Glossary of Software Engineering Terms

Document ID: SEGLOSSARY
Version: 1.0d
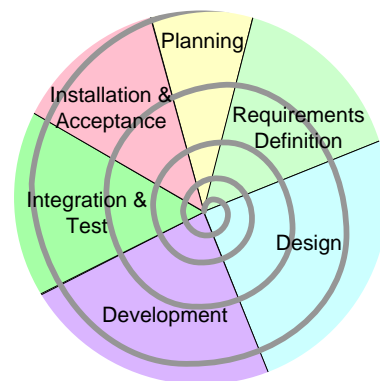
INDEX OF TERMS

ACCEPTANCE CRITERIA

The criteria that the software component, product, or system must satisfy in order to be accepted by the customer.

ACCEPTANCE PROCESS

The process used to verify that a new or modified software product is fully operational and meets the customer's requirements.

ACCEPTANCE TESTING

Formal testing conducted by the customer to determine whether or not a software product or system satisfies the documented acceptance criteria. Successful completion of acceptance testing defines the point at which the customer will accept the product as a successful implementation.

ACTIVITY

A major unit of work to be completed in achieving the objectives of a software project. An activity incorporates a set of tasks to be completed, consumes resources, and results in work products. An activity may contain other activities in a hierarchical manner. All project activities are described in the Project Plan.

ACTOR

A person or system that interacts with the software application in support of a specific process or to perform a specific operation or related set of operations. See *Use Case*.

ALGORITHM

A set of well-defined rules for the solution to a problem in a finite number of steps. Generally implemented as a logical or mathematical test or calculation. See *Business rules*.

ANOMALY

A nice word for "bug." Anything observed in the operation of software that deviates from expectations based on design documentation or user references.

APPLICATION

One or more software executables designed to fulfill a specific set of business functions individually or in cooperation with other applications.

ASSESSMENT

A formal examination of a deliverable, generally by a quality assurance reviewer, for the presence of a specific set of attributes and structural elements. An assessment is not an in-depth examination of content, as the content of a deliverable may be outside the reviewer's domain of expertise. See *Review*.

ASSUMPTION

A condition that is generally accepted as truth without proof or demonstration.

ATTRIBUTE

A piece of information describing part of a particular entity.

AUDIT

An independent examination of software or software documentation to assess compliance with predetermined criteria.

AUTHENTICATION

The ability of each party in a transaction to verify the identity of the other parties.

BANDWIDTH

The capacity of a communications channel.

BASELINE

A set of software components and documents to that has been formerly reviewed and accepted, that serves as the basis for further development or current production, which can be changed only through formal change control procedures. See *Release version*.

BATCH PROCESSING

A method of collecting and processing data in which transactions are accumulated and stored until a specified time when it is convenient or necessary to process them as a group.

BUSINESS PROCESSES

The unique ways in which organizations coordinate and organize work activities, information, and knowledge to produce a product or service. For example, in a sales environment, the information used and steps taken to record a new customer order is considered a business process.

BUSINESS PROCESS COMPLEXITY

A project risk factor that takes into consideration the complexity of the business process or processes under automation. Project risk is considered low when all processes involve fairly simple data entry and update operations. Project risk is considered medium when a minority of the business processes under automation are complex, involving multiple steps, exchanges with external systems or significant validation/processing logic. Project risk is considered high when a majority of the business processes under automation are considered to be complex.

BUSINESS PROCESS MATURITY

A project risk factor that takes into consideration the maturity and stability of the business process or processes to be automated. Project risk is considered low when standard business processes that have been stable and in place for a significant period of time are being automated. Project risk is considered medium when one or more nonstandard but stable business processes, generally unique to the customers situation, are being automated. Project risk rises significantly when the development team is attempting to automate one or more new or unusual business processes.

BUSINESS RULE

A logical or mathematical test that determines whether data entered in a database

complies with an organization's method of conducting its operations.

CLIENT

1. The user point-of-entry for an application. Normally a software executable residing on a desktop computer, workstation, or laptop computer. The user generally interacts directly only with the client, using it to input, retrieve, analyze and report on data.
2. A device or application that receives data from or manipulates a server device or application.

CODE REVIEW

A meeting at which source code is presented for review, comment, or approval.

COMPONENT

One of the parts that make up a system. A component may be hardware, software, or firmware and may be subdivided into other components.

COMPUTER SOFTWARE

Detailed, pre-programmed instructions that control and coordinate the work of computer hardware and firmware components in an information system. See *Application*, *Operating system*.

COMPUTER-AIDED SOFTWARE ENGINEERING (CASE)

The automation of step-by -step methodologies for software and systems development to reduce the amount of repetitive work required of the analyst or developer.

