Testing

June 2014 Paper II

Regression testing is primarily related to

(A) Functional testing
(B) Development testing
(C) Data flow testing
(D) Maintenance testing

(D) Maintenance testing

Regression testing is primarily related to (D) Maintenance testing. It involves retesting a software application after changes have been made to ensure that existing functionalities are not affected by those changes.

Paper III June 2014

Software testing is

(A) the process of establishing that errors are not present.
(B) the process of establishing confidence that a program does what it is supposed to do.
(C) the process of executing a program to show that it is working as per specifications.
(D) the process of executing a program with the intent of finding errors.

(D) the process of executing a program with the intent of finding errors.

Software Testing is the process of executing a program or system with the intent of finding errors. Or, it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results.

Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails.

Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible

Software testing is a process of executing a program or application with the intent of finding errors (the software bugs.)

It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided it's design and development.

Paper III June 2014

Which one of the following is not a definition of error?

(A) It refers to the discrepancy between a computed, observed or measured value and the true, specified or theoretically correct value.
(B) It refers to the actual output of a software and the correct output.
(C) It refers to a condition that causes a system to fail.
(D) It refers to human action that results in software containing a defect or fault.

(C) It refers to a condition that causes a system to fail.

Definition of Error : One common definition of a software error is a mismatch between the program and its specification. In other words, we can say, a software error is present in a program when the program does not do what its end user expects.

Categories of Software Errors:

- **User interface errors such as output errors or incorrect user messages.**
- **Function errors**
- **Hardware defects**
- **Incorrect program version**
- **Requirements errors**
- **Design errors**
- **Documentation errors**
- **Architecture errors**
- **Module interface errors**
- **Performance errors**
- **Boundary-related errors**
- **Logic errors such as calculation errors, State-based behavior errors, Communication errors, Program structure errors, such as control-flow errors.**

Paper II December 2014

Which one of the following is used to compute cyclomatic complexity ?

(A) The number of regions – 1
(B) E – N + 1, where E is the number of flow graph edges and N is the number of flow graph nodes.
(C) P – 1, where P is the number of predicate nodes in the flow graph G.
(D) P + 1, where P is the number of predicate nodes in the flow graph G.

(D) P + 1, where P is the number of predicate nodes in the flow graph G.

Cyclomatic complexity is a software measurement, used to indicate the complexity of a program.

Cyclomatic complexity for aflow graph G is V(G)=E-N+2, where N is the number of nodes and E is the number of edges in the flow graph.

Paper II June 2015

In which testing strategy requirements established during requirements analysis are validated against developed software?

(A) Validation testing
(B) Integration testing
(C) Regression testing
(D) System testing

(A) Validation testing

In Validation testing it is checked that a software system meets specifications and that it fulfills its intended purpose. i.e. strategy requirements established during requirements analysis are validated against developed software.

Paper III June 2015

Verification:

(A) refers to the set of activities that ensure that software correctly implements a specific function.
(B) gives answer to the question - Are we building the product right ?
(C) requires execution of software
(D) both (A) and (B)

(D) both (A) and (B)

Verification refers to the set of activities that ensure that software correctly implements a specific function and gives answer to the question - Are we building the product right ? but it does not requires execution of software.

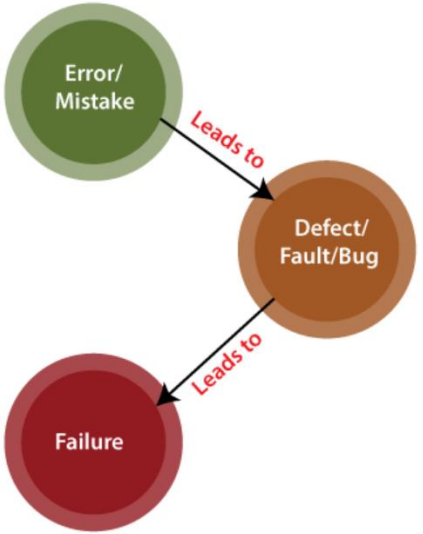
Paper II December 2015

In software testing, how the error, fault and failure are related to each other?

