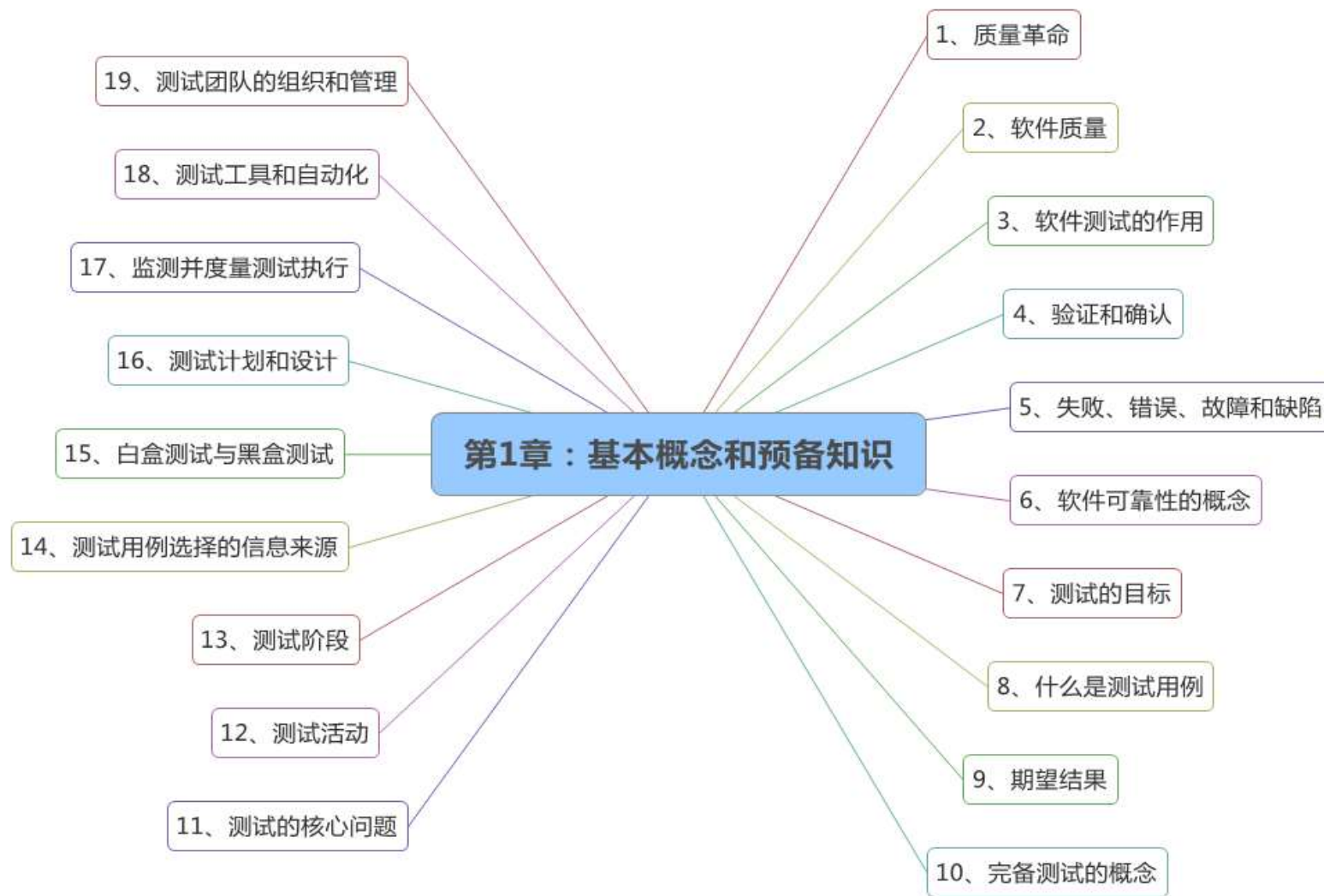


# 基本概念和预备知识

## Fundamentals

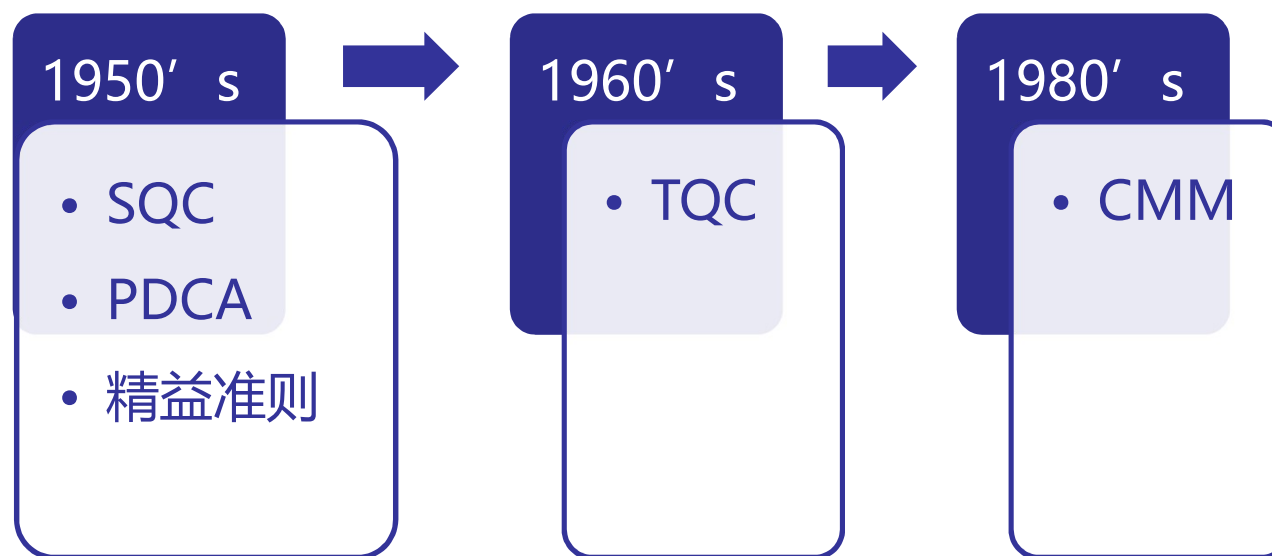
---

