

# Software Testing



# **Software testing:**

**“The Process of executing a program with the intent of finding errors”**

**(Myers)**

**"Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results."**

**(Hetzel)**

# **Software testing:**

**“The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component”**

**(IEEE Standard 610.12-1990)**

# **Software testing:**

**“The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item.”**

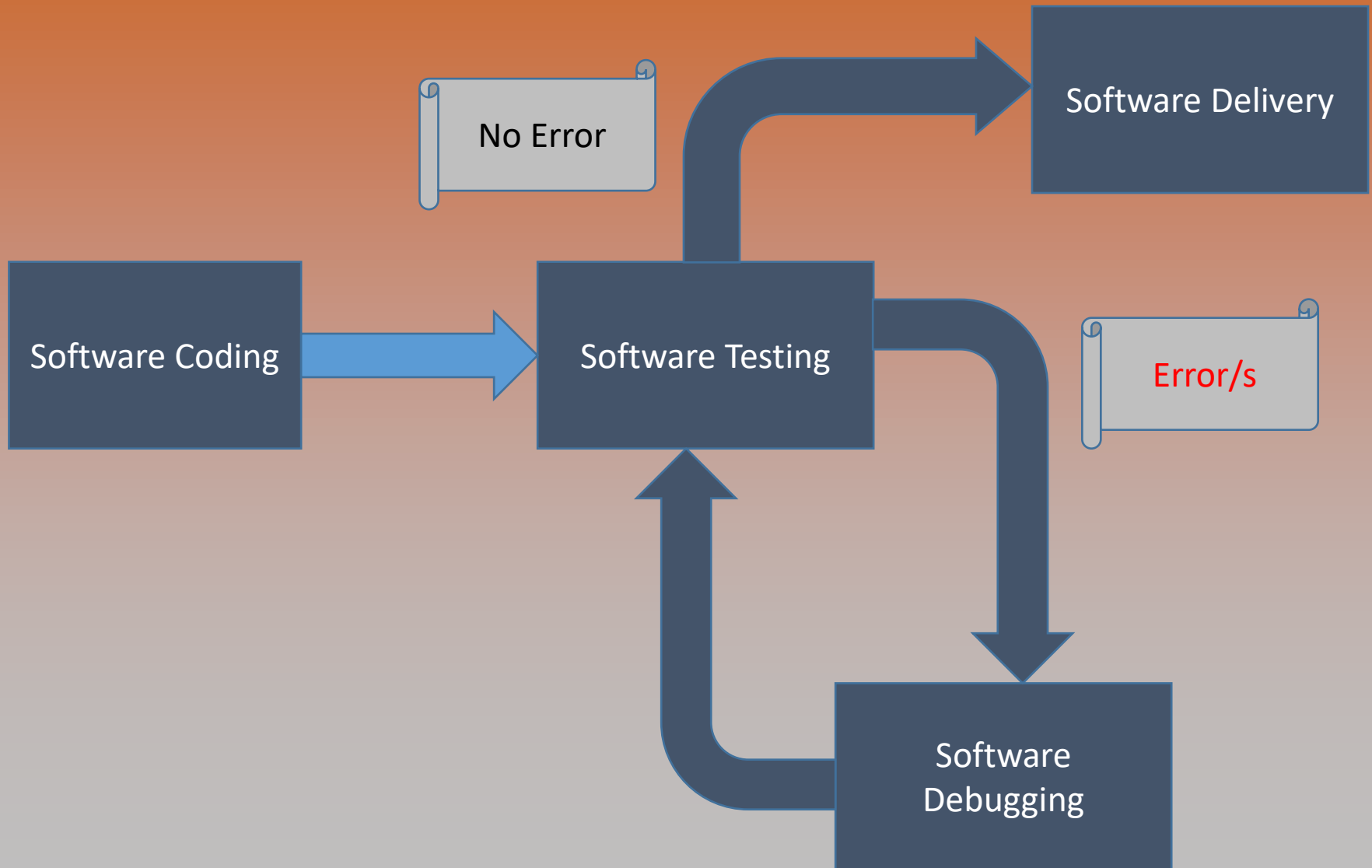
**(IEEE Standard 829-1998 )**

# Testing is not easy!

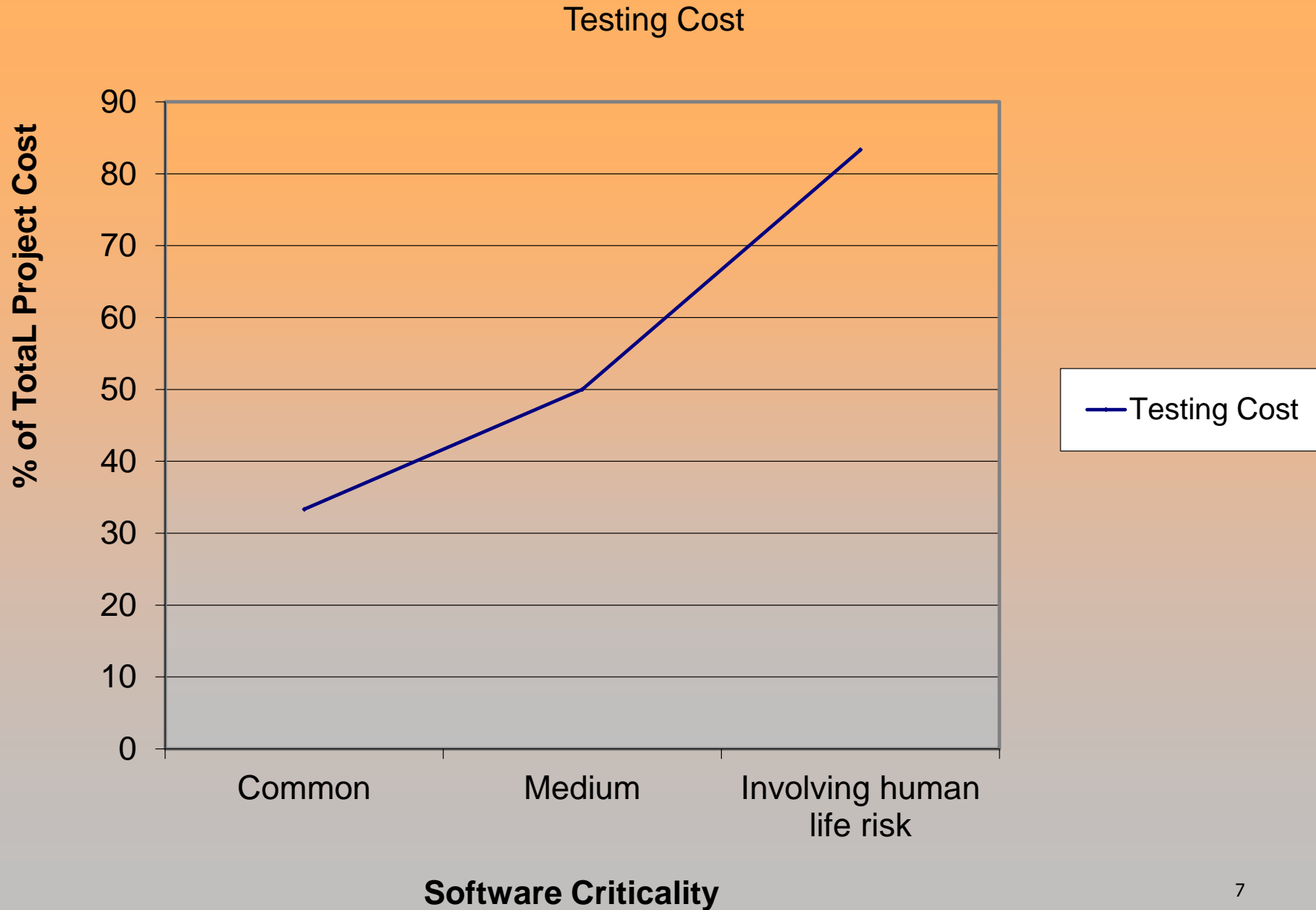
- Takes time, must be scheduled
- Takes resources, it costs and it costs much.
- May include heavy debugging, can cause the re-development of the product.
- There is no net to catch all the bugs (Exhaustive vs Selective testing).



