

## Session 2

### Preliminary to Software Testing

chengbaolei@suda.edu.cn

## Objectives

- ◆ Review of Session 1
  - ◆ Why we need SQA
  - ◆ Definition of SQA
  - ◆ QA VS. QC
  - ◆ Testing overview: what, who, when

◆ A question:

Testing is a subset of quality assurance or  
quality control?

## Objectives

- ◆ In this session, you will learn:
  - ◆ Comparison of some concepts
  - ◆ Testing life cycle
  - ◆ Model of testing
  - ◆ Types of testing

## Comparison of some concepts

- ◆ What is “defect”?
- ◆ Verification / Validation
- ◆ Testing / Debugging
- ◆ White-box testing / Black-box testing

## What is “defect”?

- **Error** (错误) : occurs in the process of **writing** a program.
- **Fault** (故障) : is the manifestation of one or more errors. (**internal state**)
- **Failure** (失效) : occurs when faulty piece of code is executed leading to an incorrect state that propagates to the program's **output**.
- **Incident** (事故) : **no message** is displayed when failure occurs.



# Software Testing and Quality Assurance

- void test(int iLength, char iData[ ])

```
0 {  
1   int x=0;  
2   int y=0;  
3   for(int loop=0; loop<iLength; loop++)  
4   {  
5     if("A"==iData[loop])  
6       x++;  
7     else  
8       if("a"== iData[loop])  
9         x++;  
10      else  
11        y++;  
12    }  
13   printf ("%d,%d\n", x, y);  
14 }
```

```
char iData[]="John"  
char iData[]="Amy"  
char iData[]="Anna"
```

# Software Testing and Quality Assurance

## First Bug- 1947

### Harvard University Mark II Aiken Relay Calculator

9/9

0800 Action started  
1000 . stopped - action ✓  
1300 (032) MP - NC { 1.2700 9.037 847 025  
023 PR0 2 9.037 846 795 const  
1300 (032) 1.130476415 (023) 4.615925059 (-2)  
const 2.130476415  
const 2.130476415

Relays 6-2 in 033 failed special speed test  
in relay . 10.000 test .

1100 Started Cosine Tape (Sine check)  
1525 Started Multi Adder Test.

1545



Relay #70 Panel F  
(moth) in relay.

1600 Action started.  
1700 closed down .



Photo # NH 96919-KN Commodore Grace M. Hopper, 1984

## What is “defect”?

- Software defect definition
  - All kinds of software problems
  - Exist in code, **data** and **documents**
  - Out of accord with user expectation

## What is “defect”?

- Defect example:
  - Software function cannot be executed correctly
  - Shortcoming
    - running slowly
  - Inconsistency
    - Ctrl+S can't save all applications
  - User interface design defect
    - a button should show 5 words on it, but only 3 words could be seen

# Software Testing and Quality Assurance

## Bug Kinds (diff. kinds of errors)

- **Syntactic** errors:

- Mal-formed program:

```
int square(int x) {  
    return x*x  
} *** syntax error at line 2  
          ';' expected
```

- **Semantic** errors:

- **Symbol** errors
- **Type** errors
- Other semantic errors:  
(e.g. uninitialized vars)

```
int square(int x) {  
    return n*n;  
} *** symbol error at line 2  
          undefined variable "n"
```

```
int square(float x) {  
    return x*x;  
} *** type error at line 2  
          function returns float, not int
```

- **Logical** errors:

- Compiler: "no errors"

```
int square(int x) {  
    return x+x;  
}
```

no errors found!!!

