

CS605 - Software Engineering II Glossary By **www.virtualians.pk**

Abstraction : Abstraction - (1) the level of technical detail of some representation of software; (2) a cohesive model of data or an algorithmic procedure

Adaptive maintenance : Adaptive maintenance - activity associate with changing an application to make it conform to changes in its external environment

Agile development : Agile development (also referred to as agile process model) - an adapted version of software engineering that emphasizes customer communication, incremental software delivery, informal methods and work products, and highly motivated teams.

Analysis : Analysis - a set of activities that attempt to understand and model customer needs and constraints

Architectural design : Architectural design - an activity that attempts to layout the module "floor plan" for the software

Architecture : Architecture - the overall structure of software components, the data and/or content that components manipulate, and the relationships between them

Automatic code generation : Automatic code generation - tools that generate source code from a representation of software that is not source code

Baseline : Baseline - a point at which some deliverable produced during the software engineering process is put under formal change control

Behavioral modeling : Behavioral modeling - representing the mode of behavior (called states) of an application and the events that cause transitions from state to state

Beta testing : Beta testing - testing that is conducted by the user

Black box testing : Black box testing - testing that does not focus on the internal details of the program but uses external requirements

Business risks : Business risks - the set of potential business problems or occurrences that may cause the project to fail

CASE : CASE - Computer-aided software engineering,

Change control : Change control - an umbrella process that enables a project team to accept, evaluate, and act on changes in a systematic manner

Change control authority (CCA) : Change control authority (CCA) - the person(s) who have responsibility for deciding whether a change is to be made

Change management : Change management - a set of software engineering actions that helps ensure that changes are properly identified, controlled, and reported

Classes : Classes - a basic construct in object-oriented methods that categorizes elements of the problem

Cohesion : Cohesion - an informal measure of the degree to which a software component implements a single, focused function

Complexity : Complexity - a quantitative measure of a program's complexity

Configuration : Configuration - the collection of programs, documents and data that must be controlled when changes are to be made

Configuration audit : Configuration audit - an activity performed by an SQA group with the intent of ensuring that the change control process is working

Configuration control : Configuration control - the control of changes to programs, documents or data

Coupling : Coupling - an informal measure of the degree to which a software component is connected to other components, to data, and to the external environment

Cyclomatic complexity : Cyclomatic complexity - a measure of the logical complexity of an algorithm, used in white-box testing

Data design : Data design - an activity that translates the data model developed during analysis into implementable data structures

Data flow diagram (DFD) : Data flow diagram (DFD) - a modeling notation that represents a functional decomposition of a system

Data modeling : Data modeling - an analysis method that models data objects and their relationships

Defect amplification : Defect amplification - when a defect is introduced early in the software process and remains undetected, it often is amplified into multiple defects later in the software process

Defect removal efficiency (DRE) : Defect removal efficiency (DRE) - a nondimension ratio (between 0 and 1) that provides an indication of the degree to which errors are removed from software before it is released to end-users

Design specification : Design specification - a document that describes the design

Domain analysis : Domain analysis - an object-oriented software engineering activity that attempts to identify classes that are relevant to an entire application domain, rather than a specific application

Effort : Effort - the work-time product (e.g., person-days) associated with a project

Enhancement : Enhancement - an extension of functional or performance requirements

Errors : Errors - a lack of conformance found before software is delivered to the customer

Estimation : Estimation - a project planning activity that attempts to project effort and cost for a software project

Extreme programming : Extreme programming - an agile process model that emphasizes scenario-based planning, incremental delivery, refactoring, pair programming and continuous testing.

Factoring : Factoring - a technique that distributes control and work in a top-down manner within a software architecture (used a part of structured analysis)

Formal methods : Formal methods - a software engineering approach in which specification and design are described using mathematically-based formal notation

Formal technical reviews : Formal technical reviews - a structured meeting conducted by software engineering with the intent of uncovering errors in some deliverable or work product

Function points : Function points - a measure of the utility delivered by an application

Functional decomposition : Functional decomposition - a technique used during planning, analysis and design; creates a functional hierarchy for the software

GQM (Goal, Question, Metric) paradigm : GQM (Goal, Question, Metric) paradigm - a technique for defining meaningful metrics for any part of the software process

Grammatical parse : Grammatical parse - a technique that is used during analysis and intended to help isolate basic data objects and function

High-order tests : High-order tests - black-box tests conducted once the software has been integrated

Integration testing : Integration testing - a testing step that constructs the software while testing it

Interface design : Interface design - a software engineering action that establishes the structure and workflow for a user interface; follows three "golden rules:" place the user in control, reduce the user's memory load, make the interface consistent

Interoperability : Interoperability - the degree to which one application communicates or interfaces with another

ISO 9001:2000 : ISO 9001:2000 - a quality assurance standard that applies to software engineering

JAD : Joint application development (JAD) - a specific FAST technique

Levels of abstraction : Levels of abstraction - the degree of detail with which some representation of the software is presented

Line-of-code metrics : Line-of-code metrics - measures of quality or productivity that are normalized using lines of code produced

Load testing : Load testing - a testing task that determines how software (often a WebApp) will respond to various loading conditions

Loop testing : Loop testing - a white box testing technique that exercises program loops