# Testing – Debugging Cycle



# Testing cost!



# Why do we spend so much?

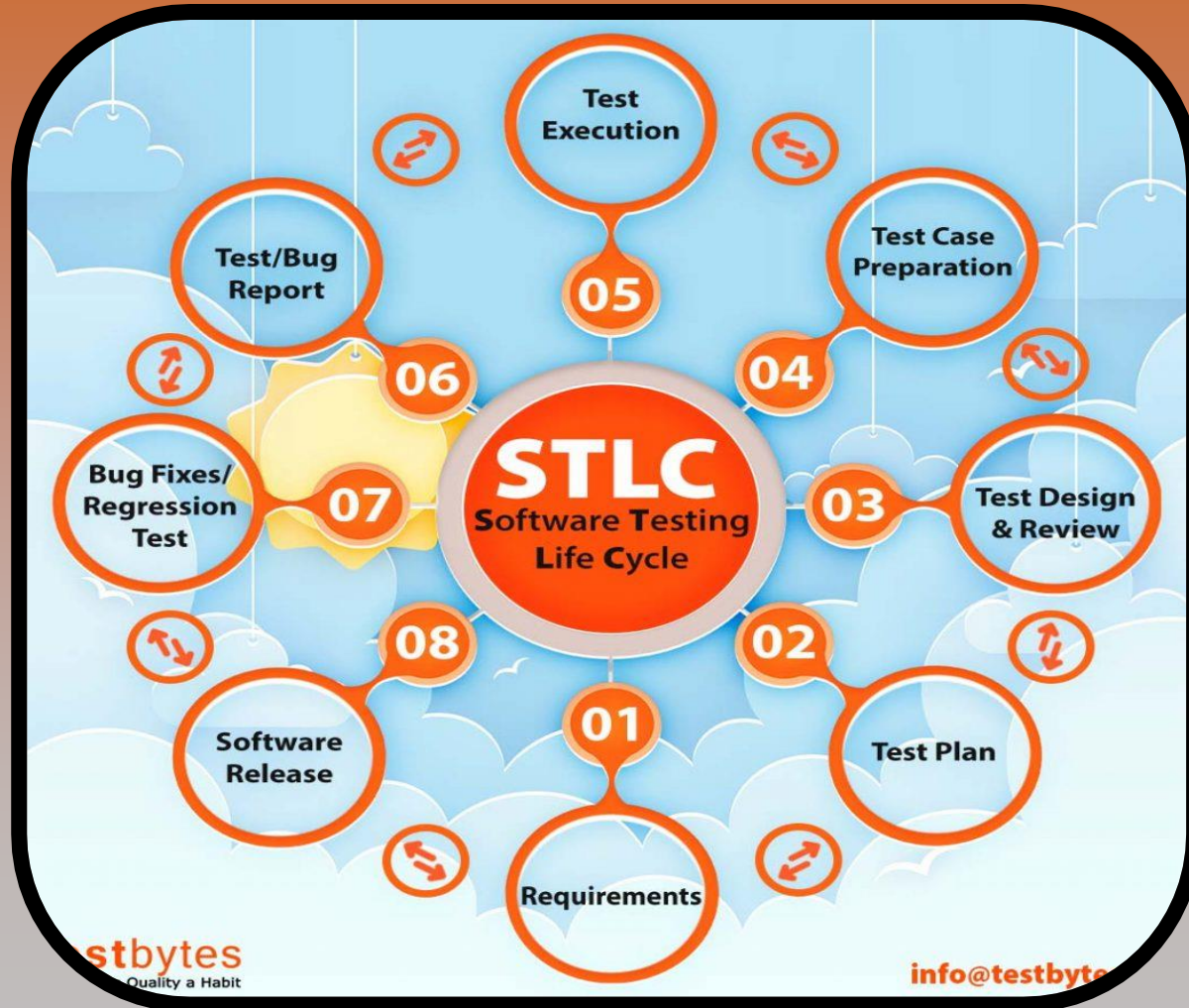
- Why is the testing being done?
  - Find failures
  - Certification
  - Safety: criticality
  - Usability
  - Performance
  - Acceptance
  - Load
  - Compatibility
  - Security