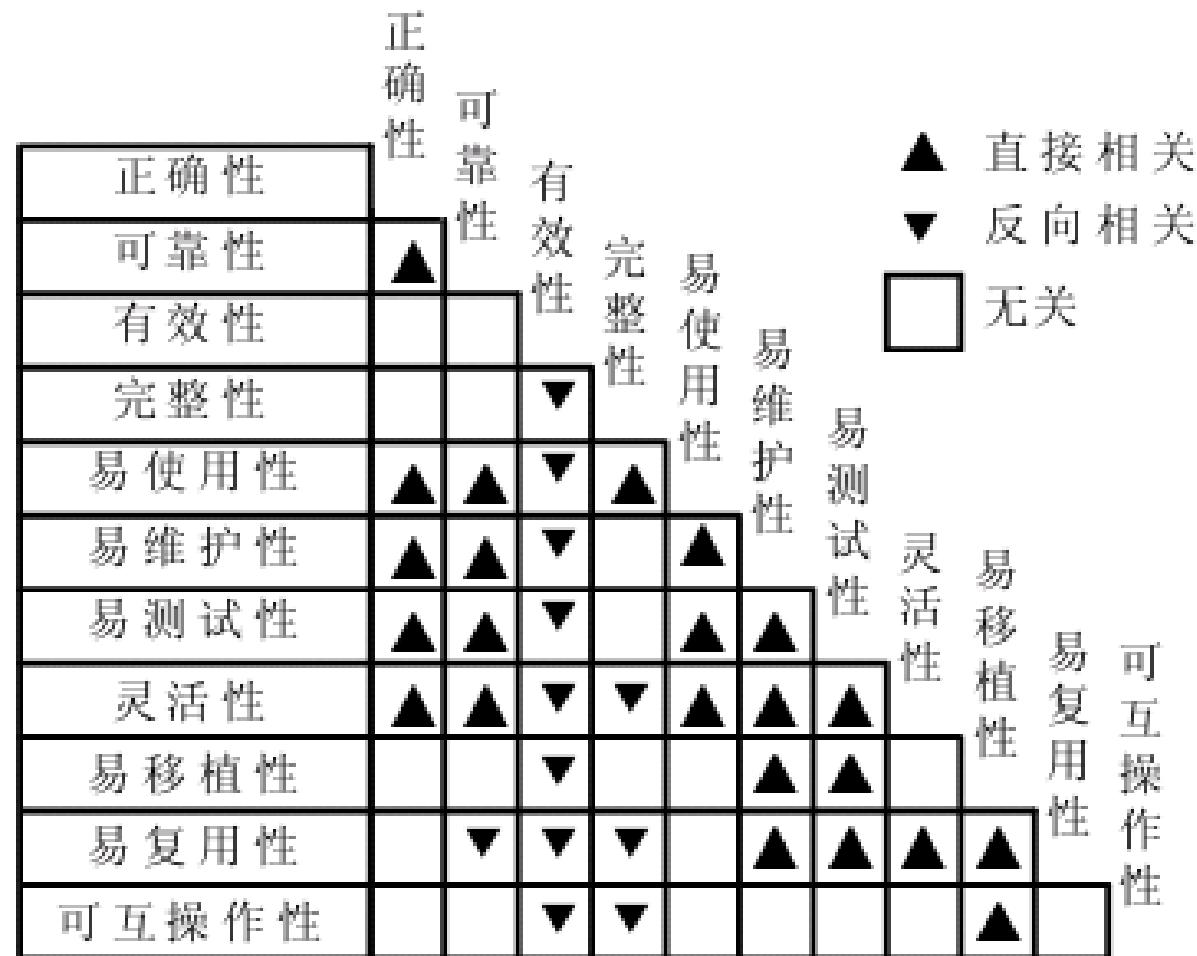
- Software Quality Attributes
  - McCall (1977)
  - Boehm (1978)

- McCall's Quality Model - 1977
  - Product operations (basic operational characteristics).
  - The product operations perspective identifies quality factors that influence the extent to which the software fulfils its specification:
    - Correctness, the functionality matches the specification.
    - Reliability, the extent to which the system fails.
    - Efficiency, system resource (including cpu, disk, memory, network) usage.
    - Integrity, **protection from unauthorized access.**
    - Usability, ease of use.(Usability requirements deal with the scope of staff resources needed to train a new employee and to operate the software System)

- Product revision (ability to change).
- The product revision perspective identifies quality factors that influence the ability to change the software product, these factors are:
  - Maintainability, the ability to find and fix a defect.
  - Flexibility, the ability to make changes required as dictated by the business.
  - Testability, the ability to Validate the software requirements.

- Product transition (adaptability to new environments).
- The product transition perspective identifies quality factors that influence the ability to adapt the software to new environments:-
  - Portability, the ability to transfer the software from one environment to another.
  - Reusability, the ease of using existing software components in a different context.
  - Interoperability, the extent, or ease, to which software components work together.

## 软件质量要素之间的关系



- 某软件系统的需求文档中包括规格：编程遵守公司的编码标准和指南，该需求属于哪个软件质量因素？  
A. 可维护性    B. 可测试性    C. 正确性    D. 灵活性
- 软件质量的主要特性中，在功能改变和扩充的情况下，软件能够正常运行的能力是哪一个？  
A. 可靠性    B. 安全性    C. 可扩展性    D. 功能性
- 某软件系统的需求文档中包括规格：普通用户可以通过Internet访问GIS（地理信息系统）文件，但不得在所访问的地图中插入更改，该需求属于哪个软件质量因素？  
A. 可维护性    B. 完整性    C. 正确性    D. 灵活性
- 某软件系统的需求文档中包括规格：系统可在稍后经过修改后适合其它的同行单位使用，该需求属于哪个软件质量因素？  
A. 可靠性    B. 安全性    C. 正确性    D. 灵活性

