



软件测试实践

贺依娜

he_yina@founder.com.cn

2003 - 10 - 19



Founder R&D



内容提要

- 一． 测试基本理论的回顾
- 二． 测试阶段和流程介绍
- 三． 基于测试流程上的缺陷管理系统
- 四． 缺陷数据的分析





一、测试基本理论的回顾

1. 软件测试的目的、作用
2. 验证与确认
3. 黑盒测试与白盒测试
4. 软件测试的衡量标准
5. 软件测试人员的素质要求



1.1 软件测试目的和作用

○ 目的

- 1. 测试是程序的执行过程，目的在于发现错误；
- 2. 一个好的测试用例在于能发现至今未发现的错误；
- 3. 一个成功的测试是发现了至今未发现的错误的测试。

-- 《the art of software testing》

--grenford j. myers

○ 作用

- 1、测试是为了要找出缺陷，但同时，也可以通过对缺陷的度量和统计，分析缺陷产生的原因和缺陷的分布特征，分析产品的质量、工作效率、诊断开发过程中的问题，并通过改进各个开发过程提高过程能力，最终降低缺陷数量和缺陷密度。
- 2、没有发现错误的测试也是有价值的，完整的测试是评定测试质量的一种方法。



软件测试的意义

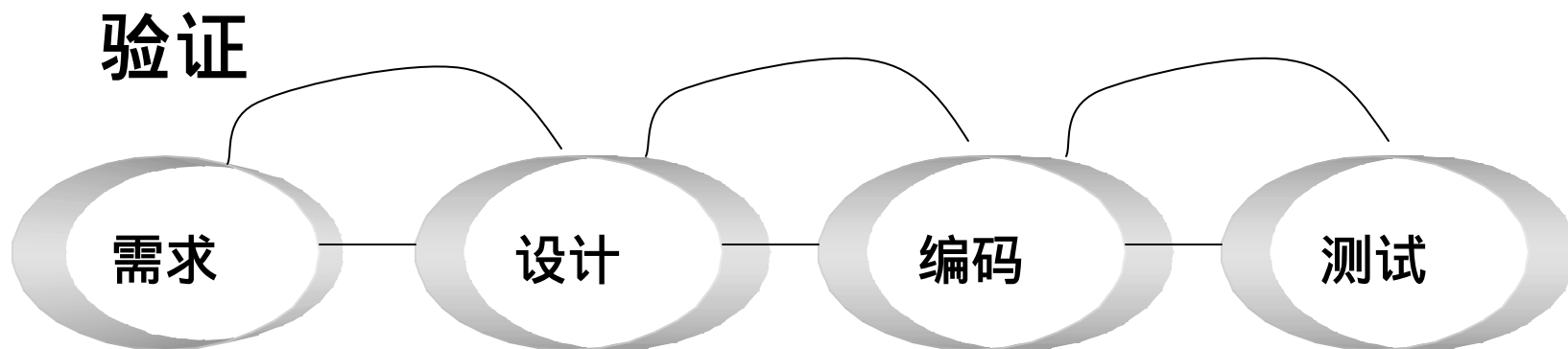
- 迄今为止，软件质量仍然主要靠软件测试来验证和确认，而且由于测试工作特别耗费资源，在软件开发的总成本中，用在测试上的开销要占30%到50%。在极端的情况下，例如在关系到人的生命安全的软件中（如飞机控制或核反应监控等软件），测试费用可能相当软件生存周期所有其它阶段费用总和的三到五倍。此外，据美国工业界的统计，对商品化的程序来说，测试在时间和费用两方面的花费都要占整个软件开发周期总开销的50%左右。
- 但软件测试并不是质量保证的“安全网”。因此，我们应该从两个方面来保证和提高软件质量：首先，应尽量在开发期间减少错误，通过软件过程来建立软件质量；次之，应通过分析和测试来发现和修复软件产品中残存的错误。