# **Testing Activities**

- 1. Test planning and design**
- 2. Verification of test plans, designs and cases**
- 3. Test case selection, creation and verification**
- 4. Planning expected results**
- 5. Data preparation**
- 6. Execution and recording of results**
- 7. Analysis of results**
- 8. Wrap-up**

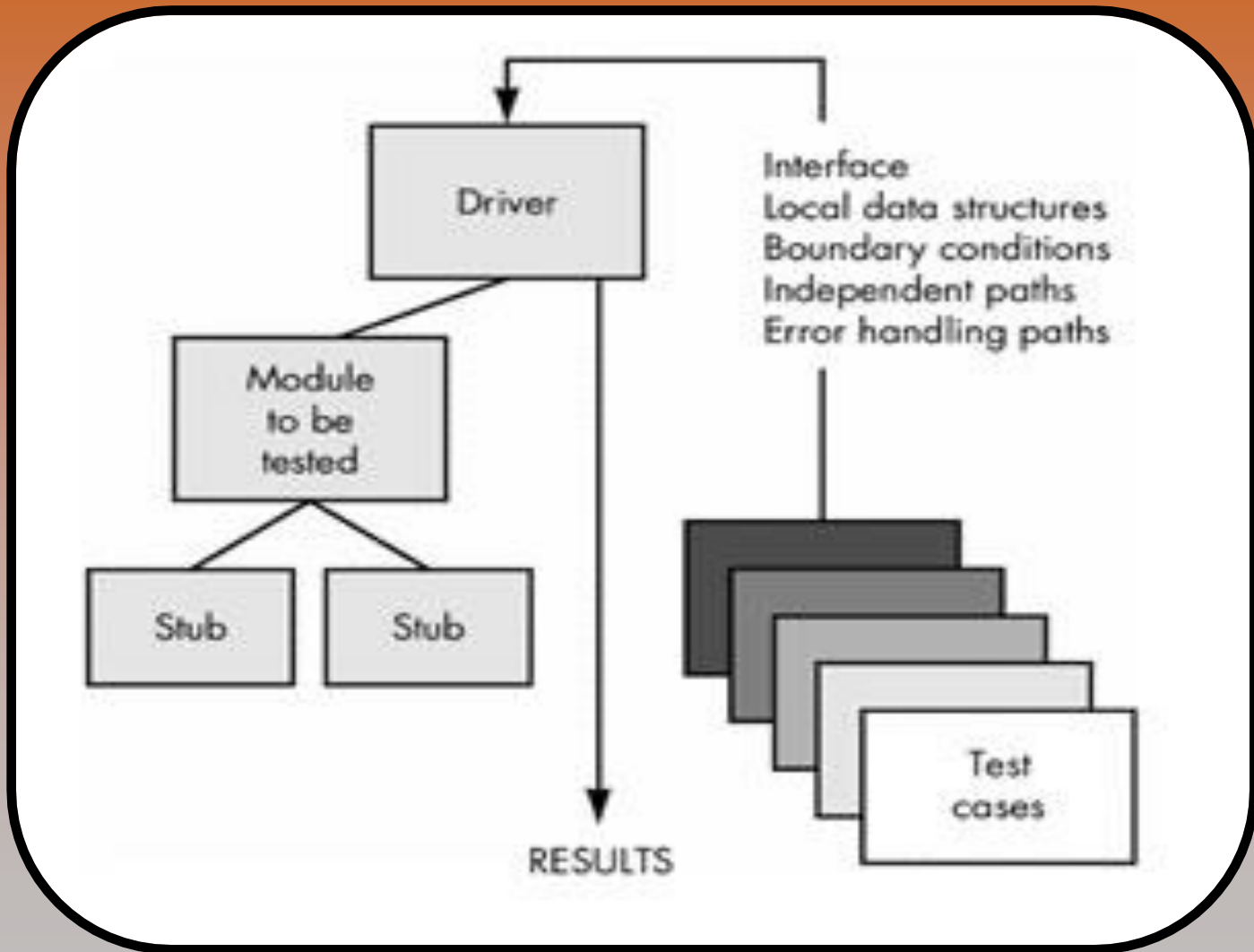
# Software Testing Life Cycle:



