ILLINOIS TECH | College of Computing

ITMD 536 Software Testing & Maintenance

Chapter 4
Test Design Techniques

ILLINOIS TECH | College of Computing

Objectives

- What is Test Design Technique?
- What is the test case, and test procedure?
- What is test condition, test case and test procedure?
- What is traceability and expected results?
- What black-box and white-box testing?

4.1. Test Design Techniques

- By **design** we mean to create a plan for how to implement an idea and **technique** is a method or way for performing a task. So, **Test Design** is creating a set of inputs for given software that will provide a set of expected outputs.
- Procedure used to derive and/or select test cases.

4.1 The Test Development Process

- Before a test is executed we need to know what is our end goal what are we trying to test, the inputs, the results that should be produced by those inputs and how we can get ready to run the test.
- Review test conditions, test cases, and test procedures (or scripts) follow the Test Documentation Standards (IEEE 829)
- Test cases are document in the test case specification.

ILLINOIS TECH | College of Computing

4.1 The Test Development Process

- **Test case specification:** A document specifying a set of test cases (objective, inputs, test actions, expected results, and execution preconditions) for a test item.
- Test procedures are documented in a Test Procedure Specification (also known as a test script or a manual test script).

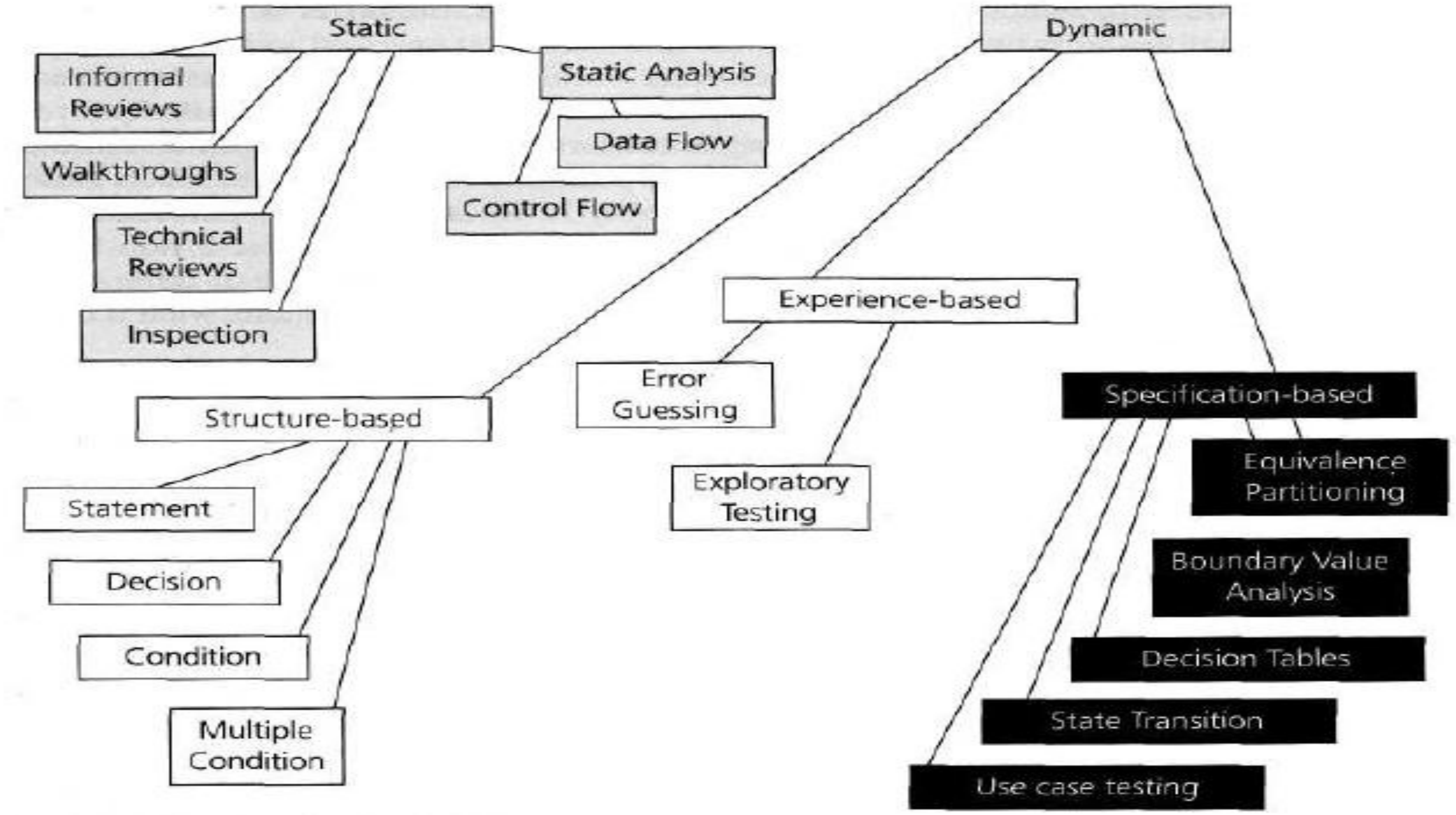


FIGURE 4.1 Testing techniques

4.1.1 Formality of test documentation

- Formal testing would have extensive documentation and well controlled. It would include the detail of exact and specific input and expected outcome of the test.
- The thoroughness of test documentation also depends on time constraints.

4.1.2 Test analysis: identifying test conditions

- Test analysis is the process of looking at something that can be used to derive test information.
- This basis of the tests is called the "test basis". This could be a system requirement, a technical requirement or the code itself (for structural testing) or a business process.
- Test basis provides us the information as to what could be tested – these are the test conditions.
- A test condition is simply something that we could test.
- List of test conditions are the decision outcomes (True and False).

ILLINOIS TECH | College of Computing

4.1.2 Test analysis: identifying test conditions

- Testing technique helps us select a good set of test from the total number of all possible tests for a given system.
- Tests might based on risk, models of the system, likely failures, compliance requirements, expert advice or heuristics (I find).
- Test conditions should be linked back to their sources in the test basis and it’s called traceability.
- **Traceability:** The ability to identify related items in documentation and software, such as requirements with associated tests.

ILLINOIS TECH | College of Computing

4.1.2 Test analysis: identifying test conditions

- Traceability is important because:
- Verify function or feature changes
- How many tests will be affected with any change in requirement.
- Before new release we need to have the tests results with specified requirements that are tested and passed.

4.1.2 Test analysis: identifying test conditions

- **Horizontal traceability:** The tracing of requirements for a test level through the layers of test documentation (e.g. test plan, test design specification, test case specification and test procedure specification or test script).
- **Vertical traceability:** The tracing of requirements through the layers of development documentation to components.
- Test conditions are documented in the IEEE 829 called a Test Design Specification.

IEEE 829 Standard: Test Design Specification Template

- Test design specification identifier
- Features to be tested
- Approach refinements
- Test identification
- Feature pass/fail criteria

4.1.3 Test design: specifying test cases

- A test case needs to have input values, of course, but just having some values to input to the system is not a test!
- You need to know what the system is supposed to do with the inputs, you can't tell whether your test has passed or failed
- The test case also specifies the environment and other things that must be in place before the test can be run (the preconditions) and any things that should apply after the test completes (the post conditions).

