# SNS COLLEGE OF TECHNOLOGY

Kurumbapalayam (Po), Coimbatore – 641 035

**An Autonomous Institution**

Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## COURSE NAME: 23ITO201- Software Testing (OPEN ELECTIVE )

## III  YEAR / VI  SEMESTER

1

## Unit 2  - TEST CASE DESIGN STRATEGIES

## Topic  :  TEST CASE DESIGN STRATEGIES

# EMPATHY IN TEST CASE DESIGN STRATEGIES

**Equivalence Partitioning**

Groups data to test representative values

**Boundary Value Analysis**

Focuses on edge cases and limits

**Decision Table Testing**

Maps complex conditions to actions

**State Transition Testing**

Verifies system state changes

**Error Guessing**

Uses intuition to predict defects

# Why We Can't Test Everything?

Understanding the constraints of software testing and the need for strategic approaches.

### THE CHALLENGE
## Testing Constraints

- 🕐 Time is limited
- 👥 Resources are limited
- ∞ Inputs can be infinite

### THE SOLUTION
## Design Strategies

- Reduce number of test cases
- Increase defect detection
- Test smartly and efficiently

# What is a Test Case Design Strategy?

A **systematic approach** used by testers to design effective test cases that uncover defects with **minimum effort**.

### Maximize Coverage

Ensure the widest possible test coverage with the smallest set of test cases.

### High-Yield Areas

Focus testing efforts on parts of the software where defects are most likely to occur.

### Uncover Defects

Systematically reveal hidden bugs that random testing might miss.

# Equivalence Partitioning (EP)

Efficiently testing groups of input data

## DEFINITION

Divides input data into **valid** and **invalid** groups (partitions) and tests **one value** from each group.

## WHY USE EP?

- Reduces number of test cases
- Avoids redundant testing
- Improves test coverage

## Example Scenario

Condition: Age must be between **18** and **60**

| PARTITION TYPE | TEST DATA (INPUT) |
|---|---|
| ✅ Valid Group (Between 18-60) | 25 |
| ❌ Invalid Group (Below 18) | 17 |
| ❌ Invalid Group (Above 60) | 61 |

# Boundary Value Analysis (BVA)

Focusing on the edge cases where defects hide

### DEFINITION

A technique that focuses on testing **boundary (edge) values** rather than values in the center, as this is where defects commonly occur.

### WHY BVA IS IMPORTANT?

🐞 Most defects occur at boundaries

</> Developers often mistake limits

### 🧪 Example Scenario

🔑 Condition: Password length: **8 to 16** characters

| TEST CASE | VALUE | RESULT |
|---|---|---|
| Min Boundary | 8 | Valid |
| Max Boundary | 16 | Valid |
| Just below min | 7 | Invalid |
| Just above max | 17 | Invalid |

💡 **KEY INSIGHT**
Always test the four critical points: **Minimum**, **Maximum**, **Just below min**, and **Just above max** to catch off-by-one errors.

# Decision Table Testing

Managing complex business logic and combinations

## DEFINITION

A technique used when the system behavior depends on a **combination of multiple conditions** or inputs.

## WHY USE THIS STRATEGY?

- Handles complex business rules
- Avoids missing combinations
- Improves requirement clarity

## Logic Matrix Example

Scenario: Discount Eligibility

| Conditions / Rules | Case 1 | Case 2 | Case 3 |
|---|---|---|---|
| Premium User? | Yes | Yes | No |
| Amount > ₹500? | Yes | No | Yes |
| Action: Discount | ✓ YES | ✗ NO | ✗ NO |

*Showing 3 representative test cases from the full matrix

### KEY INSIGHT

Each column (or row depending on orientation) in a decision table represents **one unique test case**, ensuring all logical combinations are verified.