## 【判断题】

- 1、故障是指编码错误的客观存在，然而编码错误仍处于内部状态，不一定被执行且尚未传递到外部输出。
- 2、软件缺陷不仅仅存在于代码中，还有可能存在于数据、文档等各类软件项中。
- 3、在系统设计过程中，应对各种要素进行折衷以便得到总体上用户和开发人员都满意的质量标准。

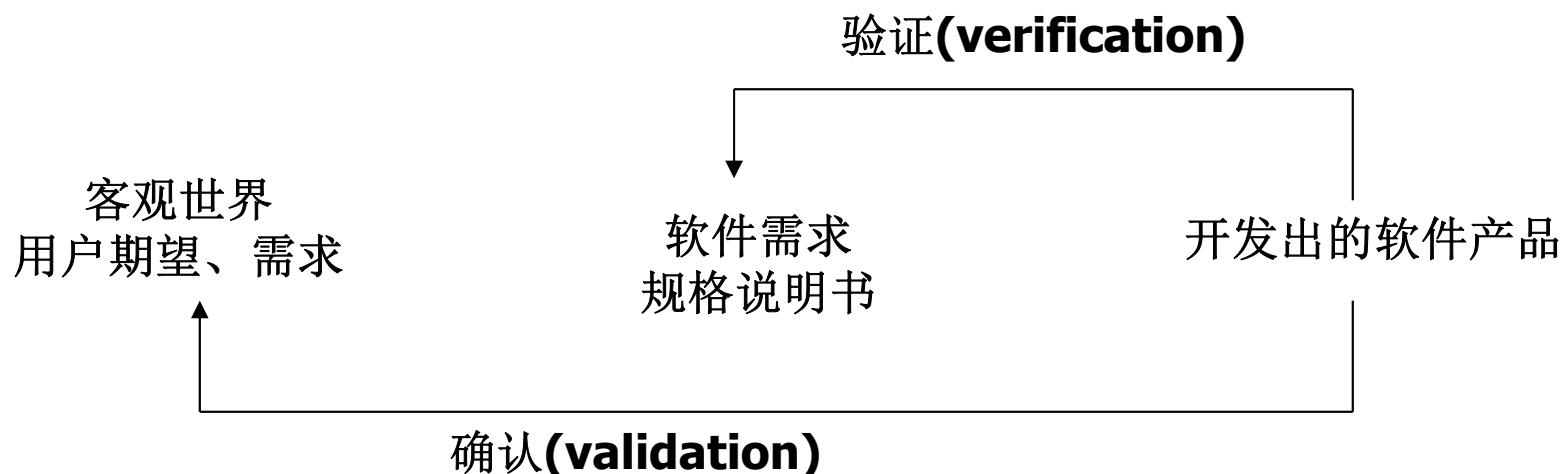
软件缺陷的含义广泛，以下哪些是软件缺陷的表现形式？（ ）

- A.系统崩溃
- B.实际结果和预期结果不一致
- C.响应时间过长
- D.界面不美观
- E.数据精度不

# Software Testing and Quality Assurance

## Verification vs. validation

- **Verification** (验证) : The software should conform to its specification  
(Are we building the product right?)
- **Validation** (确认) : The software should do what the user really requires  
(Are we building the right product?)



# Software Testing and Quality Assurance

## Verification vs. validation

S.N.	Verification	Validation
1	Are you building it right?	Are you building the right thing?
2	Ensure that the software system meets all the functionality.	Ensure that functionalities meet the intended behavior.
3	Verification takes place first and includes the checking for documentation, code etc.	Validation occurs after verification and mainly involves the checking of the overall product.
4	Done by developers.	Done by Testers.
5	Have static activities as it includes the reviews, walkthroughs, and inspections to verify that software is correct or not.	Have dynamic activities as it includes executing the software against the requirements.