IEEE 829 Standard: Test Case Specification Template

- Test case specification identifier
- Test items
- Input specifications
- Output specifications
- Environmental needs
- Special procedural requirements
- Interface dependencies

4.1.4 Test implementation: specifying test procedures or scripts

- Test cases need to be grouped and executed in sequential order to get the valuable results.
- The document that describes the steps to be taken in running a set of tests is called a test procedure in IEEE 829 and is also referred to as a test script.
- **Test script** Commonly used to refer to a test procedure specification, especially an automated one.

4.1.4 Test implementation: specifying test procedures or scripts

- **Test Execution schedule** A scheme for the execution of test procedures. The test procedures are included in the test execution schedule in their context and in the order in which they are to be executed.
- A regression script may always be the first to be run when a new release of the software arrives, as a smoke test or sanity check.

4.1.4 Test implementation: specifying test procedures or scripts

- Writing the test procedure is another opportunity to prioritize the tests, to ensure that the best testing is done in the time available. A good rule of thumb is 'Find the scary stuff first'.

IEEE 829 Standard: Test Procedure Specification Template

- Test procedure specification identifier
- Purpose
- Special requirements
- Procedure steps

4.2.1 Introduction

- There are two main categories of testing:
- Static and
- Dynamic
- **Dynamic** techniques are divided into three categories:
- Specification-based (black-box/behavioral techniques),
- Structure-based (white-box or structural techniques),
- Experienced-based

4.2.2 Static testing techniques

- **Static** testing techniques do not execute the code being examined and are generally used before any tests are executed on the software.
- They could be called non-execution techniques.
- Tests the source code.

ILLINOIS TECH | College of Computing

4.2.3 Specification-based (black-box) testing techniques

- ***Dynamic*** testing will look at the specification-based testing also know as ***black-box*** testing or input/output driven testing.
- ***Black-box*** testing either functional or non-functional, without reference to the internal structure of the component or system.

4.2.4 Structure-based (white-box) testing techniques

- Structure-based testing is also known as ‘***white-box***’ or ‘glass-box’ testing.
- White-box testing is based on an analysis of the internal structure of the component or system.

ILLINOIS TECH | College of Computing

4.2.5 Experience-based testing techniques

- **Experienced-based test design technique**
- Procedure to derive and /or select test cases based on the tester's experience, knowledge and intuition.
- In experience-based techniques people's knowledge, skills and background are a prime contributor to the test conditions and test cases.
- The experience of both technical and business people is important, as they bring different perspective to the test analysis and design process.

4.2.6 Where to apply the different categories of techniques

- Specification-based techniques are appropriate at all levels of testing (component testing through to acceptance testing) where a specification exists.
- Structure-based techniques can also be used at all levels of testing.
- Experienced-based techniques are used to complement specification-based and structure-based techniques, and are also used when there is no specification, or the specification is inadequate or out of date.

ILLINOIS TECH | College of Computing

4.3 Specification-Based or Black-Box Techniques

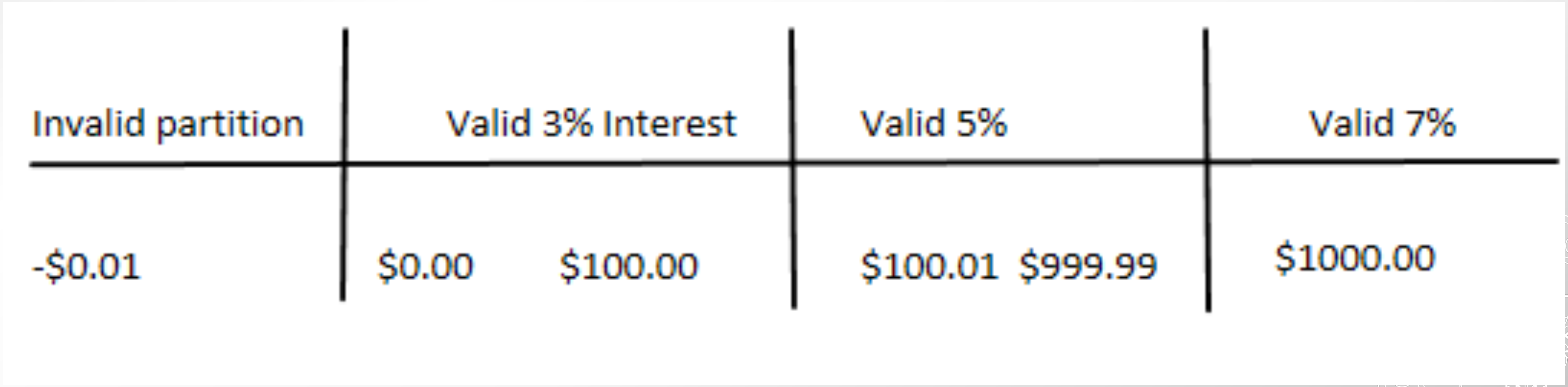
- **Black-box test design technique:** Produces to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.
- There are four specification-based techniques:
- Equivalence partitioning;
- Boundary value analysis;
- Decision tables;
- State transition testing.

4.3.1 Equivalence partitioning and boundary value analysis

- **Equivalence partitioning:** A black box test design technique in which test cases are designed to execute representatives from equivalence partitions. In principle test cases are designed to cover each partition at least once.
- **Equivalence partition** A portion of an input or output domain for which the behavior of a component or system is assumed to be the same, based on the specification.

4.3.1 Equivalence partitioning and boundary value analysis

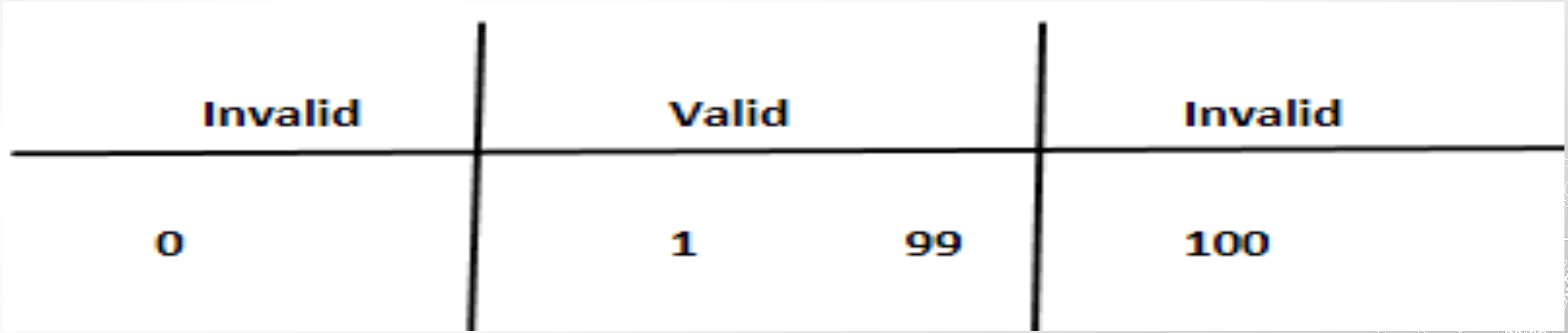
- Bank account with $0-100 has 3% interest and so on.. see example



Invalid partition	Valid 3% Interest	Valid 5%	Valid 7%
-$0.01	$0.00 $100.00	$100.01 $999.99	$1000.00

Boundary Value Analysis

- **Boundary Value Analysis(BVA):** A black box test design technique in which test cases are designed based on boundary values.



Invalid	Valid		Invalid
0	1	99	100

Boundary Value

- **Boundary Value:** An input value or output value which is on the edge of an equivalence partition or at the smallest incremental distance on either side of an edge, for example the minimum or maximum value of a range.