# What to test?

- Any part of a system can be tested
  - choice of part constrains choices in other dimensions
- Units are typically the smallest parts tested
  - a unit can be a function, a class, or a work assignment
- Some parts of the software may not be tested at all
  - e.g. exception handlers, debug code, commercial components, library functions
- For some testing
  - stubs and drivers may have to be written (and tested)

# Stub and Driver:



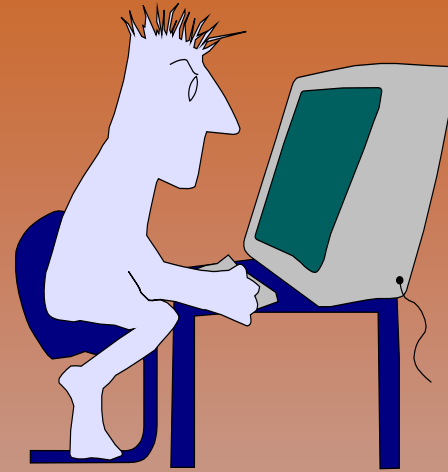
# Who should do the testing?



***Developer***

**Understands the system  
but, will test "gently"  
and, is driven by "delivery"**

**Constructive Approach**



***Independent tester***

**Must learn about the system,  
but, will attempt to break it  
and, is driven by "quality"**

**Destructive Approach**

# **Skills of a Tester:**

- **Not all testers are developers**
  - **e.g. testers at Corel who are photographers (Corel Draw testing - end user testing).**
  - **e.g. testing buddies at Microsoft.**
  - **Testers may also be developers.**
- **Some testers may need specialized domain knowledge related to the application area.**

# **Skills of a Tester:**

- **Testers must have**
  - **ability to select good test cases**
  - **ability to withstand the most severe deadline and resource pressures**
  - **ability to organize and analyze large volumes of data**
  - **ability to sort out complex version control issues**

- **Inexperienced testers often fail to report:**

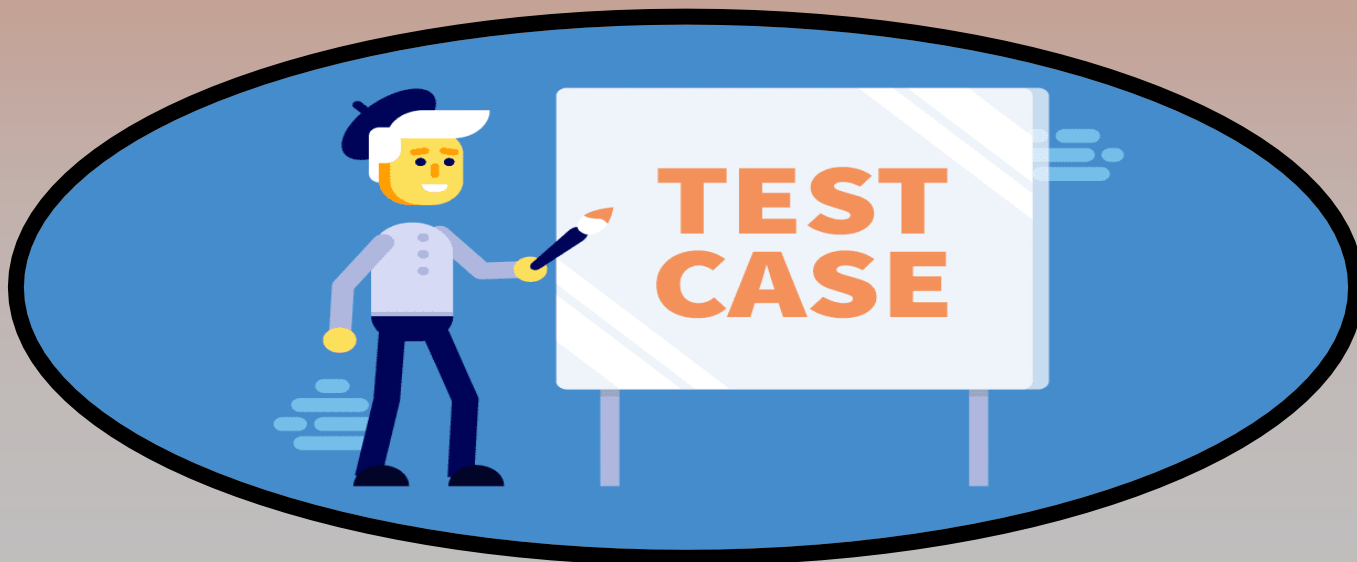
- timing problems
- transient bugs
- what they can't quickly replicate
- what they think can be attributable to their misunderstanding
- what they think the reader of the report might consider minor
- lots of other problems