**1 meaning;  
2 purpose;  
3 time;  
4 participant;  
5 activities**

## Testing vs. debugging

### **Testing :**

**It involves the identification of bug/defect in the software without correcting it.**

**Testing is performed in the testing phase.**

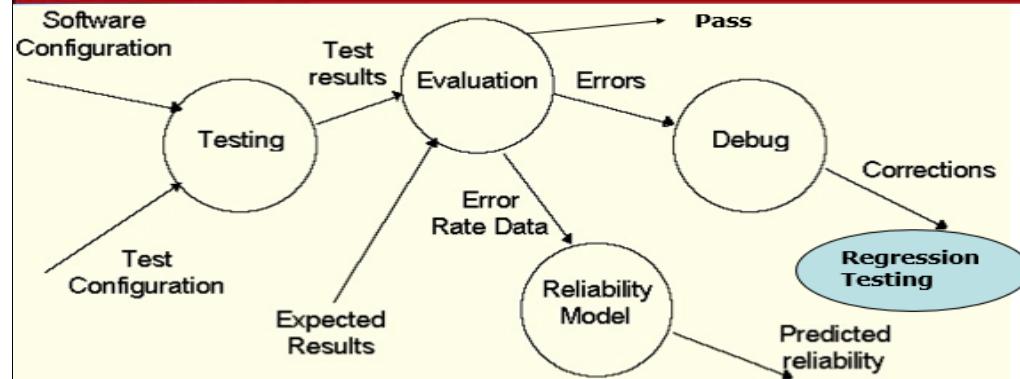
### **Debugging :**

**It involves identifying , isolating and fixing the problems/bugs .**

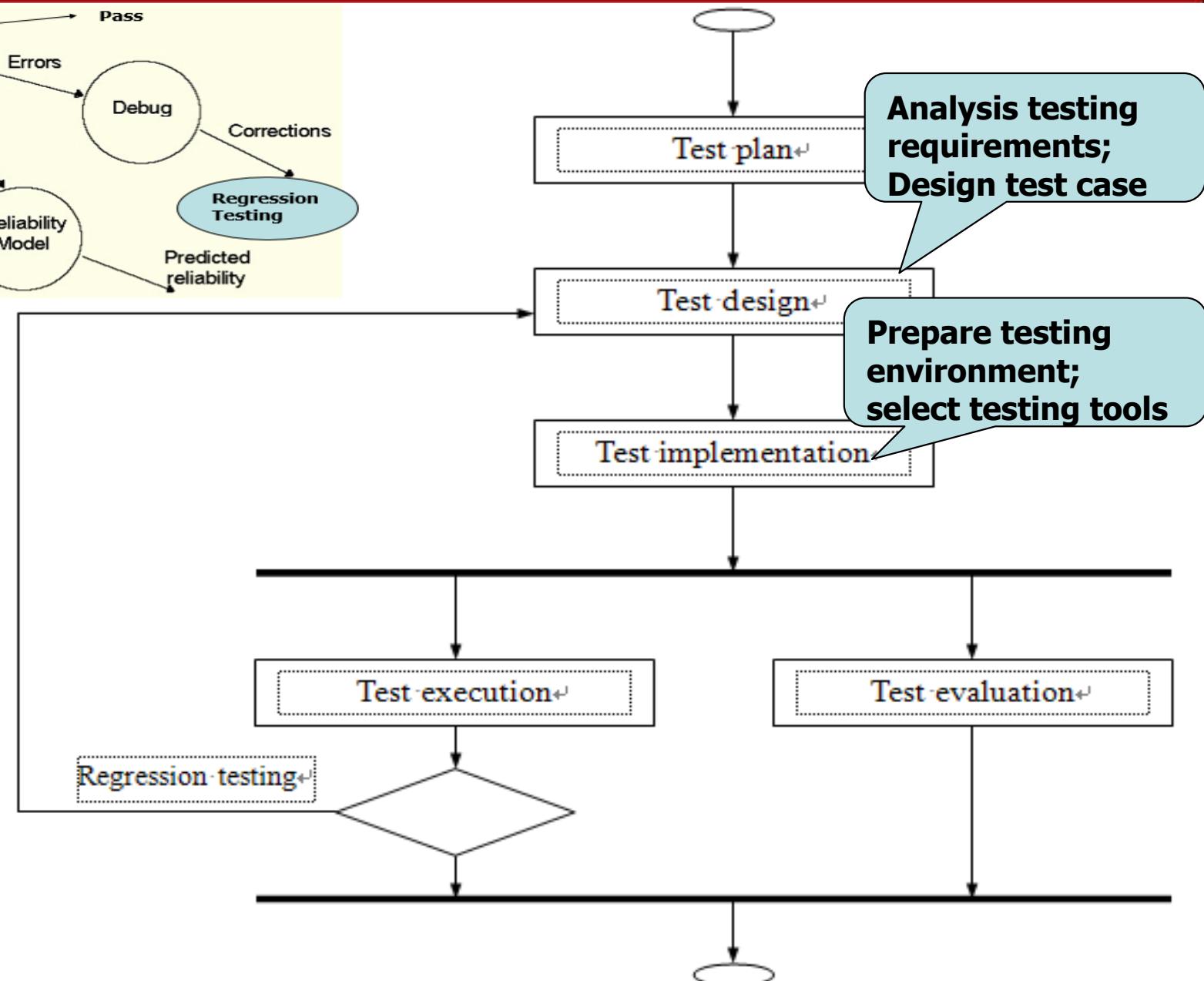
**Debugging can be performed in the development phase while conducting Unit Testing or in phases while fixing the reported bugs.**