Extending equivalence partitioning and boundary value analysis

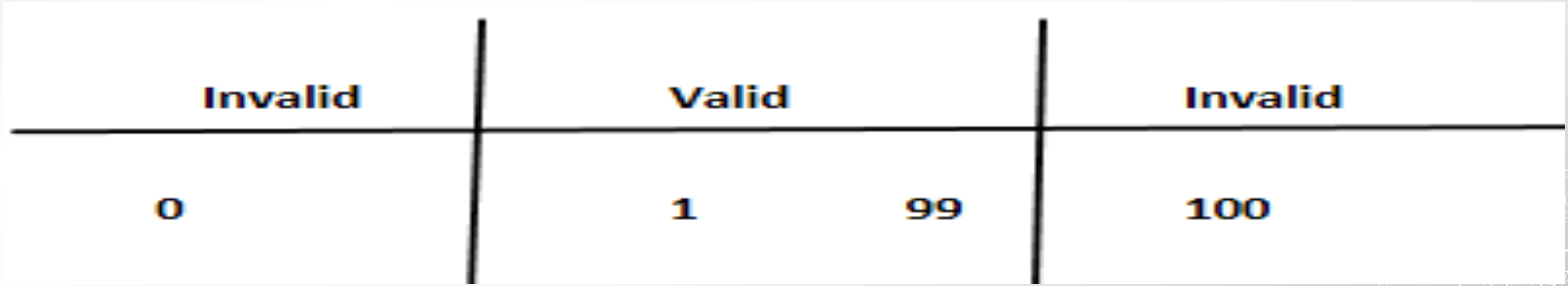
- Equivalence partitions and boundaries
- Digits (0-9) invalid partition containing non-digits
- Number of digits, 3 invalid values 2 and 4 digits
- Range of extension 100 to 699 invalid values 099 - 700

Designing Test Cases

- A set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement.

Why do both equivalence partitioning and boundary value analysis?

- Technical, because every boundary is in some partition, if you did only boundary value analysis you would also have tested every equivalence partition.



Invalid	Valid		Invalid
0	1	99	100

4.3.2 Decision Table Testing

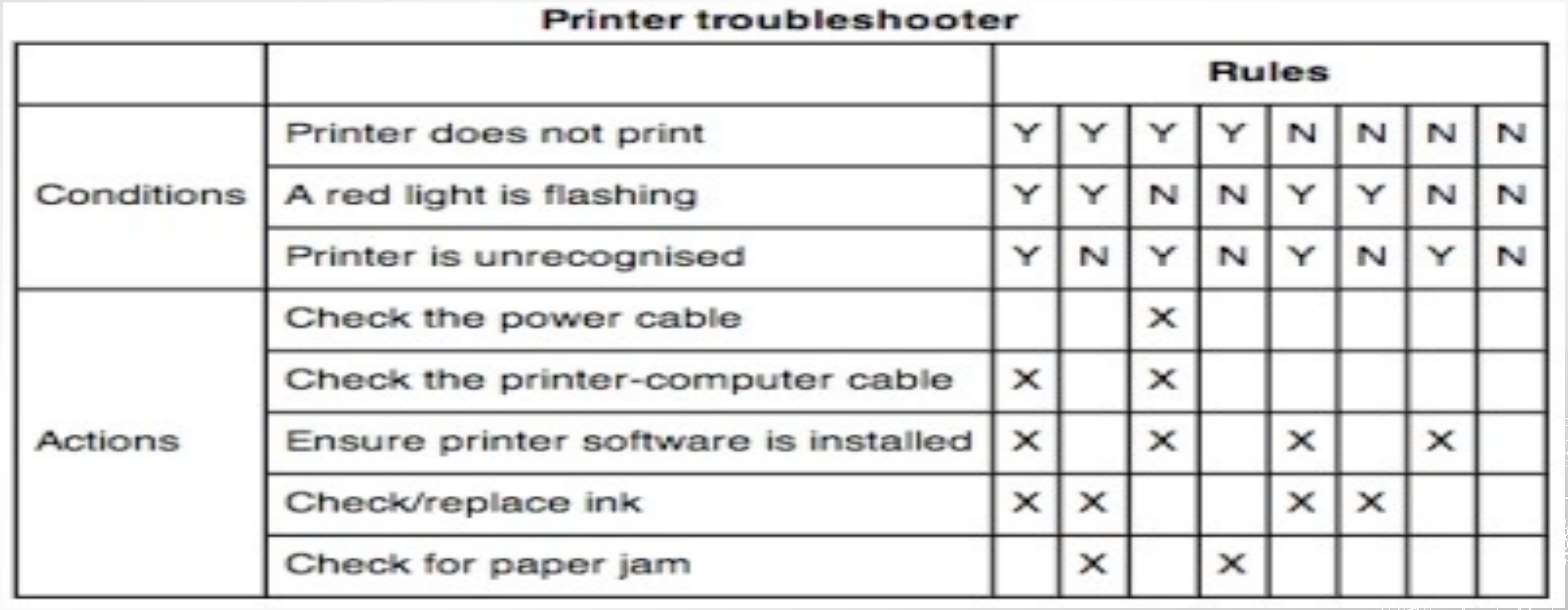
- **Decision Table Testing:** A black box test design technique in which test cases are designed to execute the combinations of inputs and/or stimuli (cases) shown in a decision table.
- Decision Table Testing and Transition Testing are more focused on business logic and business rules.
- **Decision Table:** A table showing combinations of inputs and/or stimuli (causes) with their associated outputs and/or actions (effects),which can be used to design test cases.
- A decision table is a good way to deal with combinations of things (inputs). This technique is also called cause-effect' table.