# Test Case:

**“A test case is a scenario to be executed with the intention of finding a particular error/defect.”**

**“Inputs to test the system and the predicted outputs from these inputs if the system operates according to its specification.”**



# A good test case:

- A good test case:
  - has a high probability of finding an error
  - is not redundant
  - should be “best of breed”
  - should be neither too simple nor too complex



# Successful Test:



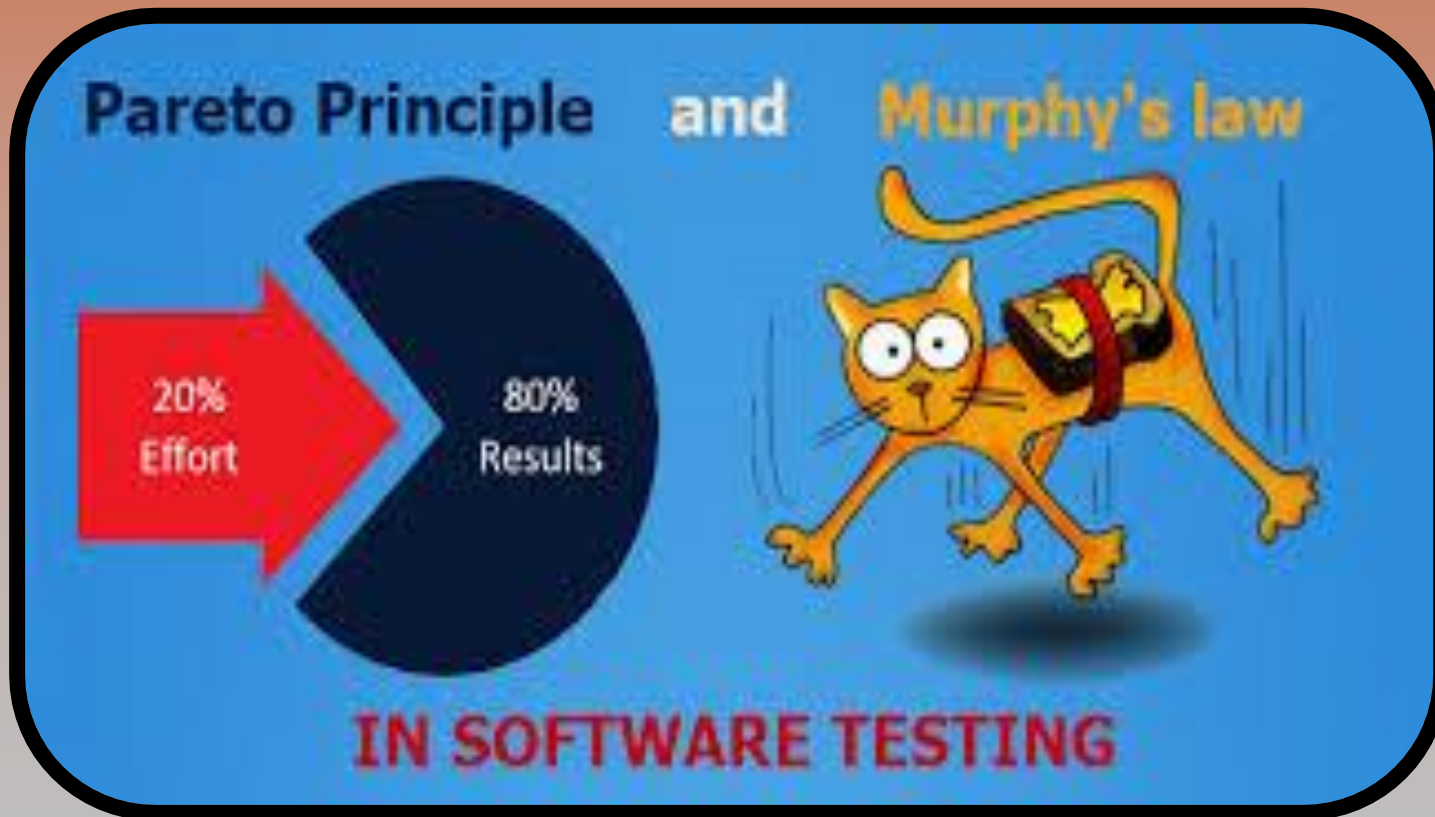
- A test which causes a program to behave in an anomalous way, is said to be a successful test case.
- Tests show the presence not the absence of defects.
- Only exhaustive testing can show a program is free from errors. However, exhaustive testing is impossible.

