

Fun. point

COCOMO (effort, time ...)

Black box testing (test cases) (types)

DFD (Diagram)

Cyclomatic complexity

T(6th Sm.)-Computer Sc.-H/CC-13/CBCS

2021

COMPUTER SCIENCE — HONOURS

Paper : CC-13

(Software Engineering)

Full Marks : 50

The figures in the margin indicate full marks.

Candidates are required to give their answers in their own words
as far as practicable.

Answer question no. 1 and any four from the rest.

| open system | closed system |
|---|--|
| 1. highly compatible with other systems | 1. low compatibility |
| 2. easily extendable | 2. difficult to modify or extend |
| 3. source code is often available | 3. source code is not available |
| 4. strong community involvement | 4. controlled by a single organization |

2x5

1. Answer any five questions :

- (a) Write the differences between an *open* and a *closed* system.
- (b) What do you understand by software fault? → It is an error or defect in a programme that causes it to produce incorrect or unexpected results. Faults can occur at various stages of the software development process, from the initial design to the final development. Common faults are: coding errors, design flaws, and requirements errors.
- (c) What do you mean by the term 'V and V activities', and what is its role in the lifecycle of a software? → V & V activities means Verification and Validation is a very important part of the SDLC. It helps in testing the quality of the software. Verification: It is the process of checking that software achieves its goal without any bugs, and verifies that the product developed fulfills the requirements. Validation: Also known as dynamic testing, checks whether the software product is up to the mark. It checks what we are developing is the right product. It is the validation of what is expected product.
- (d) Why is unit testing performed on software? → It focuses on individual units or components of a software system. It validates if each unit of the software works as intended & meets the requirements. Unit testing is done by the developer, it is performed early in the development process before the code is integrated to be tested as a whole system. Unit testing is done after each time the code is changed to ensure that the new code does not break the existing functionality. Helps to fix any error earlier in the development process.
- (e) What is context diagram?
- (f) Why is the low coupling desirable? → Easy to modify as changes made in one module is less likely to impact other modules, making it easier to extend & maintain. Also can be built & find as they do not propagate easily.
- (g) What is Logical DFD?
- (h) Write the functionality of a stub. → Logical Data Flow Diagram mainly focuses on the system process. It illustrates how data flows in the system. It is used in various organizations for the smooth running of system. In building software system, it is used to describe how data is moved from one entity to another.

2. (a) What are the advantages of the Prototype Model? Regardless of its merits, why is it a costly model to emulate? → Advantages are:-

1. reduces risk of incorrect user requirements.
2. reduces rework.
3. allows for rapid development.
4. testable, makes user requirement changes.

It is costly to emulate because:-

1. prototyping tools are expensive.
2. time consuming process, which increases development cost.

(b) Discuss the Evolutionary Model in brief, along with its suitable domain of application.

Evolutionary model is based on making a initial product & then evolving over time. It follows incremental & iterative approach with frequent customer feedback. It is used in cases where requirements are not specified properly or a subject to change in future. Eg. Mobile apps, websites.

3. (a) What is the role of an SRS? What are the components of a good SRS?

Components of a good SRS are:-

1. functional requirements of the system.
2. non-functional requirements of the system.
3. goals of implementation.
4. use cases.

SRS is a formal report, which acts as a representation of the software. It enables customer to verify whether the system is according to their requirement.

Role:-

1. provides understanding of what the system is expected to do.
2. helps the developer.
3. provides customer feedback.

use cases as a product validation check.

→ (b) Discuss the different types of 'coupling' in detail.

Types of coupling:-

1. Data coupling → If modules communicate through data.
2. Stamp coupling → If modules communicate through a common variable.
3. Control coupling → If data of one module is used to direct the order of execution in second module.
4. Common coupling → If modules share data through global data items.
5. Content coupling → If modules share codes.
6. External Coupling → If modules share DB, libraries etc.

4. (a) A software project of type Semi-detached comprises of 3750 KLOC. Compute the development effort using Basic COCOMO.

(b) Describe the features of organic type of software as specified in COCOMO.

→ Small team, good communication & collaboration, well understood requirements, flexible & simple constraints, relatively small & etc.

(c) Write the advantages of decision tree in problem design.

→ Easy to understand & interpret,
Simple to use & implement,
Transparency in decision making.
Capable of handling large datasets.

Given: 3750 KLOC
Semi-detached

Formula: $E = A \times (KLOC)^B$ [where A & B are constants
KLOC = kilo line of codes]

Development Time = $C \times (E)^D$ [where C & D are constants
 $E = \text{Effort}$]

$$E = 3 \times 3750^{0.42}$$

$$= 3 \times 10067.3466$$

$$= 30202.04$$

$$T_{dev} = C + E^{0.85}$$

$$= 2.5 \times 30202.04^{0.85}$$

$$= 2.5 \times 86.984$$

$$= 92.459$$

Please Turn Over

Q.5 Black box testing focuses on testing the functionality of the software, ensuring the software functionality works as intended.

- a. Black box testing is done by a separate team of testers.
- b. No knowledge about internal code.
- c. focus of functionality & behaviour of the software.
- d. e.g - functional testing,
system testing,
acceptance testing.

6-digits

valid : 123456
987654

invalid :

c1: less than 6 num → 123,790
c2: more than 6 num → 12345670, 1792186
c3: alpha numeric → ab123, 12d3
c4: special character → -123, 92;4@

T(6th Sm.)-Computer Sc.-H/CC-13/CBCS

(2)

5. (a) Why is Black Box testing also known as functional testing? How is it different from traditional White Box testing?

(b) Design an equivalence class partitioning for a program that accepts as an input, an OTP of 6-digits, and considers all other types of inputs as invalid.

(c) How is testing related to software quality? →
1. find & fix bugs improves reliability.
2. verification & validation ensures software meets user needs & specification.
3. Ensures protection against threats.
4. Evaluates how system performs under diff. conditions
5. Improvement with continuous testing & feedback. (1+2)+4+3

6. (a) Design a level-0 DFD and level-1 DFD for library management system.

→ A physical DFD shows how data flows through a system's physical component such as hardware, software, files & people.
1. Actions or functions that manipulate data.
2. Physical storage locations for data.
3. Movement of data flow between process, data storage, & external entities (user or other systems).

7. (a) Briefly discuss different Software Quality Assurance (SQA) activities in brief.

(b) How are the terms 'reliability' and 'maintainability' related to software quality?

8. (a) Briefly discuss different stages of SDLC.

(b) Define : Computer based systems engineering.

Q.7(b)

High reliability means the software consistently operates correctly, providing users with dependable service & minimum downtime. It enhances the trust & satisfaction.

High maintainability means the software can be easily updated, extended, or fixed. Resulting low maintenance cost.

↑ Reliability & Maintenance

→ SQA management plan - make a plan for how to carry out the SQA throughout the project.
Set check points - Evaluate performance of the project on the basis of collected data at different check points.
MultiTesting strategy - Do not rely on single testing approach.
Measure change impact
Manage good relations

Q.8(a) Phases of SDLC :-

- 1. Communication → Planning of project goals, timeline, resources.
- 2. Requirement Gathering → Gathers the need of the customer, what the software achieves.
- 3. Feasibility Study → Evaluates the project's feasibility to ensure viability.
- 4. System Analysis → Documenting SRS.
- 5. Coding → writing codes based on the SRS.
- 6. Testing → Testing software to identify & fix bugs.
- 7. Integration → Combining individual software modules & testing them as a group.
- 8. Deployment → Deploying software to the production environment.
- 9. Maintenance → bug fixes, updates for future.