CONFIGURATION MANAGEMENT

A process that effectively controls the coordination and implementation of changes to software components.

CONSTRAINT

A restriction, limitation, or regulation that limits a given course of action.

CONTEXT DIAGRAM

Overview data flow diagram depicting an entire system as a single process with its major inputs and outputs.

CONTROL (RISK)

A method or means used to mitigate hazards and reduce risk.

CONVERSION

The process of changing from the old system to the new system.

CRITICAL SUCCESS FACTORS (CSFs)

A set of specific operational conditions shaped by the business environment that are believed to significantly impact the success potential of an organization or business function. In a software development effort, critical success factors are composed of

assumptions and dependencies that are generally outside the control of the development team.

CUSTOMER RESOURCES

The number of subject matter experts for each Use Case (UC) in an application under development. This project risk factor is considered low when more than one SME is available perUC. A high risk ensues when outside SMEs are involved with a software development effort. See *operational data area, subject matter expert, Use Case.*

DATA

Streams of raw facts representing events before they have been organized and arranged into a form that people can understand and use. See *information.*

DATA DICTIONARY

A structured description of database objects such as tables, indexes, views and fields, with further descriptions of field types, default values and other characteristics.

DATA ELEMENT

See *field.*

DATA ENTITY

A data representation of a real world object or concept. Usually represented as a row in a database table, such as information about a specific Product in inventory.

DATA FLOW DIAGRAM

A primary tool in structured analysis that graphically illustrates a system's component processes and the flow of data between them.

DATA TYPE

A description of how the computer is to interpret the data stored in a particular field. Data types can include text or character string data, integer or floating point numeric data, dates, date/time stamps, true/false values, or Binary Large Objects (BLOBs) which can be used to store images, video, or documents.

DATABASE

A set of related data tables and other database objects, such as a data dictionary, that are organized as a group. A collection of data organized to service many applications at the same time.

DATABASE OBJECT

A component of a database, such as a table or view.

DATABASE ADMINISTRATOR

Person(s) responsible for the administrative functions of databases, such as system security, user access, performance and capacity management, and backup and restoration functions.

DATABASE MANAGEMENT SYSTEM (DBMS)

Software used to create and maintain a database and enable individual business applications to extract the data they need without having to create separate files or data definitions for their own use.

DEFAULT

An initial value assigned to a field by the application when a new database record is created. Used to facilitate data entry by pre-entering common values for the user.

DELIVERABLE

A specific work product, such as requirements or design documentation, produced during a task or activity to validate successful completion of the task or activity. Sometimes, actual software is delivered.

DESIGN DOCUMENT

A less formal term for Software Design Specification (SDS).

DESIGN ELEMENT

A specification for a software object or component that fulfills, or assists in the fulfillment of a functional element. A part of the system design specification.

DESIGN STAGE

A stage in the software development lifecycle that produces the functional and system design specifications for the application under development.

DEVELOPER SKILLS/RESOURCES

The availability of developers and other resources with appropriate skills is a significant factor in project success. When developers and resources are readily available, the likelihood of project success is very high. Most development firms manage multiple projects, allowing some contention between projects for developers and other resources. This project risk factor is considered high when one or more developers with specific skill sets, or resources with specific capabilities, need to be acquired before the project can continue.

DEVELOPMENT STAGE

A stage in the software development lifecycle during which a software product is created from the design specifications and testing is performed on the individual software units produced. See *Unit testing*. Also referred to as the programming stage or implementation stage.

DOCUMENTATION

Information made available to: 1) assist end-users in the operation of a software application, generally in the form of on-line help, or 2) assist developers in locating the correct root procedure or method for a specific software function, generally in the form of an implementation map. Note that printed manuals are rarely delivered with software anymore; on-line documentation is more consistently available from within the application and is easier to use.

ENCRYPTION

The coding and scrambling of messages to prevent unauthorized access to or understanding of the data being stored or transmitted.

END-USER REVIEW

The review of a deliverable for functional accuracy by a Subject Matter Expert who is familiar with the software product under design or development. See *Review*.

ENTITY

A collection of attributes related to and describing a specific subject, such as Products.

ENTITY-RELATIONSHIP DIAGRAM

A diagram illustrating the relationship between various entities in a database.

EXECUTABLE

A binary data file that can be run by the operating system to perform a specific set of functions. In Windows, executables carry the extension .EXE and can be launched by double-clicking on them.

EXTERNAL INTERFACE

In database applications, an external interface is a defined process and data structure used to exchange data with other systems. For example, an order processing application may have an interface to exchange data with an external accounting system.