1.2 验证和确认

验证（Verification）：

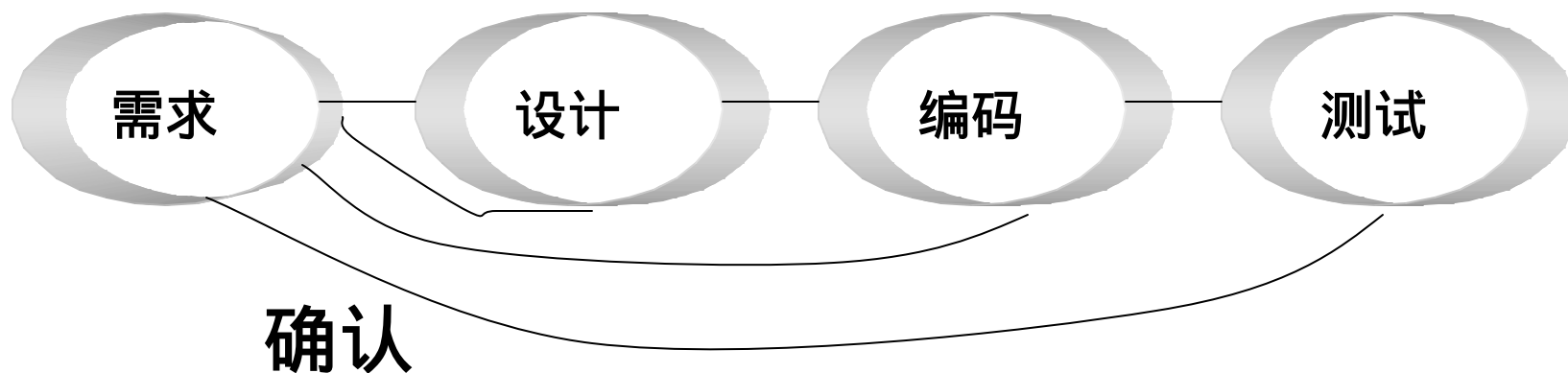
在软件生存期各个阶段，验证是指检测各个阶段结束时的工作产品是否满足对上一阶段的结束后的工作产品所定义的规格的验证过程。



1.2 验证和确认

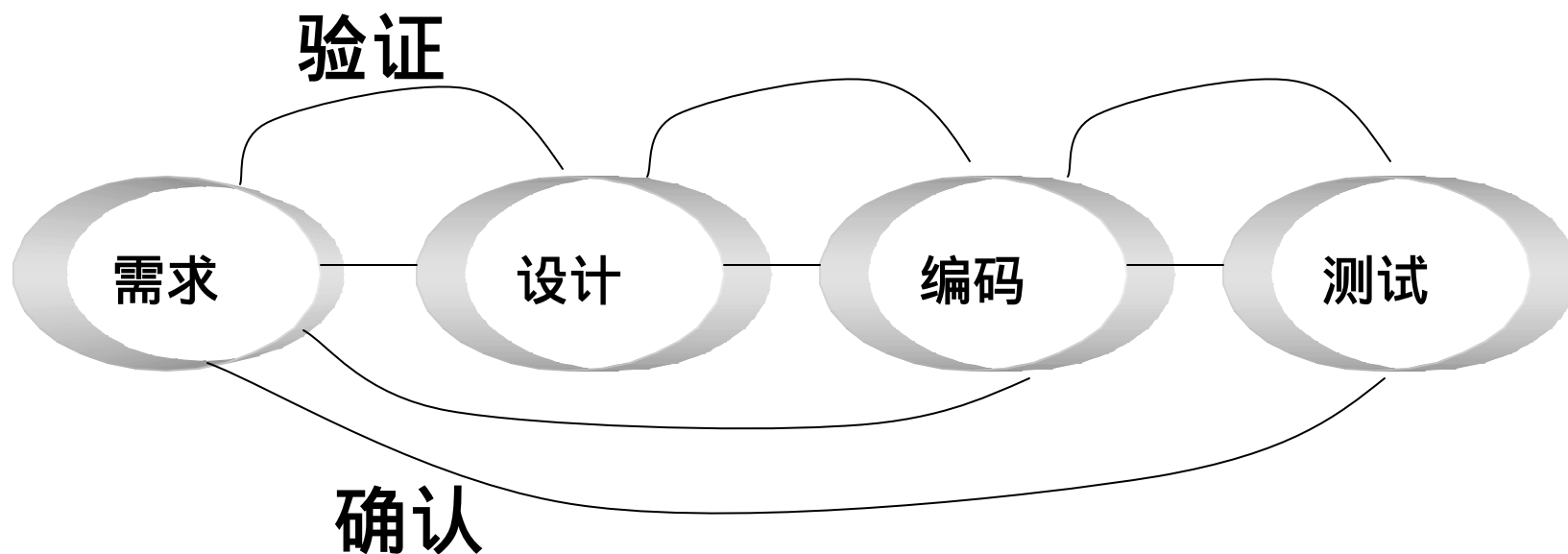
确认 (Validation) :

