Assignment-2 Software Testing & Automation

__/__/____

8. Evaluate the effectiveness of statement coverage and branch coverage as metrices for structural testing. Which one provide a more comprehensive assessment and why?

⇒ Statement Coverage ensures that every executable statement in the program is executed at least once during testing. It is simple to implement and helps detect dead code. However, it does not guarantee that all logical decisions are tested.

Branch coverage, also known as decision coverage, ensures that each possible outcome of every decision point is executed. It is more through than statement coverage, as it also covers control flow path that may not covers control flow path that may not be reached by statement coverage alone.

While statement coverage is useful for basic structural validation, branch coverage is more effective in detecting logical errors and ensuring full decision testing.

9. Propose a strategy for combining path testing with data flow testing to acheive a more robust structutral test suite. Justify your approach.

⇒ Strategy :-

(i) Construct the Control Flow Graph (CFG) – Begin by modeling the program using a control flow graph to visualize all possible execution paths.

(ii) Path Testing – Determine the independent paths using cyclomatic complexity.

(iii) Analyze Define-Use (DU) Chains – Identify variables that are defined and later used.

(iv) Overlay DU Chains on Paths – Ensure the selected basis paths also include all relevant data flows.

__/__/____

(v) ~~Prioti~~: Prioritize Critical Paths — Give priority to paths that involve complex decisions or variable usage patterns.

10. Critique the limitations of purely structural testing approach When might they fail to uncover defects, and other testing techniques could complement them?

⇒ Structural testing focuses on the internal code structure, not on functional correctness or user expectations. It may fail to detect missing functionality, usability issues, or requirements mismatches. for example, if a function is absent but the struct is correct, structural testing won't catch it. Complementary techniques like functional testing, exploratory testing, and user acceptance testing (UAT) help identify such defects.

11. Assess the trade-offs b/w performing full regression testing and employing selective regression test selection technique. In what situations would each be more appropriate?

⇒ Full regression testing ensures through re-validation but is time-consuming and resource-intensive, making it suitable when major changes occur or in safety-critical systems. Selective regression test selection targets only impacted areas, saving time but risking missed defects if selection is imperfect. It is ideal in agile enviorments or when time is limited. The choice depends on risk, impact, and project constraints.

12. Differentiate b/w code-based and model-based techniques for regression test selection. Elaborate on a scenario where one would be prefered over the other.

⇒ Code-based techniques select test cases based on changes in the source code, focusing on modified functions or classes. Model-based techniques use design model to trace changes and select affected tests.

13. Design a basic framework for evaluating the overall effectiveness of structural testing effort, considering factors beyond just test coverage.

⇒ A basic evaluation framework should include the following key components :

(i) Test Coverage Metrices – Evaluate statement, branch, and path coverage.
(ii) Defect Detection Rate – Measure number and saverity of defects found.
(iii) Code Complexity – Use metrices like cyclomatic comptexity to assess test adequacy.
(iv) Execution Efficiecy – Track test run time and resource usage.
(v) Maintainability – Assess ease of updating tests when code changes.

This framework provides a balanced assessment of structural testing effectiveness.

14. Justify the need for test case optimization in regression testing. Suggest two common techniques for achieving this optimization.

⇒ Test case optimization is crucial to reduce execution time and cost without compromising fault detection. As the regression suit grows, running all test cases becomes impractical. Two common techniques are :

(i) Test Case Prioritization – Execute tests with the highest risk or recent changes first
(ii) Test case Minimization – Remove redundant test cases that do not contribute new information.

These ensures efficient, targeted, and faster regression cycles.

10/10

28/7