# 第1章：基本概念和预备知识

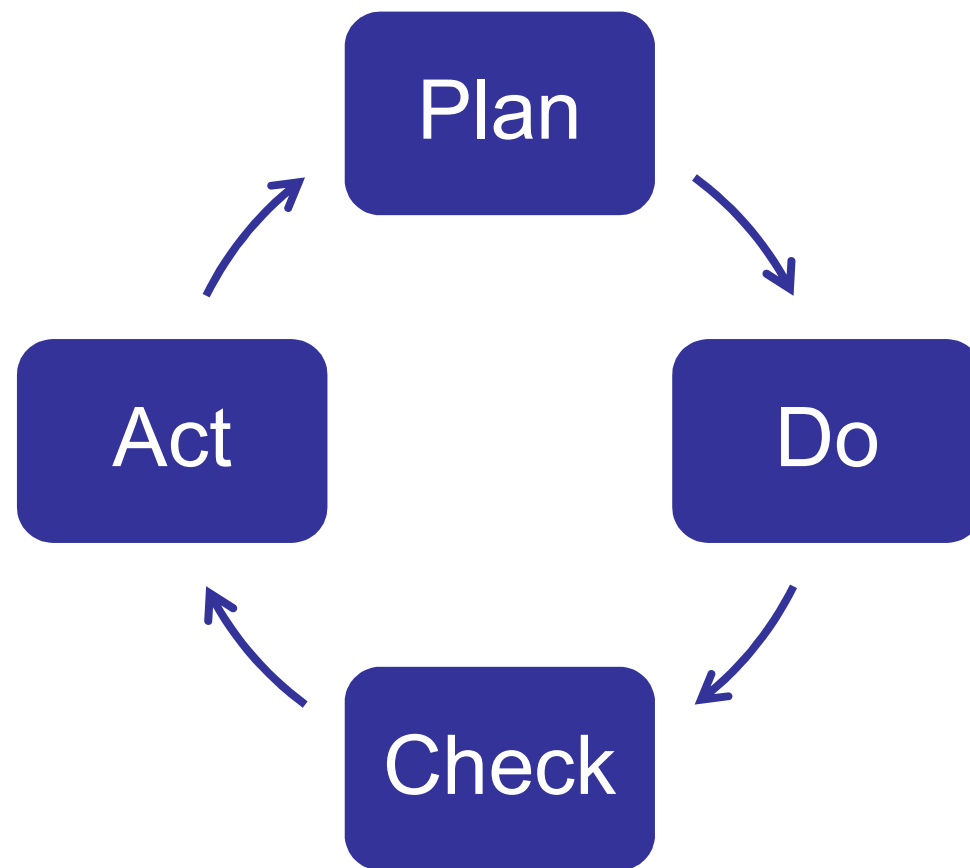


# 1.1 质量革命

**质量是任何一个公司能长久生存的关键因素！**



# 1.1 质量革命



Shewhart循环 (PDCA)

