RUDRA BARAD
18DCS007

UNIT TEST 2
SUB: SE
DATE: 16/4/21

Q11. COUPLING - Coupling is the measure of the degree of interdependence blw modules

CLASSIFICATION OF COUPLING

(1) DATA COUPLING - components are independent to each other and communicating through data.
Example. customer billing system

(2) STAMP COUPLING - complete data structure is passed from one module to another module.

(3) CONTROL COUPLING - If modules communicate by passing control info., then they are said to be control coupled.

(4) EXTERNAL COUPLING - modules depend on other modules, external to software being developed or to particular type of hardware

(5) COMMON COUPLING - modules have shared data such as global data structures. Changes in global data mean tracing back to all modules which access the data to evaluate the effect.

(6) CONTENT COUPLING - one module can modify the data of another module or control flow is passed from one module to other module.

→ COHESION - It is measure of degree to which elements of module are functionally related. A good software ~~software~~ design will have high cohesion.

CLASSIFICATION OF COHESION

(1) FUNCTIONAL COHESION - functional cohesion performs the task & functions. It is an ideal situation.

(2) SEQUENTIAL COHESION - An element outputs some data that becomes the input for other element.

(3) COMMONICATION COHESION - 2 elements operate on same input data / contribute towards same output

(4) PROCEDURAL COHESION - elements of procedural cohesion ensure the order of execution

(5) TEMPORAL COHESION - elements are related by their timing involved. It contains code for initializing all parts of system

(6) LOGICAL COHESION - elements are logically related and not functionally. Eg. disk and network

(7) COINCIDENTAL COHESION - elements are not related. Elements have no conceptual relationship other than location in source code.

Q12. Significance and importance of CMM-certification for any software organization are:

I. CONSISTENCY - provides proven approach that enables driven organizations to drive out real benifits in terms of improved project predectibility

II. COST SAVING - It delivers cost savings (efficient error detection) reducing cost of spending less on network.

III. SELF IMPROVEMENT - heat of competition is now driving significant interest in CMMI

IV. MARKET DEMAND - companies have adopted CMMI approach for best practices & reaping benifit to it.
- trying best to meet customers demands competitions

V. PERFORMANCE - purpose of CMMI is to improve upon the performance of existing organizational standard process & procedures & not to redefine them.

VI. PROCESS IMPROVEMENT - CMMI driven improvement project will deliver framework to standardize process, ensuring best business practices shared & adopted.

(4)

→ No, its not possible for an organization to cheive higher level of CMM without acheiving lower one as CMM follows hierarchical structure.

Q13

BLACK BOX TESTING	STRUCTURAL TESTING
→ In this testing the internal structure of software is not known	→ In this testing tester knowns the internal structure of software
→ It can be reffered as external testing	→ It is known as internal software testing
→ It is applicable to high level of testing of software	→ It is applicable to the lower levels of software testing
→ It is least time consuming	→ It is more time consuming
→ It is not suitable for algorithm testing	→ It is suitable for algorithm testing

TYPES OF WHITE BOX TESTING

(1) Statment coverage
(2) Branch coverage
(3) Path coverage
(4) Condition coverage
(5) Mutation testing
(6) Data flow based testing

CYCLOMATIC COMPLEXITY:

→ Cyclomatic complexity of a code part is quantitative measure of the no. of linearly independent paths in it. It is a software metric used to indicate the complexity of program

→ It is computed using CFG of program. The nodes of graph are lines of code and edges represent the order of executions of statements.

→ Formula to find cyclomatic complexity:

$e - n + 2p$

e = no. of edges
p = no. of disconnected parts of graphs
n = no. of nodes

→ Another way of computing CYC:
- inspect the CFG
- find no. of bounded areas in graph
- V(G) = no of bounded areas + 1

→ We can write a program to find no. of bounded regions of a CFG

→ CYC provides a lower bound on the no. of test cases to design.

Miracle Page Date

Q14. KEY COMPONENTS OF DEVOPS

(1) BUILD
(2) CODE
(3) TEST
(4) PLAN
(5) MONITOR
(6) DEPLOY
(7) OPERATE
(8) RELEASE

(1) BUILD - without devops the cost of consuption of resources was evaluted based on predicted individual usage with fixed hardware allocation.
- but with devops, the usage of cloud, sharing of resources comes into the picture & build is dependent upon the user's need.

(2) CODE - many good practices like widely used git enables the code to be used, which ensure not only writting the code for business but also helps to track changes.

(3) TEST - Testing can be done by automation which deserves the time for testing and so that time to deploy the code of production can be reduced.

(4) PLAN - devops use agile methodology if plan the development unplanned works always reduces productivity.

(5) MONITOR - It is useful in tracking the system accurately so that health of application also can be checked.

(6) DEPLOY - A cloud management platform enables users to capture accure insights and view the optimization scenarion

(7) OPERATE - The team's operate in a collaborative way where both the teams participate actively throughout the service lifecycle.

(8) RELEASE - most of process involved in release management commonly specify to ~~do~~ the deployment in the production environment.