

The background of the slide features a light blue world map. Overlaid on the map are several geometric shapes: a large white triangle with horizontal blue lines on the left, a solid green triangle pointing downwards on the right, and a small blue diamond at the bottom center. A horizontal bar with segments of dark blue, teal, and blue runs across the middle of the slide.

# 软件测试与质量管理

## Software Testing and Quality Management

### 第2章 软件测试基础

# 上节要点

---

## 软件测试概述

- 软件测试的发展
- 软件测试的定义
- 为什么要进行软件测试
- 软件测试的目的
- 软件测试的误区

# 主要内容

- 2.1 软件测试的对象
- 2.2 软件测试原则
- 2.3 软件测试分类
- 2.4 软件测试过程模型
- 2.5 软件测试方法介绍
- 2.6 白盒测试用例设计
- 2.7 黑盒测试用例设计

## 2.1 软件测试对象

- 1. 软件定义

软件=程序+数据+文档

### 2. 软件测试对象

——软件测试不等于程序测试。

——软件测试贯穿于软件定义和开发的整个过程。

——软件开发过程中所产生的需求规格说明、概要设计规格说明、详细设计规格说明以及源程序都是软件测试的对象。

## 2.2 软件测试原则

- 软件测试，从不同的角度出发会派生出两种不同的测试原则：
- 从用户的角度出发，就是希望通过软件测试能充分暴露软件中存在的问题和缺陷。
- 从开发者的角度出发，就是希望测试能表明软件产品不存在错误，已经正确地实现了用户的需求。
- 为了达到上述原则，就应注意以下几点：



## 2.2 软件测试原则

- 一 所有的软件测试都应追溯到用户需求。
- 二 应当把“尽早地和不断地进行软件测试”作为软件测试者的座右铭。
- 三 完全测试是不可能的，测试需要终止。
- 四 测试无法显示软件潜在的缺陷。
- 五 充分注意测试中的群集现象。
- 六 程序员应避免检查自己的程序。
- 七 尽量避免测试的随意性。