EXTERNAL INTERFACE COMPLEXITY

The level of complexity associated with an external interface. A simple interface is generally unidirectional, with limited, stable logic defining the structure of the exchanged data. A standard export from a database to a spreadsheet is considered a simple interface. A complex interface may be bi-directional, or may have extensive, adaptive logic defining the structure of the exchanged data. The transmission of labor data to a corporate payroll system, with its attendant validation and transaction confirmation requirements, is considered a complex interface.

FEASIBILITY STUDY

A process that determines whether the solution under analysis is achievable, given the organization's resources and constraints.

FIELD

Synonym for a data element that contains a specific attribute's value; a single item of information in a record or row.

FOCUS

The application object to which the user-generated input (usually keyboard and mouse) is directed.

FOREIGN KEY

A field or set of fields in a table whose value must match a primary key in another table

when joined with it. Also see *primary key* and *join*.

FORM

A screen formatted to facilitate data entry and review. Utilizes data entry fields, option selection tools, and control objects such as buttons and menu items.

FORMAL ITERATION PROCESS

In this process, draft deliverables are generated for formal review and comment. As participants review the draft deliverables, they are responsible for reporting errors found and concerns they may have to the PDR.

FUNCTIONAL AREA

Any formally organized group focused on the development, execution, and maintenance of business processes in support of a defined business function.

FUNCTIONAL ELEMENT

A definition that specifies the actions that a software component, product, or system must be able to perform.

FUNCTIONAL TESTING

Also known as end-user testing. Testing that focuses on the outputs generated in response to selected inputs and execution conditions.

FUNCTION POINT ANALYSIS

A software measurement process that focuses on the number of inputs, outputs, queries, tables, and external interfaces used in an application. Used for software estimation and assessment of developer productivity.

GROUP

During report generation, one or more records that are collected into a single category, usually for the purpose of totaling. Also used to identify a collection of database users with common access privileges. See *privileges*.

HAZARD

Any source or situation with potential to cause injury or harm to workers or the public, harm to the environment, incurred liability, or damage to or loss of property.

HARDWARE

Physical computer equipment and peripherals used to process, store, or transmit software applications or data.

HIERARCHICAL MENU

A menu with multiple levels, consisting of a main Menubar that leads to one or more levels of sub menus from which choices or actions are made.

HYPERTEXT MARKUP LANGUAGE (HTML)

A programming tool that uses HyperText to establish dynamic links to other documents

stored in the same or remote computers.

IMPLEMENTATION ELEMENTS

A specific software component created to fulfill a specific function defined in the functional and system design documents.

IMPLEMENTATION STAGE

A stage in the software development lifecycle during which a software product is created from the design specifications and testing is performed on the individual software units produced. See *unit testing*. Also referred to as the programming stage or development stage.

INCREMENTAL DEVELOPMENT

A software development technique where multiple small software development lifecycles are used to develop the overall software product in a modular fashion.

INDEX

A specialized data structure used to facilitate rapid access to individual database records or groups of records.

INFORMAL ITERATION PROCESS

One of the generic processes of the Shell Method SDLC. During this process, the project participants work together to gather additional information and refine stage inputs into draft deliverables. Activities of this stage may include interviews, meetings, the generation of prototypes, and electronic correspondence. All of these communications are deemed informal, and are not recorded as minutes, documents of record, controlled software, or official memoranda.

INFORMATION

Data that has been shaped into a form that is meaningful and useful to humans.

INFORMATION SYSTEM

Interrelated components working together to collect, process, store, and disseminate information to support decision-making, coordination, control, analysis, and/or visualization in an organization.

INHERITANCE

A feature of object-oriented programming where a specific class of objects receives the features of a more general class.

INITIAL DATA LOAD

When a new database application is first brought online, certain sets of data are preloaded to support operations. In some cases, a large amount of data is transferred from one or more legacy systems that the new database application is replacing. The initial data load figure is calculated as the sum of all records in operational and support data areas on day zero of the applications production lifecycle. This figure is used as a baseline for estimating development effort, server hardware requirements and network loads.

INITIAL RISK

The risk before controls are established and in place, or the risk posed by hazards that have not been adequately controlled; for activities with established control systems, the risk before Laboratory-instituted controls are implemented, or the risk posed because of the likelihood of failure of the established controls.

INNER QUERY

Synonym for *sub query*. See *sub query*.

INSERTION POINT

The position of the cursor within a block of text.

INSPECTION

Also termed *desk checking*. A quality assurance technique that relies on visual examination of developed products (usually source code or design documentation) to detect errors, violation of development standards, and other problems.

