**Static Testing vs Sanity Testing**

| **Feature** | **Static Testing** | **Sanity Testing** |
| --- | --- | --- |
| **Definition** | Testing without executing code | quick test to check specific functionality after changes |
| **Purpose** | Find defects early without running code | Verify minor bug fixes or changes work as expected |
| **Execution Timing** | Early in development | After receiving a new build or minor changes |

## **Retest vs Regression Testing**

| **Feature** | **Retest** | **Regression Testing** |
| --- | --- | --- |
| **Definition** | Verify that specific defect fixes are successful | Ensure new changes do not break existing functionality |
| **Test Cases** | Executes only failed test cases | Executes both old and new test cases |

## **Acceptance Criteria vs Acceptance Testing**

| **Feature** | **Acceptance Criteria** | **Acceptance Testing** |
| --- | --- | --- |
| **Definition** | Preconditions and requirements that must be met | Testing to verify software meets acceptance criteria |
| **Purpose** | Define when software is acceptable for delivery | Verify software meets defined acceptance requirements |
| **Focus** | Set by stakeholders before testing | Actual validation against criteria |

## **Decision Table Testing vs Decision Testing**

| **Feature** | **Decision Table Testing** | **Decision Testing** |
| --- | --- | --- |
| **Definition** | Testing based on decision tables which map inputs to outputs | Testing based on decision logic or conditions in code |
| **Purpose** | Ensures all possible input combinations are tested | Focuses on individual decision points and branches |
| **Focus** | Covers combinations of inputs systematically | Covers decisions or branches one at a time |

## 

## **State Transition vs Statement Coverage**

| **Feature** | **State Transition Testing** | **Statement Coverage** |
| --- | --- | --- |
| **Definition** | Tests software behavior through various states & transitions | Measures percentage of executed statements in code |
| **Purpose** | Validate that state changes occur correctly | Validate that code statements are executed during testing |
| **Focus** | Behavior-based testing | Code coverage metric |

## **Data Testing vs Database Testing**

| **Feature** | **Data Testing** | **Database Testing** |
| --- | --- | --- |
| **Definition** | Testing the accuracy and validity of data | Testing the database functionalities and structures |
| **Purpose** | Data correctness, transformation, and integrity | Database performance, query execution, and data storage |

## **Test Planning vs Test Strategy**

| **Feature** | **Test Planning** | **Test Strategy** |
| --- | --- | --- |
| **Definition** | Process of defining what to test, when, who, and how | Overall approach and objectives for testing across projects |
| **Purpose** | Detailed, project-specific tasks and schedules | High-level guideline and principles for testing |

## **Unit Testing vs Component Testing**

| **Feature** | **Unit Testing** | **Component Testing** |
| --- | --- | --- |
| **Definition** | Testing individual code units | Testing a group of integrated units as a single component |
| **Purpose** | Focus on smallest testable parts of application | Focus on combined behavior of multiple units together |
| **Focus** | Behavior-based testing | Code coverage metric |