## 2.3 软件测试分类

---

- 按照开发阶段划分

1. 单元测试

2. 集成测试

3. 确认测试

4. 系统测试

5. 验收测试

## 2.3 软件测试分类（续一）

- 按照测试实施组织划分

1. 开发方测试

2. 用户测试

3. 第三方测试



## 2.3 软件测试分类（续二）

- 按照测试技术划分

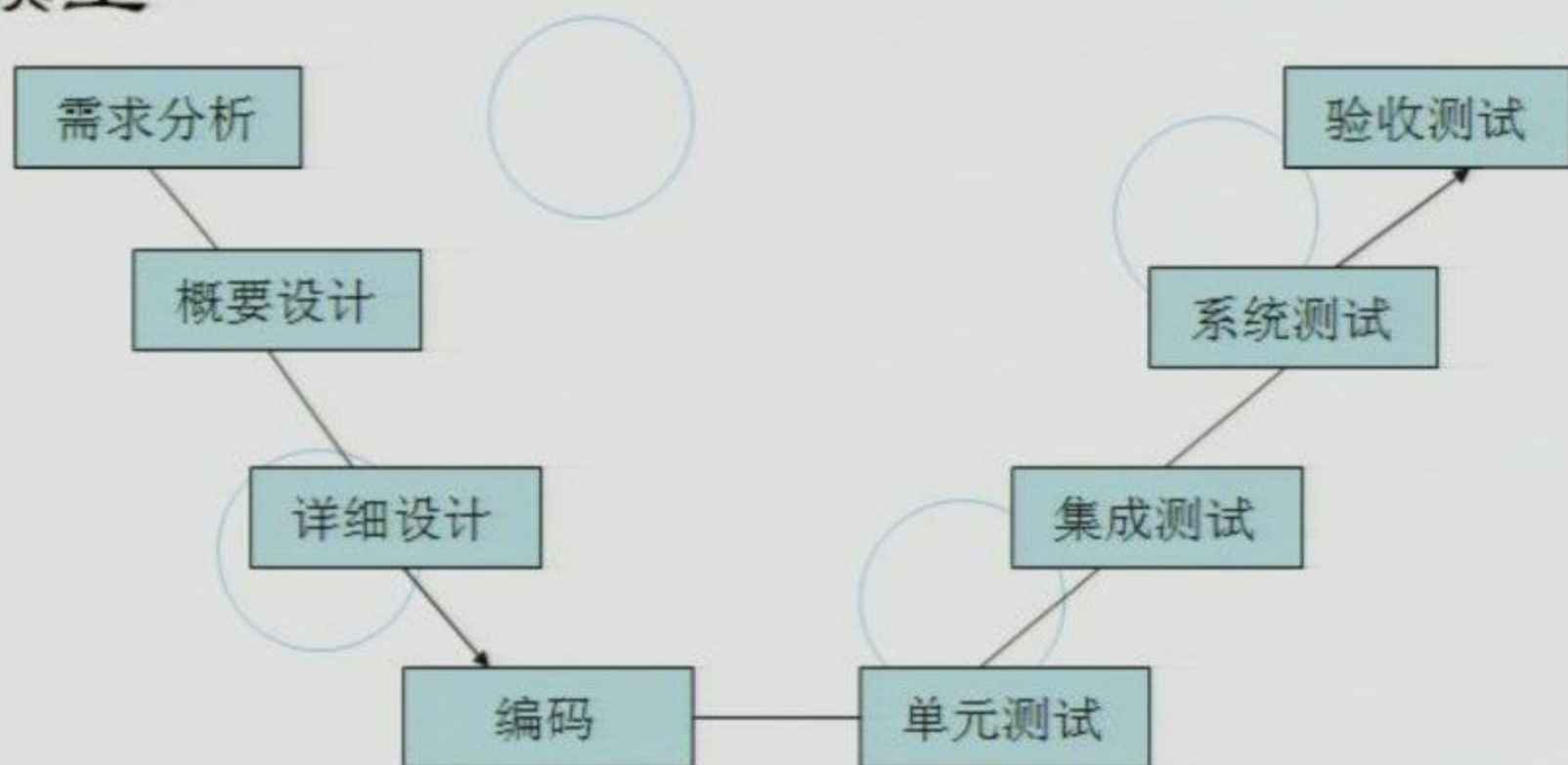
1. 白盒测试

2. 黑盒测试

3. 灰盒测试

## 2.4 软件测试过程模型

- V模型



## 2.4 软件测试过程模型（续一）

在传统开发过程中测试不受重视，仅把它作为在需求分析、概要设计、详细设计及编码之后的一个阶段。尤其在瀑布模型中。

- (1) 每个开发活动都有右边的测试活动相对应。
- (2) 软件开发过程是一个自顶向下，逐步细化的过程。
- (3) 测试过程是依相反顺序安排的自底向上，逐步集成的过程。

