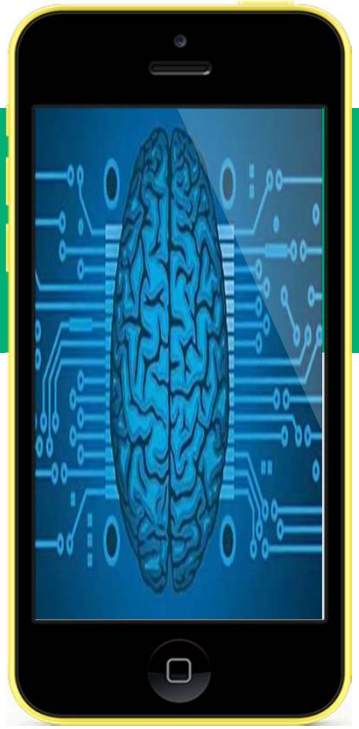

软件测试的基本知识

目 录

CONTENTS

01

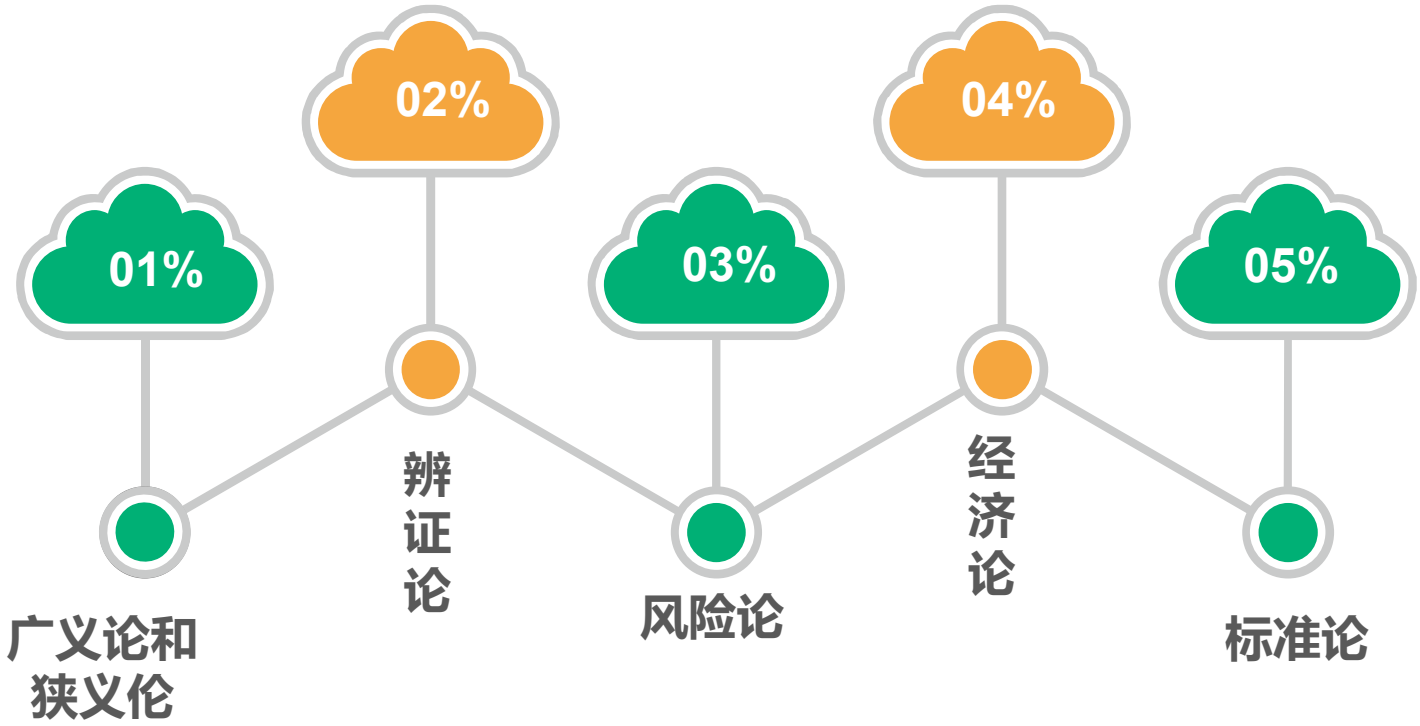
测试的几种观点



我们都是有主见的人？

01

软件测试的几种观点



1.1 测试的狭义和广义论

以过程为导向

以结果为导向（开发结束，维护以前）

1.2 辩证论

正向思维：验证软件是正常工作的

Dr.Bill Hetzel

反向思维：证明软件是工作的