(A) Error leads to failure but fault is not related to error and failure
(B) Fault leads to failure but error is not related to fault and failure
(C) Error leads to fault and fault leads to failure
(D) Fault leads to error and error leads to failure

(C) Error leads to fault and fault leads to failure

An Error is a mistake made in the code; that's why we cannot execute or compile code. The Fault is a state that causes the software to fail to accomplish its essential function. If the software has lots of defects, it leads to failure or causes failure.



Comparison basis	Bug	Defect	Error	Fault	Failure
Definition	It is an informal name specified to the defect.	The **Defect** is the difference between the actual outcomes and expected outputs.	An **Error** is a mistake made in the code; that's why we cannot execute or compile code.	The **Fault** is a state that causes the software to fail to accomplish its essential function.	If the software has lots of defects, it leads to failure or causes failure.
Raised by	The **Test Engineers** submit the bug.	The **Testers** identify the defect. And it was also solved by the developer in the development phase or stage.	The **Developers and automation test engineers** raise the error.	**Human mistakes** cause fault.	The failure finds by the manual test engineer through the **development cycle**.

Paper II July 2016

The cyclomatic complexity of a flow graph V(G), in terms of predicate nodes is:

(A) P + 1
(B) P - 1
(C) P - 2
(D) P + 2

Where P is number of predicate nodes in flow graph V(G).

Paper II July 2016

The cyclomatic complexity of a flow graph V(G), in terms of predicate nodes is:

(A) P + 1
(B) P - 1
(C) P - 2
(D) P + 2

Where P is number of predicate nodes in flow graph V(G).

Paper II August 2016 (Re-test)

For a program of k variables, boundary value analysis yields ..... test cases.

(A) 4k – 1
(B) 4k
(C) 4k + 1
(D) 2k – 1

(C) 4k + 1

For a program of k variables boundary value analysis yields 4 * k + 1 test Robustness testing yields 6 * k + 1 worst case testing yields 5 * k.

Paper III January 2017

Which of the following statement(s) is/are TRUE with regard to software testing?

I. Regression testing technique ensures that the software product runs correctly after the changes during maintenance.

II. Equivalence partitioning is a white-box testing technique that divides the input domain of a program into classes of data from which test cases can be derived.

(1) only I
(2) only II
(3) both I and II
(4) neither I nor II

(1) only I

Regression testing is a type of software testing that verifies that software previously developed and tested still performs correctly even after it was changed or interfaced with other software.

Equivalence partitioning is black box testing approach and not a white box testing

Equivalent Partioning or Equivalence Class Partitioning is a black box technique (code is not visible to tester) which can be applied to all levels of testing like unit, integration, system, etc. In this technique, you divide the set of test condition into a partition that can be considered the same.

Paper III January 2017

Which of the following are facts about a top-down software testing approach?

I. Top-down testing typically requires the tester to build method stubs.
II. Top-down testing typically requires the tester to build test drivers.

(1) only I
(2) Only II
(3) Both I and II
(4) Neither I nor II

(1) only I

Top down Testing: In this approach testing is conducted from main module to sub module. if the sub module is not developed a temporary program called STUB is used for simulate the submodule.

Bottom up testing: In this approach testing is conducted from sub module to main module, if the main module is not developed a temporary program called DRIVERS is used to simulate the main module.

Paper III January 2017

Complete each of the following sentences in List-I on the left-hand side by filling in the word or phrase from the List-II on the right-hand side that best completes the sentence:

List-I	List-II
I. Determining whether you have built the right system is called	A. Software testing
II. Determining whether you have built the system right is called	B. Software verification
III. ..... is the process of demonstrating the existence of defects or providing confidence that they do not appear to be present.	C. Software debugging
IV. ..... is the process of discovering the cause of a defect and fixing it.	D. Software validation

Codes:

	I	II	III	IV
(1)	B	D	A	C
(2)	B	D	C	A
(3)	D	B	C	A
(4)	D	B	A	C

Paper III January 2017

Complete each of the following sentences in List-I on the left-hand side by filling in the word or phrase from the List-II on the right-hand side that best completes the sentence:

List-I	List-II
I. Determining whether you have built the right system is called	A. Software testing
II. Determining whether you have built the system right is called	B. Software verification
III. ..... is the process of demonstrating the existence of defects or providing confidence that they do not appear to be present.	C. Software debugging
IV. ..... is the process of discovering the cause of a defect and fixing it.	D. Software validation

Codes:

	I	II	III	IV
(1)	B	D	A	C
(2)	B	D	C	A
(3)	D	B	C	A
(4)	D	B	A	C

Paper II November 2017

What is the normal order of activities in which traditional software testing is organized?