# 1.1 质量革命

**计划 (Plan)** 阶段，看哪些问题需要改进，逐项列出，找出最需要改进的问题

**执行 (Do)** 阶段，实施改进，并收集相应的数据

**检查 (Check)** 阶段，对改进的效果进行评价，用数据说话，看实际结果与原定目标是否吻合

**处理 (Act)** 阶段，如果改进效果好，则加以推广；如果改进效果不好，则进行下一个循环

Shewhart循环 (PDCA)

## 1.2 软件质量

什么是  
软件质  
量？

- 1、**抽象的观点**：设想质量是可以识别但很难定义的
- 2、**用户的观点**：把质量认知为对目标的适应性，即回答关键问题 “产品是否满足用户的需求和期望？”
- 3、**制造的观点**：理解为与规范的一致性；质量等级=与规范吻合程度
- 4、**产品的观点**：质量依赖产品内在的特性；内在品质决定外在质量
- 5、**基于价值的观点**：质量依赖于客户愿意为它支付的货币数额



## 1.2 软件质量

McCall = 质量因素 + 质量标准

软件质量  
概念

**质量因素：**一个质量因素描述了系统的一个行为特征，包括：正确性、可靠性、效率、可测性、可维护性、可复用性

**质量标准：**是一个质量因素的属性，与软件的开发相关联；例如可维护性的属性之一“模块化”

## 1.2 软件质量

### 软件质量模型

**ISO9126**：6个质量特性（广泛、相对独立）：功能性、可靠性、易用性、效率，可维护性、可移植性

**CMM**：软件能力成熟度模型，5个成熟度级别和核心过程领域

**软件测试过程质量模型**：测试过程改进模型（TPI），测试成熟度模型（TMM），均用来指导测试过程的持续改进



## 1.3 软件测试的作用

### 软件测试的作用及方法

软件测试是为了软件质量的**评估**和**提高**而进行的验证过程

**静态测试**：由**人工**完成，**基于文档、软件模型和源代码**；形式包括代码评审、审查、走查、算法分析、正确性证明；推断在测试运行期间发生的所有可能出现的行为

**动态测试**：测试由**计算机**完成，**基于源代码**；形式为实际执行程序，以暴露出程序可能的缺陷，以及观察程序的行为和性能特征。关键：测试用例集合的选取

## 1.4 验证和确认

**验证** 确定给定开发阶段的产品是否满足该阶段开始之前所定义的需求。检查开发阶段的**正确性**的活动称为**验证活动**

(  
ve  
rifi  
cat  
io  
n)  
建立软件开发过程每一个实现阶段和其规范之间的通信

验证活动的**目标**：**正确地构建了产品**

**验证活动**寻求的质量属性是在系统开发的每一个主要阶段的一致性、完备性和正确性

**验证活动**会采用静态测试和动态测试

## 1.4 验证和确认

**确认** 确认产品是否实现了**预期用途**（是否满足用户的全部期望）。**确认活动**关注最终产品，从用户角度进行测试

(validation)