Glenord J.Myers

1.2 风险论

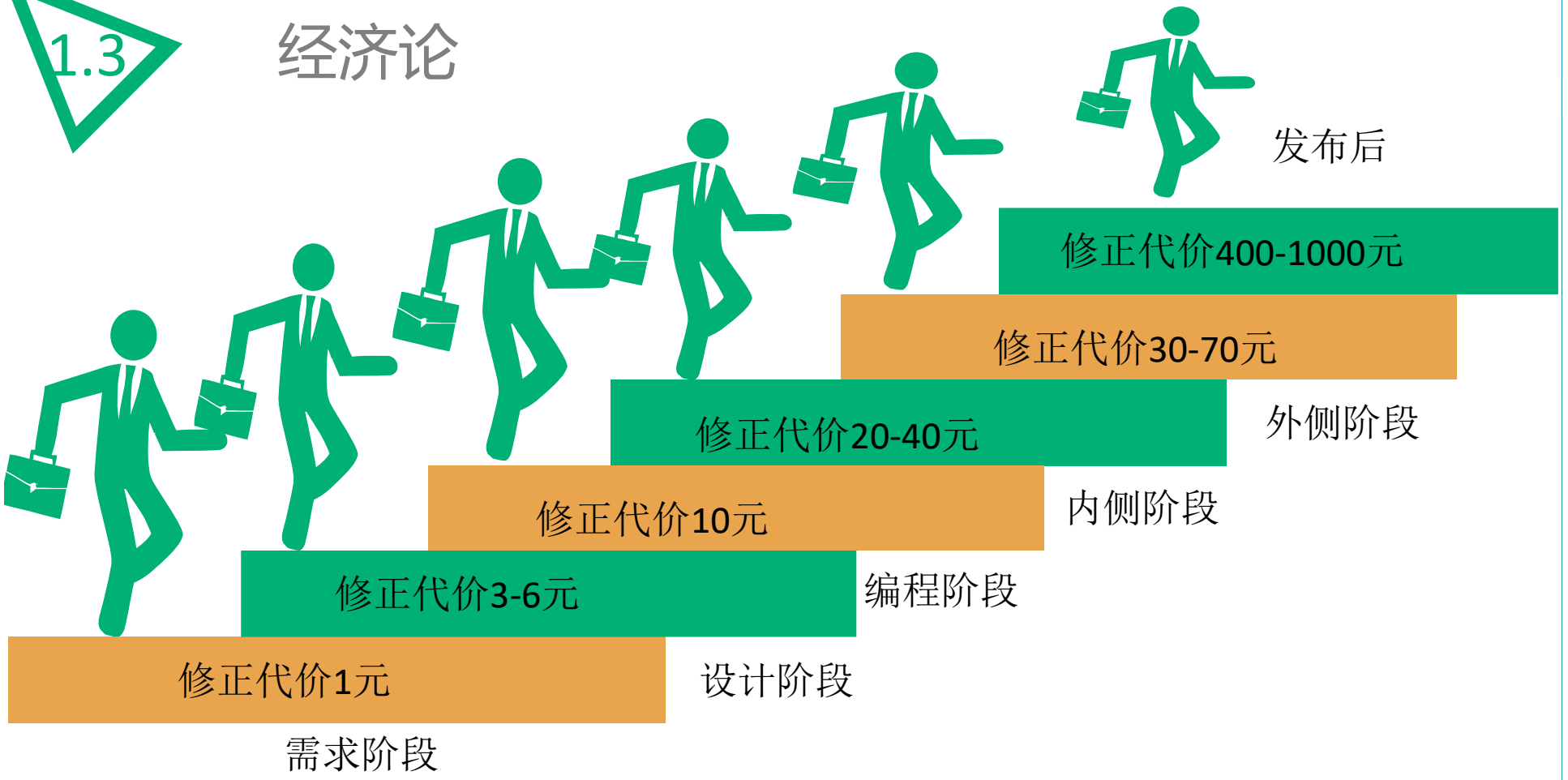
软件测试的风险论认为测试是对软件系统中潜在的各种风险进行评估的活动

巴莱多定律： 也叫二八定律

二—— 高——做——测

八—— 低——不做（少做）——少测（不测）

1.3 经济论


发布后
修正代价400-1000元
修正代价30-70元
外侧阶段
修正代价20-40元
内侧阶段
修正代价10元
编程阶段
修正代价3-6元
设计阶段
修正代价1元
需求阶段

