

Categories of Test Techniques and Their Characteristics

Black-box testing techniques	Description
Equivalence partitions	<ul style="list-style-type: none"> - Partitions are called equivalence classes - An equivalence partition containing valid values is called a "valid equivalence partition". - An equivalence partition containing invalid values is called "invalid equivalence partition". - Each value must belong to one and only one equivalence partition - Any partition can be divided into sub-partitions if necessary - Coverage is measured as the number of equivalence partitions tested by at least one value, divided by the total number of identified equivalence partitions, usually expressed as a percentage. - When invalid equivalence partitions are used in test cases, they must be tested individually
Limit value analysis	<ul style="list-style-type: none"> - limits value analysis is an extension of equivalence partitions, but can only be used when the partition is ordered, composed of numerical or sequential data. The minimum and maximum values of a partition are its boundary values. - Some variants of this technique identify three boundary values per boundary: the values below, on and just above the boundary. - Boundary value analysis can be applied to all levels of testing. This technique is typically used to test requirements that require a series of numerical values - The boundary coverage of a partition is measured as the number of boundary values tested, divided by the total number of boundary values identified, usually expressed as a percentage.
Decision table testing	<ul style="list-style-type: none"> - Each column corresponds to a decision rule that defines a unique combination of conditions that results in the execution of the actions associated with that rule. - Condition and action values are usually displayed as Boolean values (true or false) or discrete values (e.g. red, green, blue), but can also be numbers or ranges of numbers - These different types of conditions and actions can be grouped in the same table. - The table can be reduced by removing columns containing impossible combinations of conditions, columns containing combinations of possible but infeasible conditions, and columns that test combinations of conditions that do not affect the outcome. - The common minimum coverage for decision table testing is to have at least one test case per decision rule in the table. - Coverage is measured as the number of decision rules tested by at least one test case, divided by the number of decision rules, usually expressed as a percentage.
Testing state transitions	<ul style="list-style-type: none"> - A state transition diagram shows the possible states of the software, as well as how the software enters, exits, and evolves between these states. - A transition is triggered by an event - The change of state can lead to an action of the software - A state transition table shows all valid and potentially invalid transitions between states - State transition charts normally show only valid transitions and exclude invalid transitions. - Tests can be designed to cover a sequence of multiple states, to exercise all states, to exercise all transitions, to exercise specific sequences of transitions, or to test invalid transitions. - Coverage is typically measured as the number of identified states or transitions tested, divided by the total number of identified states or transitions in the test object, normally expressed as a percentage.
Use case testing	<ul style="list-style-type: none"> - Use cases are associated with actors and subjects - Each use case specifies a behavior that a subject can perform in collaboration with one or more actors - A use case can be described by interactions and activities, as well as by preconditions, postconditions and natural language - A use case can include possible variations of its basic behavior, including exceptional behavior and error handling - Tests are designed to exercise the defined behaviors - Coverage can be measured by the percentage of use case behaviors tested divided by the total number of use case behaviors, usually expressed as a percentage.

White-Box testing techniques	Description
Test and coverage of the instructions	<ul style="list-style-type: none"> - Tests exercise potential executable instructions in the code. - The percentage of executable instructions that have been executed by a test suite.
Decision Testing and Coverage	<ul style="list-style-type: none"> - Decision testing exercises decisions in code and tests the code that is executed based on the results of the decision. - Coverage of decision results.

Experience-based Testing techniques	Description
Error estimation	<p>Error estimation is a technique used to anticipate errors, defects, and failures, based on the tester's knowledge, including:</p> <ul style="list-style-type: none"> - How the application has worked in the past - What types of errors developers tend to make - What failures have occurred in other applications - A methodical approach to the error estimation technique is to create a list of possible errors, defects, and failures, and design tests that will expose these failures and the defects that caused them.
Exploratory tests	<ul style="list-style-type: none"> - In exploratory testing, informal (non-predefined) tests are designed, executed, recorded and evaluated dynamically during test execution. - The results of the tests are used to learn more about the component or system, and to create tests for parts that might need more testing. - Exploratory testing is sometimes done using test sessions to structure the activity. - Exploratory testing is most useful when there are few or inadequate specifications or significant time constraints on testing.
Tests based on checklists	<ul style="list-style-type: none"> - In checklist-based testing, testers design, implement, and execute tests to cover the test conditions listed in a checklist. - During testing, testers create a new checklist or add to an existing checklist, but testers can also use an existing checklist without modification. - Checklists can be created to support different types of tests, including functional and non-functional tests. - In the absence of detailed test cases, checklist-based tests can provide guidelines and a degree of consistency. - Because they are high-level lists, there is likely to be some variability in the actual tests, which could result in greater test coverage but lower reproducibility.