(a) Integration Testing
(b) System Testing
(c) Unit Testing
(d) Validation Testing

Code:

(1) (c), (a), (b), (d)
(2) (c), (a), (d), (b)
(3) (d), (c), (b), (a)
(4) (b), (d), (a), (c)

(2) (c), (a), (d), (b)

The correct order of testing activities in software testing is:
1. Unit testing: It includes the testing of every single module or component of the software.
2. Integration testing: It includes the testing of two or more combined modules of the software.
3. Validation testing: It answers the question, "Are we building the right product?". It ensures that client's requirements must be fulfilled by the developed software.
4. System testing: In this testing, the whole system is tested as a single component.

Hence, the correct order of testing is Unit testing, Integration testing, Validation testing & System testing.

Paper II November 2017

Which of the following testing techniques ensures that the software product runs correctly after the changes during maintenance?

(1) Path Testing
(2) Integration Testing
(3) Unit Testing
(4) Regression Testing

(4) Regression Testing

option1: Path testing is used to design the test cases.

option2: Integration testing includes the testing of two or more combined modules of the software.

option3: Unit testing includes the testing of every single module or component of the software.

option4: Regression testing is used to test modified parts of code & parts affected by the code to ensure that software doesn't have any defect after modifications.

Paper III November 2017

Which of the following statements is/are FALSE with respect to software testing?

S1 : White-box tests are based on specifications; better at telling whether program meets specification, better at finding errors of ommission.

S2 : Black-box tests are based on code; better for finding crashes, out of bounds errors, file not closed errors.

S3 : Alpha testing is conducted at the developer's site by a team of highly skilled testers for software that is developed as a product to be used by many customers.

(1) Only S1 and S2 are FALSE.
(2) Only S1 and S3 are FALSE.
(3) Only S2 and S3 are FALSE.
(4) All of S1, S2, and S3 are FALSE.

(1) Only S1 and S2 are FALSE.

S1: Black-box tests are based on specifications; better at telling whether the program meets specification, better at finding errors of omission. Hence the statement I is False.

S2: White-box tests are based on code; better for finding crashes, out-of-bounds errors, file not closed errors. Hence the statement II is False.

S3: Alpha testing is conducted at the developer's site by a team of highly skilled testers for software that is developed as a product to be used by many customers. Hence the statement III is True.

Paper III November 2017

Consider the method mcq ( ) :

```
int mcq (boolean a, boolean b, boolean c, boolean d)
{
int ans=1;
if (a) {ans = 2;}
else if (b) {ans = 3;}
else if (c) {
if (d) {ans=4;}
}
return ans ;
}
```

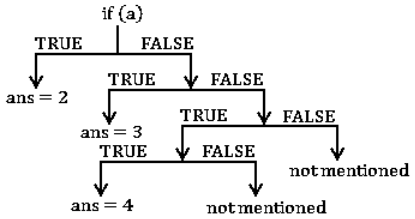
If

M1 = Number of tests to exhaustively test mcq ( );
M2 = Minimum number of tests to achieve full statement coverage for mcq ( ); and
M3 = Minimum number of tests to achieve full branch coverage for mcq ( );
then (M1, M2, M3) = ...