在软件生存周期各个阶段，确认是指检测各个阶段结束时的工作产品是否满足在软件生存周期初期在系统需求文档中描述的各项软件规格的确认过程。



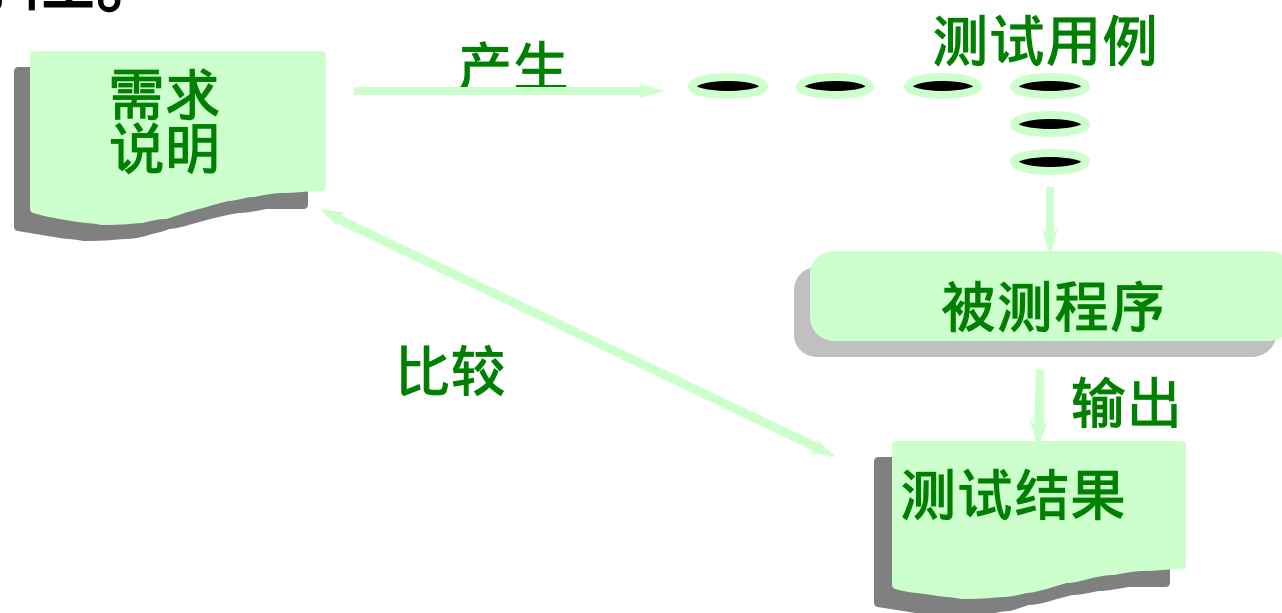
确认和验证的比较

- 验证是检测每一阶段形成的工作产品是否与前一阶段定义的规格相一致，
- 确认是检测每一阶段的工作产品是否与最初定义的软件需求规格相一致。
- 软件测试既可用于验证，又可用于确认。



1.3.1 黑盒测试 (Black-box testing)