Using Decision Tables for Test Design

- Simple Cause and Effect Table
- Name
- Focusing Question

Cause	Effect

Using Decision Tables for Test Design



Printer troubleshooter

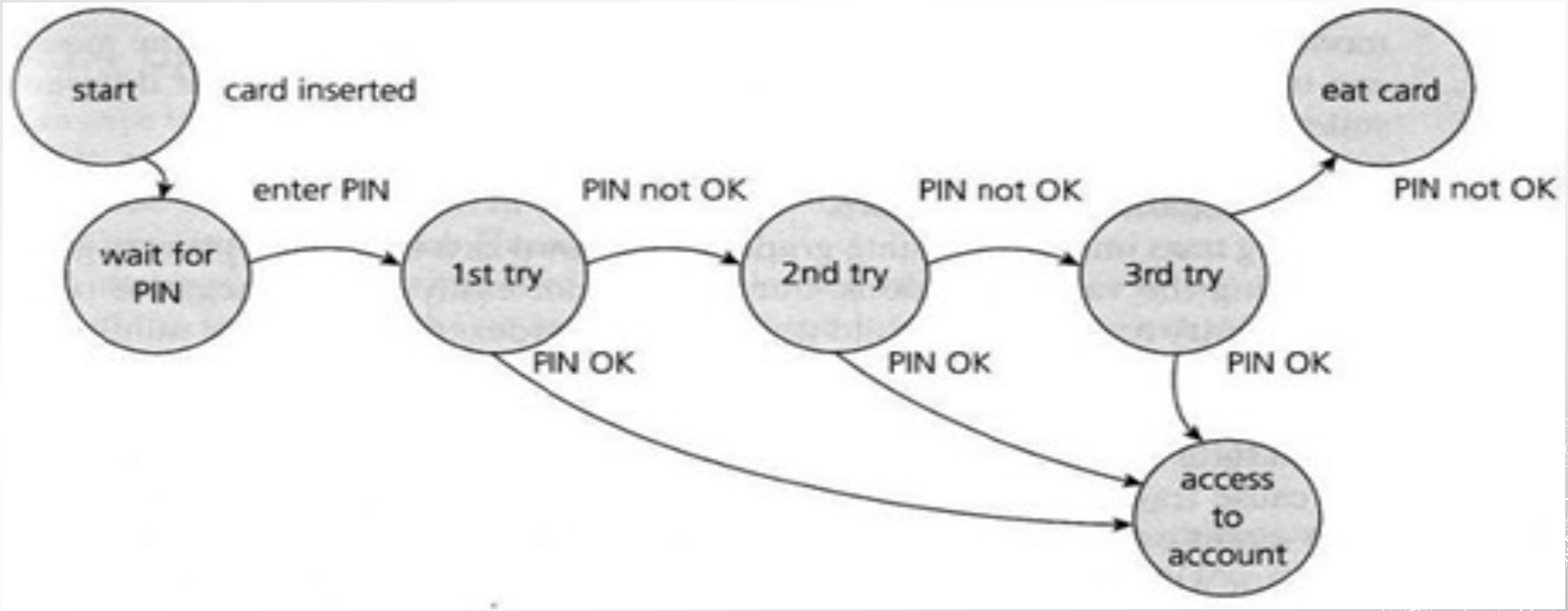
		Rules							
Conditions	Printer does not print	Y	Y	Y	Y	N	N	N	N
	A red light is flashing	Y	Y	N	N	Y	Y	N	N
	Printer is unrecognised	Y	N	Y	N	Y	N	Y	N
Actions	Check the power cable			X					
	Check the printer-computer cable	X		X					
	Ensure printer software is installed	X		X		X		X	
	Check/replace ink	X	X			X	X		
	Check for paper jam		X		X				

4.3.3 State Transition Testing

- **State Transition Testing:** A black box test design technique in which test cases are designed to execute valid and invalid state transitions.
- State transition testing is used where some aspects of the system can be described in what is called a ‘finite state machine’.
- **State Diagram:** A diagram that depicts the states that a component or system can assume, and shows the events or circumstances that cause and/or result from a change from one state to another.
- A finite state system is often shown as a state diagram.

ILLINOIS TECH | College of Computing

State Diagram for PIN Entry – withdraw $100 at ATM



State Transition Model has Four Parts

- States software may occupy (open/closed or funded/insufficient funds);
- Transaction form one State to another State (not all transactions are allowed);
- States software may occupy (open/closed or funded/insufficient funds);
- Transaction form one State to another State (not all transactions are allowed);

State Transition Model has Four Parts

- Events that cause a transaction (closing a file or withdrawing money);
- Actions that result from a transition (an error message or being given your cash).

Testing for Invalid Transactions

- **State Table:** A grid showing the resulting transitions for each state combined with each possible event, showing both valid and invalid transitions

ILLINOIS TECH | College of Computing

Testing for Invalid Transactions

- State Table for the PIN

	Insert Card	Valid Pin	Invalid Pin
S1 – Start State	S2	-	-
S2 – Wait for Pin	-	S6	S3
S3 – 1st Try Invalid	-	S6	S4
S4 – 2nd Try Invalid	-	S6	S5
S5 – 3rd Try Invalid	-	-	S7
S6 – Access Account	-	?	?
S7 – Eat Card	S1 for new card	-	-

4.3.4 Use Case Testing

- **Use Case Testing:** A black box test design technique in which test cases are designed to execute scenarios of use cases.
- Use case testing is a technique that helps us identify test cases that exercise the whole system on a transaction by transaction basis from start to finish.
- Use cases are a sequence of steps that describe the interactions between the actor and the system.
- Use cases can uncover integration defects.
- Use cases describe the process flows through a system based on its most likely use.

- Use cases are a sequence of steps that describe the interactions between the actor and the system.
- Use cases can uncover integration defects.
- Use cases describe the process flows through a system based on its most likely use.

Partial Use Case for PIN Entry

	Step	Description
Main Success Scenario A: Actor S: System	1	A: Insert Card
	2	S: Validates card and asks for PIN
	3	A: Enters PIN
	4	S: Validates PIN
	5	S: Allows access to account
Extensions	2a	Card not valid S: Display message and reject card
	4a	PIN not valid S: Display message and ask for re-try (twice)
	4b	PIN invalid 3 times S: Eat card and exit

Structure-Based or White-Box Techniques

- **White-box test design technique:** Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.
- White-box technique focus on the structure of a software component, such as statements, decisions, branches or even distinct paths.