(1) (16, 3, 5)
(2) (8, 5, 3)
(3) (8, 3, 5)
(4) (16, 4, 4)

(1) (16, 3, 5)

M1: Number of tests to exhaustively test MCQ()



Given total $ conditions. Total exhaustively test MCQ =2*2*2*2=16

M2:Minimum number of tests to achieve full statement coverage for MCQ();

Total 4 conditions. If any one condition is TRUE remaining automatically will get False except C and D

Condition-A	Condition-B	Condition-C	Condition-D
True	False	False	False
False	True	False	False
False	False	True	True

M3: Minimum number of tests to achieve full branch coverage for MCQ():

Condition-A	True	Print ans=2	return Ans			
Condition-B	False	True	Print ans=3	return Ans		
Condition-C&D	False	False	True	True	Print ans=4	return Ans
Condition-C&D	False	False	True	False	return Ans	
Condition-C	False	False	False	return Ans		

Paper II December 2018

Consider the following method:

```
int f(int m, int n, boolean x, boolean y)
{
   int res = 0;
   if (m<0) {res = n - m;}
   else if (x || y)
   {
      res = -1;
      if(n == m){res = 1;}
   }
   else {res = n;}
   return res;
} /*end of f */
```

IF P is the minimum number of tests to achieve full statement coverage for f(), and Q is the number of tests to achieve full branch coverage for f(), then (P,Q) =

(1) (3,4)
(2) (4,3)
(3) (2,3)
(4) (3,2)

(1) (3,4)

To achieve full statement coverage, all the statements must be executed.

Statement coverage = (number of statement executed)/ total number of statements

In above code, there are three test cases required to achieve full statement coverage.

Case 1: when if condition becomes true,

```
int f(int m, int n, boolean x, boolean y)
{
int res=0;
if(m<0) {res=n-m;}
return res;
}
```

These statements will be executed.

Case 2: when other two cases of else will execute, then all remaining statements will also execute. By, three tests, full statement coverage is achieved.

Branch coverage: It is decision coverage and covers both true and false conditions.

Decision coverage = (number of decisions exercises)/ total number of decision outcomes.

It checks for true and false cases. In the above given code, there will be 4 tests required to achieve full branch coverage.

So, value of (P, Q) = (3, 4)

Paper II June 2019

In the context of software testing, which of the following statements is/are NOT correct?

P : A minimal test set that achieves 100% path coverage will also achieve 100% statement coverage.

Q: A minimal test set that achieves 100% path coverage will generally detect more faults than one that achieves 100% statement coverage.

R: A minimal test set that achieves 100% statement coverage will generally detect more faults than one that achieves 100 % branch coverage.

(a) R only
(b) Q only
(c) P and Q only
(d) Q and R only

(a) R only

True : A minimal test set that achieves 100% path coverage will also achieve 100% statement coverage.

True: A minimal test set that achieves 100% path coverage will generally detect more faults than one that achieves 100% statement coverage

False : A minimal test set that achieves 100% statement coverage will generally detect more faults than one that achieves 100% branch coverage.

Paper II June 2019

Software products need adaptive maintenance for which of the following reasons?

(a) To rectify bugs observed while the system is in use.
(b) When the customers need the product to run on new platforms.
(c) To support the new features that users want it to support.
(d) To overcome wear and tear caused by the repeated use of the software.

(b) When the customers need the product to run on new platforms.

To rectify bugs observed while the system is in use ----corrective maintenance
When the customers need the product to run on new platform-------adaptive maintenance
To support the new features that users want it to support-----perfective maintenance
To overcome wear and tear caused by the repeated use of the software----preventive maintenance

Paper II June 2019

Software validation mainly checks for inconsistencies between

(a) use cases and user requirements.
(b) implementation and system design blueprints.
(c) detailed specifications and user requirements.
(d) functional specifications and use cases.

(c) detailed specifications and user requirements.

Software validation : is performed to ensure that the product actually meets the user's needs and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.

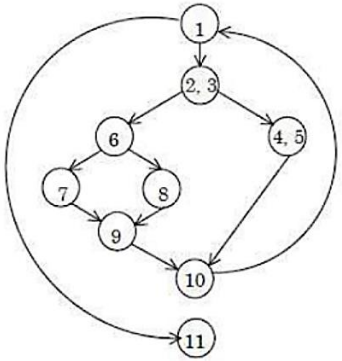
Software validation is achieved through a series of tests that demonstrate conformity with requirements. A test plan outlines the classes of tests to be conducted, and a test procedure defines specific test cases that are designed to ensure that all functional requirements are satisfied, all behavioral characteristics are achieved, all content is accurate and properly presented, all performance requirements are attained, documentation is correct, and usability and other requirements are met (e.g., transportability, compatibility, error recovery, maintainability). After each validation test case has been conducted, one of two possible conditions exists:

(1) The function or performance characteristic conforms to specification and is accepted or

(2) a deviation from specification is uncovered and a deficiency list is created.