- 黑盒测试是从用户观点出发的测试，它又称功能测试、数据驱动测试或基于规格说明书或用户手册的测试。它所依据的是程序的外部特性。



黑盒测试的准则

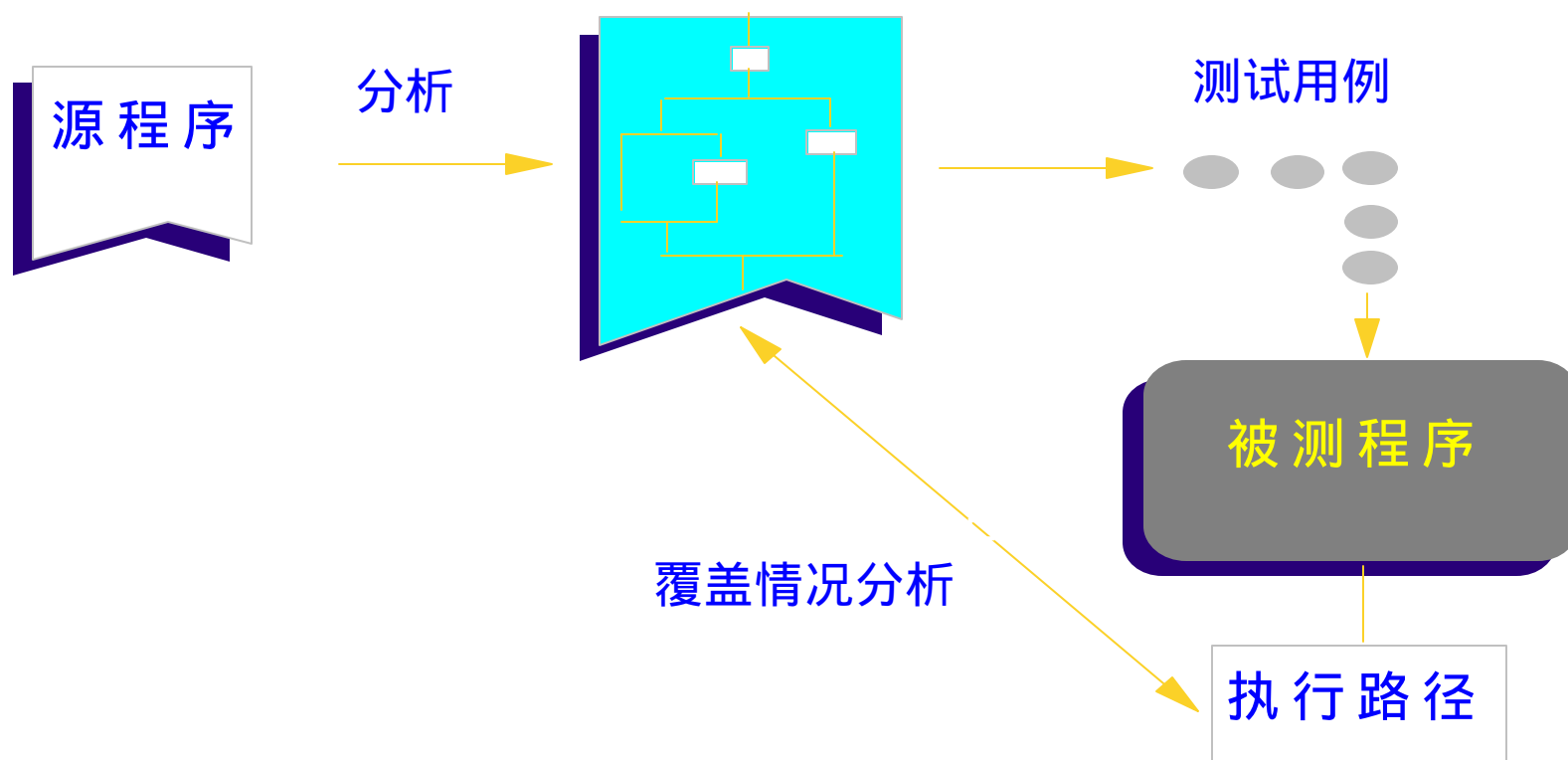
- 基于需求规格说明书的测试（需求矩阵）
 - 需求列表
 - 设计列表: 概要设计和详细设计
 - 测试用例与需求的对照表

需求	设计	编码	测试
rq1	de1	f1	tc1
rq1	de2	f2	tc2
rq1	de3	f3	tc3
rq1	de3	f4	tc4
rq2	de4	...	tc5
rq3	de5	...	tc6
rq4	de6		tc7



1.3.2 白盒测试 (White-box testing)

- 白盒测试又称结构测试、逻辑驱动测试或基于程序的测试，其测试过程如图所示。





白盒测试覆盖准则

○语句覆盖

- -在测试时，设计若干测试用例，运行被测程序，使程序中的每个可执行语句至少执行一次。分支覆盖-在测试时，设计若干测试用例，运行被测程序，使程序中的每个判断真假的分支至少遍历一次。

