

# Testing

# Lecture objectives

- Testing
- Defect/Bug
- Successful Test
- Limitations of Testing
- Effective Testing Methodology
- Testing Vs. Development
- Testing Stages
- Verification Methods

# Testing

- The correct approach to testing a scientific theory is not to try to verify it, but to seek to refute the theory. That is to prove that it has errors. (Popper 1965)

# Defect/Bug

- A defect is a variance from a desired product attribute. These attributes may involve software/system specifications as well as user expectation. Anything that may cause customer dissatisfaction, is a defect.

# Testing

- ❖ The goal of testing is to expose defects in a software/system before it is put to use.
- ❖ A tester tries to break the system. The objective is to show the presence of a defect not the absence of it.
- ❖ Testing cannot show the absence of a defect. It only increases your confidence in the system.
- ❖ This is because exhaustive testing is not possible – it is simply too expensive and needs virtually infinite resources.

# Typical Residual Defect Rates

- Some typical residual defect rates (bugs left over after the software has shipped) per kloc (one thousand lines of source code):
  - **1 - 10 defects/kloc:** Typical industry software.
  - **0.1 - 1 defects/kloc:** High-quality validation.
  - **0.01 - 0.1 defects/kloc:** The very best, safety-critical validation. NASA and companies like Praxis can achieve this level.

# Successful Test

- “A test is said to be successful if it discovers an error”

# Limitations of testing

a	b	Expected result
"cat"	"dog"	False
""	""	True
"hen"	"hen"	True
"hen"	"heN"	False
""	""	False
""	"ball"	False
"cat"	""	False
"HEN"	"hen"	False
"rat"	"door"	False
" "	" "	True



# Code Example

```
public class Comparison {  
    boolean isStringsEqual(String a, String b) {  
        boolean result=false;  
  
        if (a.length() != b.length())  
            result = false;  
  
        else if(a.length() == b.length())  
        {  
            for (int i = 0; i < a.length(); i++) {  
                if (a.charAt(i) == b.charAt(i))  
                    result = true;  
  
                else  
                    result = false;  
            }  
        }  
  
        return result;  
    }  
  
    public static void main(String[] args) {  
        Comparison comp = new Comparison(); // Create an instance of the class  
        boolean result = comp.isStringsEqual("cat", "rat");  
        System.out.println("Are the strings equal?" + result);  
    }  
}
```

## Output:

Are the strings equal? true

# Code Example

```
public class Comparison {  
    boolean isStringsEqual(String a, String b) {  
        boolean result = true;  
  
        if (a.length() != b.length())  
            result = false;  
        else if (a.length() == b.length()) {  
            for (int i = 0; i < a.length(); i++) {  
                if (a.charAt(i) != b.charAt(i)) {  
                    result = false;  
                    break; // exit the loop if a mismatch is found  
                }  
            }  
        }  
  
        return result;  
    }  
  
    public static void main(String[] args) {  
        Comparison comp = new Comparison(); // Create an instance of the class  
        boolean result = comp.isStringsEqual("cat", "rat");  
        System.out.println("Are the strings equal?" + result);  
    }  
}
```

## Output:

Are the strings equal? False

# Limitations of Testing

- In order to prove that a formula or hypothesis is incorrect all you have to do to show only one example in which you prove that the formula or theorem is not working.
- On the other hand, million of examples can be developed to support the hypothesis but this will not prove that it is correct.