Paper II December 2019

Comprehension:
Answer the following questions (91 - 95) based on flow graph F.
A flow graph F with entry node (1) and exit node (11) is shown below:



Flowgraph F

How many nodes are there in the longest independent path?
(1) 6 (2) 7 (3) 8 (4) 9

(3) 8

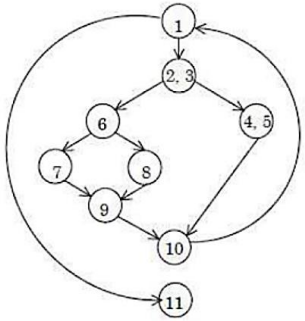
A linearly independent path is any path that introduces at least one new edge that is not included in any other linearly independent path. Data From the graph shown above, following are possible independent paths.

Path 1: 1-11
Path 2: 1-2-3-4-5-10-1-11
Path 3: 1-(2,3)-6-8-9-10-1-11
Path 4: 1-(2,3)-6-7-9-10-1-11
Each of the above path introduces a new edge.
And Path 3 and Path 4 are longest independent paths with 8 nodes.

Note that, a path like 1-(2,3)-4-5-10-1-2-3-6-8-9-10-1-11 is not an independent path because it includes already specified edges and doesn't exclusively add new edges.

Paper II December 2019

Comprehension:
Answer the following questions (91 - 95) based on flow graph F.
A flow graph F with entry node (1) and exit node (11) is shown below:



Flowgraph F

How many regions are there in flow graph F?
(1) 2 (2) 3 (3) 4 (4) 5

(3) 4

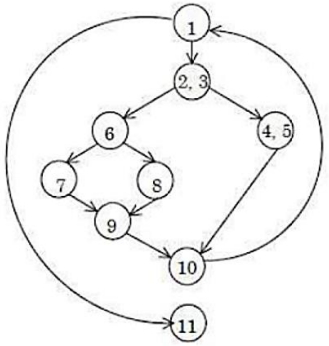
In the given flow graph F 4 regions are present.

- $(1,(2,3),(4,5),10)$
- $6,7,8,9$
- $1,(2,3),6,8,9,10,(4,5)$
- outer side region

Total 4 regions are present.

Paper II December 2019

Comprehension:
Answer the following questions (91 - 95) based on flow graph F.
A flow graph F with entry node (1) and exit node (11) is shown below:

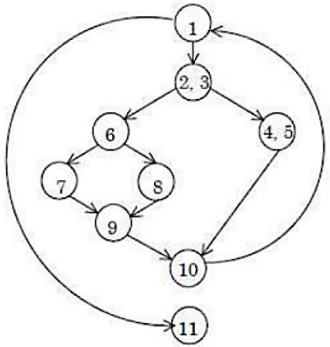


Flowgraph F

How many nodes are there in flow graph F?
(1) 9 (2) 10 (3) 11 (4) 12

Paper II December 2019

Comprehension:
Answer the following questions (91 - 95) based on flow graph F.
A flow graph F with entry node (1) and exit node (11) is shown below:

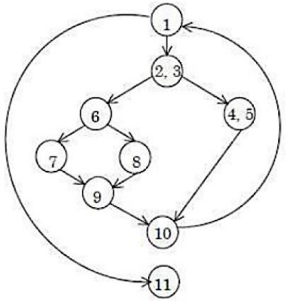


Flowgraph F

How many nodes are there in flow graph F?
(1) 9 **(2) 10** **(3) 11** **(4) 12**

Paper II December 2019

Comprehension:
Answer the following questions (91 - 95) based on flow graph F.
A flow graph F with entry node (1) and exit node (11) is shown below:



