

# IT2251 SOFTWARE ENGINEERING AND QUALITY ASSURANCE

2MARKS

## Unit-1

1. Define software engineering by IEEE?

Software engineering is :( 1) The application of a systematic, disciplined ,quantifiable approach to the development, operation and maintenance of s/w i.e. application of engineering to s/w. (2) The study of approaches as in ( 1).

2. List out the tasks focused on development phase?

- Software design
- Code generation
- Software testing

3. Specify the stages in Software Engineering as a problem solving loop?

- Status Quo
- Problem definition
- Technical development
- Solution integration

4. What is WIN-WIN Spiral Model?

Customer wins by getting the system product that satisfies majority of customer needs and developer wins by working to realistic and achievable budgets and deadlines.

5. What are the advantages of 4G techniques?

- Reduction in software development time
- improved productivity
- Provides good solution

6. What are FAST?

Facilitated Application Specification Techniques creates a joint team of customers and developers who work together to identify the problem, propose elements of solution and negotiate different approaches.

- Delay estimates till late in the project
- Decomposition techniques
- Use empirical models

8. How will you compute the 3-point or expected value of size?

$$S = (S_{opt} + 4S_{in} + S_{pess}) / 6$$

Where  $S_{opt}$  = Optimistic value

$S_{pess}$  = Pessimistic value

Sin= Most likely value

9. What are the information domain characteristics on FP estimates?

Number of inputs

Number of outputs

Datafiles

Inquiries

External interfaces

Complexity adjustment factors

10. What is estimated effort in COCOMO?

Estimated effort =  $NOP/PROD$

NOP ? New Object Point

PROD? Productivity rate

11. What are the task regions in a Spiral Model?

Customer communication

Planning

Engineering

Risk analysis

Construction & release

Customer evaluation

12. List out the tools in 4G.

Nonprocedural language for dBquery

Report generation

Data manipulation

Code generation

Spread sheet capability

High level graphics

Screen interaction & definition

Automated generation of HTML

13. Draw the resource pyramid for software development

People

Reusable S/W

components

H/W & S/W tools

14. Write the software equation for effort calculation.

$$E = [LOC \times B^{0.333} / P]^3 \times (1/t^4)$$

t ? project duration in months/years

B ? Special skill factors

P ? Productivity parameters

15. What are the generic functions in automated estimation tools?

Sizing project deliverables

Selecting project activities

Predicting staffing levels

Predicting software effort

Predicting software cost

Predicting software schedules

16. What are umbrella activities?

Software project tracking and control

Formal reviews

SQA

SCM

Document preparation & production

Reusability management

Measurement

Risk management

17. Draw the waterfall model

Analysis

Design

Code

Test

Support

18. What are object points?

The number of Screens

Reports - Components to build the application

19. What is Empirical Estimation Model & give the general structure of it?

A Typical Estimation Model is derived using Regression Analysis on data collected from past software projects.

Structure is  $E = A + B \times (ev)^C$

A, B, C? constants

ev ? Estimation variable (LOC/FP)

20. What is a process?

Process defines a framework for a set of key process areas that must be established for effective delivery of s/w engineering technology.

21. What are the changes encountered in support (maintenance) phase?

Correction.

Adaptation.

Enhancement.

Prevention.

22. What is a process model or a software engineering paradigm?

It is a development strategy that includes process, methods and tools and three generic phases (definition, development, support). A process model is chosen based on the nature of application, the methods, tools used and controls and deliverables that are required.

23. What is Chaos model?

It describes software development as a continuum from the user to developer to the technology.

24. When is prototyping paradigm suitable?

When the customer defines only objectives, but doesn't specify detailed i/p, processing or o/p. when developer is unsure of the efficiency of an algorithm or the adaptability of OS etc.

25. What is RAD model?

The RAD model is a "high-speed" adaptation of the linear sequential model in which rapid development is achieved by using component-based construction.

26. What are the evolutionary s/w process models?

The Incremental Model.

The Spiral model.

27. What is a software scope?

Software scope describes data and control to be processed, function, performance, constraints, interfaces and reliability. Cost and schedule estimates are functionally oriented. Performances encompass processing and response time requirements. Constraints specify limits placed on s/w by external h/w, available memory etc.

28. What are meta-questions?

Are you the right person to answer these questions?

Are answers official?

Am I asking too many questions?

Are my questions relevant to the problem?

Can anyone else provide additional information?

Should I be asking you anything else?