IN- STAGE ASSESSMENT PROCESS

One of the generic processes of the Shell Method SDLC. This is the formal quality assurance review process for each stage.

INSTALLATION AND ACCEPTANCE STAGE

A software lifecycle stage that consists of the testing, training, and conversion efforts necessary to place the developed software application into production. See *production*.

INTEGRATION AND TEST STAGE

A software development lifecycle stage where all the components of the final version of the software are brought together for testing.

INTEGRITY

The degree to which a software component or application prevents unauthorized access to, or modification of, programs or data.

INTEGER

A whole number, usually limited to a range of -32,000 to +32,000.

INTERFACE

A formal connection point defined between two independent applications for the purpose of data exchange. Also see *user interface*.

INTERFACE TESTING

A testing technique that evaluates whether software components pass data and control correctly to one another.

INTERSECTION

A group of data elements included in two or more tables as part of a Join operation. See *Join*.

ITERATIVE DEVELOPMENT MODEL

A model that develops a limited subset of an application during through a series of short iterations. Each iteration is composed of six development stages: 1) Planning, 2) Requirements Definition, 3) Design, 4) Development, 5) Integration & Test, and 6) Installation & Acceptance. The intent of the model is to develop the application as a series of layers, each building on the foundations laid by the previous iterations. Each iteration is scope-limited to allow completion of all six stages in a 6 to 8 week period. This is the model supported by the Shell Method™.

JOIN

A database operation or command that links the rows or records of two or more tables by one or more columns in each table.

JOINT APPLICATION DESIGN (JAD)

A design technique that brings users and IT professionals into a facilitated meeting for the purpose of interactively designing an application.

KEY FIELD

A field used to identify a record or group of records by its value. See also *primary key* and *foreign key*.

KEY PROCESS AREA

A software engineering process identified by the Software Engineering Institute Capability Maturity Model that is an essential to an organization's ability to develop consistently high-quality software products.

KICKOFF PROCESS

One of the generic processes of the Shell Method SDLC. The purpose of the kickoff process is to review the output of the previous stage, go over any additional inputs required by that particular stage, examine the anticipated activities and required outputs of the current stage, review the current project schedule, and review any open issues.

KILOBYTE (KB)

One thousand bytes (actually 1024 storage positions). Used as a measure of storage capacity.

KNOWLEDGE MANAGEMENT

The process of systematically managing and leveraging the stores of knowledge in an organization.

LIFECYCLE

A set of software development activities, or stages that function together to guide the development and maintenance of software products. Each stage is finite in scope, requires a specific set of inputs, and produces a specific set of deliverables.

LOGICAL DESIGN DOCUMENT (LDD)

A high-level description of the storage requirements of a software application. Storage requirements include a top-level description of the major database entities, initial user communities, and estimated metrics.

MAINTENANCE

The process of supporting production software to detect and correct faults, optimize performance, and ensure appropriate availability to end-users.

MASTER TABLE

A table containing data on which detail data in another table depends. Master tables have a primary key that's matched to a foreign key in a detail table and often have a one-to-many relationship with detail tables.

MEGABYTE (MB)

Approximately one million bytes. A unit of computer storage capacity.

MEGAHERTZ (MHZ)

A measure of the clock speed of the CPU in a computer. One megahertz equals one million cycles per second.

METADATA

Data that describes the structure, organization, and/or location of data. In essence, metadata is "data about data."

METHODOLOGY

A set of processes, procedures, and standards that defines an engineering approach to the development of a work product.

METRICS

Numeric data representing measurements of business processes or database activity.

MILESTONE

In project management, a scheduled event of significance for which an individual or team is accountable. Often used to measure progress.

MODEL

An abstract representation that illustrates the components or relationships of a specified application or component.

MODULE

A functional part of an application that is discrete and identifiable with a specific subject. Also known as a *Component*.

MODULE TESTING

The process of testing individual software modules or sets of related modules to verify the implementation of the software.

MULTIUSER

Concurrent access to a single database by more than one user, usually through the use

of client workstations.

NORMALIZATION

The process of creating small stable data structures from complex groups of data during the design of a relational database.

OBJECT CODE

Program instructions that have been translated into machine language so that they can be executed by the computer. See *executable*.

OFFICE AUTOMATION SYSTEM (OAS)

A combination of software applications such as word processing, electronic mail, and calendaring, that is designed to increase the productivity of data workers in the office.

ONLINE ANALYTICAL PROCESSING (OLAP)

A technology that operates on nonrelational, multidimensional databases often known as data cubes. Data cubes are often created via specialized processing from relational databases. The objective of OLAP is to allow the end user to perform highly flexible analysis and reporting.

