Strategic Test Design Techniques

Optimising Software Quality through Systematic Coverage and Defect Detection

Corporate Training Series | Quality Assurance Excellence

Testleaf
Always Ahead

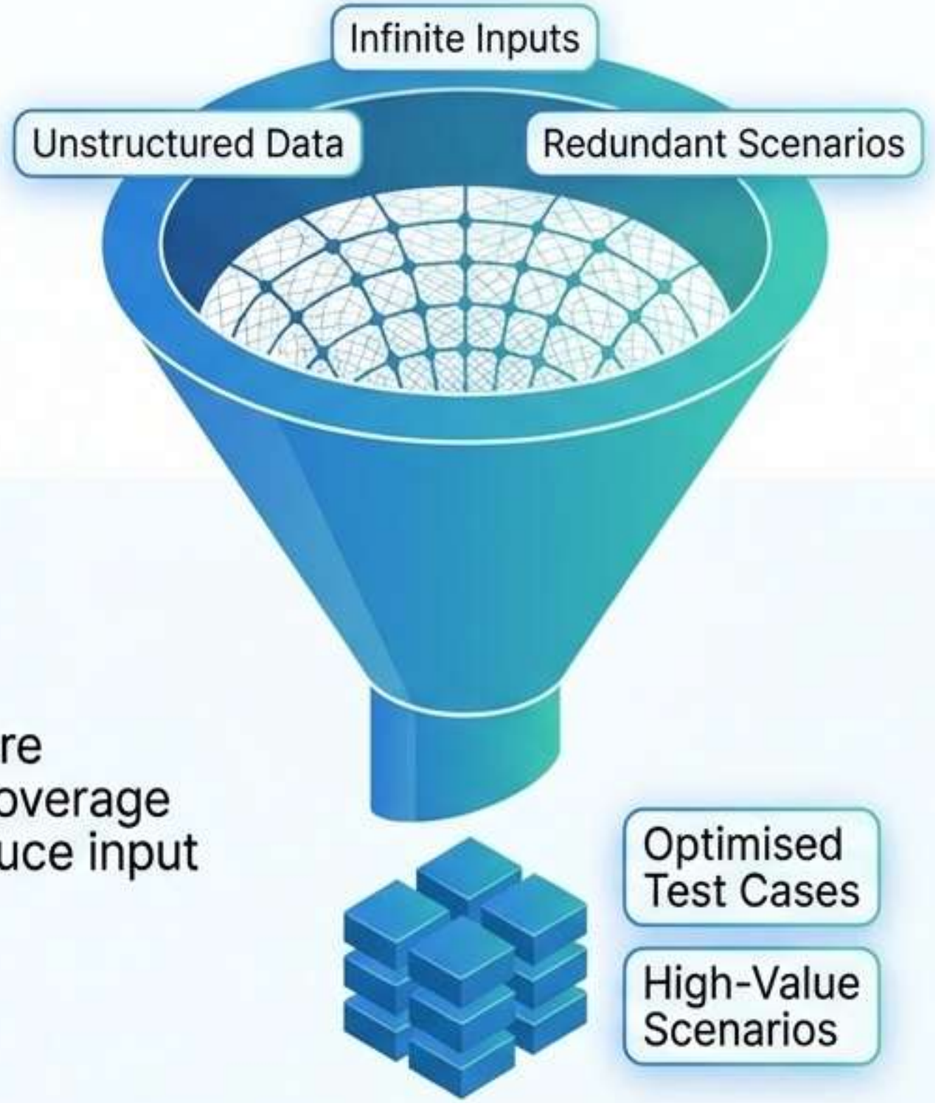
The Necessity of Test Design

The Golden Rule

Testleaf
Always Ahead

The Challenge:

Exhaustive testing is theoretically impossible due to infinite input possibilities.



The Solution:

Test Design Techniques scientifically reduce the input domain while maintaining coverage.

What:
Systematic methods to create effective test cases.

Why:
To ensure smart coverage and reduce input size.

How:
Identifying specific scenarios to verify requirements.

The Testing Landscape: Static vs. Dynamic

Testleaf
Always Ahead

Static Techniques

Definition: Testing documentation without execution.

Goal: Verification of design and requirements.

Types: Reviews (Walkthrough, Inspection), Static Analysis.