02

软件测试的目的与原则




软件测试目的与原则

（1） 找出缺陷，及时改进

（2）改善程序的有效性

（3）是软件质量的评估有效方法

2.2 测试的原则

把“尽早地和不断地进行软件测试”作为软件开发者的座右铭。

严防寄生虫现象

严防杀虫剂现象

并非所有的软件缺陷都能修复

难以说清的软件缺陷

需求规格说明书不断变化

测试用例的设计

软件测试充分性准则

03

软件测试的分类

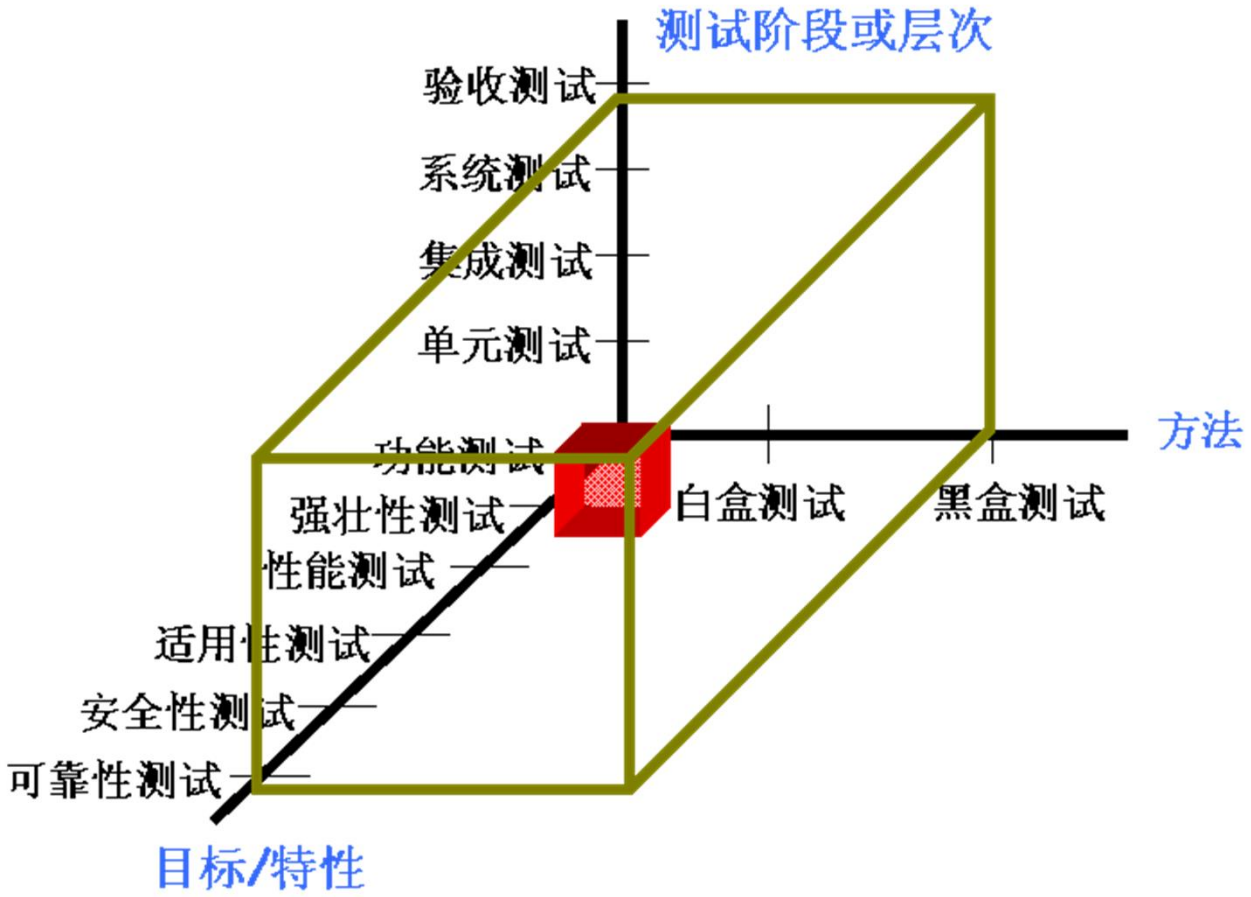
2.2

测试的原则

错误 鉴于二者之 遗漏

单元测试一般选用白盒测试，集成测试一般采用灰盒测试，系统测试和确认测试一般选用黑盒测试