## 2.4 软件测试过程模型（续二）

- W模型

由于各种原因，开发的每一个环节都可能产生错误，如果坚持各个阶段的技术评审，就能够尽早发现和预防错误。

图2-2为软件开发与测试的W模型，形象地说明了软件测试与开发的这种同步性。



## 2.4 软件测试过程模型（续三）

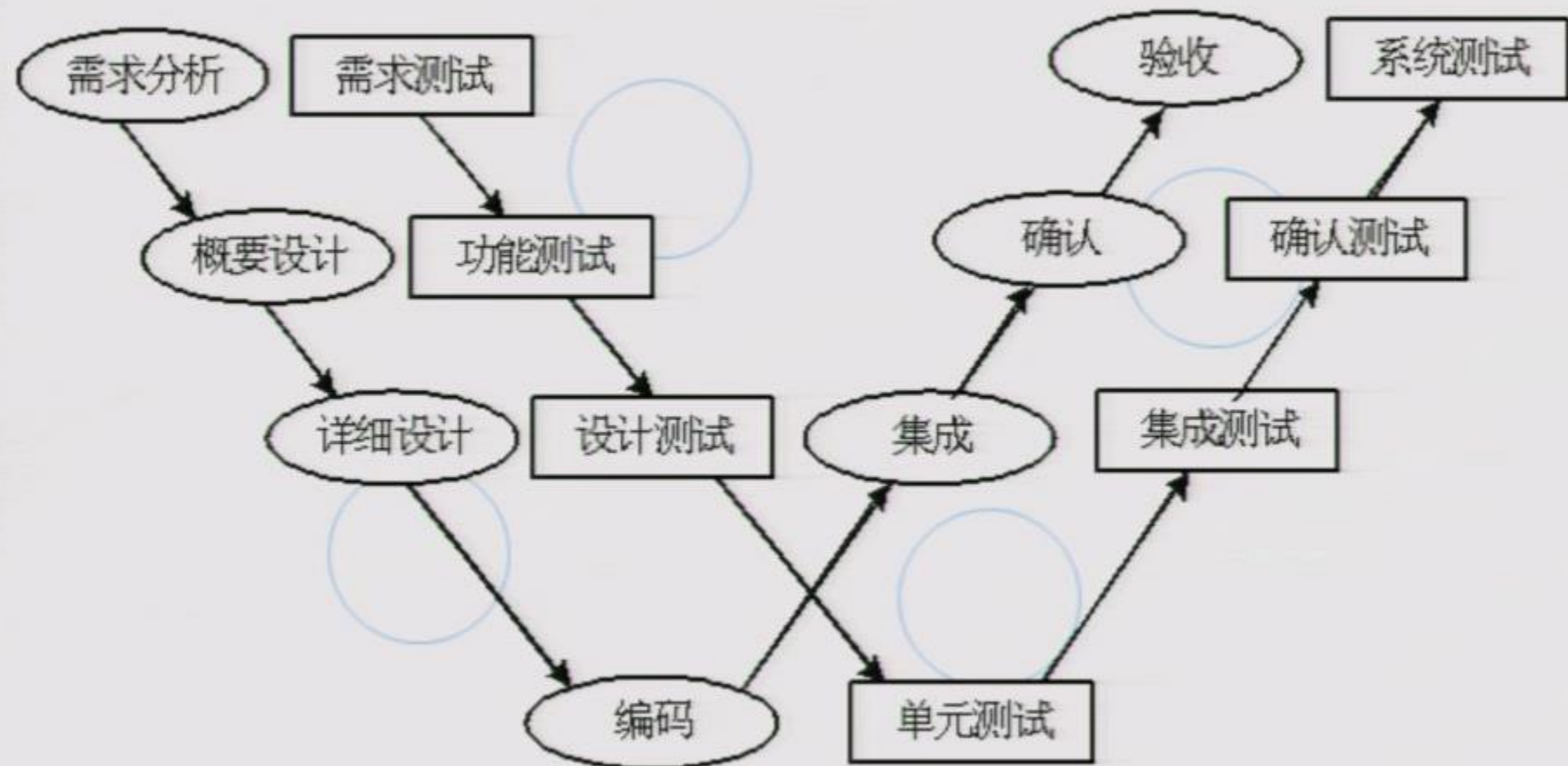


图2-2 W模型示意图



## 2.4 软件测试过程模型（续四）

- H模型

与前两种模型相比，H模型充分地体现了测试过程。如图2-3所示的H模型揭示了：

- 1、软件测试不仅仅指测试的执行，还包括很多其他的活动。

- 2、软件测试是一个独立的流程，贯穿产品的整个开发周期，与其它流程并发进行。

- 3、软件测试要尽早准备，尽早执行。

- 4、软件测试根据被测物的不同是分层次的，不同层次的测试活动可以是按照某个次序先后进行的，但也可能是反复的。

## 2.4 软件测试过程模型（续五）

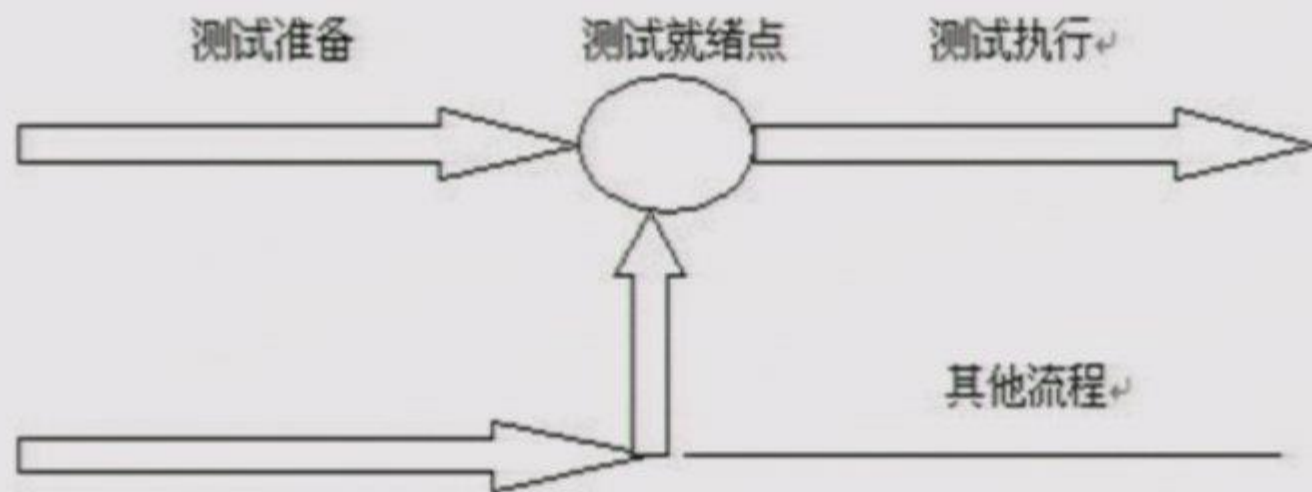


图2-3 H模型示意图

## 2.5 软件测试方法介绍

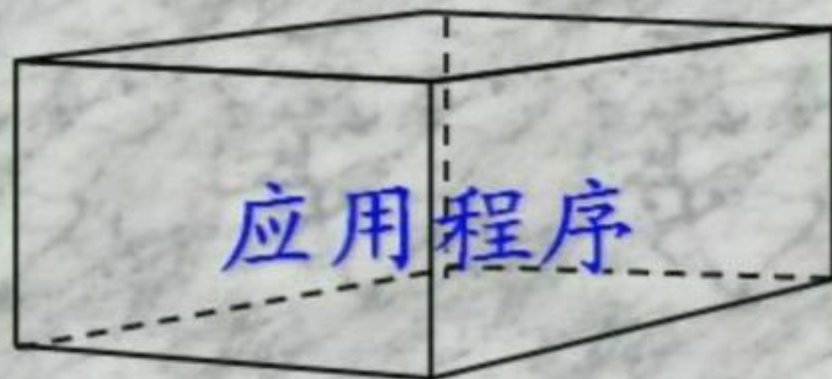
### 一、白盒测试

- 白盒测试将被测程序看作一个打开的盒子，测试者能够看到被测源程序，可以分析被测程序的内部结构，此时测试的焦点集中在根据其内部结构设计测试用例。
- 白盒测试要求是对某些程序的结构特性做到一定程度的覆盖，或者说这种测试是“基于覆盖率的测试”。