# State Transition Testing

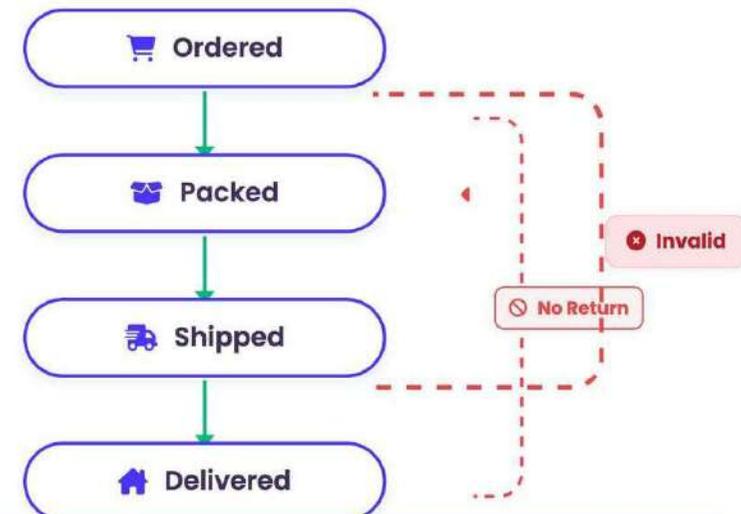Validating system behavior across different stages

### DEFINITION

A technique used to verify how the system behaves when transitioning from **one state to another** based on input events.

### WHY USE THIS STRATEGY?

↻ Applications change states

⊘ Invalid transitions cause defects

## 📋 Order Processing Flow

● Valid  ● Invalid

🛒 Ordered

📦 Packed

⊗ Invalid

🚚 Shipped

🚫 No Return

🏠 Delivered

💡 **KEY INSIGHT**

Effective testing requires verifying both **valid state transitions** (success path) and **invalid state transitions** (negative testing).

# Error Guessing

Leveraging tester intuition and past experience

## DEFINITION

An experience-based testing technique where testers **predict possible defects** based on their knowledge of how the application works and where failures typically occur.

### The Human Element

"Breaking the system where logic says it should work"

## ⚠ Common Error Scenarios

**Empty Input Fields**
Leaving mandatory fields blank to check validation.

**Invalid Data Types**
Entering text in number fields or vice versa.

**Special Characters**
Using characters like @, #, $, % in names or addresses.

**Multiple Clicks**
Rapidly clicking 'Submit' to trigger duplicate transactions.

**Network Interruption**
Disconnecting internet during a file upload or transaction.

# Comparison Summary

Quick reference guide to selecting the right strategy

| ICON | STRATEGY | PRIMARY FOCUS AREA | KEY CHARACTERISTIC |
|------|----------|--------------------|--------------------|
| | **Equivalence Partitioning** EP | Input Groups | Tests representative values from valid and invalid data partitions. |
| | **Boundary Value Analysis** BVA | Edge Values | Focuses on min, max, and edge boundaries where defects hide. |
| | **Decision Table Testing** DT | Conditions & Actions | Validates logic when system behavior depends on multiple rule combinations. |
| | **State Transition Testing** STT | System States | Checks valid and invalid paths between different system states. |
| | **Error Guessing** EG | Human Behavior | Uses intuition and experience to predict likely error scenarios. |

# ACTIVITY

**Problem Scenario (Given to Students)**
**Online Registration Form**

• Username: 5–15 characters

• Password: Minimum 8 characters, must include a number

• Age: 18–60

• Email: Valid email format

• Submit button enabled only after valid inputs

**Task 1: Identify Test Case Design Strategies (10 mins)**
Each group should:
1. List applicable **test case design strategies**
2. Map each strategy to the given scenario
**Expected Strategies**
• Equivalence Class Partitioning (ECP)
• Boundary Value Analysis (BVA)
• Decision Table Testing
• State Transition Testing
• Error Guessing