Table 1 Testing Spectrum

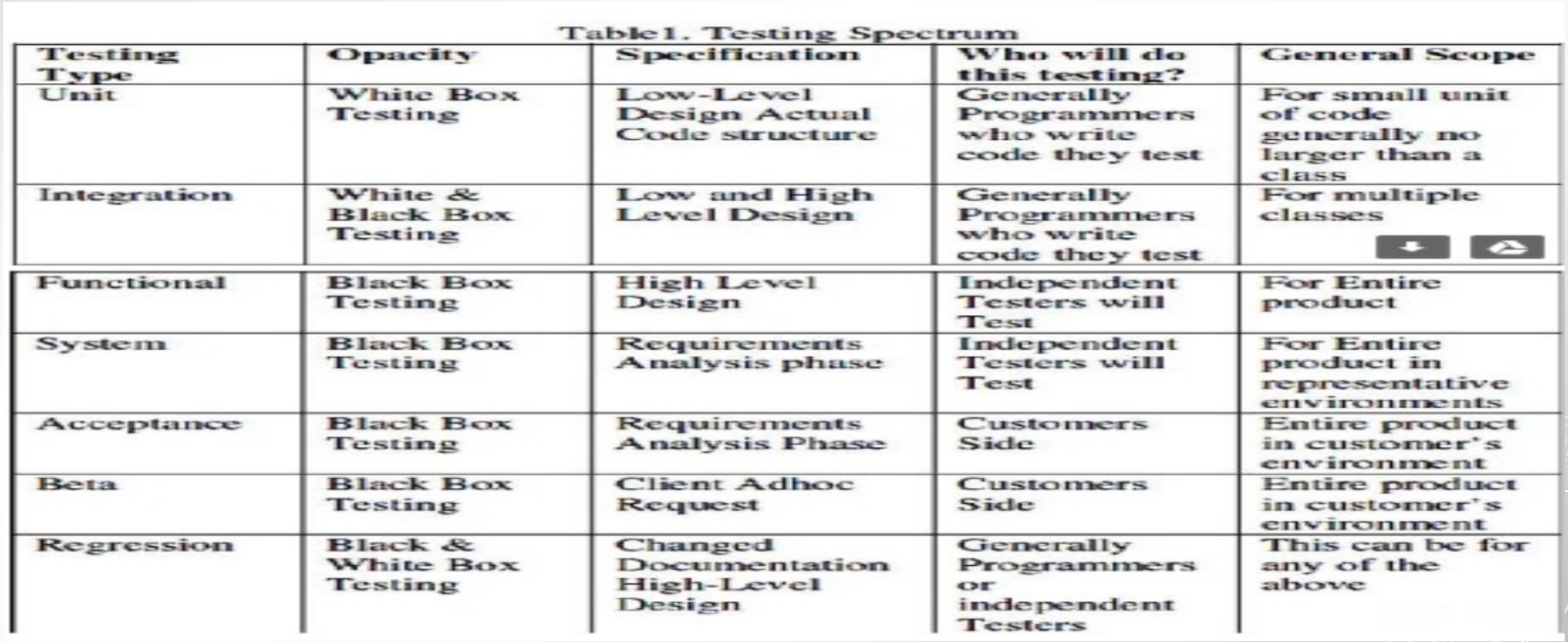
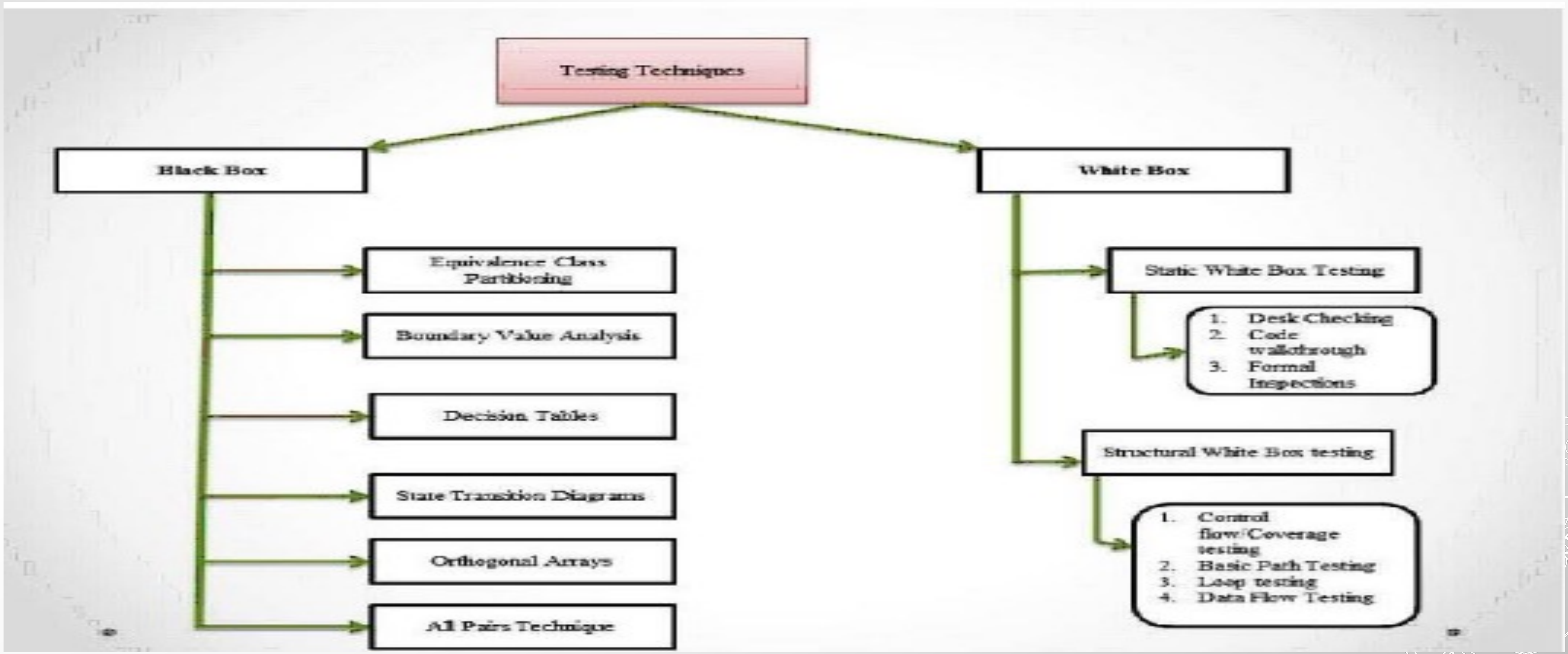


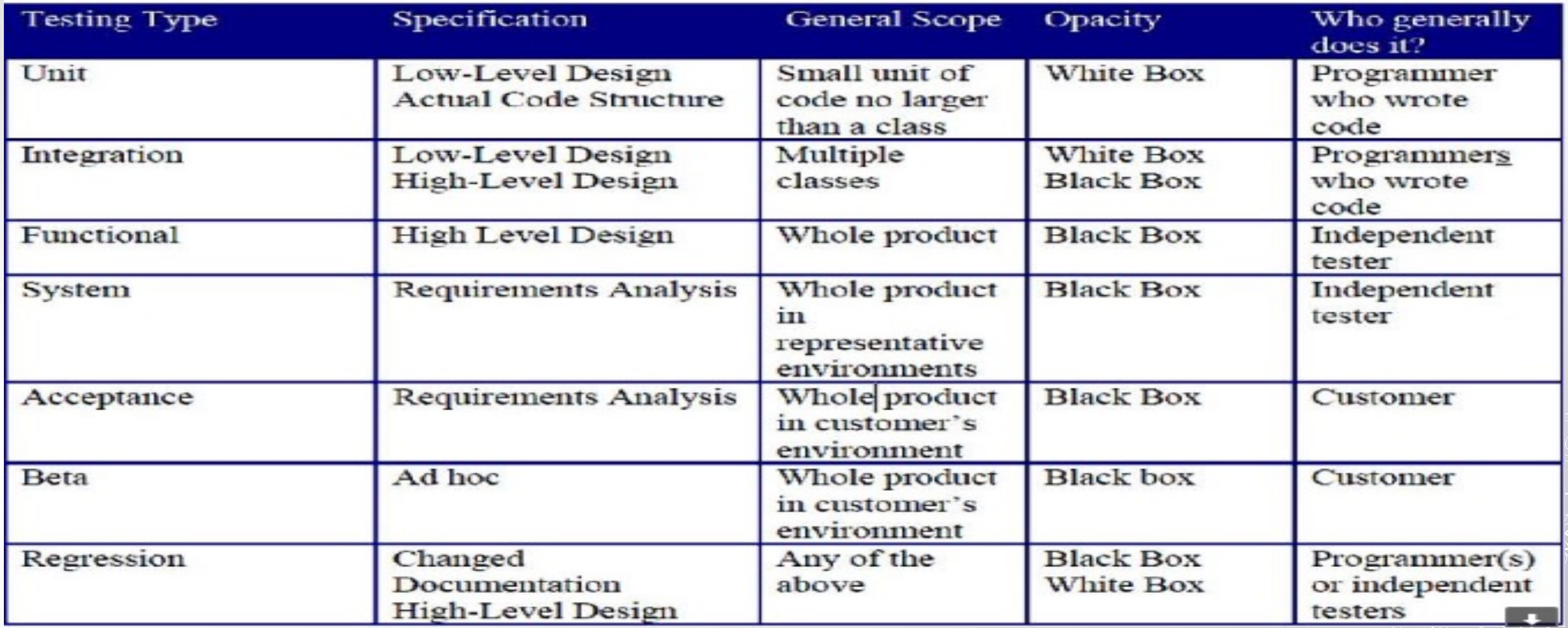
Table1. Testing Spectrum

Testing Type	Opacity	Specification	Who will do this testing?	General Scope
Unit	White Box Testing	Low-Level Design Actual Code structure	Generally Programmers who write code they test	For small unit of code generally no larger than a class
Integration	White & Black Box Testing	Low and High Level Design	Generally Programmers who write code they test	For multiple classes
Functional	Black Box Testing	High Level Design	Independent Testers will Test	For Entire product
System	Black Box Testing	Requirements Analysis phase	Independent Testers will Test	For Entire product in representative environments
Acceptance	Black Box Testing	Requirements Analysis Phase	Customers Side	Entire product in customer's environment
Beta	Black Box Testing	Client Adhoc Request	Customers Side	Entire product in customer's environment
Regression	Black & White Box Testing	Changed Documentation High-Level Design	Generally Programmers or independent Testers	This can be for any of the above