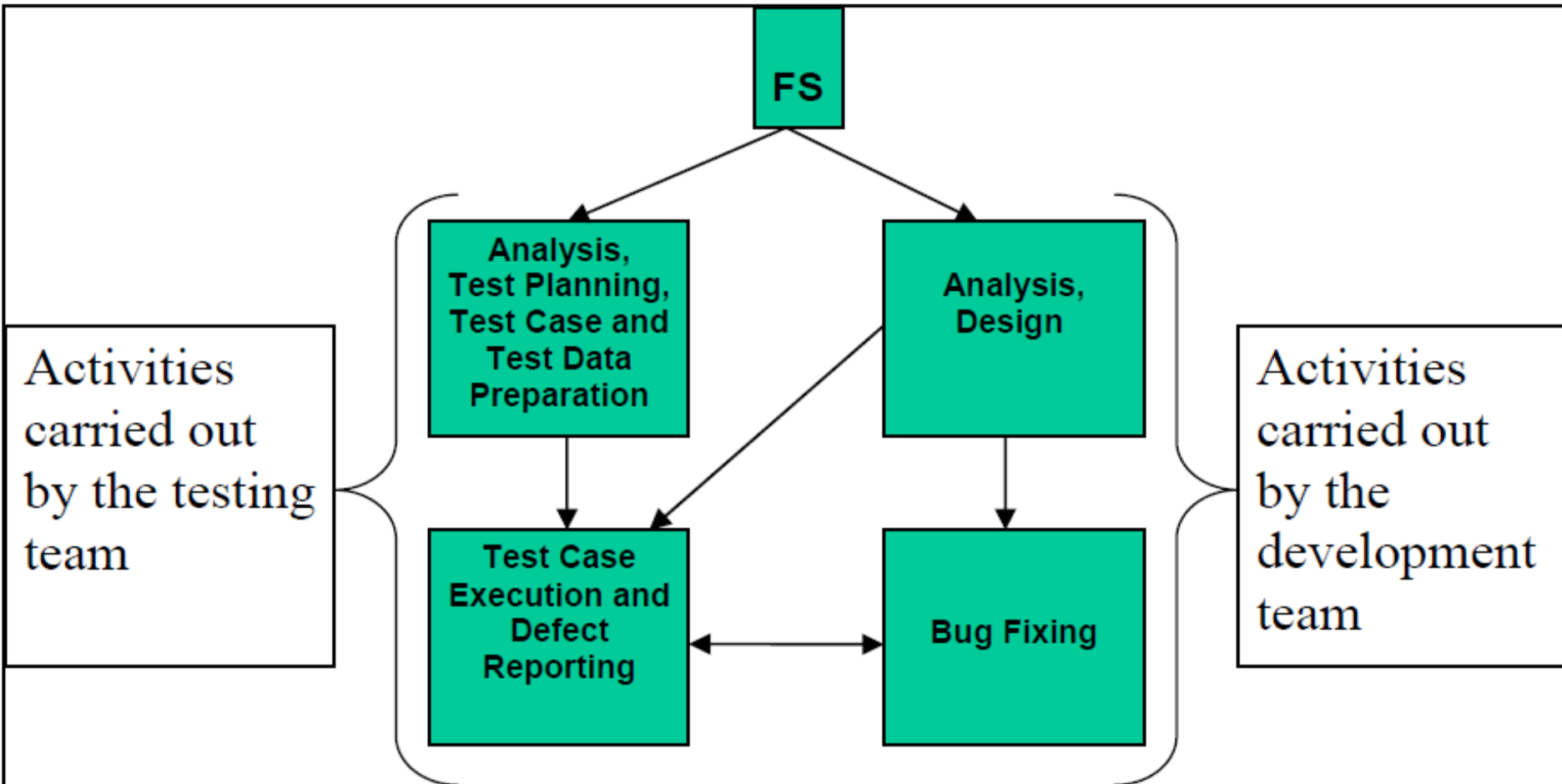
# Effective Testing Methodology

- Goal is to reveal as many defects as possible to increase the confidence level in system operation

# Test Cases and Test Data

- A test case involves
  1. Input and output specification
  2. Steps to perform the function
  3. Expected results that the system produces
- However, test data includes inputs that have been devised to test the system.