## 2.5 软件测试方法介绍（续一）

- 白盒测试



白盒测试需要完全了解程序结构和处理过程，它按照程序内部逻辑测试程序，检验程序中每条通路是否按预定要求正确工作。也被称为程序员测试。

## 2.5 软件测试方法介绍（续二）

- 测试者从系统内部进行监测。他们关心的是“怎样做的”而非“做了什么”。
  - 测试编码的分支和条件判定
  - 跟踪编程者的某条逻辑路径。
- 以逻辑为导向。测试者关心的是程序中控流的所有可能路径的执行情况。
- 几乎都是由开发人员来完成的。



## 2.5 软件测试方法介绍（续三）

- 优点

能够对程序内部特定部位进行覆盖测试。

- 缺点

无法检验程序的外部特征，无法对未实现需求的程序欠缺部分进行测试。

- 用途

一般用在测试过程的早期阶段，由程序员负责设计和执行。

## 2.5 软件测试方法介绍（续四）

### 二、黑盒测试

- 把软件当作一个黑盒子，不考虑程序内部结构和特征，而基于产品功能来规划测试，检查程序各功能是否实现，并检查其中错误的测试方法。
- 黑盒测试又称功能测试或数据驱动测试，是基于用户需求的测试。

## 2.5 软件测试方法介绍（续五）

- 黑盒测试



黑盒测试是在程序接口进行测试，它只是检查程序功能是否按照规格说明书的规定正常使用。也被称为用户测试。



## 2.5 软件测试方法介绍（续六）

- 测试者是从系统之外进行监测。他们所关心的是“做了什么”而非“怎样做的”。
- 是以需求和详细使用说明为导向，从功能角度考虑的测试。
- 为了测试系统，人们把所有可能的输入组合输入系统，然后检测输出。有效和无效输入都要考虑到。