Testing Technique

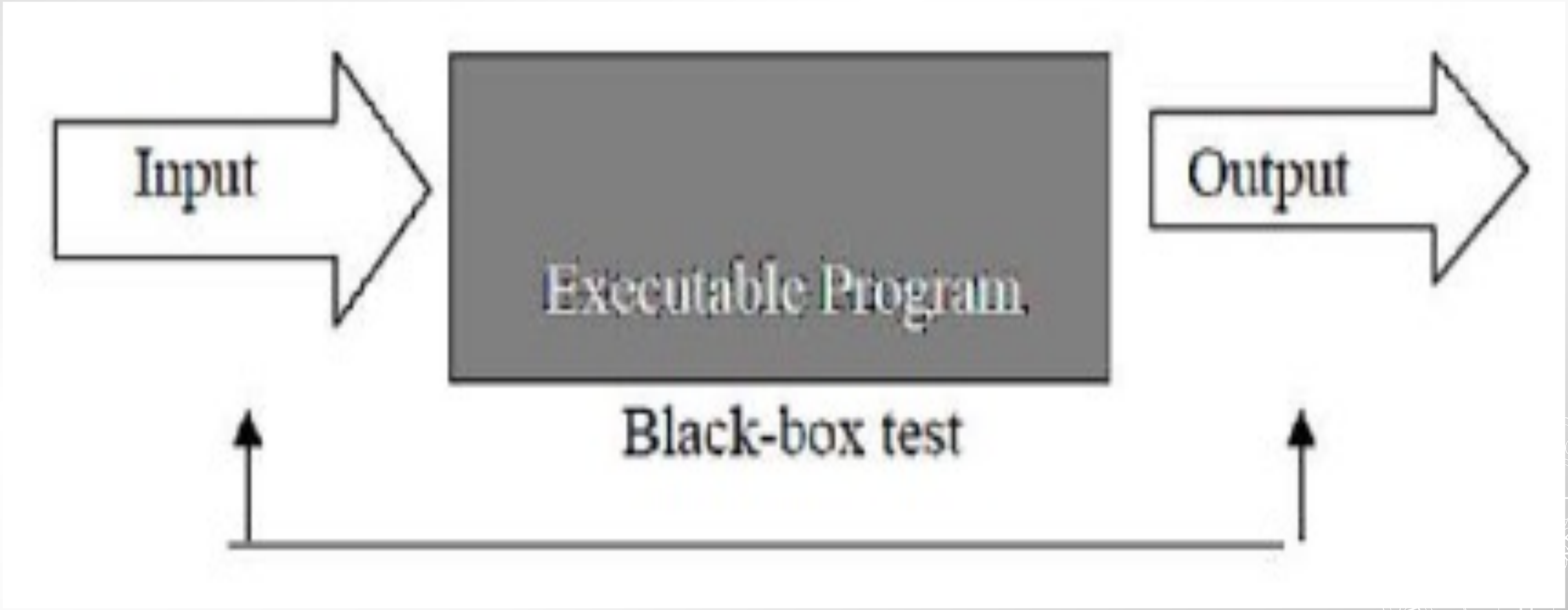


Black-Box Testing Levels



Testing Type	Specification	General Scope	Opacity	Who generally does it?
Unit	Low-Level Design Actual Code Structure	Small unit of code no larger than a class	White Box	Programmer who wrote code
Integration	Low-Level Design High-Level Design	Multiple classes	White Box Black Box	Programmers who wrote code
Functional	High Level Design	Whole product	Black Box	Independent tester
System	Requirements Analysis	Whole product in representative environments	Black Box	Independent tester
Acceptance	Requirements Analysis	Whole product in customer's environment	Black Box	Customer
Beta	Ad hoc	Whole product in customer's environment	Black box	Customer
Regression	Changed Documentation High-Level Design	Any of the above	Black Box White Box	Programmer(s) or independent testers

Black-Box Test



ILLINOIS TECH | College of Computing

Pair-wise Combination of Parameters- Sample array

	A	B	C
1	1	1	3
2	1	2	2
3	1	3	1
4	2	1	2
5	2	2	1
6	2	3	3
7	3	1	1
8	3	2	2
9	3	3	3

ILLINOIS TECH | College of Computing

Orthogonal Arrays

- **Orthogonal Array Testing Strategy (OATS**) is a systematical, statistical way of testing pair-wise interactions by deriving suitable small set of test cases from a large number of scenarios.
- The testing strategy can be used to reduce the number of test combinations and provide maximum coverage with a minimum number of test cases. OATS utilizes an array of values representing variable factors that are combined pair-wise rather than representing all combinations of factors and levels.

ILLINOIS TECH | College of Computing

4.4 Structure-Based or White-Box Techniques

- **White-box test design technique:** Procedure to derive and/or select test cases based on an analysis of the internal structure of a component.
- **White-box testing:** Testing is based on an analysis is of the internal structure of the component or system
- White box testing techniques are:

1. Static white box testing a. Desk checking b. Code walkthrough c. Formal Inspections

2. Structural White box testing a. Control flow/ Coverage testing b. Basic path testing c. Loop testing d. Data flow testing

ILLINOIS TECH | College of Computing

4.4.1 Using structure-based techniques to measure coverage and design tests

- **Coverage (test coverage)** The degree, expressed as a percentage, to which a specified coverage item has been exercised by a test suite.

What is a test coverage?

- Test coverage measures in specific way the amount of testing performed by the set of tests. The basic coverage measure is:

$$\text{Coverage} = \frac{\text{Number of coverage items exercised}}{\text{Total number of coverage items}} \times 100\%$$

Types of coverage

- Coverage can be measured at component-testing level, integration-testing level, or at a system- or acceptance testing levels.
- Component is at the code level
- Integration covers interfaces
- System or user covers requirements
- Specification-based techniques:
- EP: percentage of equivalence partitions exercised. (valid & invalid)
- BVA: percentage of boundaries. (V&I)
- Decision tables: business rules or decision table.

Type of coverage

- State transition testing:
- Percentage of states visited
- Percentage of (valid) transitions exercised (Chow's 0-witch coverage)
- Percentage of pairs of valid transitions and longer series of transitions
- Percentage of invalid transitions exercised

ILLINOIS TECH | College of Computing

How to measure coverage

Coverage measurement tool alters the code by inserting instrumentation:

1. Decide on the structural element to used (coverage items need to counted).

2. Count the structural elements or items.

3. Instrument the code.

4. Run the tests for which coverage measurement is required.

5. Using the output from the instrumentation, determine the percentage of elements of or items exercised.

How to measure coverage

- **Structure based test case design:**
- If you are targeting to cover 95% and you only have 87% then you need to add the additional test cases.

4.4.2 Statement coverage and statement testing

- **Statement coverage:** The percentage of executable statements that have been exercised by a test suite.

4.4.3 Decision coverage and decision testing

- **Decision coverage:** The percentage of decision outcomes that have been exercised by a test suite. 100% decision coverage implies both 100% branch coverage and 100% statement coverage.

ILLINOIS TECH | College of Computing

4.4.4 Other structure-based techniques

- **Branch coverage:** The percentage of branches that have been exercised by the test suite. 100% branch coverage implies both 100% decision coverage and 100% statement coverage.