○条件覆盖

- -在测试时，设计若干测试用例，运行被测程序，使程序中的每个条件的可能取值至少满足一次。

○条件分支覆盖

- -在测试时，设计足够的测试用例，使得判断中每个条件的所有可能取值至少出现一次，并且每个判断本身的判定结果也至少出现一次。

○路径覆盖

- - 设计足够多的测试用例，要求覆盖程序中所有可能的路径。



1.4 软件测试的衡量标准

- 需求的覆盖
 - 需求追溯表/需求矩阵
- 缺陷数量
 - 多、新
- 缺陷重现率
 - BUG能按照一定的测试过程稳定重现
- 效率
 - 平均每人天发现的BUG数（5个/人天）
- 成本
 - 少。合理的测试人力和软、硬件资源安排
- 重用价值
 - 测试的数据或者样例可以重用





1.5测试人员的素质要求

- 对于测试人员的职业素质要求：
 - 1、责任感
 - 坚持原则、不放弃
 - 有问题及时汇报
 - 2、沟通能力、移情能力
 - 与用户、项目组的沟通
 - 3、独立的判断和自学习能力
 - 坚持观点，不随声附和
 - 喜欢探寻、钻牛角尖不一定是坏事情
 - 4、耐心、自我督促
 - 5、团队精神
 - 与开发组整体为一个目标开展活动，有时需要妥协