29. What are called components?

Components are software building blocks that are created and reused. They must be catalogued for easy reference, standardized for easy application and validated for easy integration.

30. What is called as SEE(S/w Engineering Environment)?

SEE incorporates hardware and software. H/w provides a platform that supports the tools (s/w) required to produce work products that are an outcome of good software engineering practice.

31. List out the reasons for divergent estimates in a S/w project.

The scope of the project is not adequately understood.

Productivity data used in problem based estimation techniques is inappropriate for the application or obsolete.

32. What is (a) Boehm simple model (b) Kemerer model?

(a) Boehm model is:  $E = 3.2 * (KLOC)^{1.05}$

E is effort in person-months, KLOC is Kilo Lines OF Code.

(b) Kemerer model is  $E = 60.62 * 7.728 * 10^{-8} FP^3$

FP is Function Points.

33. What constitutes productivity parameter in s/w equation?

Overall process maturity & management practices.

Level of programming languages used.

Extent to which good s/w engg practices are used.

State of software environment.

Skills and experience of the s/w team.

Complexity of application.

1. An estimate of size(in KLOC).

2. An indication of project duration in calendar months or years.

35. Calculate the minimum development time and effort for a CAD s/w with has

33200 lines of code and with a productivity parameter of 12,000, with special skill factors equals 0.28 using a s/w equation.?

$T_{min} = 8.14 (LOC / P)^{0.43}$

Here  $LOC = 33200$  and  $P = 12,000$

$T_{min} = 8.14(33200/12000)^{0.43}$

= 12.6 calendar months.

$E = 180Bt^3$

=  $180 * 0.28 * (1.05)^3$

= 58 person-months. (note :  $1.05 = 12.6 / 12$  years)

[ Nb: For small programs with (KLOC = 5 to 15) ,  $B = 0.16$  .For programs greater than 70 KLOC ,  $B = 0.39$  ].

36. What is MOI model of leadership?

Weinberg suggests a MOI model of leadership:

Motivation- Encourage technical people to produce their best quality.

Organization- ability to mold existing process to transfer initial concept to a final product.

Ideas or Innovation- Encourage people to create .

37. Specify a technique to identify risk.

Risks can be identified by risk item check list. The checklist focuses on some subset of known and predictable risks in product size, business impact, customer characteristics, process definition, development environment, technology to be built ,staff size and experience.

38. What are the s/w risk components?

Performance risk.

Cost risk.

Support risk.

Schedule risk.

39. How does a risk projection (estimation) rates the risk?

Probability that the risk occurs really. Consequences of the problems associated with the risk.

40. If 18 components are to be developed from scratch and average component is 100 LOC ,and if cost for each LOC is \$14.00 ,with a risk probability of 80% , what is the risk exposure?

Thus  $RE = 0.80 * 25,200 = \$20,200$ .

41. What is a referent point or break point?

The risk referent level has a single point called break point at which the decision to proceed with the project or terminate are equally weighted.

42. What is Pareto 80-20 rule?

Experience indicates that 80% of the overall project risk can be accounted for by only 20% of the identified risks. 20% includes the risk that lead to highest risk exposure.

43. what is a S/w project scheduling?

S/w project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort specific s/w engineering tasks.

44. List out the uses of PERT and CPM.

Determines the critical path. Establish the "most-likely" time estimates for individual tasks.

Calculate "boundary times" that define time-window for a particular task.

45. What are the boundary times ?

Early start time of a task.

Latest start time of a task.

Earliest finish time.

Latest finish time.

The total float.

46. What is a time-line chart or Gantt chart?

It is a chart listing the various work tasks ,the duration of each tasks and associated milestones .It can be developed for the entire product or for a product function or for an individual.

## UNIT-II

1. What is control abstraction and give an example?

Control abstraction is used in software design and implies a program control mechanism without specifying internal details. Example is synchronization semaphore used to coordinate activities in an OS.

2. List out the different notations to specify the control hierarchy.

Tree like diagrams.

Warnier-Orr diagrams

Jackson diagrams.

3. What are the extra-functional properties a good architecture design must satisfy?  
other system characteristics.

4. What is (a) coincidental cohesion (b) logical cohesion.

Modules that perform a set of tasks that are loosely related to one another are said to have coincidental cohesion. A module that performs tasks that are logically related is said to have logical cohesion.

5. Specify the major issues in designing a distributed system.

Specifying the topology of communication network.

Rules for accessing shared communication channel.