# Software Testing and Quality Assurance

## Testing Life Cycle



**PDCA**



## White-box vs. Black-box Test

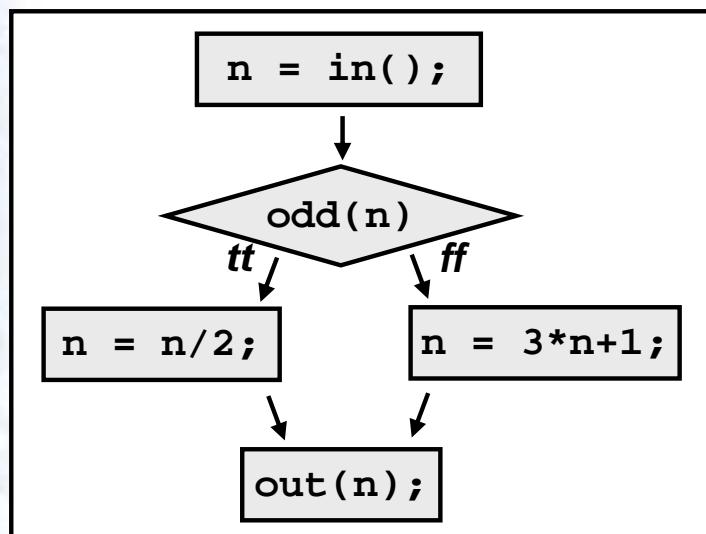
***Test Case:***

- To test whether a component conforms to project specification – **Black Box Testing**
- To test whether a component conforms to development design – **White Box Testing**

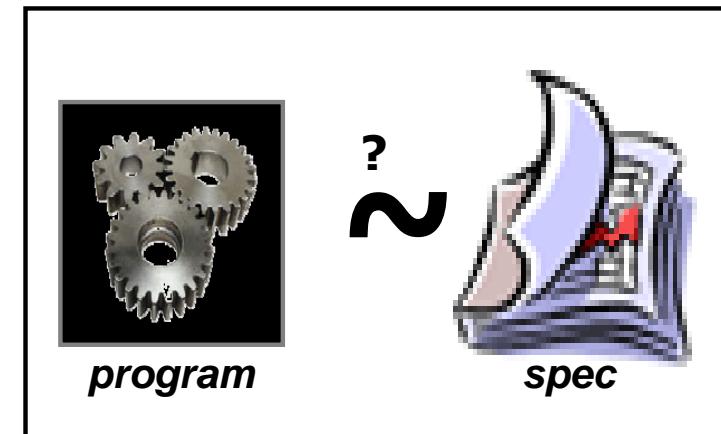
# Software Testing and Quality Assurance

## White-box vs. Black-box Test

- **White-box Testing:**
  - *structural testing*
  - *internal testing*
- Test focus:
  - **source code / design**



- **Black-box Testing:**
  - *behavioral testing*
  - *external testing*
  - *input-output testing*
- Test focus:
  - **specification (or intention)**



**Complementary Approaches!!!**

## Objectives

- ◆ In this session, you will learn:
  - ◆ Comparison of some concepts
  - ◆ Testing life cycle
  - ◆ Model of testing
  - ◆ Types of testing

## Objectives

- ◆ In this session, you will learn:
  - ◆ Comparison of some concepts
  - ◆ Testing life cycle
  - ◆ Model of testing
  - ◆ Types of testing