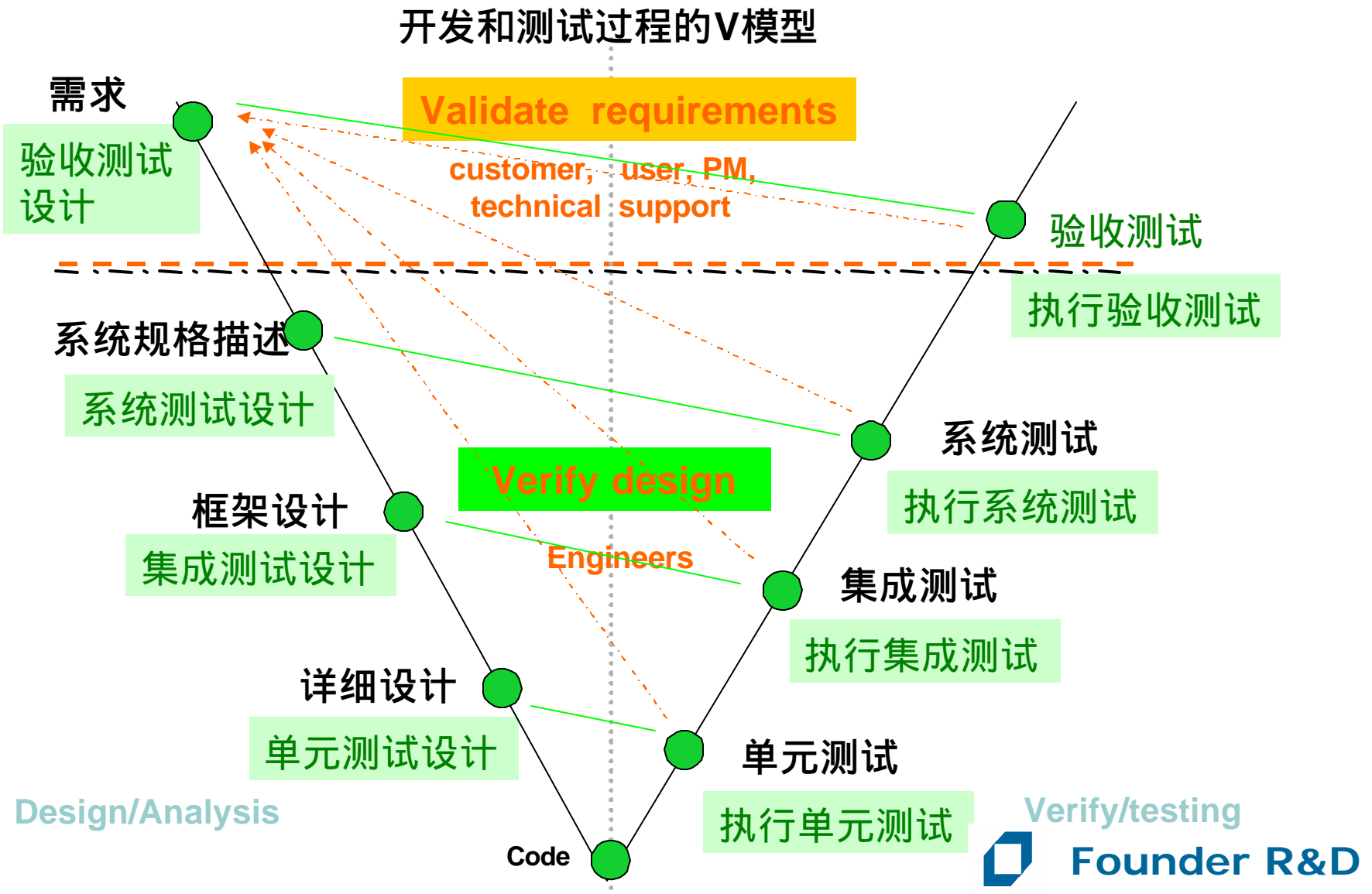


二、一般的测试过程介绍

1. V-Model
2. 测试阶段介绍
3. 测试流程介绍



2.1 V模型 (Improved)





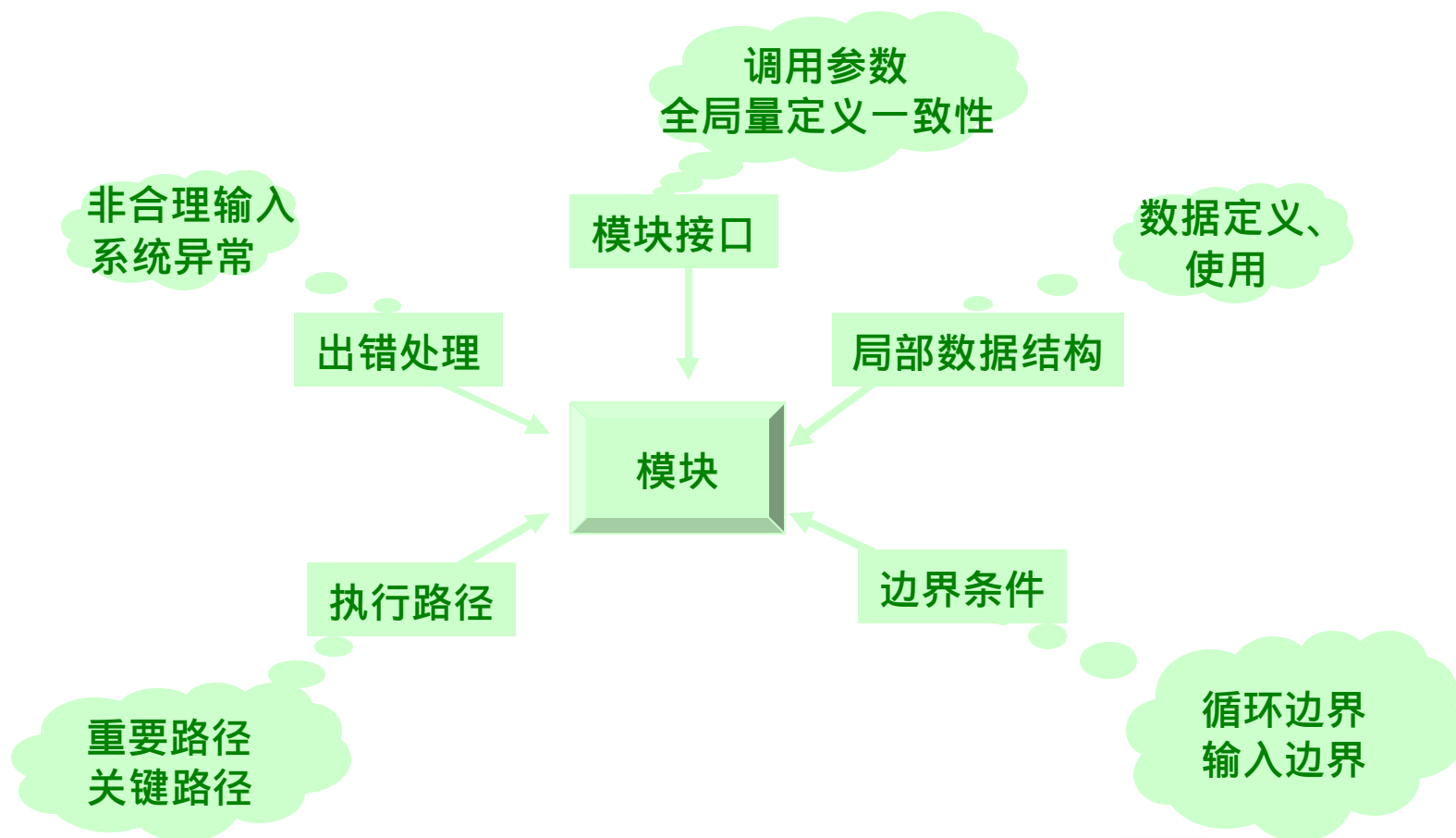
2.2.1 单元测试阶段

- 单元测试的内容：
 - 1、模块接口测试
 - 2、检查局部数据结构能否保持完整性
 - 3、模块边界条件测试
 - 4、模块执行路径测试
 - 5、检查模块内部错误处理是否有效
- 单元测试方法：
 - 白盒测试为主



单元测试或模块测试

对模块进行正确性检验的测试工作，测试用例以白盒测试为主、黑盒测试为辅。





单元测试检查单

- ❑ 逻辑和算法：正确实现了逻辑和算法
- ❑ 数据结构（全局和局部）：使用了全局数据结构？哪些？如果有，作了哪些关于全局数据的假设？这些假设正确吗？使用了局部数据？在算法执行的所有步骤期间，保持局部数据的完整性了吗？
- ❑ 接口：来自调用模块的数据匹配被调用的模块的期望接收的数据？被调用模块的数据匹配调用的模块提供的数据？
- ❑ 独立路径：标识了所有穿过模块的独立路径？执行了吗？
- ❑ 边界条件：了解边界条件吗？进行了测试确保该模块在其边界条件上的适当的操作了吗？
- ❑ 出错处理：所有出错处理路径均执行到了吗？





2.2.2 集成测试阶段

- 集成测试 (Integration test)：将通过单元测试的多个模块组合成更大的模块或子系统或产品，然后进行测试。
- 测试内容：各单元的接口是否吻合、代码是否符合规定的标准、界面标准是否统一等。
- 人员安排：既要求参与的人熟悉单元的内部细节，又要求他们能够从足够高的层次上观察整个系统。一般由有经验的测试人员和主要的软件开发来完成集成测试的计划。