Allocating processing functions to various processing nodes.

Rules for process communication and synchronization.

Allocate n/w functionality b/w h/w & s/w components.

6. What are the five criteria to evaluate a modular design?

Modular decomposability.

Modular composability

Modular understandability.

Modular continuity.

Modular protection.

7. List out the advantages of horizontal partitioning?

- Software produced is easier to test.

- Software that is easy to maintain.

- Propagation of fewer side effects.

- Software that is easier to extend.

8. What are the different types of software reuse?

Application system reuse.

Component reuse.

Function reuse.

9. What is coupling? Explain content coupling.

Coupling is a measure of interconnection among modules in a software structure. It depends on the interface complexity b/w modules and what data is passed across the interface.

10. What are Petrinets?

Petrinet are graphical representation and is used to model the concurrent systems. It is a bipartite directed graph and two types of nodes are: places and transitions. Used to specify the requirements and design of real-time and distributed systems.

11. What is refinement in design, according to Wirth?

In each step of refinement, one or several instructions of the given program are decomposed into more detailed instructions. This successive decomposition terminates when all instructions are expressed in terms of any programming

12. Define software architecture.

Software architecture alludes to “overall structure of the software and the ways in which the structure provides conceptual integrity for a system”. Architecture is the hierarchical structure of program components, the manner in which the components interact and structure of data that are used by these components.

13. What is called factoring or vertical partitioning?

Factoring suggests that control and work should be distributed top-down in the program structure. Top level modules should perform control functions. Low level modules include workers, performing all input, computation and output tasks.

14. What is information hiding?

The principle of information hiding suggests that modules be characterized by design a decision that hides from all others. The modules should be specified and designed so that information within a module is inaccessible to other modules that have no need for such information.

15. What is cohesion? List out the various types.

A cohesive module performs a single task within a s/w procedure, requiring little interactions with procedures being performed in other parts of a program. The types of cohesion include:

Coincidental cohesion.

Logical cohesion.

Temporal cohesion.

Communication cohesion.

Sequential cohesion.

Functional cohesion.

Informational cohesion.

16. What are HIPO diagrams?

HIPO diagrams are design representation schemes for top-down software development and as external documentation aids. HIPO diagrams contains a visual table of contents ,a set of overview diagrams and a set of detailed diagrams.

17. What are real-time systems?

Real time systems provide specified amount of computation within fixed time intervals. Real time systems typically sense and control external devices, respond to external events and share processing time between multiple tasks.

18. List out the three steps in Jackson structured programming.

The problem structure is mapped into a program structure by:

- i. The problem is modelled by specifying the input and output data structures using tree-like diagrams.
- ii. The i/p-o/p model is converted into a structural model for the program by identifying the points of correspondence solves the problem.

19. What are the advantages of software reuse?

- Increased reliability.
- Reduced process risk.
- Effective use of specialists.
- Standards compliance.
- Accelerated development.

20. What are structured flowcharts?

Structured flowcharts are restricted to composition of certain basic forms. The basic forms are characterized by single entry into and single exit from the form and the forms can be nested to any arbitrary depth.

## UNIT-III

1.What is an indicator?

An indicator is a metric or combination of metrics that provide insight into the software process, a software project or product itself. An indicator provides insight that enables the project manager or s/w Engineer to adjust process, project

.

2. List out the uses of project indicators?

Project indicators enable a software project manager to:

- Assess the status of an on-going project.
- Track potential risks.
- Uncover problem areas before they go “critical”.
- Adjust workflows.
- Evaluate project team’s ability to control quality of software products.

### 3. What are process indicators?

It enables a software engineering organization to gain insight into the efficacy of the existing process. They help to assess what works and what doesn't. It provides indicators that lead to long term software process improvement.

### 4. Give examples of private metrics.

Private metrics include defect rates by individual, defect rates by module and errors found during development.

### 5. What is PSP?

The personal software process (PSP) is a structured set of process descriptions, measurements and methods that can help engineers to improve their personal performance. It provides the forms, scripts and standards that help them to estimate and plan their work.

### 6. What is SSPI/ Statistical Software Process Improvement?

SSPI uses software failure analysis to collect information about all errors

### 7. What are direct and indirect measures of software?

Direct measures of S/w engineering process include cost and effort applied. It includes LOC produced, execution speed, memory size. Indirect measures of the product include functionality, quality, complexity, efficiency, reliability and maintainability.

### 8. What is the information domain considered in FP metrics?

Number of user inputs.

