

## SECTION 8 - GLOSSARY

**Actor:** In object-oriented development, this defines a coherent set of roles that external entities such as humans, machines or other systems can play while interacting with the system being developed.

**Actors List:** A list that identifies all the actors pertaining to the system being modeled.

**Application:** Automated part of a business system not including the platform or cross-application technical architecture.

**Architecture:** How a system or program is structured; how its parts interact as well as what protocols are used for communication of one component with another.

**Availability:** The portion of time that a system which is scheduled to operate actually can be used as expected.

**Bug:** A bug is a program defect that is encountered in operation when the program malfunctions, either under test or in use.

**Business process:** A set of business activities oriented toward a given outcome always has at least one customer

**Business Requirement:** "What" the system needs to do in order for the project sponsor to carry out their business. There should be no reference to how it will be done.

**Business Transaction:** A unit of work of the business process.

**Change request:** A formal request to modify a system.

**Class model:** Also known as business object model, represents a structural view of the real-world business concepts for the system under development.

**Code Review:** A form of inspection in which a programmer shows his or her code to peers.

**Component Repository:** a database or catalog that contains meta-data on shared services. Also referred to as Repository of Shared Services.

**Critical Path:** The succession of activities in a project that determines the shortest possible execution time.

**Data Flow:** Data required or produced by a process.

**Data Model:** Representation of the structure of the entity-type, the relationship-type and attribute-type of a functional domain. Usually consists of an entity-relationship diagram and descriptions of entity-types, relationship-types and attribute types.

**Database:** A set of related data or files that support an application.

**Defect:** Improper program conditions that are generally the result of an error. Not all errors produce program defects, as with incorrect comments or some documentation errors.

**Entity-type:** An object of interest about which an organization wants to store information.

**Errors:** These are human mistakes, and their quantification often depends on an understanding of the programmer's intentions. In the case of typographical or syntactic errors their causes are generally clear, but the nature and cause of design errors is much harder to establish precisely, particularly after the fact.

Event: An occurrence with no duration of interest to the system.

Failures. A failure is a malfunction of a user's installation. It may result from a bug, incorrect installation, a communication line hit, a hardware failure, and so forth resulting from system events.

GUI: Graphical User Interface, an icon-driven interface.

Integration Test: A test that ensures that the components and modules of a system communicate with one another in accordance with the technical design.

Interaction Model: Represents how objects or instances of classes defined in the class model interact in order to carry out specific scenarios described by the use case model.

IRMC: The Information Resource Management Commission. A body created by the NC General Assembly to oversee the management of the State's technical infrastructure and use of technology.

Issue: A problem to be solved or a decision that hasn't been made.

Module: A unit of program development that is independently compiled.

NCSB: North Carolina Service Broker, see "service broker".

OPA: organizational process asset – an artifact that defines some portion of the organization's software project environment.

Open point: An area where information is needed but not yet available.

Performance: A quality attribute that measures a systems ability to provide expected benefits and process all its events within a specified timeframe.

Project Controls: Activities to ensure that a project remains within its scope and comes in on-time and budget.

Prototype: A model of the design of a future application to demonstrate some of its properties.

Quality Assurance: A program of periodic project reviews to assure that the project will be on time, within budget, and provide the expected functionality.

Relationship-type: A property associating two or more entity-types.

Reliability: A quality measuring a system's ability to produce the correct response.

Roll-out: The activity required to make a fully-tested system operational.

Service Broker: the statewide standard middleware infrastructure that enables communication between applications across disparate platforms.

SDT: DENR's Systems Development Team under the ITS umbrella. It is headed by the SDT Manager/Chief.

Shared Service: an executable application logic or information that is accessible (reusable) to applications across the State or department regardless of the language it is written in or the platform on which it resides. Shared Services are consumed and exposed via the NC Service Broker API.

SLC: software life cycle – the project-specific sequence of activities that is created by mapping the Activities of the IEEE Standard 1074-1997 onto a selected software life cycle model (SLCM).

SLCM: software life cycle model – the framework, selected by each using organization, on which to map the activities of IEEE Standard 1074-1997 to produce the software life cycle (SLC).

SLCP: software life cycle process – the project-specific description of the process that is based on a project's software life cycle (SLC) and the Organizational Process Assets (OPA).

Sponsor: Primary beneficiary of a system project to whom the project team looks for direction and issue resolution.

STA: Statewide Technical Architecture – the document that describes the state's vision for adaptive, service-oriented applications. It consists of principles, best practices and standards that apply to technology components.

Standard: A mandated practice or format.

Test Data: Data input to a program during a test.

Unit Test: Testing of programming work unit to derive error-free code.

Use Case: In object-oriented development, this is a sequence of steps or transactions performed by an actor in dialog with the system.

Use Case model: Depicts the functionality of a system from the perspective of entities interacting with the system.

User Test: A test of the functional requirements of the complete system under operational conditions.

Walkthrough: A technique for verifying a design where the builder of the design explains how it is intended to work to his or her peers.

Work Plan: A list of activities to be undertaken in a project along with an estimate of required resources and a time schedule.

Workflow: A sequential mapping of key tasks or procedures involved in a particular process which is developed using a commonly accepted set of flowcharting symbols.

End of Section 8