ONLINE TRANSACTION PROCESSING (OLTP)

OLTP most commonly refers to large-scale database applications, such as order entry and payroll systems, which use transaction processing to assure data integrity. See *transaction*.

OPEN DATABASE CONNECTIVITY (ODBC)

A set of software drivers and database functions that allow different applications to access client/server RDBMSs, desktop database files, text files, and spreadsheets for the purpose of exchanging and manipulating database information.

OPERATING SYSTEM

System software that manages and controls the activities of the computer. The operating system acts as the interface between applications and the computer hardware.

OPERATIONAL DATA AREA

A component in a database application that supports the maintenance of data associated with a major operational process. For example, in an order entry system, the maintenance of customer data and the entry of orders would be considered operational data areas. See *support data area*.

OPERATIONAL TRANSACTION LOAD

The quantity of transactions per unit of time, for all users, in all operational data areas of a database application. This figure is commonly expressed as the number of transactions per day for all operational data areas, and is used to estimate server and network capacity requirements. See *operational data area, simultaneous users, transaction*.

ORGANIZATION

A formally defined structure that takes resources from the environment and processes

them to produce outputs.

ORPHAN

A child element in the chain of traceability that has no parent element associated with it.

OUTER JOIN

A SQL Join operation in which all rows of the joined tables are returned, whether or not a match is made between columns.

OUTER QUERY

A synonym for the primary query in a statement that includes a subquery. See *subquery*.

PARAMETER

A value passed to an application to direct performance. For example, a query could be set up with a parameter that limits the returned records to those falling after a specific date. Changing the value of the parameter changes the returned selection of records.

PEER REVIEW

See *Technical Review*.

PERMISSIONS

A synonym for privileges. See *privileges*.

PLANNING STAGE

The first stage of the Shell Method Software Development LifeCycle (SDLC). During this stage, the project plan is developed, along with the SQA plan, the SCM plan, and other documents as necessary to define the controls to be placed on the project. See *Project plan*.

PRIMARY DEVELOPER REPRESENTATIVE (PDR)

A person who acts as the primary point of contact and principal approver for the development community, and is the focal point for the development team. The PDR works directly with the PER to coordinate developer responses to requirements and issues generated by the end-user community. The PDR manages the entire lifecycle, either for a specific component or for the entire project, and is responsible for ensuring that appropriate technical experts conduct technical reviews in a timely manner. On large projects, multiple PDRs may exist, each dedicated to representing the development team for one or more components or subsystems. In this case, a lead PDR is assigned to coordinate activities with the lead PER.

PRIMARY END-USER REPRESENTATIVE (PER)

A person who acts as the primary point of contact and principal approver for the end-user community. In essence, the PER acts as the focal point for the end user community. The PER works directly with the Primary Developer Representative to identify subject matter experts, coordinate meetings, conduct reviews, and track identified issues, consolidating and coordinating the responses of all end-users into a cohesive set of communications. On large projects, multiple PERs may exist, each dedicated to representing the end user community for one or more components or subsystems. In this case, a lead PER is assigned to coordinate activities with the lead PDR.

PRIMARY KEY

A field or fields whose individual or combined values uniquely identify a record in a database.

PRIVILEGES

The authorities assigned to an end user by the database administrator or database owner to perform operations on data objects.

PROCEDURE

A written description or diagram of a course of action to be taken to perform a given task.

PRODUCTION

The time period after the new system is installed and any data conversion efforts are complete. The system is now being used for normal operations.

PRODUCTION INITIATION PLAN (PIP)

The PIP references the source code, describes coding standards and conventions as well as the executable generation process and/or instantiation of the application database engine and appropriately populated database tables. Sometimes referred to as the deployment plan.

PROGRAMMING STAGE

See *implementation stage*.

PROJECT

A concerted effort that is focused on developing or maintaining a specific software product or system. A project has a fixed scope, structure, and delivery schedule. See *Work Breakdown Structure*.

PROJECT MANAGER

The individual with total business responsibility for all activities of a project, including the structure of the activities, resource utilization, and schedule management.

PROJECT PLAN

A document that describes a technical and management approach to be followed for a project. The plan typically describes the scope of work, the methods to be used, the project structure and initial schedule, as well as a list of deliverables and other key events required for the project to be considered a success.

PROTOTYPING

The process of building an experimental system quickly and inexpensively for demonstration and evaluation so that end-users can better determine the requirements of an application.

PSEUDOCODE

A combination of programming language constructs and natural language used to define

an algorithm or business rule. Pseudocode is often used as a communications bridge between end-users and analysts or programmers.

QUALITY

Satisfaction of customer criteria; conformance to design specifications.