ILLINOIS TECH | College of Computing

4.5 Experience Based Techniques

- There is a definite role for non-systematic techniques, i.e. tests based on a person’s knowledge, experience, imagination and intuition. The reason is that some defects are hard to find using more systematic approaches.

Control flow

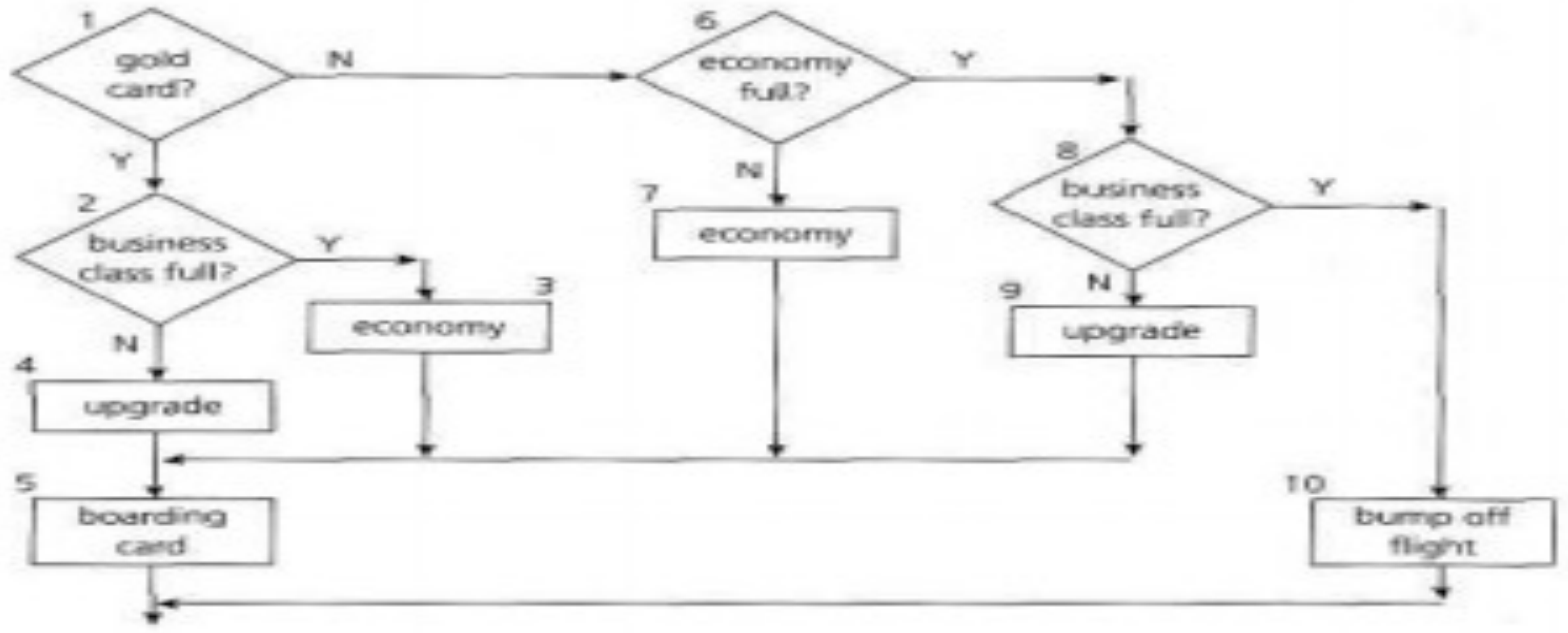


FIGURE 4.5 Control flow diagram for flight check-in

ILLINOIS TECH | College of Computing

4.5.1 Error guessing and fault attacks

- Error guessing is a technique that should be always used as complement to other formal techniques.
- **Fault attack:** Directed and focused attempt to evaluate the quality, especially reliability, of a test object by attempting of force specific failures to occur.

4.5.2 Exploratory testing

- **Exploratory testing:** An informal test design technique where the tester actively controls the design of the test as those test are performed and uses information gained while testing to design new and better tests.

4.6 Choosing Test Techniques

- The choice of which test techniques to use depends on a number of factors, including the type of system, regulatory standards, customer or contractual requirements, level of risk, type of risk, test objective, documentation available, knowledge of the tester, time and budget, development life cycle, use case models and pervious experience to find different defects.

Decision which technique to be used

- Model used: If specification contains a state transition diagram, - use state transition testing.
- Tester knowledge/experience: Tester knowledge is very useful and use it.
- Likely defects: This knowledge is gained through experience.
- Test objective: This should be through testing and use structure-based technique.
- Documentation: Documentation influences the testing technique e.g. decision tables, state graphs etc.
- Life cycle model: Sequential life cycle leads to more formal technique.

External factors that influence which technique should be used

- Risk: Exploratory testing is appropriate.
- Customer/contractual requirements: Sometimes you need to use contract specific technique for testing.
- Regulatory requirements: For example FAA airline industry, FDA food and drug administration, healthcare HIPAA law.
- **HIPAA** is the acronym for the Health Insurance Portability and Accountability Act that was passed by Congress in 1996.
- **HIPAA** does the following: Provides the ability to transfer and continue health insurance coverage for millions of American workers and their families when they change or lose their jobs
- Time and budget: Ultimately how much time and money is available for testing. Depending on the time and budget you can select which type of testing can be used.

4. Test Design Techniques-Question 1

1. In which document described in IEEE 829 would you find instructions for the steps to be taken for a test including set-up, logging, environment and measurement?

   a. Test plan

   b. Test design specification

   c. Test case specification

   d. Test procedure specification

4. Test Design Techniques - Question 2

2. With a highly experienced tester with a good business background, which approach to defining test procedures would be effective and most efficient for a project under server time pressure?

a. A high-level outline of the test conditions and general steps to take.

b. Every step in the test spelled out in detail.

c. A high-level outline of the test conditions with the steps to take discussed in detail with another experienced tester.

d. Detailed documentation of all test cases and careful records of each step taken in the testing.

ILLINOIS TECH | College of Computing

4. Test Design Techniques-Question 3

3. Put the test cases that implement the following test conditions into the best order for the test execution schedule, for a test that is checking modifications of customers on a database.

1. Print modification customer record.
2. Change customer address: house number and street name.
3. Capture and print the on-screen error message.
4. Change customer address: postal code.
5. Confirm existing customer is on the database by opening that record.
6. Close the customer record and close the database.
7. Try to add a new customer with no details at all.
   - a. 5,4,2,1,3,7,6
   - b. 4,2,5,1,6,7,3
   - c. 5,4,2,1,7,3,6
   - d. 5,1,2,3,4,7,6

4. Test Design Techniques-Question 4

4. Why are both specification-based and structure-based testing techniques useful?

a. They find different types of defect.

b. Using more techniques is always better.

c. Both find the same types of defect.

d. Because specifications tend to be unstructured.

4. Test Design Techniques-Question 5

5. What is a key characteristic of structure-based testing techniques?

a. They are mainly used to assess the structure of a specification.

b. They are used both to measure coverage and to design tests to increase coverage.

c. They are based on the skills and experience of the tester.

d. They use a formal or informal model of the software or component.

4. Test Design Techniques-Question 6

6. Which of the following would be an example of decision-table testing for a financial application applied at the system-test level?

a. A table containing rules for combination of inputs to two fields on a screen.

b. A table containing rules for interfaces between components.

c. A table containing rules for mortgage applications.

d. A table containing rules for chess.