## Model of testing

- ◆ The V model
- ◆ The W model

## The V Model

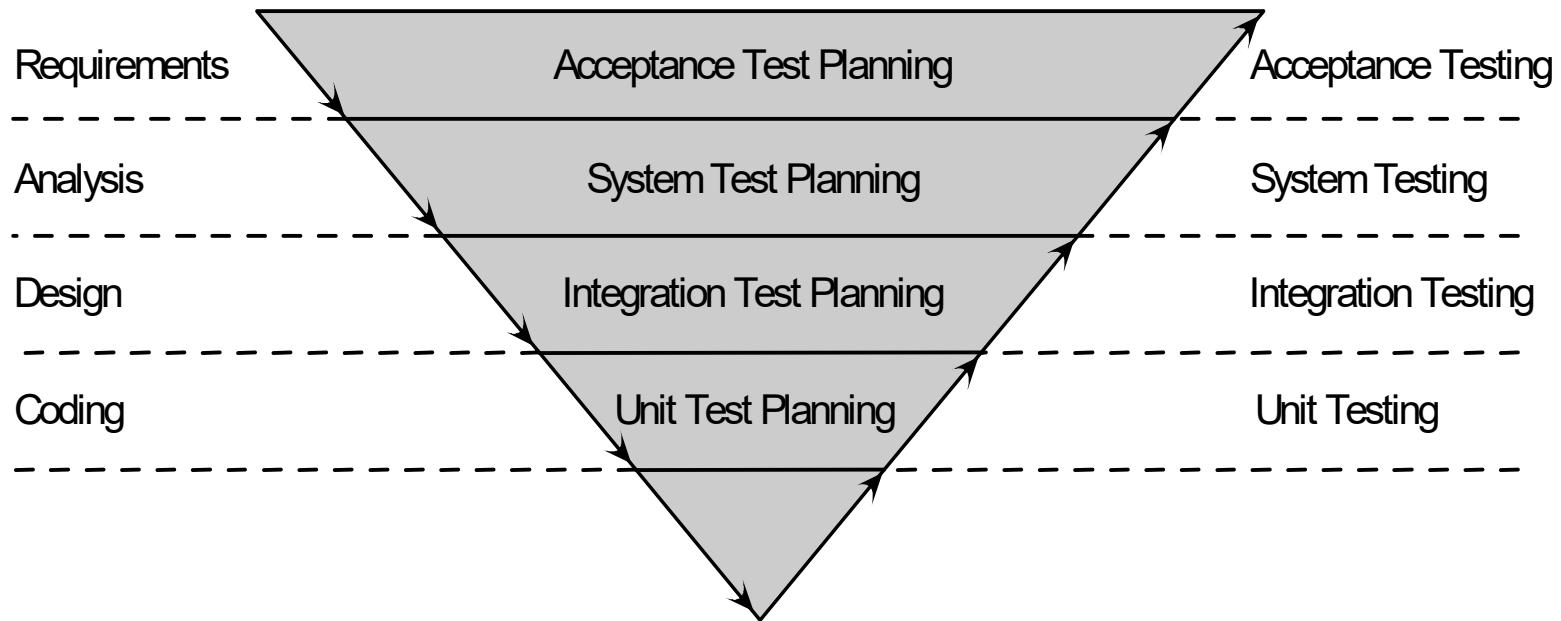
- ◆ To reduce the cost of correcting defects, the defects must be detected early in the development life cycle.
- ◆ The earlier, the better.

## The V Model

- ◆ The V model proposes an approach to software development in which both the software development process and the software test process begin simultaneously.
- ◆ When the project starts, the development team starts the software development process and the **testing team starts planning** for the test process.
- ◆ This planning is based on the documents created during the development process.

# Software Testing and Quality Assurance

## The V Model (Contd.)



- ◆ The V model places the development phases such as requirements, analysis, design, and coding on one side of the V.
- ◆ The various types of testing such as unit, integration, system, and acceptance, are placed on the other side of the V.

## The V Model (Contd.)

- ◆ What is the disadvantages of the V model?
- ◆ The V model focuses only on **dynamic testing** and does not mention the benefits and effectiveness of static testing techniques such as reviews, inspections, and static code analysis.