QUALITY ASSURANCE ASSESSMENT

The assessment of a deliverable for the presence of required internal and supporting elements. Generally performed by a Quality Assurance Reviewer who is not a member of the development team or end user base. See *Assessment*.

QUERY

A statement structured to direct the retrieval or manipulation of data in a database.

RAPID PROTOTYPING

Like *prototyping*, only uglier.

RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

An RDBMS is a database management application that can create, organize, and store data. The RDBMS treats data as if they were stored in two-dimensional tables. It can relate data stored in one table to data in another as long as the two tables share a common data element, or key. See *primary key, foreign key*.

RECORD

A group of related fields. A single row of a relational database table that contains each field defined for the table.

REFERENTIAL INTEGRITY

A set of rules governing the relationships between parent and child tables within a relational database that ensures data consistency.

REGRESSION TESTING

Structured retesting of a software component or application to verify that any modifications made have not caused unintended effects and that the software still complies with its specified requirements.

RELATIONAL DATA MODEL

See *RDBMS*.

RELIABILITY

The ability of a software application or component to perform its required functions under design-compliant conditions for a specified period of time.

RELEASE VERSION

A software application or component that has been tested and found to be in compliance with design documentation, and placed into production. See *production*.

REQUIREMENT

A condition or capability needed by the customer to solve a problem or achieve an objective. This condition or capability must be met or possessed by the developed software before it will be accepted by the customer.

REQUIREMENTS STAGE

A stage in the software lifecycle that immediately follows the planning stage. During this stage, the requirements for a software product are defined and documented. The output of this stage is a *Requirements Specification*.

REQUIREMENTS SPECIFICATION

A deliverable that specifies the manual and automated requirements for a software product in non-technical language that the customer and end-users can understand. The requirements specification focuses on what functions the application is to perform, not how those functions will be executed.

REQUIREMENTS TRACEABILITY

A process that acts as a bridge between the different stages of the software development lifecycle, and provides an auditable trail to verify that each requirement is fulfilled and tested. For example, the design elements in a design document must be traceable to "parent" requirements in the predecessor requirements document as well as to "child" elements in subsidiary documents, such as the test plan and implementation document. Traceability may be implemented as a table or spreadsheet describing the relationships between application requirements, functional elements, design elements, implementation elements, and test cases. In this case, the table is termed the Requirements Traceability Matrix (RTM). Traceability may also be implemented as an automated process that directly examines the related documents and reports valid traceability or errors in traceability.

RETIREMENT

Permanent removal of an application or software system from its operational environment.

REVERSE ENGINEERING

The process of examining an existing application that has characteristics that are similar to a desired application. Using the existing application as a guide, the requirements for the new application are defined, analyzed, and extracted all the way back to specifications. From this point, the specifications are altered to comply with any new customer requirements and the new application is developed.

REVIEW

The examination of a deliverable for specific content by a reviewer with expertise in the domain of the deliverable. The reviewer examines the content for correctness, consistency, completeness, accuracy, readability, and testability.

REUSABILITY

The degree to which a software application, component, or other work product can be used in more than one computer program or software system.

RISK

The possibility of suffering loss.

RISK MANAGEMENT

An approach to problem analysis that is used to identify, analyze, prioritize, and control risks.

RULE

A specification that determines the data type and data value that can be entered in a column of a table. Rules are classified as validation rules and business rules. See *business rule*.

SECURITY

Policies, procedures, and technical measures used to prevent unauthorized access, alteration, deft, or destruction of information systems or data.

SELF-JOIN

A SQL Join operation used to compare values within the columns of one table. Self-joins join a table with itself, requiring that the table be assigned two different names, one of which must be an alias.

SIMULTANEOUS USERS

A quantity of users and/or external systems connected to a multiuser database application for the purpose of exchanging and/or maintaining data.

SOFTWARE

Computer programs, procedures, and associated documentation pertaining to the operation of an application. The detailed instructions that control the operation of a computer system.

SOFTWARE CONFIGURATION MANAGEMENT

See *configuration management*.

SOFTWARE DESIGN DOCUMENT (SDD)

A document that defines the storage mechanisms, interfaces, internal processes, and outputs of a software application.

SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

A set of software development activities or stages that function together to guide the development and maintenance of a software product. The generally accepted stages for software development are Planning, Requirements, Design, Development, Integration and Test, Installation and Acceptance, and Maintenance. Each stage is finite in scope, requires a specific set of inputs, and produces a specific set of deliverables. The document that describes the SDLC may be called the SDLC Description or the Verification and Validation Plan (VVP).

SOFTWARE METRICS

Objective assessments (in the form of quantified measurements) of a software

application or component.