# **Testing Principles:**

- 1. All tests should be traceable to customer requirements.**
- 2. Tests should be planned long before testing begins.**
- 3. The Pareto principle applies to software testing.**
- 4. Testing should begin “in the small” and progress toward testing “in the large”.**
- 5. Exhaustive Testing is not possible.**
- 6. To be most effective, testing should be conducted by an independent third party.**

# Pareto Principle:

**“80% of all the errors uncovered during testing will likely to be traceable to 20% of all the program components.”**



# **Parts of a test case:**

- |                                 |                                |
|---------------------------------|--------------------------------|
| <b>1. Test Case No.</b>         | <b>6. Detailed Description</b> |
| <b>2. Test case Description</b> | <b>7. Steps / Actions</b>      |
| <b>3. Pre-Requisites</b>        | <b>8. Expected Results</b>     |
| <b>4. Extra Instructions</b>    | <b>9. Actual Results</b>       |
| <b>5. Dependencies</b>          | <b>10. Test case Status</b>    |

## Test Case Example1 (simple test)

Test Case #: 2.2

Test Case Name: Change PIN

Page: 1 of 1

System: ATM

Subsystem: PIN

Designed by: ABC

Design Date: 28/11/2004

Executed by:

Execution Date:

Short Description: Test the ATM Change PIN service

### Pre-conditions

The user has a valid ATM card - The user has accessed the ATM by placing his ATM card in the machine