**Boundary Value Analysis is most effective when:**

A. Input conditions are random

B. System has complex workflows

C. Input values have limits or ranges

D. Output depends on time

✅ **Answer:** C

**4. Which strategy is best suited for testing systems with different states?**

A. Error Guessing

B. State Transition Testing

C. Boundary Value Analysis

D. Equivalence Class Partitioning

✅ **Answer:** B

# Advantages & Conclusion

Why structured test design matters for delivery

## 👍 Key Advantages

**01** 🏋️ **Reduces overall testing effort**

**02** 🐛 **Improves defect detection rate**

**03** ⏱️ **Saves valuable time and project costs**

**04** ☂️ **Increases effective test coverage**

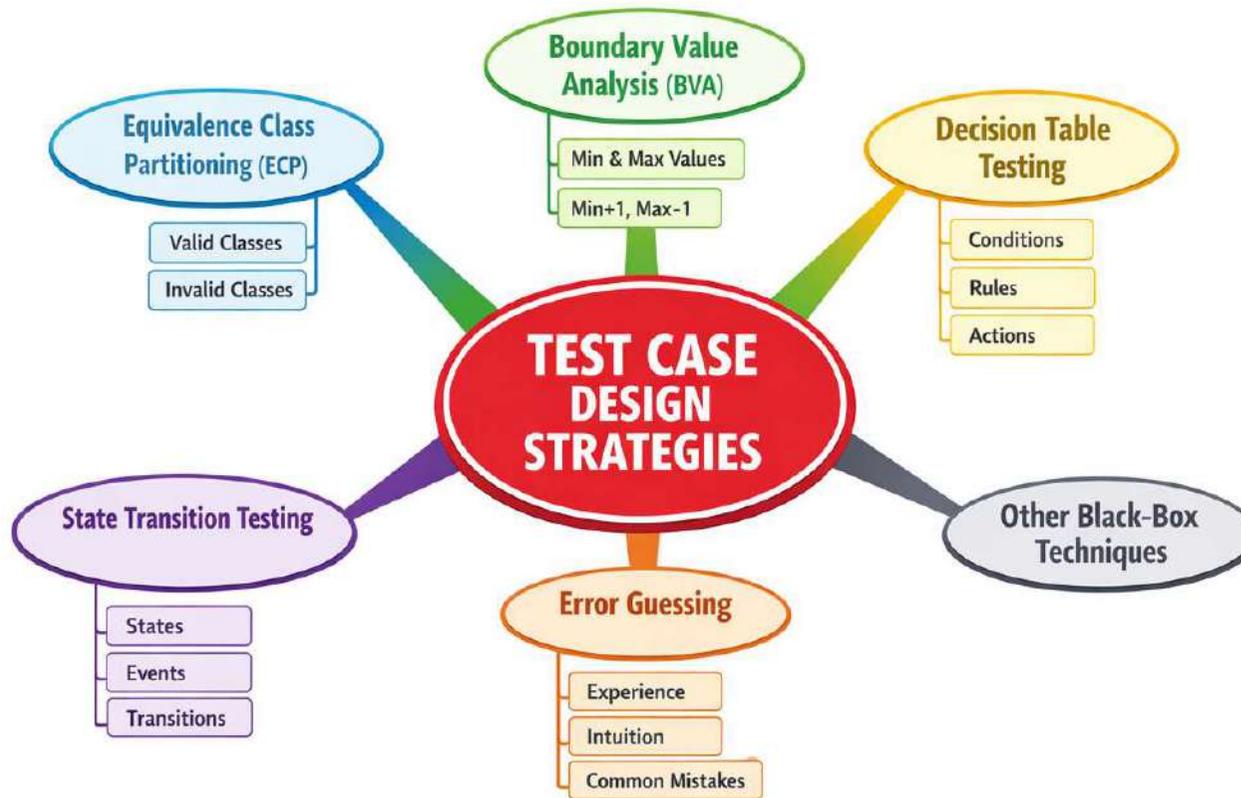**05** ⭐ **Produces high-quality, reliable software**

## " Final Takeaway

Test Case Design Strategies transform testing from a random activity into a systematic engineering discipline. They help testers prioritize scenarios that matter most, ensuring efficient testing cycles without compromising on quality.

### "TEST SMARTLY, NOT EVERYTHING"

# Mind map

| TEXT BOOKS | |
|---|---|
| 1. | Srinivasan Desikan and Gopalaswamy Ramesh, ―Software Testing – Principles and Practices, Pearson Education, 2006. |
| REFERENCES | |
| 1. | Ilene Burnstein, ―Practical Software Testing, Springer International Edition, 2003 |
| 2. | Edward Kit, Software Testing in the Real World – Improving the Process‖, Pearson Education, 1995 |
| 3. | Boris Beizer, Software Testing Techniques – 2nd Edition, Van Nostrand Reinhold, New York, 1990. |
| 4. | Aditya P. Mathur, ―Foundations of Software Testing _ Fundamental Algorithms and Techniques, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008. |

# Thank You