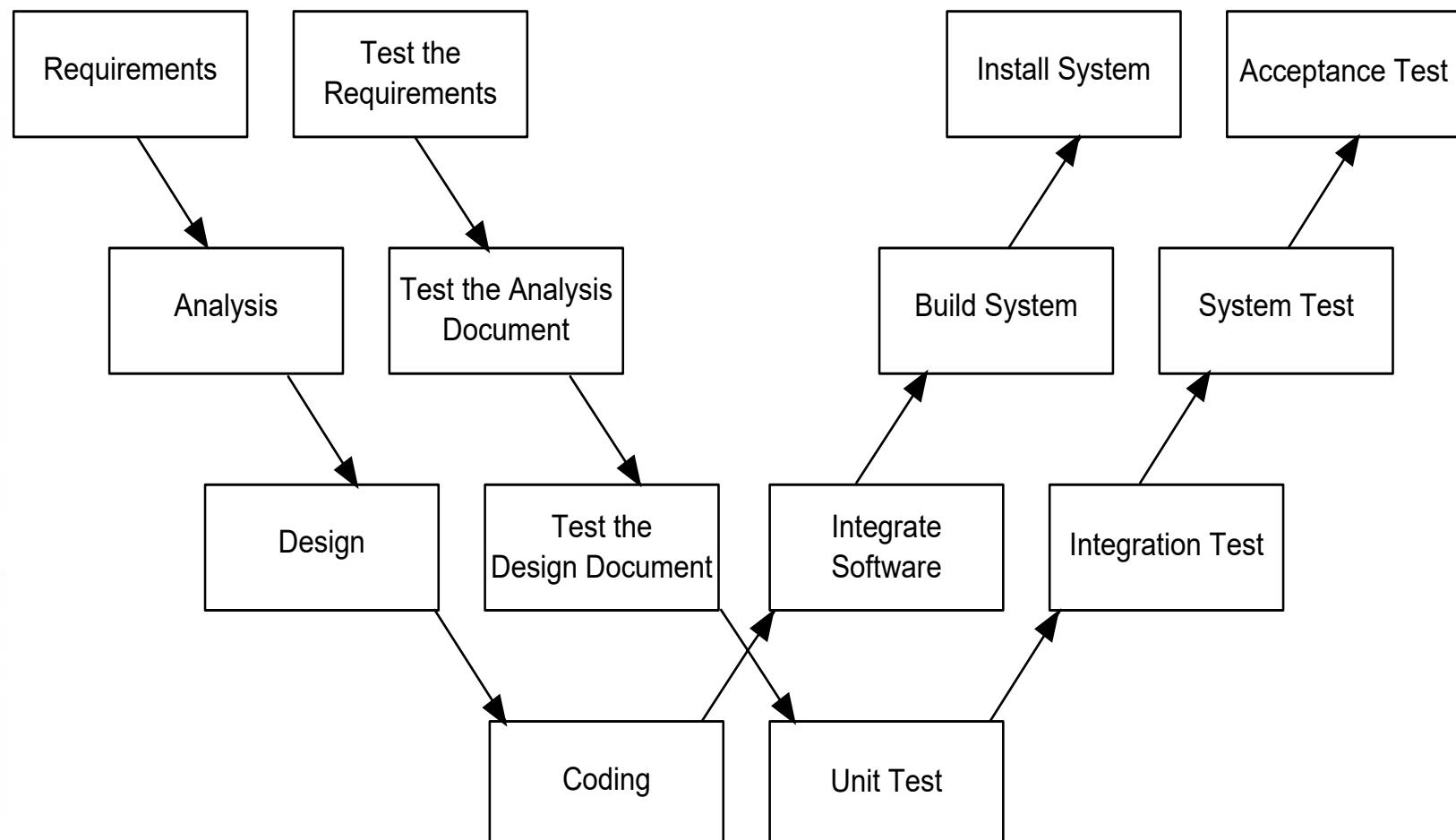
## The W Model

- ◆ The W model is an extension of and addresses the limitations of the V model.
- ◆ It focuses on a testing methodology where **static testing** techniques are applied during the various stages of development. This results in tests being performed in parallel with the development process.
- ◆ These techniques **are cheaper and more effective** than dynamic testing.

# Software Testing and Quality Assurance

## The W Model (Contd.)

- ◆ The following figure shows the W model of testing.



# Software Testing and Quality Assurance

## The W Model (Contd.)

- ◆ The following table describes the test activities performed in each phase of the development life cycle.

Phase of the development life cycle	Test Activity
Requirements	Test the requirements Acceptance Test Plan
High-Level Design	Test the design documents System Test Plan Integration Test Plan
Detailed Design	Test the design documents Unit Test Plan
Coding	Unit Testing
Integration	Integration Testing System Testing
Acceptance	Acceptance Testing

## The W Model

- ◆ The W model places the test process on an equal footing with the development process.
- ◆ The W model adopts both of static testing and dynamic testing.
- ◆ Recall verification & validation.

## Objectives

- ◆ In this session, you will learn:
  - ◆ Comparison of some concepts
  - ◆ Testing life cycle
  - ◆ Model of testing
  - ◆ Types of testing

## Types of testing

- Types of testing
  - C1:Source of test generation
  - C2:Life cycle phase in which testing takes place
  - C3:Goal of a specific testing activity

## Types of testing

- C1:Source of test generation

Artifact	Technique
Requirements (informal)	Black-box
Code	White-box
Requirements and code	Black-box and white-box
Formal model: graphical or mathematical specification	Model-based specification
Component's interface	Interface testing

## Types of testing

- C2:Life cycle phase

Phase	Technique
Coding	Unit testing
Integration	Integration testing
System integration	System testing
Post system, pre-release	Alpha/Beta-testing /Acceptance testing
Maintenance	Regression testing