Flowgraph F

What is the cyclomatic complexity of flow graph F?
(1) 2 (2) 3 (3) 4 (4) 5

(3) 4

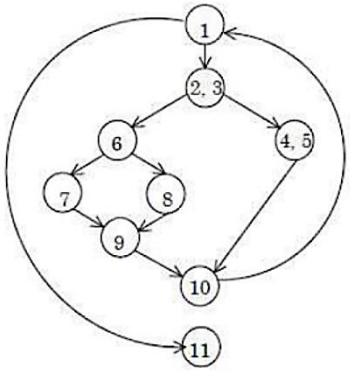
There are three ways of calculating the cyclomatic complexity M, E - Number of Edges N - Number of Nodes P - number of predicate nodes (1, (2,3), 6)

M = E – N + 2
M = 9 - 11 + 2
M = 9 - 13
M = 4

Or
M = p + 1
M = 3 + 1 = 4

Paper II December 2019

Comprehension:
Answer the following questions (91 - 95) based on flow graph F.
A flow graph F with entry node (1) and exit node (11) is shown below:



Flowgraph F

How many predicate nodes are there and what are their names?

(1) Three: (1, (2,3), 6)
(2) Three: (1, 4, 6)
(3) Four: ((2,3), 6, 10, 11)
(4) Four: ((2,3), 6, 9, 10)

(1) Three: (1, (2,3), 6)

In the given graph, the nodes with a condition or with 2 or more outgoing edges are – 1, (2.3), 6. Hence, there are 3 predicate nodes.

Paper II November 2020

Which of the following is/are behavioral testing technique(s)?

(A) Equivalence Partitioning
(B) Graph-Based Teating Method
(C) Boundery Value Analysis
(D) Data flow Testing
(E) Loop Testing

Choose the correct answer from the options given below:

a) (B) and (D) only
b) (A), (B) and (C) only
c) (D) and (E) only
d) (A), (C) and (E) only

b) (A), (B) and (C) only

Black-box testing, also known as behavioural testing, focuses on the software's functional specifications. In other words, black-box testing allows a software developer to create sets of input conditions that completely exercise all of a program's functional requirements. The methods for behavioural testing are as follows.

- **graph-based testing methods**
- **equivalence partitioning**
- **boundary value analysis**
- **comparison testing**
- **orthogonal array testing**

Paper II November 2021

Given below are two statements, one is labelled as Assertion A and the other is labelled as Reason R

Assertion A : Software developers do not do exhaustive software testing in practice.

Reason R : Even for small inputs, exhaustive testing is too computationally intensive (e.g., takes too long) to run all the tests.

In light of the above statements, choose the correct answer from the options given below

a) A is false but R is true
b) A is true but R is false
c) Both A and R are true and R is the correct explanation of A
d) Both A and R are true but R is NOT the correct explanation of A

c) Both A and R are true and R is the correct explanation of A

- **It is a Technique used in software development that tests all the possible combinations of a data set.**
- **Exhaustive Testing means complete testing, when all the testers in your team are exhausted and when all the planned tests haven been executed.**
- **Exhaustive process is a testing process and it cannot be done by the developers.**
- **In any project Exhaustive Testing is impossible and it is a part of Quality Assurance Testing.**

Paper II November 2021

In software testing, beta testing is the testing performed by _____.

a) potential customers at the developer's location
b) potential customers at their own locations
c) product developers at the customer's location
d) product developers at their own locations

b) potential customers at their own locations

Beta testing is the testing performed by potential customers at their own locations or performed by the end-user of the product.

Paper II October 2022

Alpha and Beta testing are forms of

a) White-Box Testing,
b) Black-Box Testing
c) Acceptance Testing
d) System Testing

c) Acceptance Testing

Types of Testing

Black Box Testing / Functional / Behavioral / I/O driven Testing

- Equivalence class Testing
- Boundary Value Testing
- Acceptance Testing
  - Alpha Testing
  - Beta Testing
- State based Testing
- Cause Effect Testing
- Pair Wise Testing

White Box Testing / Implementation / Structural / Logic – driven / Glass box Testing

- Control Flow Testing
- Data Flow Testing
- Basic Path Testing
- Statement Coverage
- Branch Coverage
- Path Coverage

Paper II October 2022

Fault base testing technique is

a) **Unit testing**
b) **Beta testing**
c) **Stress testing**
d) **Mutation testing**

d) Mutation testing

- **Mutation testing is a fault simulation technique.**

- **In this, the fault is introduced in the program by creating the mutant of the actual program.**

- **Mutant programs are the version of the actual program.**

- **When the testing is performed on actual program and its mutant program with the same test cases than the actual program and mutant program should generate at least one different output which leads the mutant program to get failed and show the effectiveness of the test cases.**

Paper II June 2023

_____ is intended to show that a system both conforms to its specifications and meets the expectations of the system customer.