Maintenance : Maintenance - the activities associated with changes to software after it has been delivered to end-users

Measurement : Measurement - collecting quantitative data about the software or the software engineering process

Milestones : Milestones - a point in time that is used to indicate progress during a project

Modularity : Modularity - an attribute of a design that leads to the creation of high quality program components

Navigation analysis : Navigation analysis - a Web engineering action that establishes how a user will navigate between various elements (e.g., content, functions) of a WebApp

Object-oriented analysis : Object-oriented analysis (OOA) - a technique for defined classes of objects, their relationships and basic structure

Object-oriented design : Object-oriented design (OOD) - a technique for translating the OOA model into an implementation model

Outsourcing : Outsourcing - contracting software work to a third party

Paper prototype : Paper prototype - a paper representation of an application (e.g., story boards that describe the interaction at a human interface)

Patterns : Patterns - a stylized description or characterization of a software problem or capability and/or the manner in which a solution to the problem or capability may be characterized, applied, and implemented

Portability : Portability - the ability to transport software from one target environment to another

Process framework : Process framework - a relatively small set of fundamental software engineering activities that define a software process

Project risks : Project risks - the set of potential project problems or occurrences that may cause the project to fail

Project scope : Project scope - a statement of basic requirements of the software to be built

Project tracking : Project tracking - the activity that enables a manager to understand the status of a project

Quality : Quality - the degree to which a product conforms to both explicit and implicit requirements

Quality management : Quality management - a set of software engineering actions that helps ensure that software is built in a way that achieves high quality

Re-engineering : Re-engineering - a series of activities that transform legacy systems (with poor maintainability) into software that exhibits high quality

Refactoring : Refactoring - changing software in a way that improves its internal structure but does not change its external behavior; often conducted iteratively as design evolves into code.

Reliability : Reliability - a measure of the degree to which software operates reliably over some period of time

Requirements analysis : Requirements analysis - a modeling activity whose objective is to understand what the customer really wants

Requirements engineering : Requirements engineering - the activities required to elicit, elaborate, negotiate, specify, and validate system or software requirements

Reusability : Reusability - the ability to reuse an already-existing program component in another application

Reverse engineering : Reverse engineering - trying to develop design models or an understanding of design using program code as a starting point

Risk : Risk - a potential problem or occurrence that puts a project in jeopardy

Risk analysis : Risk analysis - a technique for identifying and evaluating risks

Risk Management and Monitoring Plan (RMMP) : Risk Management and Monitoring Plan (RMMP) - a plan for mitigating, monitoring and managing risks

Scheduling : Scheduling - the activity that lays out a timeline for work to be conducted on a project

Security : Security - the ability of software to operate in a manner that is secure from internal or external attack

Security testing : Security testing - testing tasks that probe the vulnerability of both client-side and server-side software

Smoke testing : Smoke testing - an integration testing approach that constructs and tests software on a daily basis

Software engineering : Software engineering - a discipline that encompasses the process associated with software development, the methods used to analyze, design and test computer software, the management techniques associated with the control and monitoring of software projects and the tools used to support process, methods, and techniques

Software process improvement : Software process improvement (SPI) - a set of software engineering activities that attempt to improve the state of software engineering practice within an organization

Software quality assurance : Software quality assurance (SQA) - a series of activities that assist an organization in producing high quality software

Software Requirements Specification : Software Requirements Specification - a deliverable that describes all data, functional and behavioral requirements, all constraints, and all validation requirements for software

System engineering : System engineering - focuses on the analysis and design of all elements of a complete product, service, or technology for the transformation of information or control

Task analysis : Task analysis - a software engineering action that is conducted as part of user interface design; intended to better understand how a user is to interact with a system

Task set : Task set - a collection of software engineering tasks that are required to complete an activity or action that is part of a software process framework

Technical risks : Technical risks - the set of potential technical problems or occurrences that may cause the project to fail

Time-boxing : Time-boxing - a project scheduling and control technique that establishes time boundaries for the completion of a specific project task

Tools : Tools - application software used to perform software engineering tasks (e.g., design tools, testing tools);

Total quality management : Total quality management - a company commitment to develop a process that achieves high quality product and customer satisfaction

UML : UML (Unified Modeling Language) - a comprehensive diagrammatic notation for the analysis and design of software

Unit testing : Unit testing - part of the testing strategy that focuses on tests to individual program components

Usability : Usability - An informal measure of the ease with which a user interface can be learned and applied with efficiency and without errors

Use-case : Use-case - a written description that defines a very specific interaction between a user and a system, often (but not always) written in the form of a usage scenario

User hierarchy : User hierarchy - a hierarchical representation of the categories of users that will interact with any type of software

Validation : Validation - tests to ensure that the software conforms to its requirements

Web engineering : Web engineering - the application of software engineering principles, concepts, and methods (or adaptations of them) to the development of Web-based applications or systems.

White box testing : White box testing - a test case design technique that makes use of a knowledge of the internal program logic

Work breakdown structure : Work breakdown structure (WBS) - the set of work tasks required to build the software; defined as part of the process model

Work product : Work product - any cohesive and persistent information that is produced as a consequence of one or more software engineering actions of tasks