Dynamic Techniques

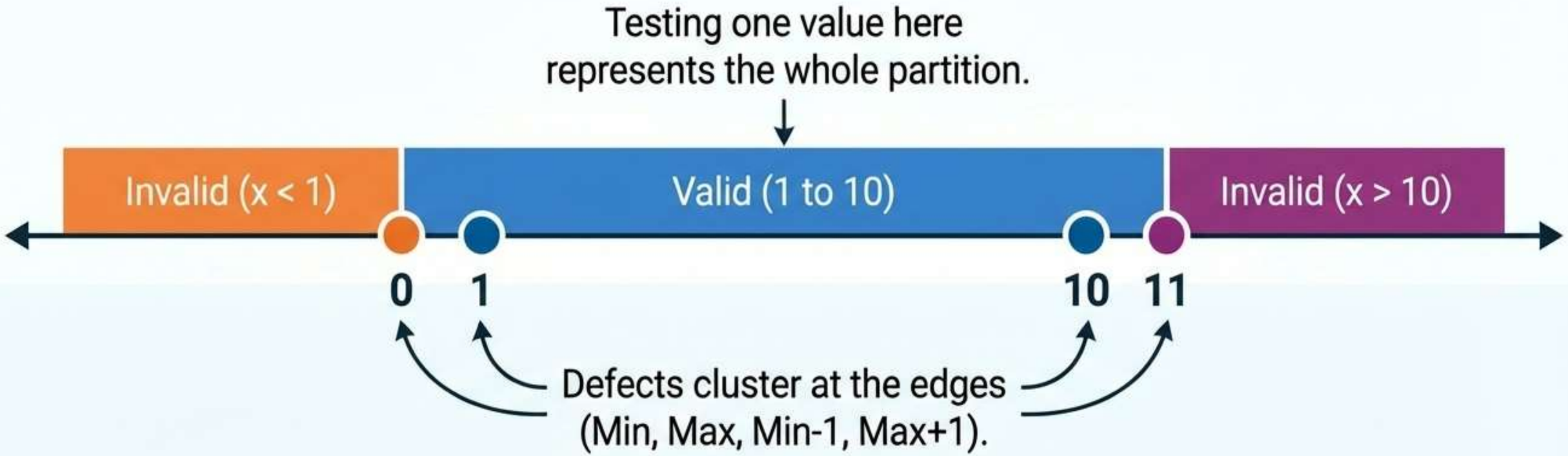
Definition: Testing by executing the system under test (SUT).

Goal: Validation of functional output.

Method: Input data -> Execution -> Verification.

Data Enumeration Techniques

Specification-Based (Black Box)



ECP: Divides data into valid/invalid partitions to reduce volume.

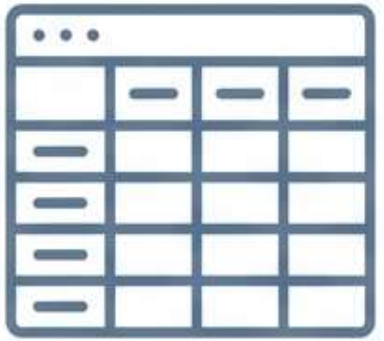
BVA: Focuses testing on the high-risk edges of those partitions.

Test Enumeration Techniques

Testleaf
Always Ahead

Validating Logic and Flow

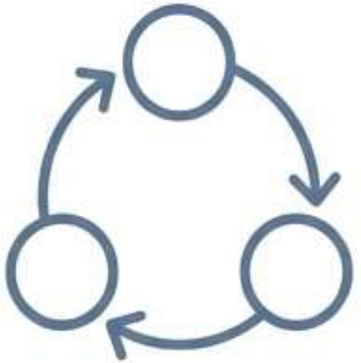
Decision Table Testing



Captures behaviour where output depends on complex combinations of inputs.

Maps Conditions (Inputs) to Actions (Outputs).

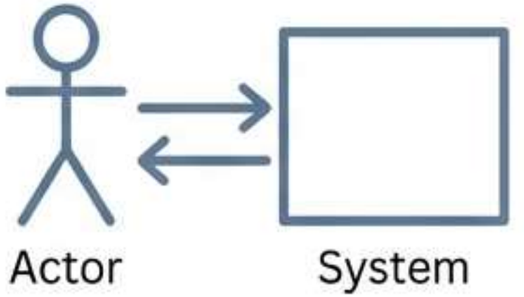
State Transition Testing



Analyses transitions between states based on events.

Essential for finite state systems (e.g., ATM, Order Processing).

Use Case Testing



Validates end-to-end flows based on user behaviours.

Focuses on 'Happy Paths' and error handling.