2.2 测试的三维空间


测试阶段或层次
验收测试
系统测试
集成测试
单元测试
功能测试
强壮性测试
性能测试
适用性测试
安全性测试
可靠性测试
白盒测试
黑盒测试
方法
目标/特性

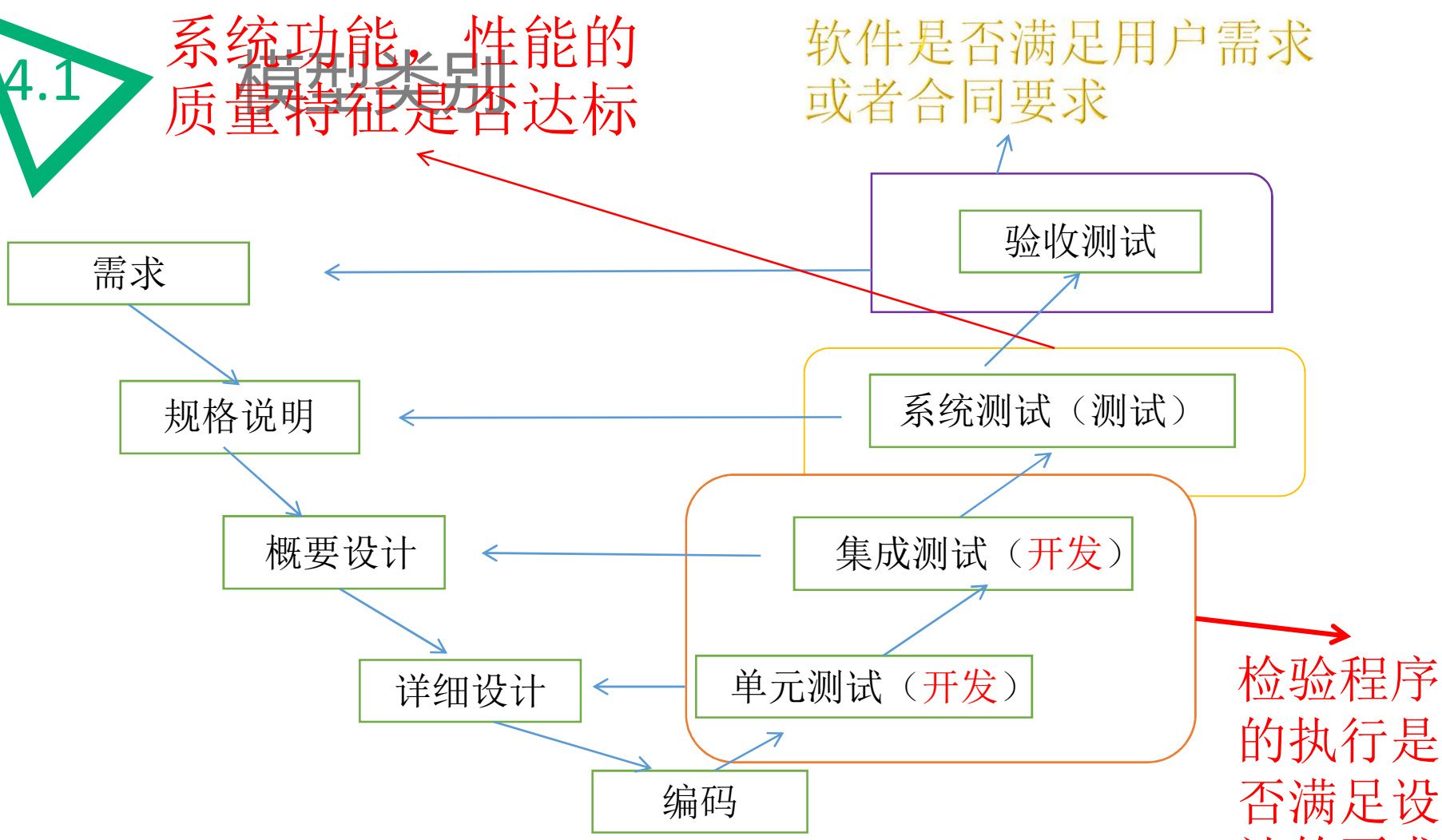
04

软件测试的模型

04

模型类别

软件测试模型是软件测试工作的框架，描述了软件测试过程所包含的主要活动，这些活动之间的相互关系等。

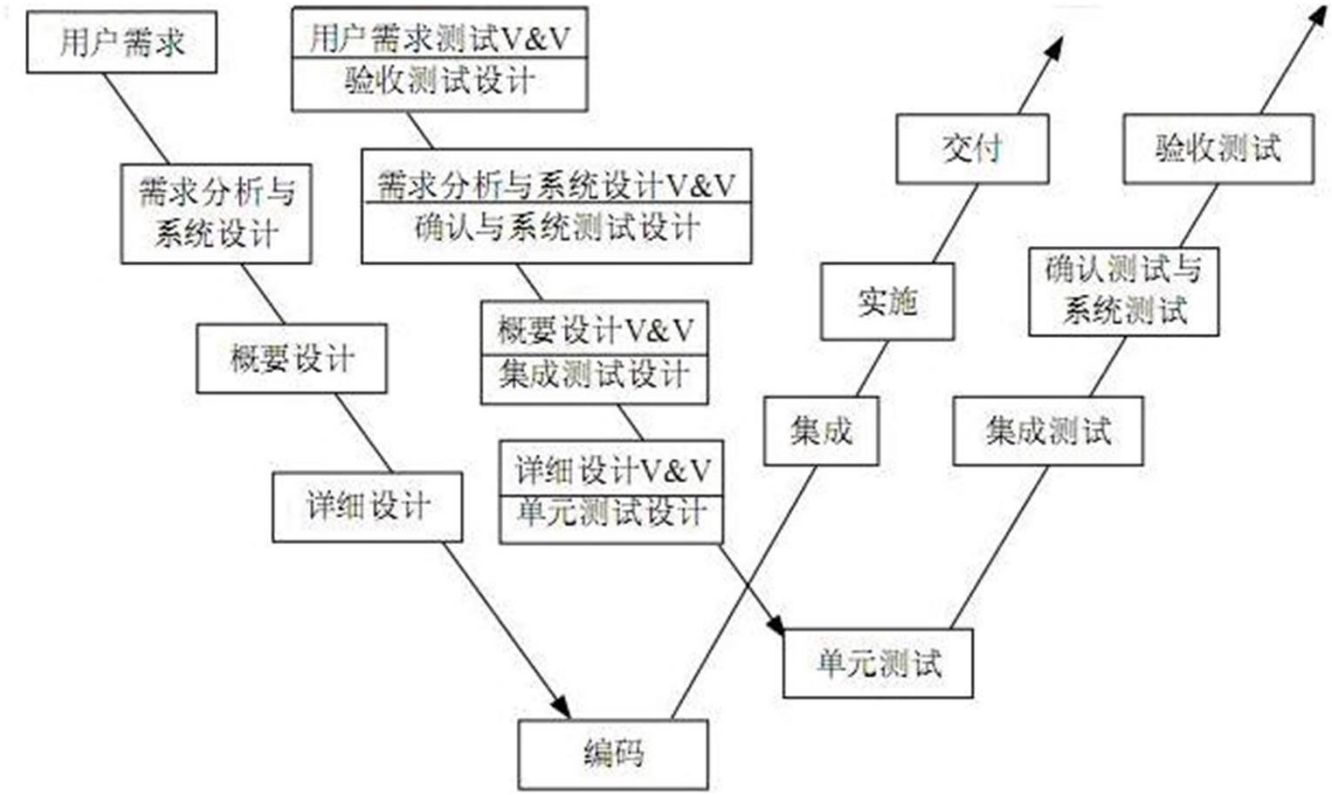

系统功能、性能的质量特征是否达标
软件是否满足用户需求或者合同要求
需求
规格说明
概要设计
详细设计
编码
验收测试
系统测试（测试）
集成测试（开发）
单元测试（开发）
检验程序的执行是否满足设计的要求
编码是V模型的黄金分割线

V模型的局限性

V模型存在一定的局限性，它仅仅把测试过程作为在需求分析、概要设计、详细设计及编码之后的一个阶段。容易使人理解为测试是软件开发的最后的一个阶段，主要是针对程序进行测试寻找错误，而需求分析阶段的隐藏的问题一直到后期的验收测试才被发现。

2.3

W模型


用户需求
用户需求测试V&V
验收测试设计
需求分析与
系统设计
需求分析与系统设计V&V
确认与系统测试设计
交付
验收测试
概要设计
概要设计V&V
集成测试设计
实施
确认测试与
系统测试
详细设计
详细设计V&V
单元测试设计
集成
集成测试
单元测试
编码