4. Test Design Techniques-Question 7

7. Which of the following could be a coverage measure for the state transition testing?

V All states have been reached.

W The response time for each transaction is adequate.

X Every transition has been exercised.

Y All boundaries have been exercised.

Z Specific sequences of transitions have been exercised.

a. X,Y and Z
b. V,X,Y and Z
c. W, X and Y
d. V, X and Z

4. Test Design Techniques-Question 8

8. Postal rates for ‘light letters’ are 25p up to 10g, 35p up to 50g plus an extra 10p for each additional 25g up to 100g.

Which test inputs (in grams) would be selected using equivalence partitioning?

a. 8, 42, 82, 102

b. 4, 15, 65, 92, 159

c. 10, 50, 75, 100

d. 5, 20, 40, 60, 80

4. Test Design Techniques-Question 9

9. Which of the following could be used to access the coverage achieved for specification-based (black-box) test techniques?

V Decision outcomes exercised

W Partitions exercised

X Boundaries exercised

Y State transitions exercised

Z Statements exercised

a. V, W, Y or Z
b. W, X or Y
c. V, X or Z
d. W, X, Y or Z

ILLINOIS TECH | College of Computing

4. Test Design Techniques-Question 10

10. Which of the following would structure-based test design techniques be most likely to be applied to?

1. Boundaries between mortgage interest rate bands.
2. An invalid transition between two different arrears statuses.
3. The business process flow for mortgage approval.
4. Control flow of the program to calculate repayments.

   a. 2, 3 and 4

   b. 2 and 4

   c. 3 and 4

   d. 1, 2 and 3

ILLINOIS TECH | College of Computing

4. Test Design Techniques-Question 11

11. Use case testing is useful for which of the following?

P Designing acceptance tests with users or customers.

Q Making sure that the mainstream business processes are tested.

R Finding defects in the interaction between components.

S Identifying the maximum and minimum values for every input filed

T Identifying the percentage of statements exercised by a sets of tests.

a. P, Q and R
b. Q, S and T
c. P, Q and S
d. R, S and T

4. Test Design Techniques-Question 12

- 12. Which of the following statements about the relationship between statement coverage and decision is correct?

  a. 100% decision coverage is achieved if statement coverage is greater than 90%

  b. 100% statement coverage is achieved if statement coverage is greater than 90%.

  c. 100% decision coverage always means 100% statement coverage.

  d. 100% statement coverage always means 100% decision coverage.

ILLINOIS TECH | College of Computing

4. Test Design Techniques-Question 13

- 13. If you are flying with a an economy ticket, there is a possibility that you may get upgraded to business class, especially if you hold a gold card in the airline's frequent flier program. If you don't hold a gold card, there is a possibility that you will get 'bumped' off the flight if it is a full and you check in late.

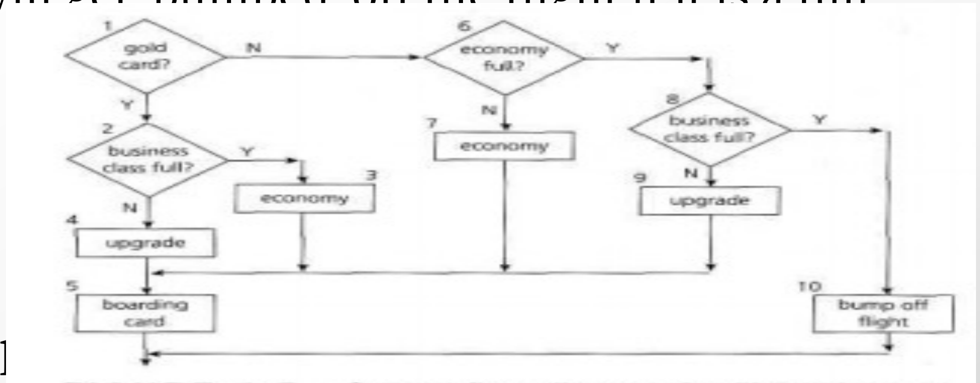


FIGURE 4.5 Control flow diagram for flight check-in

Check control flow diagram(flight check-in)

Three test have been run:

Test 1: Gold card holder who gets upgraded to business cl

Test 2: Non-gold card holder who stays in economy.

Test 3: A person who is bumped form the flight. What is the statement coverage of these three tests?

a. 60%

b. 70%

c. 80%

d. 90%

4.Test Design Techniques-Question 14

14 Why are error guessing and exploratory testing good to do?

a. They can find defects missed by specification-based and structure-based techniques.

b. They don't require any training to be as effective as formal techniques.

c. They can be used most effectively when there are good specifications.

d. They will ensure that all of the code or system is tested.

4.Test Design Techniques-Question 15

15. How do experience-based techniques differ from specification-based techniques?

a. They depend on the tester's understanding of the way the system is structured rather than on a documented record of what the system should do.
b. They depend on having older testers rather than younger testers.
c. They depend on a documented record of what the system should do rather than on an individual's personal view.
d. They depend on an individual's personal view rather than on a documented record of what the system should do

4.Test Design Techniques-Question 16

16 When choosing which technique to use in a given situation, which factor should be taken into account?
U Previous experience of types of defects found in this or similar systems.
V The existing knowledge of the testers.
W Regulatory standards that apply.
X The type of test execution tool that will be used.
Y The documentation available.
Z Pervious experience in the development language

a. V, W, Y and Z
b. U, V, W and Y
c. U, X and Y
d. V, W and Y

ILLINOIS TECH | College of Computing

4.Test Design Techniques-Question 17

- 17. Given the state diagram in Figure 4.6,which test case is the minimum series of valid transitions to cover every

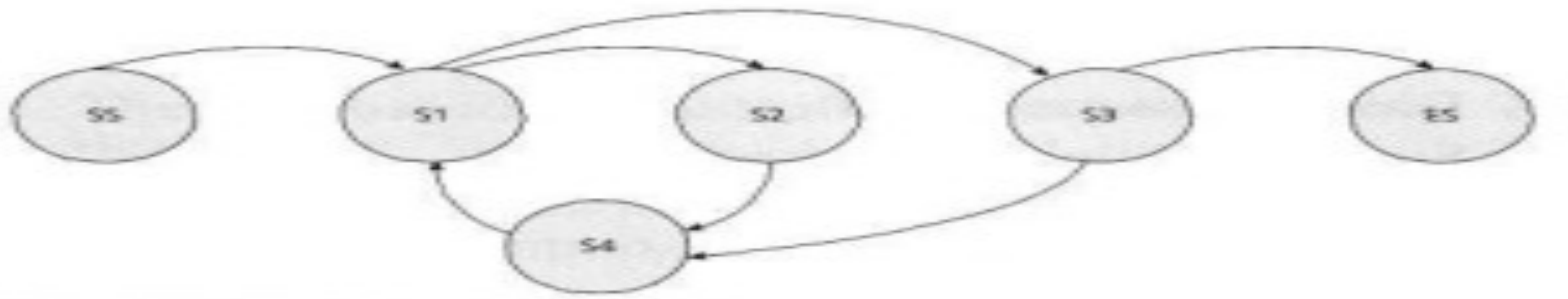


FIGURE 4.6 State diagram

a. SS – S1 – S2 – S4 – S1 – S3 – ES
b. SS – S1 – S2 – S3 – S4 – ES
c. SS – S1 – S2 – S4 – S1 – S3 – S4 – S1 – S3 – ES
d. SS – S1 – S4 – S2 – S1 – S3 – ES