SOFTWARE PROJECT MANAGEMENT PLAN

The rarest artifact in the software development universe. See *project plan*.

SOFTWARE QUALITY ASSURANCE (SQA)

A process designed to provide management with appropriate visibility into the software engineering processes being used by the project team. A formal process for evaluating the work products produced during the software development lifecycle.

SOFTWARE REQUIREMENTS DOCUMENT (SRD)

Also referred to as a Requirements Document (RD), or Software Requirements Specification (SRS). The SRS is a document (or set of related documents) that specifies the desired functionality for a software product through a series of requirements statements. The requirements statements are written in non-technical language that the customer and end-users can understand. The requirements specification focuses on what functions the application is to perform, not how those functions will be implemented.

SOURCE CODE

Software programming instructions written in a language readable by humans that must be translated into machine language before it can be executed by the computer.

SPECIFICATION

Documentation that describes software requirements, designs, or other characteristics in a complete, precise, and verifiable manner.

SPIRAL DEVELOPMENT MODEL

A development model that specifies the development and refinement of a series of successively more complex prototypes, followed by a traditional waterfall development model to place the application into production. See *Waterfall development model*.

STAGE

A partition of the software development cycle that represents a meaningful and measurable set of related tasks which are performed to obtain specific work products, or deliverables. See *deliverable*.

STAGE EXIT PROCESS

One of the generic processes of the Shell Method SDLC. The stage exit is the vehicle for securing the concurrence of principal project participants to continue with the project and move forward into the next stage of development. The purpose of a stage exit is to allow all personnel involved with the project to review the current stage deliverables, provide a forum to raise issues and concerns, and to ensure an acceptable action plan exists for all open issues.

STAKEHOLDERS

Those individuals with decision-making authority over a project or group of projects.

STANDARDS

Approved reference models and protocols as determined by standard-setting groups to prescribe a disciplined, uniform approach to software development and maintenance.

STANDARD OPERATING PROCEDURES (SOPS)

Precise, defined rules, activities, and practices developed by an organization to consistently manage normal operations.

STRUCTURED ANALYSIS

A top-down method for defining system inputs, processes, and outputs to build models of software products or systems. The four basic features in structured analysis are data flow diagrams, data dictionaries, procedure logic descriptions, and data store descriptions.

STRUCTURED QUERY LANGUAGE (SQL)

A standard data definition and data manipulation language for relational database management systems.

SUBJECT MATTER EXPERT (SME)

A person, generally a customer staff member, who is considered to be an expert in one or more operational processes that are the focus of an automation effort. SMEs are generally the primary sources of application requirements, and play very significant roles in the requirements, design, and testing stages of the software development lifecycle.

SUBQUERY

Any SQL Select statement that's included (nested) within another Select, Insert, Update, or Delete statement or nested within another subquery.

SUPPORT ACTIVITIES

Activities that make the delivery of the primary services or products of a firm possible. These activities typically focus on support of the organization's infrastructure, human resources, technologies, and logistics.

SUPPORT DATA AREA

A component in a database application that supports the maintenance of data used primarily for reference and support of operational data areas. For example, in an order entry system, the maintenance of lists of customer business types or order shipment methods would be considered support data areas. See *operational data area*.

SYSTEM

A collection of hardware, software, firmware, and documentation components organized to accomplish a specific function or set of related functions.

SYSTEM DESIGN STAGE

A stage in the software development lifecycle during which the requirements for the software product architecture, components, interfaces, and data structures are refined and expanded to the extent that the design is sufficiently complete to be implemented. Also known as the *Design stage*.

SYSTEM OWNER

The organizational unit or person that provides funding and has approval authority for the project. Also referred to as the customer or client. Typically, system owners are also system users.

SYSTEM SOFTWARE

Specialized programs that manage the resources of the computer, such as the central processor, communication links, and peripheral devices. See *operating system*.

SYSTEM TESTING

Testing of the application or information system as a whole to determine if discrete components will function together as planned and to evaluate compliance with specified requirements.

SYSTEMS ANALYSIS

The analysis of a problem that the organization will try to solve with an information system.

SYSTEMS ANALYSTS

Specialists who translate business problems and requirements into information systems requirements, acting as a bridge between the information systems department, the system owner, and end-users.

TABLE

A database object consisting of a group of rows (records) divided into columns (fields) that contain data or Null values. A table is treated as a database device or object.

TASK

The smallest accountable unit of work. A task is the lowest level of work division typically included in the Project Plan and Work Breakdown Structure. Related tasks are usually grouped to form activities. See *Work Breakdown Structure*.

TECHNICAL FEASIBILITY

Determines whether a proposed application can be implemented with the available hardware, software, and technical resources.