建立系统和用户期望之间的通信

确认活动的**目标**：**构建了正确的产品**

**确认活动**是为了确认整个系统是否满足客户的需求和期望

**确认活动**采用动态测试

## 1.5 失败、错误、故障和缺陷

**失败 (failure)** : 无论何时, 当一个系统的外部行为表现与它的系统规范不一致时, 就说产生了一个失败

**错误 (error)** : 是系统的一个状态。当系统没有执行正确行动时, 一个错误状态可以导致失败

**故障 (fault)** : 一个故障是一个错误的裁定原因。学术界常用本词

**缺陷 (defect)** : 与故障同意, 业界常用

**故障->错误->失败**

## 1.6 软件可靠性的概念(Chapter15)

软件可  
靠性

**测试可以发现软件中的所有故障吗?**

**可靠性**是在进行软件质量测试时一个有用的**定量测量**

**软件可靠性**定义为在指定的环境、指定的时间内，软件系统无故障运作的**概率**

# 1.7 测试的目标

测试的目标

**它是工作的（程序员）**：从开发者角度（心理学）来看，测试的目标是系统能够工作

**它是不能工作的（测试工程师）**：当系统能够工作后，开发者会进行更多的测试，其目的是使得系统失败（发现故障）

**减少失败的风险（客户）**：通过实施越来越多的测试，达到一个更高的目标：降低失败的风险，使其达到一个可以接受的程度

**减少测试成本（项目经理）**：实施测试的最高级别的目标是使用数量更少的测试用例来生产低风险的软件



# 1.7 测试的目标

## 测试的成本

- 1、设计、维护、执行**测试用例**的成本
- 2、分析每一个**测试用例**执行的结果的成本
- 3、将**测试用例**文档化的成本
- 4、系统实际执行及其文档化的成本

## 1.8 什么是测试用例 (test case)

无状态  
系统

测试用例是指<输入, 期望结果>这样一对数据

例如, 计算非负数平方根的程序测试用例如下: <0, 0>或者<25, 5>或者<40, 6.3245553>



## 1.8 什么是测试用例 (test case)

面向状态系统      测试用例是由**<输入, 期望结果>构成的序列**

例如, ATM的取钱过程程序测试用例如下: <检查余额, \$500.00> + <按下取钱按钮, "amount?" > + <输入 \$200.00, "\$200.00" > + <检查余额, \$300.00>;

## 1.9 期望结果

**如何得到期望结果？**

## 1.9 期望结果

程序的  
执行结  
果是什  
么？

**程序产生的数值：**本地观察的输出；远程  
存储、操作或观察的输出

**状态变化：**程序的状态变化；数据库的状  
态变化

**数值序列或集合，**它们必须组合起来解释，  
以说明结果的正确性

## 1.9 期望结果

如何得到期望结果呢？

**理想（简单）情况：**在设计测试用例时计算出来，即应该能从对程序需求的理解来计算出期望结果

**实际情况：**通过对实际的测试结果进行检测，识别出期望的结果。四个步骤：**a、**使用选择的输入执行程序；**b、**观察程序的实际执行结果；**c、验证实际结果是期望结果；****d、**使用验证的实际结果作为该测试用例随后运行的预期结果

## 1.10 完备测试的概念

**完备测试**意味着在测试阶段的结束时发现了**所有**的故障--**这几乎是不可能的**

**现有测试技术是设计的基石**

# 1.10 完备测试的概念

为什么完备测试不可能?

1、一个程序可能的输入范围非常大，以至在测试过程中不可能完全使用所有输入：a)既有合法的输入，也有非法的输入；b)程序可能有大量的状态；c)对于输入可能有时时间约束；d)一个系统的输入域可能非常大

2、设计的问题可能非常复杂以至于不能进行完备测试

3、创建一个系统所有可能的执行环境是不太可能的

2021年6月2日华为召开鸿蒙操作系统及华为全景新品发布会，全球首发鸿蒙OS正式版

前不久，鸿蒙OS2.0向首批10款机型推送了正式版公测，几天时间里就获得了1000万用户的升级适配，鸿蒙OS取得了一个“开门红”