集成测试

- 集成测试计划：集成测试计划由系统设计人员在设计阶段制定，它是和设计规格说明同时完成的。内容有：
 - 测试的描述和范围
 - 测试环境
 - 测试时间表
 - 集成次序
 - 测试用例以及测试的预期结果等
- 测试方法：集成测试阶段是以黑盒法为主。





2.4 系统测试

- 经过集成测试之后，分散开发的模块被联接起来，构成完整的程序，其中各模块间接口存在的种种问题都已基本消除。测试开始进入到系统测试的阶段。
- 为了发现缺陷并度量产品质量，按照系统的功能和性能需求进行的测试
 - 一般使用黑盒测试技术
 - 一般由独立的测试人员完成





系统测试

- 系统测试(System test):

- 应该由若干个不同测试组成，目的是充分运行系统，验证系统各部件是否都能正常工作并完成所赋予的任务。这里所谓的系统不仅仅包括软件本身，而且还包括计算机硬件及其相关的外围设备，数据及其收集传输机构，甚至掌握计算机运行的人员及其操作等。

- 通常意义上的系统测试包括：功能测试、压力测试(Stress test)、性能测试(Performance test)、容量测试(Capacity test)、用户界面测试、兼容性测试等。





系统测试

○压力测试：

- 也称为强度测试。压力测试的目的就是在软件投入使用以前或软件负载达到极限以前，通过执行可重复的负载测试，预先分析出软件可承受的并发用户极限值和性能瓶颈，以帮助软件厂商或用户优化自己的程序。

○容量测试：

- 对软件容量的测试，能让用户明白到底此软件能一次性承担多大访问量。有了对软件负载的准确预测，不仅能让用户对软件在实际使用中的性能状况充满信心，同时也可以帮助用户最经济地规划自己的网络配置，避免无谓的硬件投入，还可以减少网络系统的宕机时间和因此带来的经济损失。