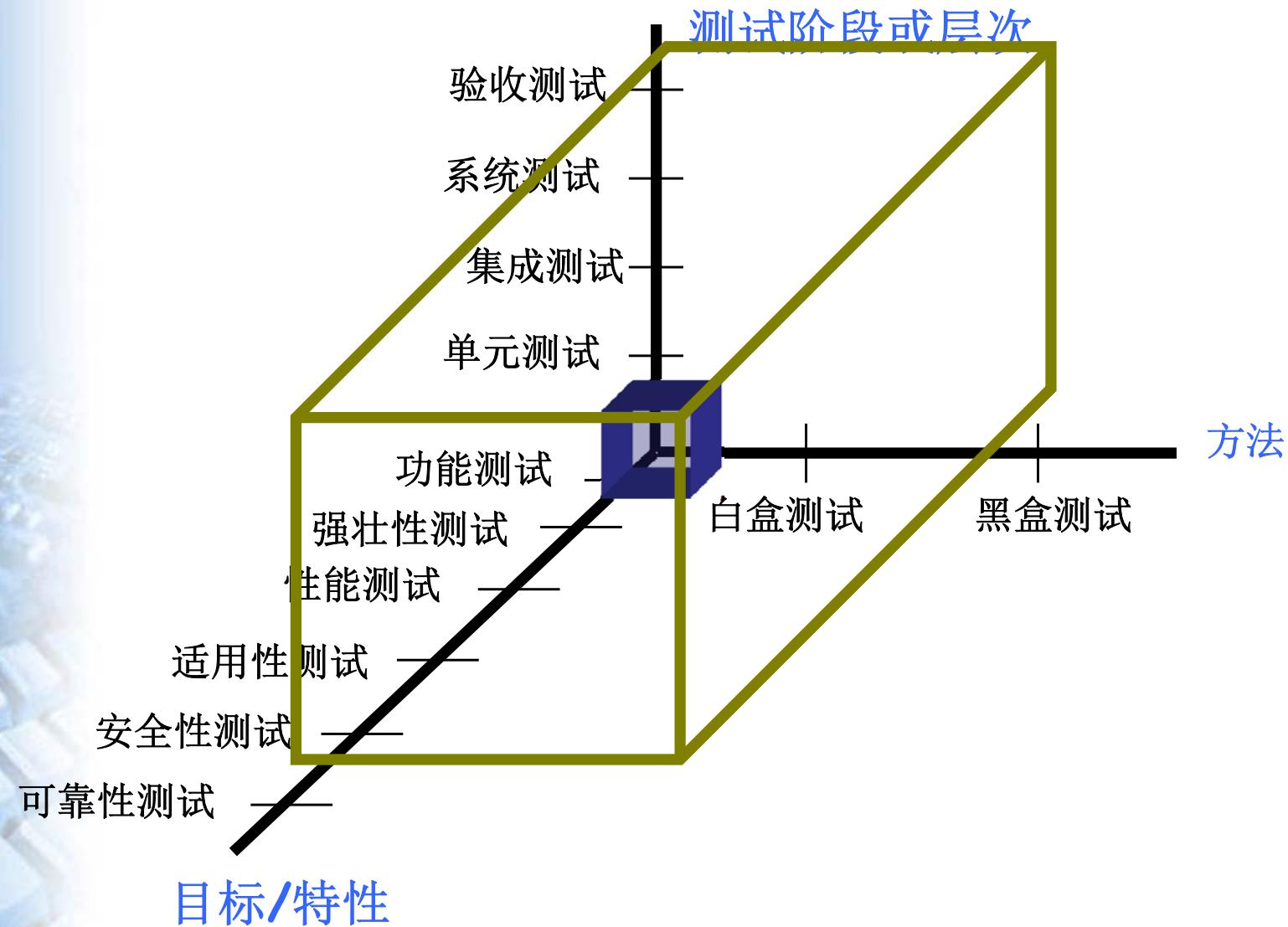
## Types of testing

- C3:Goal-directed testing

Goal	Technique
Advertised features	Functional
Security	Security
Invalid inputs	Robustness
Errors in GUI	GUI
Operational correctness	Operational
Reliability assessment	Reliability
System performance	Performance
Customer acceptability	Acceptance
Business compatibility	Compatibility
Peripherals compatibility	Configuration
Foreign language compatibility	Foreign language

# Software Testing and Quality Assurance

## Types of testing



## Summary

- ◆ In this session, you learned that:
  - ◆ Comparison of some concepts
  - ◆ Testing life cycle
  - ◆ Model of testing
  - ◆ Types of testing

◆ Questions:

- 1 Testing is a subset of quality assurance or quality control?
- 2 Testing life cycle
- 3 V model and W model