parameter to be passed. *Syn:* call by address, call by location

cf. call by name, call by value

NOTE With this method, the called module has the ability to change the value of the parameter stored by the calling module.

3.321
call by value
1. a method of passing parameters, in which the calling module provides to the called module the actual value of the parameter to be passed

cf. call by name, call by reference

NOTE With this method, the called module cannot change the value of the parameter as stored by the calling module.

3.322
call graph
1. a diagram that identifies the modules in a system or computer program and shows which modules call one another. *Syn:* call tree, tier chart

cf. structure chart, control flow diagram, data flow diagram, data structure diagram, state diagram

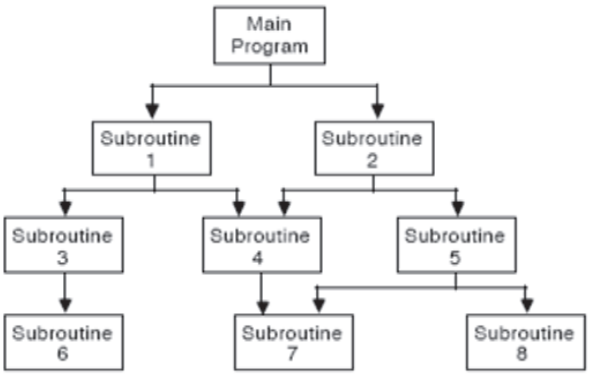

Figure 5 —Call graph

© ISO/IEC 2010 – All rights reserved
© IEEE 2010 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC IEEE 24765:2010