Software engineer – An engineer who makes high-quality software.

Software engineering- Uses techniques, methodologies, and tools to create high-quality software.

Make high quality, within budget and within time. Specify -> Design - > Program -> Validate

Software Proposal – Document that a developer submits to business for acceptance.

Diagnose problem -> Prescribe treatment -> describe plan of work

Software Process – Specification - > development -> Validation -> Evolution

Development: Defining the organization of the system and implementing the system

Waterfall, Spiral, Incremental, Agile, XP,(Types of Reusable : API, Libraries:framework, Software system)

Reduce rework cost with : Change avoidance and Change tolerance

Good Requirements are : Correct, Not ambiguous, complete, consistent, and verifiable

Requirement process: Elicitation -> Analysis -> Specification -> Validate -> SRS(Document)

Validate- Check requirements to see if complete, feasibility, ambiguity, and verifiable

Stakeholder – customers, end users, legal staff, may include developers of organization

Get requirement through interviews, panel or forum of stakeholders, observe/immerse in environment

Scenario – Description of one or more specific interactions with user and system(high level)

Use case- Description of a single general interaction of an actor with system(Eliciatation)

Specify – Natural language is using English for requirements and Structured is using a template(Specfi)

XP practices: small release, simple, refactoring, testing, pair programming, onsite customer, planning game, metaphor, collective ownership, continuous integration, 40-hour week, coding standard

CRC : Class Responsibility Collaboration (It’s a class in your system)

Object-oriented Analysis – Scenario based elements (usecasediagram), class based(class diagrams, CRC), Behavioral (Sequence/state diagram). Structured anaylsis – flow oriented (Data flow diagram, controlflo)

UML (Unified Model Language) – standard for documenting OO models. Class, Usecase, and sequence/state. (UML (activity diagrams)represents scenario based elements)

Functional decomposition – top down (system level -> sub system -> modules)(Detail Design)

Relational Database – ER Diagram, System Diagram- MVC(Detail Design)

Architectural pattern – pipes and filter, MVC, client-server, three tier architecture, layered

Coupling - # of dependencies between software units(loose), (strong)Cohesion- # of dependencies between within subsystem/ Types of UI Design – Command-Line, Textmenues, GUI, 2 issues Flow of interaction and look and feel

McCabe Complexity – program quality is directly related to the complexity of the control flow branching

Cyclomatic complexity = Edges – Nodes + 2 \*number of connected components/ # is equal to max num of linearly independent paths thru flow diag. used to determine test cases needed per path

Service oriented architecture Web services – HTTP for transport, SOAP structured data exchange, UDDI for discovery, WSDL for description, XML used everywhere. UDDI is where to find web service, WSDL teaches how to access web service, SOAP is a protocol for accessing web services.

Verification – the software should conform to its specifi (the func and non func requirements)

Validation – The software should do what customer really requires

Static Verification : inspection and review diagrams/models/source code/test plans

Dyanmic Verification: is actually tested with simulated test data.

Failure : Deviation between the specification and the actual behavior of the system./Fault: a design or code that causes abnormal behavior. (bug or defect)/ Test case set of inputs and expected results that exercises a system with the purpose of detecting faults

Blackbox- focuses on input/output behavior of software. Not based on how software is structured./White box testing is focus on internal structure of software. Test all parts of code in softW./Test driver code that simulates the part of the system that calls the component under test/Test stub is code that simulates a component that is called by the tested component/ System test > Component test > unit test

Test case developed by Partition testing – identify different groups of inputs that have common characteristics and should be processed the same way by the system, use one test case per group./Boundary value analysis – test the boundaries of the groups used in partition testing/ Path testing – exercise all possible paths through the code at least once/ Guideline- use guidelines that reflect the kinds of errors programmers often make.

Security assures CIA : Confidential, Integrity, and Availbility. SDL : Secure Development Lifecycle

SDL : Requirements -> Design -> Implementation -> Verification -> Deployment -> Maintenance

Security Vulnerability : Buffer Overflow can long input affect service/ Script injection can input with scripts execute/ numeric overflow can a large number become a negative or small number/ race condition can multiple threads cause errors/ configuration issues can software be installed improperly, causing abuse/ Programming backdoor have programmers left hooks providing entry or information/ SQL injections

Re-engineering : Restructure or rewrite part of all of a software system without changing its functionality. Easier to maintain if restructured and re documented. Techinques of Re-engineering : Regression testing – to ensure modifications don’t change functionality/ Source code translation : need new language?/ Reverse engineering: analyze code (creates documentation)/ program restructuring and data structure clean up.

Refactor – changing a software system internal structure. Without changing external behavior. Improve readability, structure, reduce complexity, easeier to modify in future. No added functions. Preventative maintenance: Reduces future maintenance cost.