W模型的局限性

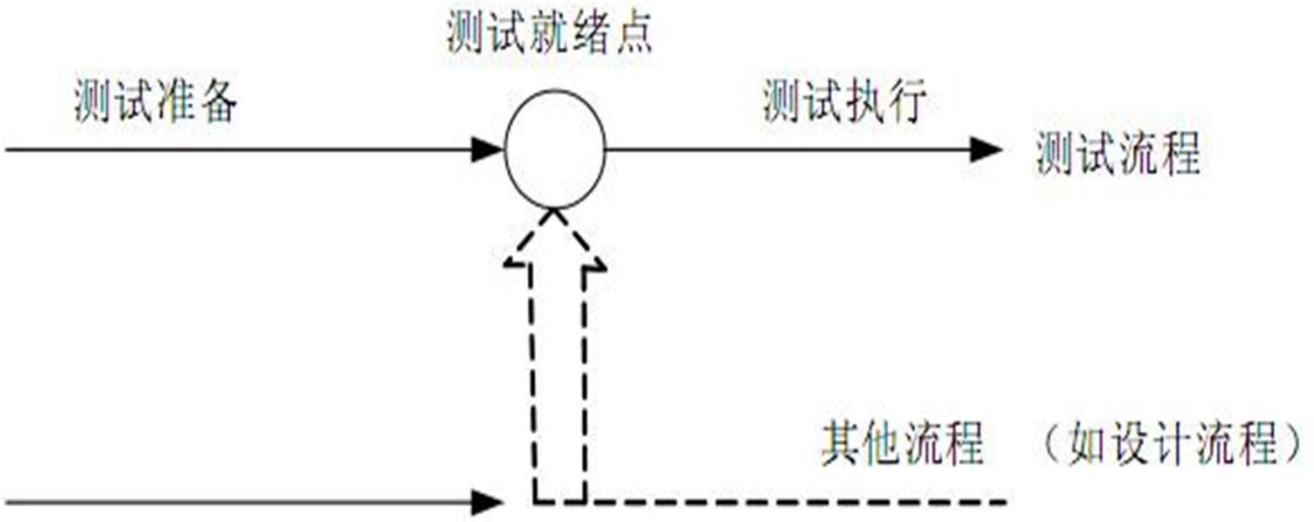
（1） 在V模型中增加软件各开发阶段应同步进行的测试，被演化为一种W模型。

（2）开发是“V”，测试也是与此相重叠的“V”。

（3）W模型体现了“尽早地和不断地进行软件测试”的原则

（4）W模型也存在局限性。在W模型中，需求、设计、编码等活动被视为串行，测试和开发活动保持着一种线性的前后关系，上一阶段结束，才开始下一个阶段工作，因此，W模型无法支持迭代开发模型。

2.4 H模型



H模型的特点

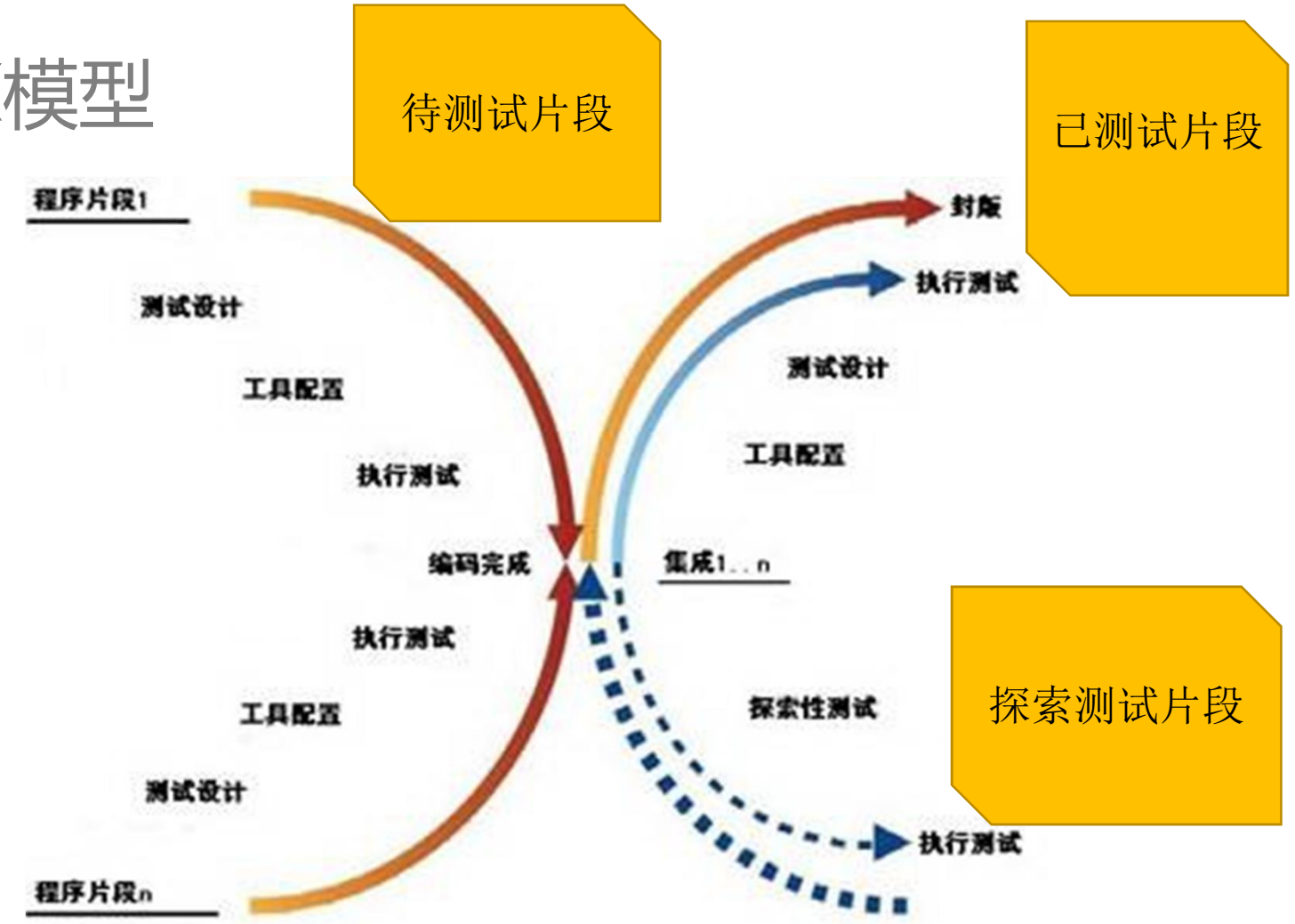
（1） 测试与开发并行。

（2）测试准备与测试执行分离，有利于资源调配，降低成本，提高效率。

（3）充分体现测试过程（不是技术）的复杂性。

（4）相应的测试也不存在严格的次序关系，单元测试、集成测试、系统测试之间具有反复迭代

2.5 X模型


待测试片段
已测试片段
程序片段1
测试设计
工具配置
执行测试
编码完成
执行测试
工具配置
测试设计
程序片段n
封版
执行测试
测试设计
工具配置
集成1..n
探索性测试
执行测试
探索测试片段