Number of user outputs.

Number of user inquiries.

Number of files.

Number of external interfaces.

### 9. What are the characteristics of feature points?

Feature a point is a superset of function point measure applied in real time and embedded software applications. It counts the information domain values and algorithms and hence used in systems where algorithmic complexity is high.

### 10. What is software quality?

Software quality is defined as conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

### 11. What are the activities of a SQA group?

Prepares an SQA plan for the project.

Participates in the development of the project's s/w process description.

Reviews s/w engg activities for compliance with defined s/w process.  
Audits designated s/w work products for compliance with those defined as part of s/w process.  
Records any noncompliance and reports to senior management.

12. What are the uses of reviews?

Point out needed improvements in the product of a single person or team.  
Confirm those parts of a product in which improvement is either not desired or not needed.  
Achieve more uniform technical work.

13. What are constraints on FTR?

Between 3 or 5 people should be involved in the review.  
Advance preparation should occur.  
The duration of the review meeting should be

14. What are the purposes of review issue list?

To identify problem areas within the product.  
To serve as an action item checklist that guides the producer as corrections are made.

15. What is software reliability?

Software reliability is defined in statistical terms as the probability of failure-free operation of a computer program in a specified environment for a specified time. A program X is estimated to have a reliability of 0.96, if it is executed 100 times and operates correctly 96 times.

16. Define Software availability.

Software availability is the probability that a program is operating according to requirements at a given time and is defined as:

$$\text{Availability} = \left[ \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \right] * 100\%$$

17. What is software safety?

Software safety is a SQA activity that focuses on the identification and assessment of potential hazards that may affect software negatively and causes an entire system to fail. If hazards can be identified early in the software engineering process, software design features can be specified that will either eliminate or control potential hazards.

18. What are the requirements addressed by ISO 9001 standard?

The requirements delineated by ISO 9001 includes topics like management responsibility, contract review, design control, process control, inspection and testing, corrective and preventive action, control of quality records, internal quality auditing, training, servicing and statistical techniques.

19. What are the contents of review summary report?

It answers the three questions:

What was reviewed?

Who reviewed it?

What were the findings and conclusions?

20. What are the quality costs?

It includes:

Prevention costs.

Appraisal costs.

Failure costs.

## UNIT-IV

1. Specify the objectives of testing.

Testing is a process of executing a program with the intent finding an as yet undiscovered error.

A successful test is one that uncovers an as yet undiscovered error.

2. What is software testability?

Software testability is simply how easily a computer program can be tested. Testability is used to mean how adequately a particular set of tests will cover the product.

3. What are the attributes of a good test?

- A good test has a high probability of finding an error.

- A good test is not redundant.

- A good test should be best of breed.

- A good test should be neither too simple nor too complex.

4. What is basis path testing?

The basis path method enables the test case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basis set of execution paths. Test cases derived to exercise the basis set are guaranteed to execute every statement in the program at least one time during testing.

5. What is an independent path?

An independent path is any path through the program that introduces at least one new set of processing statements or a new condition. In case of flow graph, an independent path must move along at least one edge that has not been traversed before the path is defined.

6. What is verification and validation?

Verification refers to the set of activities that ensure that software correctly implements a specific function. "Are we building the product right?" Validation refers to the different set of activities that ensure the s/w is traceable to customer requirements. "Are we building the right product?"

7. What is the role of an ITG?

The role of an independent test group (ITG) is to remove the inherent problems associated with the software. ITG is the part of s/w development project team in the sense that it becomes involved during specification activity and in planning and specifying test procedures. ITG group reports to the SQA organization.

8. What is antidebugging?

It is the technique to setup error handling paths and to reroute or cleanly terminate processing when an error occurs.

9. What is regression testing?

It is the re-execution of some subset of tests that have already been captured/playback tools.

10. What is cyclomatic complexity?

Cyclomatic complexity is an s/w metric that provides a quantitative measure of the logical complexity of a program. In basis path testing, the value computed for cyclomatic complexity defines the number of independent paths in the basis set of a program and provides an upper limit for the number of tests that must be conducted to ensure that all statements have been executed at least once.

11. What is a region in a flow graph?

Areas bounded by edges and nodes are called regions.

12. What is a predicate node?

Each node that contains a condition is called a predicate node, and is characterized by two or more edges emanating from it.

13. How will you compute the cyclomatic complexity?

(a) The number of regions of flow graph indicates cyclomatic complexity,  $V(G)$ .