1. Software specification
2. Software design
3. Software evaluation
4.Software validation

4. Software validation

There are many different kinds of software processes, but each and every one of them involve these four types of fundamental activities:

Software specification - defining what the system should do;

Software design and implementation - defining the organization of the system and implementing the system;

Software validation - checking that it does what the customer wants;

Software evolution - changing the system in response to changing customer needs.

Paper II June 2023

Match List I with List II

LIST I		LIST II	
A.	Scenario testing	I.	To verify the I/O behavior of text object
B.	Regression testing	II.	user acceptance methodology
C.	Component testing	III.	No new bugs after changes in program
D.	Beta testing	IV.	The documentation of a use case

Choose the correct answer from the options given below:

1. **A-IV, B-III, C-II, D-I**
2. **A-II, B-I, C-III, D-IV**
3. **A-IV, B-III, C-I, D-II**
4. **A-III, B-I, C-IV, D-II**

3. A-IV, B-III, C-I, D-II

Scenario testing is a software testing activity that uses scenarios: hypothetical stories to help the tester work through a complex problem or test system. There are two methods in scenario testing:

- **System scenarios: Scenario tests used in this method are only those sets of realistic, user activities that cover various components in the system.**

- **Use-case and role-based scenarios In the use-case and role-based scenario method the focus is specifically on how the system is used by a user with different roles and environment.**

Regression Testing is a type of testing in the software development cycle that runs after every change to ensure that the change introduces no unintended breaks.

Component testing, also known as program or module testing, is done after unit testing. In this type of testing those test objects can be tested independently as a component without integrating with other components. This testing is done by the development team.

Beta Testing is performed by real users of the software application in a real environment. Beta testing is one of the types of User Acceptance Testing.

Paper II June 2023

Given below are two statements: one is labelled as Assertion A and the other is labelled as Reason R.

Assertion A: Validity checks real need of system users.
Reason R: Completeness checks system user defined requirements.

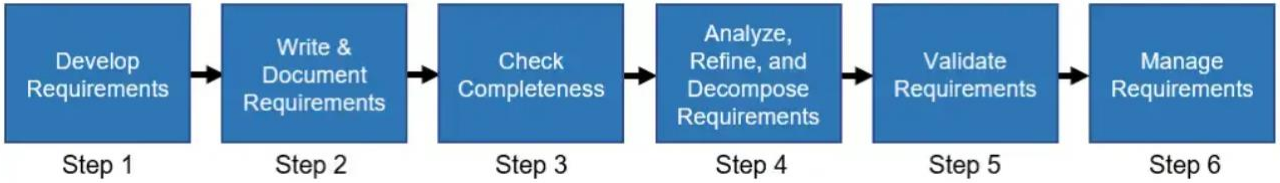
In the light of the above statements, choose the correct answer from the options given below.

1. Both A and R are true and R is the correct explanation of A
2. Both A and R are true but R is NOT the correct explanation of A
3. A is true but R is false
4. A is false but R is true

2. Both A and R are true but R is NOT the correct explanation of A

Requirements Validation Techniques are used to ensure that the software requirements are complete, consistent, and correct.

“Check Completeness” is to check that a complete set of requirements have been developed and documented that defines all system functions that are needed to satisfy the stakeholder needs with their associated performance, environmental, and other non-functional requirements.



Requirements Development Process

Paper II December 2023

Test suite is consist of:

(1) Set of defect cases
(2) Set of boundary cases
(3) Set of test cases
(4) Set of nest cases

Test suite is consist of:

(1) Set of defect cases
(2) Set of boundary cases
(3) Set of test cases
(4) Set of nest cases