The current PIN is 1234

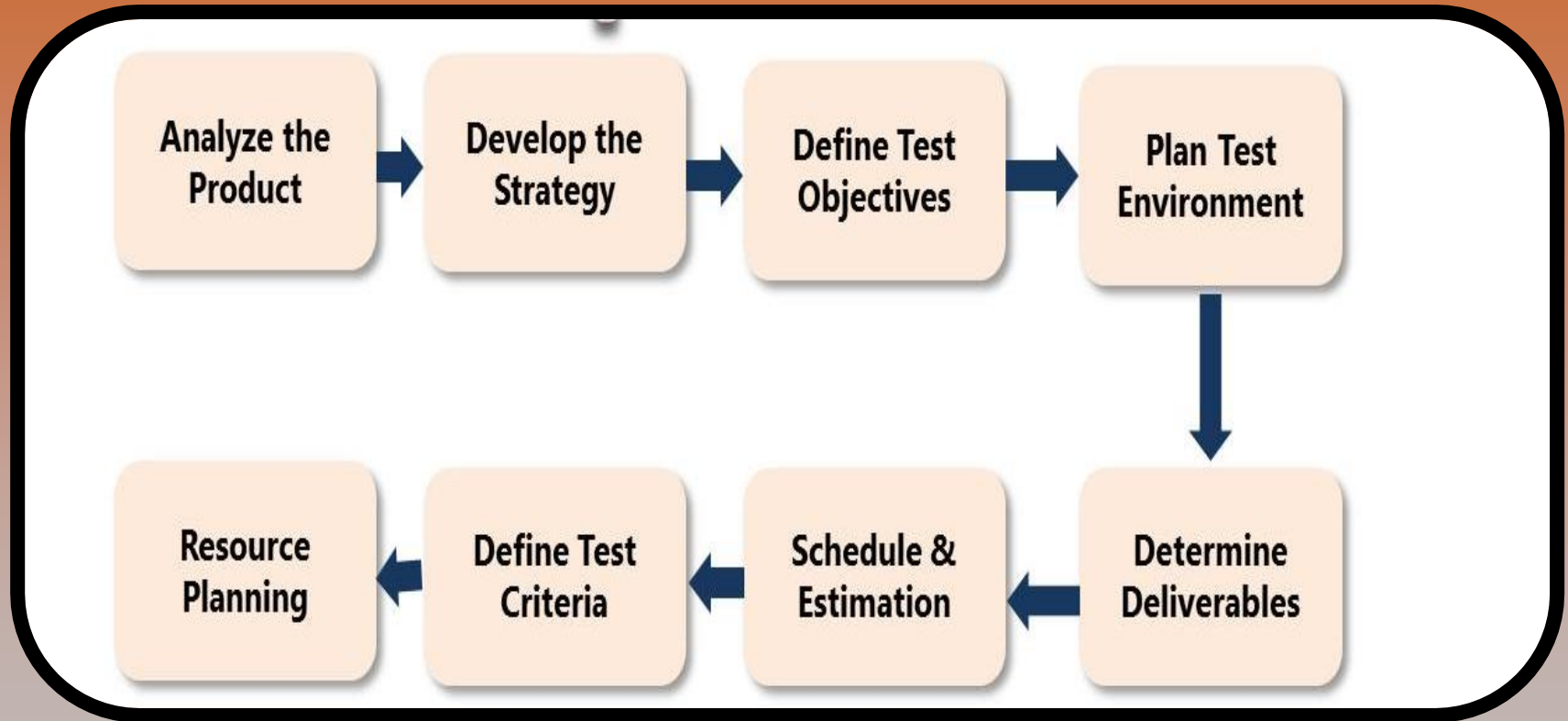
The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click the 'Change PIN' button	The system displays a message asking the user to enter the new PIN		
2	Enter '5555'	The system displays a message asking the user to confirm (re-enter) the new PIN		
3	Re-enter '5555'	The system displays a message of successful operation The system asks the user if he wants to perform other operations		
4	Click 'YES' button	The system displays the main menu		
5	Check post-condition 1			

### Post-conditions

1. The new PIN '5555' is saved in the database

# Test plan:





# Test report:

## Test Report

Test Cycle

System Test

EXECUTED	PASSED			130	
	FAILED			0	
	(Total) TESTS EXECUTED (PASSED + FAILED)				130
PENDING					0
IN PROGRESS					0
BLOCKED					0
(Sub-Total) TEST PLANNED					130
(PENDING + IN PROGRESS + BLOCKED + TEST EXECUTED)					

Functions	Description	% TCs Executed	% TCs Passed	TCs pending	Priority	Remarks
New Customer	Check new Customer is created	100%	100%	0	High	
Edit Customer	Check Customer can be edited	100%	100%	0	High	
New Account	Check New account is added	100%	100%	0	High	
Edit Account	Check Account is edit	100%	100%	0	High	
Delete Account	Verify Account is delete	100%	100%	0	High	
Delete customer	Verify Customer is Deleted	100%	100%	0	High	
Mini Statement	Verify Ministatement is generated	100%	100%	0	High	
Customized Statement	Check Customized Statement is generated	100%	100%	0	High	

# When to stop testing?

- How is the decision to stop testing made?
  - Coverage
  - Limits set by management
  - User acceptance e.g. Early adopters
  - Contractual
  - Reliability threshold
  - Fault detection rate threshold
  - Meeting a standard

# **References:**

- **Sommerville, Ian “Software Engineering 6<sup>th</sup> edition.”**
- **Shepard, Terry “Dimensions of Testing”**
- **Pressman, Roger “Software Engineering, A Practitioner’s Approach 6<sup>th</sup> edition. ”**

**Thank You!**