## 2.5 软件测试方法介绍（续七）

- 优点

能站在用户立场上进行测试。

- 缺点

不能测试程序内部特定部位，如果需求或规格说明有误，则无法发现。

- 用途

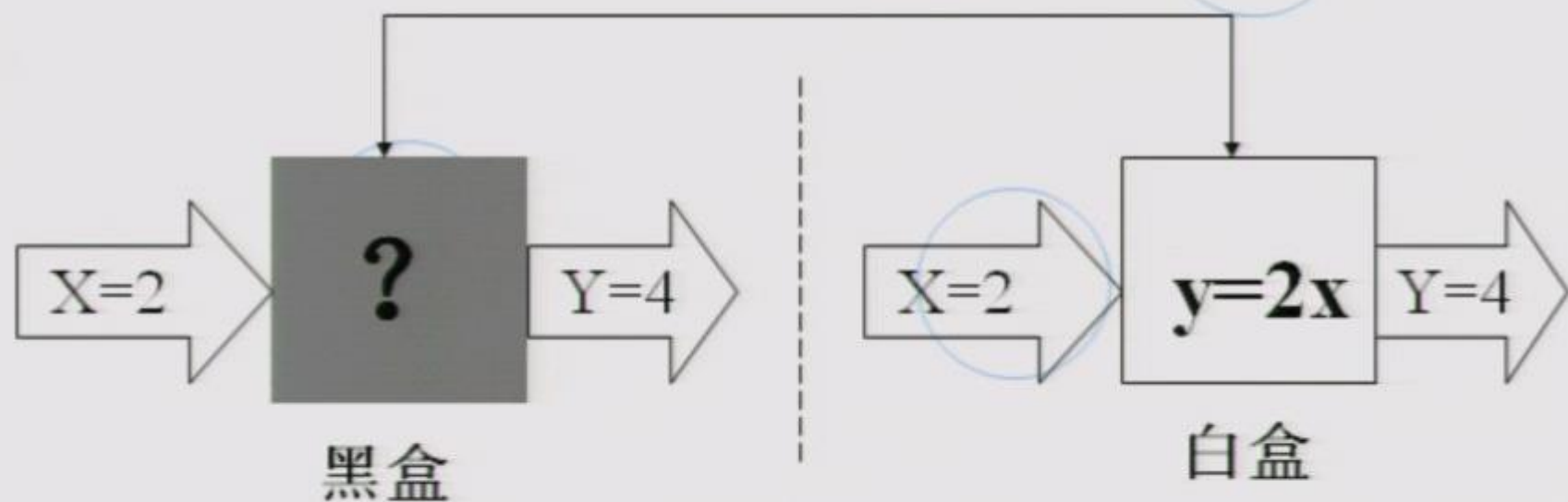
主要用于证明软件功能的正确性和可操作性；用于测试一些无法得到源程序的软件，如外购软件。



## 2.5 软件测试方法介绍（续八）

### 三、白盒测试和黑盒测试的比较

未知等式与已知等式



## 2.5 软件测试方法介绍（续九）

- 黑盒测试
  - 功能性
  - 关于行为的
  - 数据驱动的
  - 根据输入和各自的输出
- 白盒测试
  - 结构性的
  - 逻辑驱动
  - 测试基于对程序的了解

## 2.5 软件测试方法介绍（续十）

- 静态测试

- 确保系统按照组织的标准和过程运行，主要依赖于评审和非运行的手段来检查。通常包括需求评审、设计评审、代码走查和代码检查。

- 动态测试

- 在生命周期中进行测试（运行）。通常包括单元测试、集成测试、系统测试、用户的验收测试。

## 2.5 软件测试方法介绍（续十一）

- 静态测试和动态测试的比较

测试阶段	执行人	静态校验	动态校验
可行性评审	开发人员，用户	√	
需求评审	开发人员，用户	√	
设计评审	开发人员	√	
单元测试	开发人员		√
集成测试	开发人员，用户		√
系统测试	开发人员在用户的协助下完成		√
验收测试	用户		√

## 本节要点

---

- 软件测试对象
- 软件测试原则
- 软件测试分类
- 软件测试过程模型
- 软件测试方法介绍