TECHNICAL REVIEW

The review of a deliverable for techical accuracy by a qualified developer who is familiar with the software product under design or development. See *Review*.

TECHNOLOGY

Software and hardware tools and services used to develop and support a database application. In general, a development team working with well-known, mature tools is substantially more likely to succeed than a team working with newly released, unfamiliar technology. Caveat: in rare cases, unique performance requirements dictate the use of new technology for the project to have any chance of success.

TESTBED

A specific set of hardware, software, instrumentation, simulators, and other support elements needed to conduct a test of a database application.

TEST CASE

A defined set of database records, test inputs, execution conditions and anticipated results designed to exercise specific application components and verify compliance with design criteria and requirements. Contains detailed instructions for the set up, execution, and evaluation of the results for the test case.

TESTING STAGE

Also often referred to as the test and acceptance stage. A stage in the software development lifecycle where the components of a software product are executed under specified conditions, the results are observed and recorded, and an evaluation is made to determine whether or not the requirements have been met.

TEST ITEM

A software component that is the object of a test case.

TEST PLAN

A document that defines the preparations, test items, test data, and test cases for the series of tests to be performed on an application.

TEST RESULTS DOCUMENT

A document that contains a chronological record of the execution and results of the testing carried out for a software component or application.

THREE-TIER

The architecture of a database application consisting of a front end, application server, and database server. The front end handles user input and output, the application server serves front end components and handles the business rules, while the database server manages the associated data storage as directed by the application server or front end client.

TIMESTAMP

A set of date and time data attributes applied to a disk file or database record when it is created or edited.

TRACEABILITY

The degree to which a relationship can be established between two or more products of the software development lifecycle. See *requirements traceability*.

TRANSACTION

A set of processing tasks that are treated as a single activity to perform a desired result. For example, a transaction would entail all the steps necessary to insert and or modify values in multiple tables when a new invoice is created. If any of the record management tasks fail, the entire activity is canceled, or rolled back. If all of the record management activities are successful, the transaction is committed, or made permanent.

TRANSACTION ANALYSIS

A process used to divide complex data flow diagrams into smaller, individual data flow diagrams for each class of transaction that the application will process.

UNIT

See *implementation element.*

UNIT TESTING

The isolated testing of each logical path of a specific implementation element or groups of related elements. The expected output from the execution of the logical path is pre-defined to allow comparisons of the planned output against the actual output.

USABILITY

The ease with which a user can learn to operate a software application.

USE CASE

A description of a business process under automation, focused on how Actors (user and interfacing systems) interact with the process. Includes descriptions of the information the Actors send to the system, the data the Actors receive from the process, and the operations they perform using the system. See *Actor.*

USER

Those individuals who use a specific software product or system. User activities can include data entry, queries and updates, the execution of batch operations, and the generation of reports.

USER INTERFACE

The part of the application through which the end-user interacts with the system.

USER MANUAL

A document that describes a software application in sufficient detail to enable an end-user to obtain desired results. Typically includes a tutorial, a description of functions and data structures, options, allowable inputs, expected outputs, and possible error messages.

VALIDATION

1. The process of determining whether a value in a table's data cell its within the allowable range or is a member of a set of acceptable values.
2. The process of evaluating software to ensure compliance with established requirements and design criteria.

VERIFICATION

The process of evaluating an application to determine whether or not the work products of a stage of a software development lifecycle fulfill the requirements established during the previous stage. See *requirements traceability.*

VERIFICATION AND VALIDATION PLAN (VVP)

A software development plan that describes the specific processes to be used to verify

traceability and validate correct functionality of the software during the SDLC. Also referred to as the *Software Development LifeCycle (SDLC)*.

VIRTUAL ORGANIZATION

An organization that uses networks to link people, assets, and ideas to create and distribute products and services without being limited to traditional organizational boundaries or physical location.

WALK-THROUGH

The review of a functional element, design element, or implementation element by a team of subject matter experts to detect possible errors, violation of development standards, and other problems.

WATERFALL DEVELOPMENT MODEL

A software development lifecycle in which each stage is dependent upon the outputs of the previous stage. The vision is that of water flowing downhill. Once a stage is completed, the next stage begins. Previously completed stages cannot be re-initiated.

WIDOW

A parent element in the chain of traceability that has no child elements associated with it. See *Requirements traceability*.

WORK BREAKDOWN STRUCTURE (WBS)

A listing of all activities and tasks related to those activities that make up a complete project. Generally described in outline form to show the hierarchical relationship between activities and tasks.

WORK PRODUCT

A specific document or software component that results from a project activity or task. See *deliverable*.