前置模型

开发和测试相结合

对每一个交付内容进行测试

让验收测试和技术测试保持相互独立

反复交替的开发和测试

引入新的测试理念

05

软件测试的用例

5.1

测试用例

一个三角形程序，该程序完成：

输入三个整数a、b和c，程序判断由这三条边构成的三角形类型是：等边三角形、等腰三角形还是一般三角形，并打印出相应的信息。

1．是否设计了一种测试输入表示合法的一般三角形?

注意，像（2，1，4）和（9，3，9）这样的测试输入不应该回答“是”，为什么?

2．是否设计了一种测试输入表示合法的等边三角形?

3．是否设计了一种测试输入表示合法的等腰三角形?

像(5，5，10)这样的测试输入不应该回答“是”。

4. 是否至少设计了三种测试输入表示合法等腰三角形，由此检查了两条边相等的所有三种排列方案?

像（3，3，4），（4，3，3）和（3，4，3）。

5．是否设计了这样的测试输入，其中三角形的一条边长为0?

6．是否设计了一种测试输入，其中三个整数都大于0，而其中的两数之和等于第三数?

注意，如果把（2，3，5）当成一个一般三角形，则表明程序中有故障。

7．是否至少设计了三种第6类那样的测试输入，检查一条边边长等于另外两边边长之和的所有三种排列方式?

如（2，3，5），（5，3，2）和（3，5，2）。

8．是否设计了一种测试输入，表示三个整数都大于0，而其中某两个数的和小于第三个数?

注意，如果把（2，3，9）当成一个一般三角形，则表明程序中有故障。

9．是否至少设计了三种第8类那样的测试输入，检查了所有三种排列的方案？如（2，3，9），（2，9，3）和（9，2，　3）。

10．是否设计了一种测试输入表示三条边边长都为0？

即（0，0，0）。

11．是否设计了这样的测试输入，其中三角形的一条边长为负数?

12．是否至少设计了一种测试输入，其中三角形的边长不是整数?

13．是否至少设计了一种测试输入，其中三角形的边数不是3？　例如，给出两条边或四条边。

14．对于每一种测试输入，是否还给出了预期的输出?

解析：假设在字长为16为的计算机，则每一个整数的取值有 2^{16} 种，对三角形的三边，a，b，c进行群举测试，可能需要 $2^{16}\times 2^{16}\times 2^{16}\approx 3\times 10^{14}$ 种测试方法，假设测试一次需要1MS，执行共需1万年

测试用例的作用

测试用例（Test Case）是指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。

内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，最终形成文档。

简单理解，测试用例是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，用于核实是否满足某个特定软件需求。

5.2 测试用例的作用

使用测试用例的好处

1. 测试之前设计好测试用例，避免盲目测试，提高效率
2. 测试用例令软件测试的实施重点突出，目的明确
3. 软件更新，只需要修正部分案例，介绍工作量
4. 不断精化案例，提高软件效率

测试阶段和用例关系

测 试 阶 段	测 试 类 型	执 行 人 员
单元测试	模块功能测试，包含部分接口测试、路径测试	开发人员
集成测试	接口测试、路径测试，含部分功能测试	开发人员，如果测试人员水平较高可以由测试人员执行
系统测试	功能测试、健壮性测试、性能测试、用户界面测试、安全性测试、压力测试、可靠性测试、安装/反安装测试	测试人员
验收测试	对于实际项目基本同上，并包含文档测试；对于软件产品主要测试相关技术文档	测试人员，可能包含用户