鸿蒙登场，将为华为手机提供国产操作系统，带来全新的体验

不过毕竟是一个新系统，鸿蒙OS2.0开启大范围公测之后，经过一段时间的用户体验，关于鸿蒙OS的一些bug和“差评”不可避免地暴露在网络上

华为对于网络上的“差评”没有避而不谈，反而诚恳回应了用户们反馈的BUG和“差评”。主要集中在升级鸿蒙OS后，手机出现发热、卡顿、耗电等异常情况



鸿蒙的意义是华为的一场自救行动。通过鸿蒙，华为能够展开更广阔的市场布局，打破国外对华为的一系列技术封锁

鸿蒙更大的意义在于肩负“自主可控与产业安全”方面的重任

过去几十年里，“缺芯少魂”一直是我国半导体产业最大的困扰

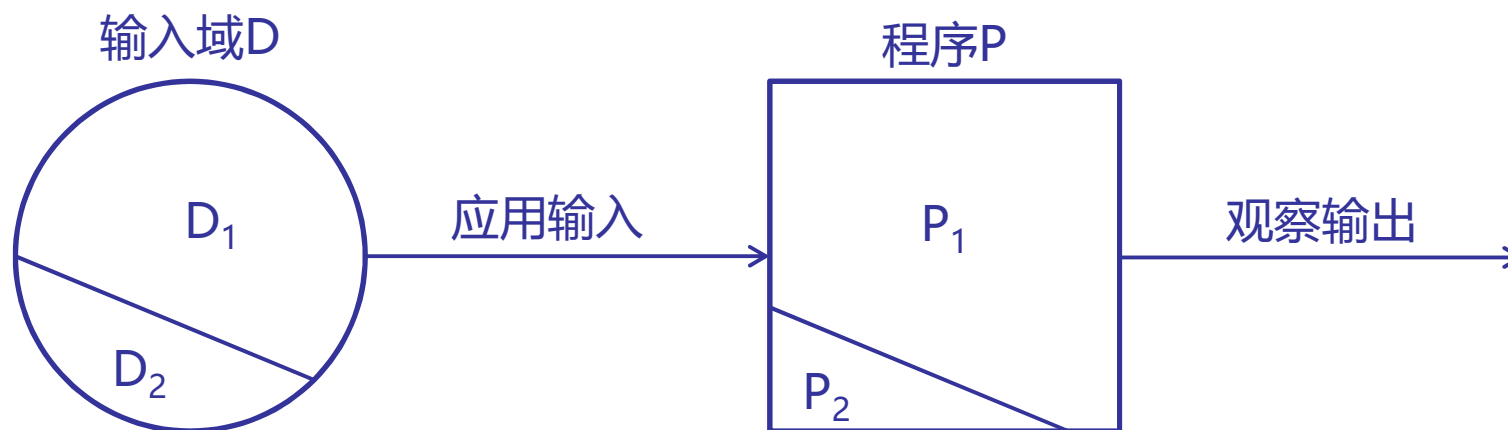
鸿蒙已经打破了软件领域的垄断

在国内芯片企业中，华为海思是国内最强的芯片研发设计企业

在芯片研发领域内，华为海思不仅研发手机芯片，其还研发人工智能芯片、通讯基带、PC芯片以及视频芯片等。最为用户熟知的还是麒麟芯片和巴龙通讯基带

希望同学们成为帮助国家解决“缺  
芯少魂”问题的一员，请奋发努力吧！

# 1.11 测试的核心问题



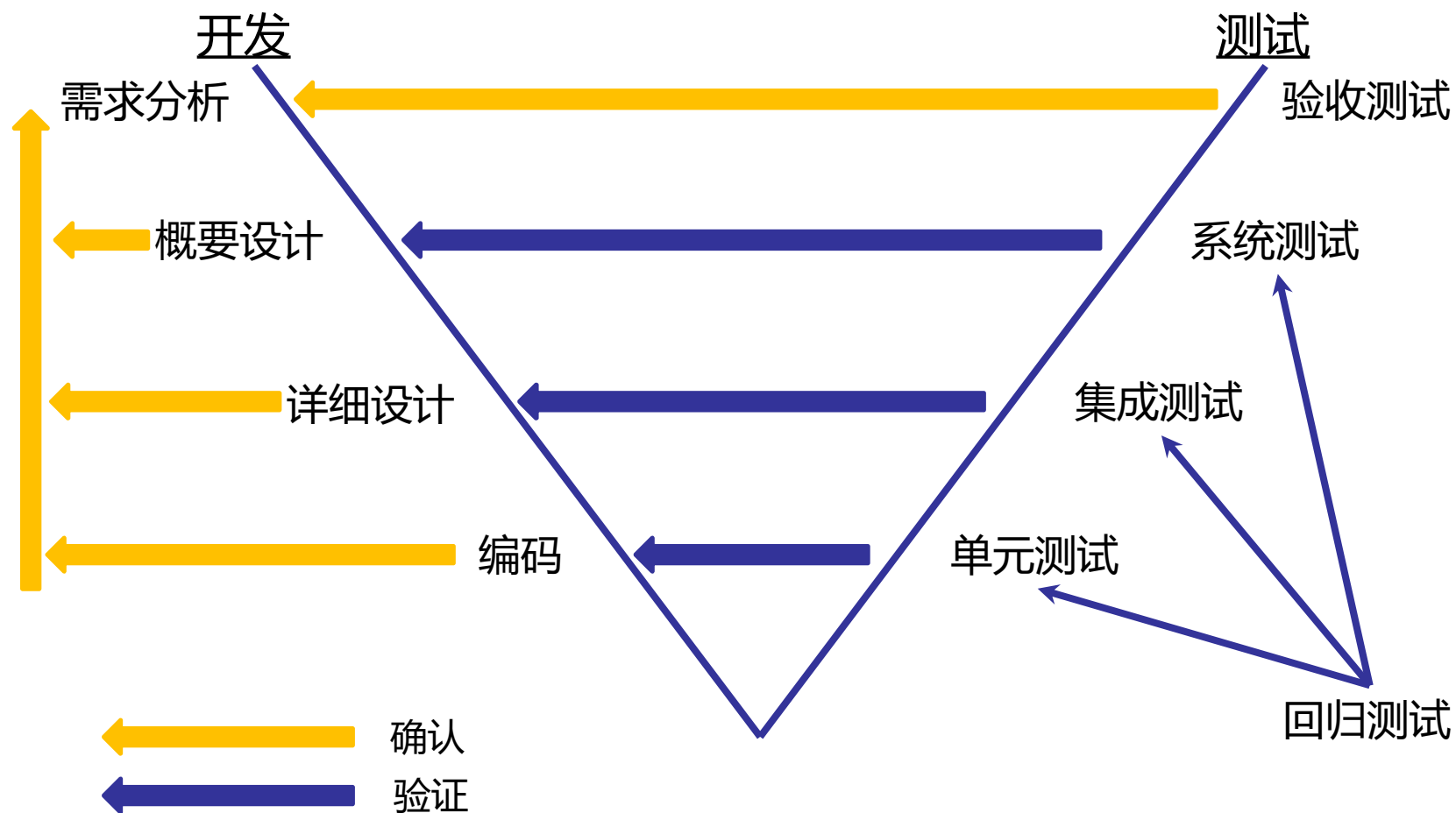
因为完备测试不可能，所以最好的方法就是从输入域内选取一个**子集**对程序进行测试