Structure-Based & Experience-Based Techniques

Testleaf
Always Ahead

Structure-Based (White Box)

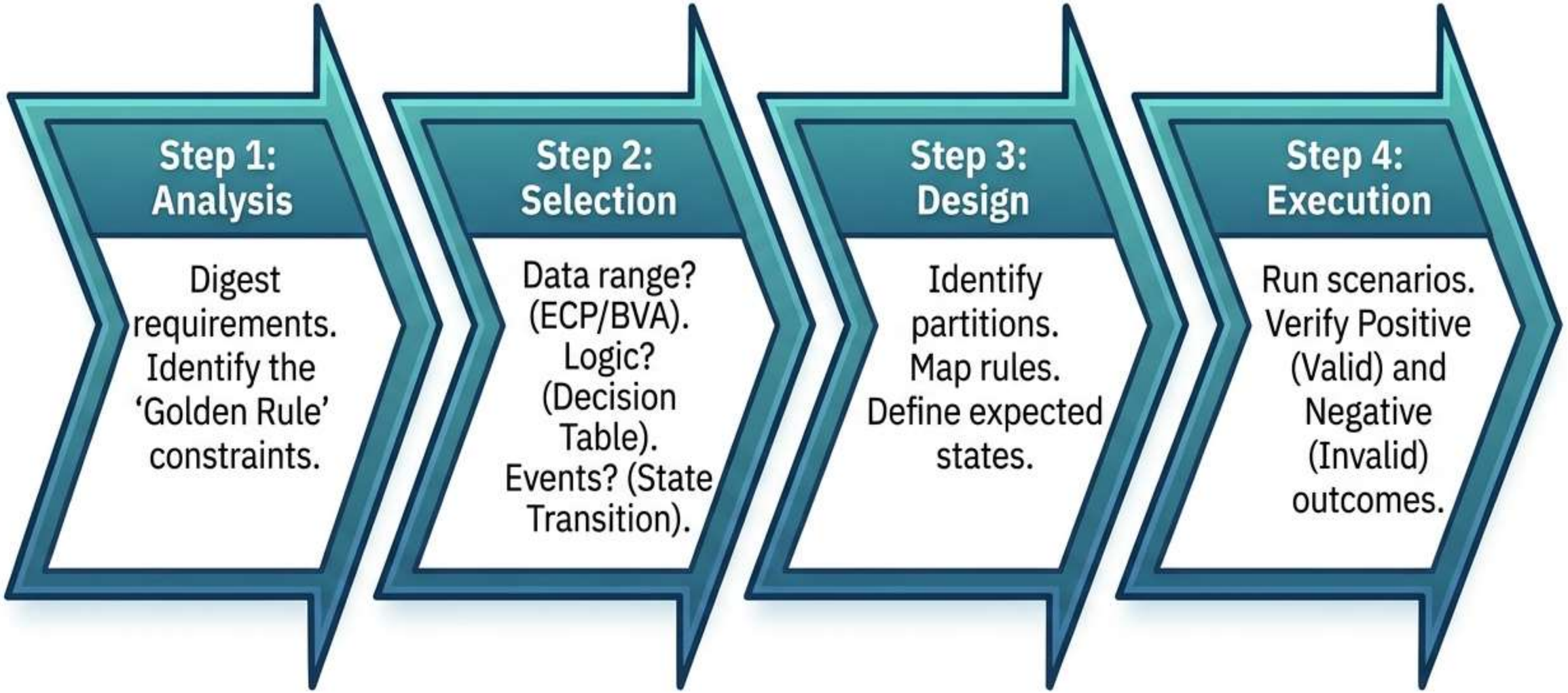
- Tests internal logic and code structure.
- Requires knowledge of implementation (Statement, Branch coverage).

Experience-Based

- Exploratory Testing: Simultaneous learning and execution. Relies on domain expertise to find high-severity bugs.
- Error Guessing: Unstructured testing based on intuition and past experience with similar failures.

Strategic Test Design Workflow

Testleaf
Always Ahead



Applied Techniques in Business Scenarios

Testleaf
Always Ahead

Scenario	Technique	Application
Age Verification (18–60)	BVA & ECP	Test Boundaries: 17, 18, 19 and 59, 60, 61. Test Partition: 30.
Loan Approval Logic	Decision Table	Inputs: Credit Score > 700 AND Income > £50k. Action: Approve/Reject based on T/F combos.
ATM PIN Entry	State Transition	States: Wait -> 1st Fail -> 2nd Fail -> Block Card. Event: Incorrect PIN.

Industry Best Practices

Testleaf
Always Ahead

- **Combine Techniques:** Use ECP for data fields and Use Cases for workflows to ensure robust coverage.
- **Prioritise Boundaries:** Defects historically cluster at the edges. Always include Min/Max +/- 1.
- **Review Early:** Apply Static techniques to requirements before test design begins.
- **Simplify Logic:** If a Decision Table is too large, break it into smaller logical subsets.
- **Leverage Domain Knowledge:** Allocate time for Exploratory Testing to find 'hidden' edge cases.

Session Summary

The Golden Rule (Scope)

Static vs. Dynamic

Data Enumeration (ECP/BVA)

Test Enumeration (Logic)

Real-World Application

You are now equipped to systematically reduce input domains and detect defects with precision.