测试用例用例的使用准则

A 有效性

B 经济性

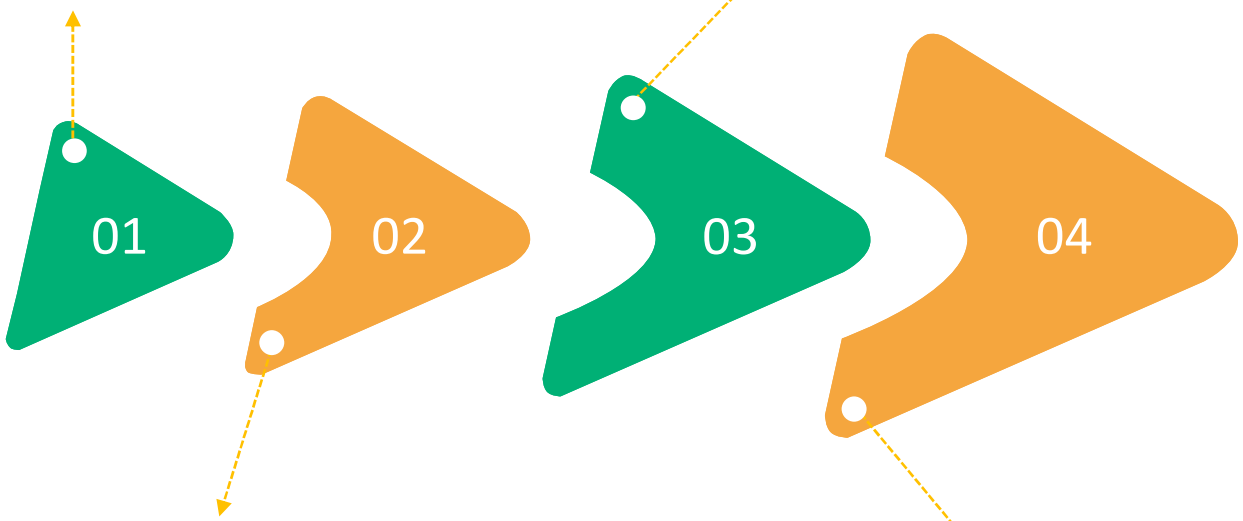
C 完备性

D 可判断性

E 可再现性

5.4 测试用例的设计步骤

制定测试用例的策略和思想，在测试计划中描述出来



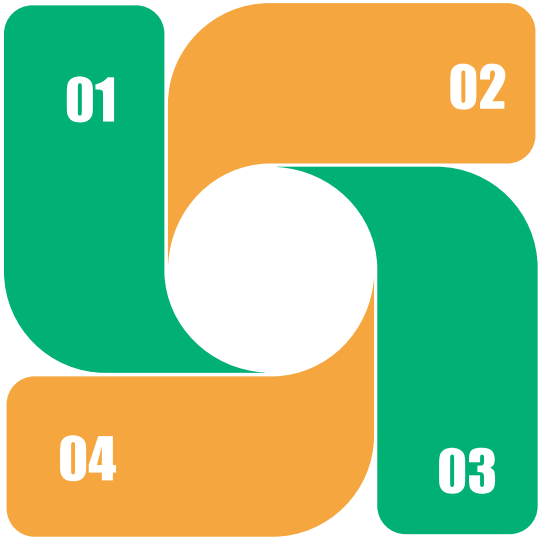
设计测试用例的框架，也就是测试用例的结构

细化结构，逐步设计出具体的测试用例

通过测试用例的评审，不断优化测试用例

5.5 测试用例的维护

产品特性没变，只是根据漏掉的缺陷来完善测试用例。这时候，增加和修改测试用例均可，因为当前被修改的测试用例对相应的版本都有效，不会影响某个特定版本所拥有的测试用例



完全新增加的特性，需要增加新的测试用例

原有产品特性发生变化，不是新功能特性，而是功能增强，这时候原有的测试用例只对先前版本有效，对当前新的版本无效。此时，绝不能修改测试用例，只能增加新的测试用例，不能影响原有的测试用例。

原有功能取消了，这时只要将与该功能对应的测试用例在新版本上置为空标志或“无效”状态，但不能删除这些测试用例，因为它们对先前某个版本还是有效的

06

软件测试停止标准

停止的总体标准

- 测试超过了预定时间，则终止测试
- 执行所有测试用例，但是没有发现错误，则终止测试
- 使用特定的测试用例设计方法作为判断测试停止的基础
- 给出测试停止的要求，例如发现20个软件故障
- 根据单位时间内差处的故障数来确定是否停止
- 经过各个阶段测试吗，都到达预期效果
- 如软件项目开发暂停

单元测试停止的标准

- 单元测试用例设计已经通过评审
- 按照单元测试计划完成了所有规定单元的测试
- 达到了测试计划中关于单元测试所规定的覆盖率的要求
- 被测试的单元每千行代码必须发现至少3 个错误
- 软件单元功能与设计一致
- 在单元测试中发现的错误已经得到修改，各级缺陷修复率达到标准

集成测试停止的标准

- 集成测试用例设计已经通过评审
- 按照集成构件计划及增量集成策略完成了整个系统的集成测试
- 达到了测试计划中关于集成测试所规定的覆盖率的要求
- 被测试的集成工作版本每千行代码必须发现2 个错误
- 集成工作版本满足设计定义的各项功能、性能要求
- 在集成测试中发现的错误已经得到修改，各级缺陷修复率达到标准

系统测试停止的标准

- 系统测试用例设计已经通过评审
- 按照系统测试计划完成了系统测试
- 达到了测试计划中关于系统测试所规定的覆盖率的要求
- 被测试的系统每千行代码必须发现1 个错误
- 系统满足需求规格说明书的要求
- 在系统测试中发现的错误已经得到修改，各级缺陷修复率达到标准

THANK YOU