(b) Also  $V(G) = E - N + 2$ , where  $E$  → number of flow graph edges,  $N$  → number of flow graph nodes.

(c)  $V(G) = P + 1$ , where  $P$  is the predicate node.

14. What is a connection matrix?

The graph matrix in which '1' represents connection exists between a pair of nodes and '0' for the absence of connection.

15. List out the different classes of loops?

Simple loops

Concatenated loops.

Nested loops.

Unstructured loops.

16. What is behavioral testing?

Black-box or behavioral testing focuses on the functional requirements of the software.

17. List out the types of errors uncovered by black-box testing?

- Incorrect or missing functions

- Interface errors.

- Errors in data structure and external data base access.
- Behaviour and performance errors.
- Initialization and termination errors.

18. What is equivalence partitioning?

It is a black-box testing method that divides the input domain of a of test cases to be developed.

19. What is a Boundary value Analysis (BVA)?

It is a testing technique that leads to the selection of test cases that exercise bounding values.  
Selects test cases at the edges of the class.

20. When is orthogonal array testing suitable?

Applied to problems in which the input domain is relatively small.

Also finds errors associated with region faults –an error category associated with a faulty logic within a software component.

21. What comprises “build” in smoke testing?

A build includes all data files, libraries, reusable modules and engineered components that are required to implement one or more product functions.

22. List out the uses of smoke testing.

Integration risk is minimized.

The quality of the end-products is improved.

Error diagnosis and correction is simplified.

Progress is easier to assess.

23. What are the characteristics of a critical module?

Addresses several software requirements.

Has a high level of control.

Is complex or error prone.

Has definite requirements.

24. How are the automatic recovery evaluated?

- Reinitialization

- Check pointing mechanisms.

- Data recovery.

- Restart.

25. What are the three categories of debugging?

Brute force.

Back tracking

Cause elimination.

## Unit 5

### TwoMark Questions and Answers

#### 1. Define Quality?

Quality is the degree of goodness of a product or service or perceived by the customer.

Quality concept is the way business organizations perform their business activities that focus on two things.

#### 2. What are the different dimensions of quality?

The different dimensions of quality are Performance, features, reliability, conformance, durability, serviceability, aesthetics, perceived quality and reputation, response.

#### 3. Define Quality planning?

Quality planning is a selection of appropriate procedures and standards from this framework, adapted for a specific software project.

#### 4. Define Quality control?

Quality control is a definition and enactment of processes that ensure the software development team have followed project quality procedures and standards.

#### 5. What is Quality improvement?

Quality improvements aims at attaining unprecedented levels of performance which are significantly better than the past level.

#### 6. Define Fan in?

Fan in is a count of the modules that call a given module.

#### 7. Define Fan out?

A count of modules that are called by a given module.

#### 8. What are the measures of software quality?

- Correctness
- Maintainability
- Integrity
- Usability

#### 9. Define software engineering process?

The total set of software engineering activities needed to a user's requirement into software.

10. Define software configuration management?

SCM is a set of activities carried out for identifying, organizing and controlling changes throughout the life cycle of computer software.

11. What are critical software process issues?

The critical software process issues are Quality, product technology, requirement instability and complexity.

12. What is meant by software configuration item?

A software configuration item is created as part of the software engineering process.

13. There are three different types of review for quality management?

- Design or program inspections
- Progress reviews
- Quality reviews.

14. What are the parameters of software configuration?

- Identification control
- Version control
- Change control
- Configuration control
- Status control

15. What are the different SCM features?

Versioning, dependency tracking and change management, requirement tracing, configuration management, audit trails.

16. What are the different types of software quality metrics?

- Fan in/fan out
- Length of code
- Cyclomatic complexity
- Length of identifiers
- Depth of conditional nesting
- Fox index

17. What are the documents required to implement quality management system in organization?

- Quality policy and quality objectives

- Quality manuals
- Document to ensure the effective planning,operation and control of its process
- Records

18. Define software quality metrics?

Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process and project. In general, software quality metrics are more closely associated with process and product metrics than with project metrics.

19. What are the software quality attributes considered during the quality planning process?

There is a wide range of potential software quality attributes which are given below and are considered during the quality planning process

- Security
- Reliability
- Understandability
- Testability
- Adaptability
- Efficiency

20. Define Baseline?

A specification or product that has been formally reviewed and agreed upon, that serves as the basis for further development, and that can be changed only through formal change control procedures.