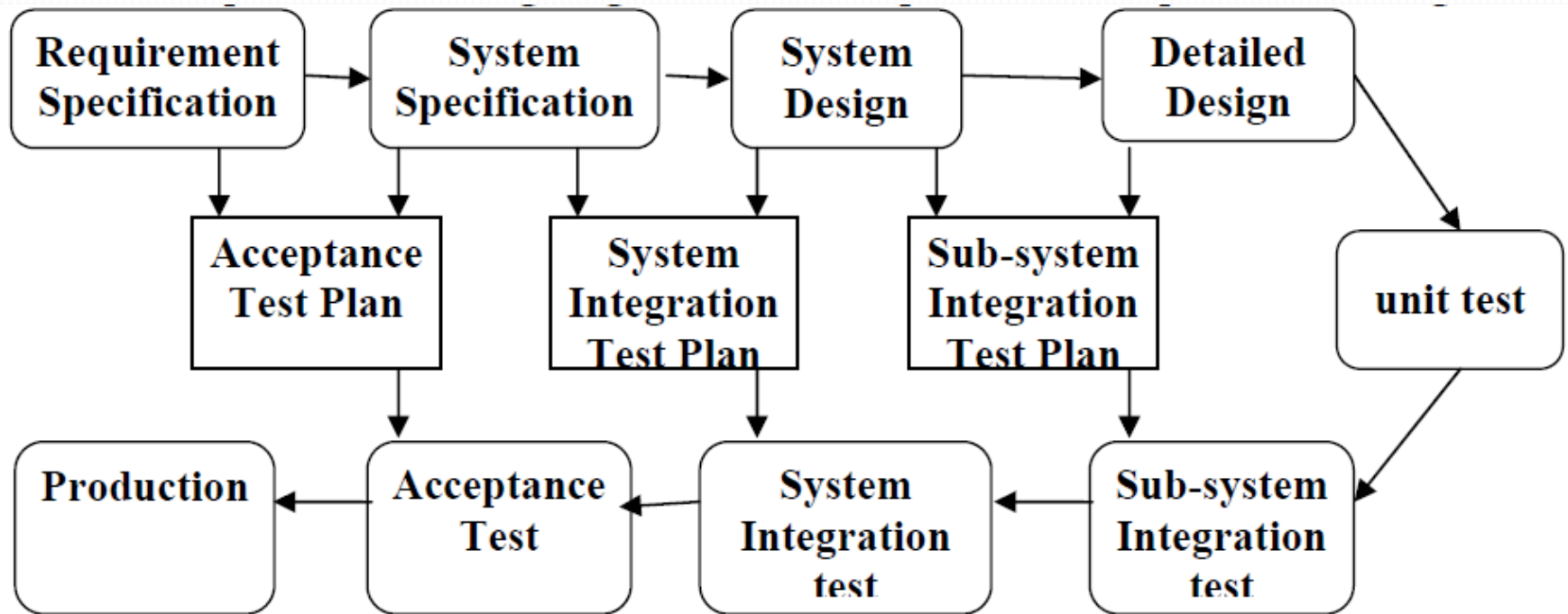
# Testing vs Development



# Developer vs Tester

Development	Testing
Development is a creative activity	Testing is a destructive activity
Objective of development is to show that the program works	Objective of testing is to show that the program does not work

# Testing in stages





# Testing stages

- ❖ Unit testing – testing individual components independent of other components.
- ❖ Module testing – testing a collection of dependent components
- ❖ Subsystem testing – testing of collection of modules to discover interfacing problems among interacting modules.
- ❖ System testing – after integrating subsystems into a system – testing this system as a whole.
- ❖ Acceptance test – validation against user expectations.
- ❖ Alpha testing – in-house testing for products.
- ❖ Beta testing – testing of product with potential customers who agree to use it and report problem before system is released for general use

## Alpha Testing

## Beta Testing

Alpha testing performed by Testers who are usually internal employees of the organization

Beta testing is performed by Clients or End Users who are not employees of the organization

Alpha Testing performed at developer's site

Beta testing is performed at client location

Alpha testing involves both the white box and black box techniques

Beta Testing typically uses black box testing

Alpha testing requires lab environment or testing environment

Beta testing doesn't require any lab environment or testing environment. Software is made available to the public and is said to be real time environment

# Verification Methods

- ❖ **Static Verification (Inspection)**
- ❖ **Dynamic Verification (Testing)**

# Inspection vs Testing

1. Inspections and testing are complementary and not opposing verification techniques.
2. Both should be used during the verification and validation process.
3. Inspections can check conformance with a specification but not conformance with the customer's real requirements.
4. Inspections cannot check non-functional characteristics such as performance, usability, etc.
5. Inspection does not require execution; maybe used before implementation.

# Inspection and Chaotic Zone