必须以一种系统的、谨慎的方式来选择输入域的子集，以便得出的推论尽可能**准确和完备**

## 1.12 测试活动（步骤）



# 1.13 测试阶段



## 1.13 测试阶段

**概念 单元测试：**一般由程序的开发者完成，针对单独的单元进行测试

**集成测试：**由开发人员和集成测试人员共同完成，其目标是构建一个合理的稳定系统，以便进行下一步系统测试

**系统测试（关键阶段）：**包括功能测试、安全测试、健壮性测试、负载测试、稳定性测试、压力测试、性能测试、可靠性测试

**回归测试：**不是一个独立的测试阶段，其贯穿于上述3个阶段，任何时候改动了系统的一个部件，均会触发回归测试的执行；关键思想是确定修改并没有给未改动的部分引入新的故障

**验收测试：**目标是为了度量产品的质量，而不是寻找缺陷；关键概念是客户对系统的期望（**UAT**-用户验收测试，**BAT**-业务验收测试）



## 1.14 测试用例选择的信息来源

**信息**    **需求和功能规范**（系统期望的行为）：**所有**程序要满足的需求

**来源**    **源代码**（系统的实际行为）：测试用例必须根据程序而设计

**输入和输出域**：考虑特殊的输入和输出值

**操作特性**：一个操作特性就是关于一个系统如何使用的定量描述，即系统在业务领域的实际运行方式，例如不同输入出现的概率

**故障模型**：曾经遇到过的故障信息形成的知识库。基于故障模型有三种测试方法：a、错误推测；b、故障植入；c、变异分析

## 1.14 测试用例选择的信息来源

### → 基于故障模型的测试方法

**a、错误推测：**第一步，评估情况并且猜测可能存在哪种类型的故障；第二步，设计特定的测试以便专门暴露这些类型的故障

**b、故障植入：**注入已知故障到程序中，执行测试套件以评估测试套件的有效性；（假设：如果测试套件发现了被植入的故障，那么也可以发现其他的故障）

**c、变异分析：**生成程序语句块的变异（故障仿真），以便判定测试套件探测故障的能力；不一定产生错误程序；如果测试套件不能发现这样的故障，则需指定额外的测试用例来揭示这类故障

# 1.15 白盒测试与黑盒测试

**概念**     **白盒测试（结构测试）**：通过关注控制流和数据流对源代码进行检查（程序内部细节）

**控制流**(chapter4)：从一条指令到另一条指令的控制流向，比如指令间顺序、方法调用、信息传递及中断

**数据流**(chapter5)：数值从一个变量、常量传播到另一个变量。变量的定义和使用决定了程序中数据流的方向

**黑盒测试（功能测试）** (chapter6, 9)：考虑从程序外可以访问的部分，即仅从输入和外部可见的输出判断程序的结果是否是期望的结果（输入域和输出域）；是系统接口层次上的测试



# 1.16 测试计划和设计

**概念**     **测试计划：** 目标是为测试的执行做好准备和组织工作。一个测试计划为所需的资源、所需的尝试、活动的时间表、预算等提供了一个框架、范围和细节。范围概括了测试活动的领域和程度，涵盖了测试的管理方面，**不是详细的技术和具体的测试用例**

**测试设计：** 仔细研究系统需求，彻底识别出要被测试的系统特征，**定义出测试用例的目标和详细行为**。根据不同的来源定义不同的测试目标，为每一个测试目标设计一个或多个测试用例。把每一个测试用例设计为一个模块化测试部件的组合体，称为测试步骤。这些测试步骤可以组合在一起生成更复杂、多步骤的测试

## 1.17 监测并度量测试执行

**监测和度量**是任何一种科学/工程活动的两个关键准则

## 1.17 监测并度量测试执行

**监  
测**

**监测**（通过指标）使得管理者掌控测试过程、进度和质量，以便管理者做出决定（纠正和预防）来降低生产成本，减少交付延误，提高软件质量；（例子：如果产品有太多缺陷以至于无法进行有意义的系统测试，那么通过暂缓产品的测试周期，测试经理可以有效地分配测试工程师时间和可能的资金）

**监测指标：**

**监测测试执行的指标：**关注测试用例执行的过程

**监测缺陷的指标：**关注测试执行结果发现的缺陷

**指标要周期性的追踪和分析**

## 1.17 监测并度量测试执行

- 度量:** 评估用于执行任务的技术有效性: 比较不同技术发现缺陷的数目
- (定量评估)**
- 评估开发活动的生产效率:** 每天设计的测试用例数量和执行测试用例的数量
  - 评估产品的质量:** 监测每周测试发现的缺陷可以观察到系统的质量等级
  - 评估产品的测试:**
    - 测试用例有效性指标:** 1、测量测试套件揭示缺陷的能力; 2、使用这个指标来改进测试设计过程;
    - 测试效果有效性指标:** 计算由客户发现而在产品发布之前没有被测试工程师发现的缺陷的数目

## 1.18 测试工具和自动化(Chapter12)

软件测试是一个**劳动力高度密集**的工作，测试用例的设计和执行，结果的分析基本上都是**人工过程**



# 1.18 测试工具和自动化(Chapter12)

## 测试自动化的好处

**提高测试人员的生产率：**提高测试工程师的编程水平（规范化的测试用例设计）

**回归测试有更好地覆盖：**自动化测试用例随时间累积，形成用例库，有利于测试用例一致性（对同组测试产生重复性结果的能力）集合的实现，最终使得测试过程更加有组织性、结构化和可重现

**减少测试阶段的周期（提高测试执行效率）：**无人值守时

**减少软件维护的成本：**回归测试的自动化可以极大降低人工成本

**增加测试用例的有效性：**比如压力测试和负载测试，这类测试基本无法人工完成

# 1.18 测试工具和自动化(Chapter12)

测试  
自动  
化的  
代价

**工具的支持（自动化测试套件）**

**不能完全取代手工测试**

**初期可能会有更大的投入**



# 1.19 测试团队的组织和管理(Chapter16)

**单元测试：**程序开发者

**系统集成测试：**集成测试组

**系统测试：**系统测试组

**验收测试：**用户组（集成测试人员、系统测试人员、客户服务人员和市场人员）+用户

**集成测试组和系统测试组完全独立于开发团队，  
有单独的人员和预算**

# 课堂提问

- 1、软件测试的核心问题是什么？**
- 2、选择测试用例的数据来源有哪些？**



# 基本概念和预备知识

**End**

---