系统测试

○性能测试：

- 对于那些实时和嵌入式系统，软件部分即使满足功能要求，也未必能够满足性能要求，虽然从单元测试起，每一测试步骤都包含性能测试，但只有当系统真正集成之后，在真实环境中才能全面、可靠地测试运行性能，系统性能测试就是为了完成这一任务。

- 压力测试、容量测试和性能测试的手段和方法很相似，有时可以交织在一起进行测试。压力测试的重点在于发现功能性测试所不易发现的系统方面的缺陷。容量测试和性能测试更着力于提供性能与容量方面的数据，以供制造者参考改进或进行广告宣传。

系统测试

功能测试举例

打印 [?] [X]

打印机

名称(N): 传真 [v] [属性(P)]

状态: 闲置

类型: Windows NT Fax Driver

位置: MSFAX:

批注:

打印范围

☒ 全部(A) ☐ 当前幻灯片(E) ☐ 选定幻灯片(S)

☐ 自定义放映(Q): [v]

☒ 幻灯片(L): []

请输入幻灯片编号或幻灯片范围。例如, 1, 3, 5-12

份数

打印份数(C): [1] [v]

☒ 逐份打印(T)

打印内容(W):

幻灯片 [v]

讲义

每页幻灯片数(E): [6] [v]

顺序: ☒ 水平(Z) ☐ 垂直(V)

☒ 灰度(G) ☐ 根据纸张调整大小(F)

☐ 纯黑白(U) ☐ 幻灯片加框(M)

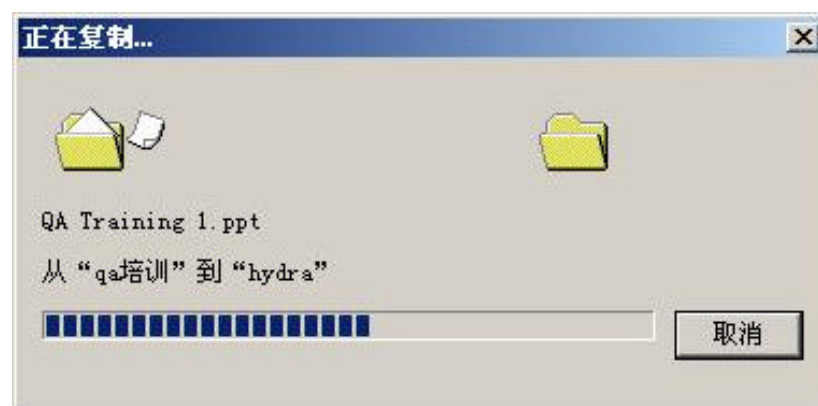
☐ 包括动画(O) ☐ 打印隐藏幻灯片(H)

[确定] [取消]



系统测试

- 用户界面的测试：
 - 符合标准和规范
 - 直观性
 - 一致性
 - 灵活性
 - 舒适性



系统测试

- 兼容性测试/配置测试
 - 向前和向后兼容
 - 多版本的测试
- 例：网站测试用例设计矩阵表

	PC				Unix/Mac			
	Win98	WinME	WinNT	Win2000	Solaris	HP-UX	OS IX	OS X
IE5								
IE5.5								
IE6								
NS4.7								
NS6.0								
...								





系统测试

其他还有一些关于测试的分类，例如：

- 健壮性测试
 - 容灾测试
 - 内存泄漏测试
 - 并发性测试
 - 安全性测试
 - 配套产品测试。
- 实际上，这些测试都是因为测试的目的不同，而在制定测试策略和测试设计的时候有不同的侧重点。





测试规范

必做的测试：

- 安装测试
- 功能测试
- 值域测试
- 界面测试
- 可用性测试
- 说明书测试
- 配置测试
- 加密问题测试
- 裸机测试

可选的测试：

- 内存泄漏测试
- 接口测试
- 性能测试
- 并发性测试
- 安全性测试
- 破坏性测试
- 配套产品测试





系统测试

○ 测试员的效率

- 平均每个工作日发现4-6个Bug
- 平均每修正3个Bug，会引进1个新的Bug
- 平均75%的Bug会在单元测试阶段解决掉
- 平均20%的Bug会在集成测试和系统测试阶段解决掉
- 平均5%的Bug会被交付给用户





2.2.4验收测试

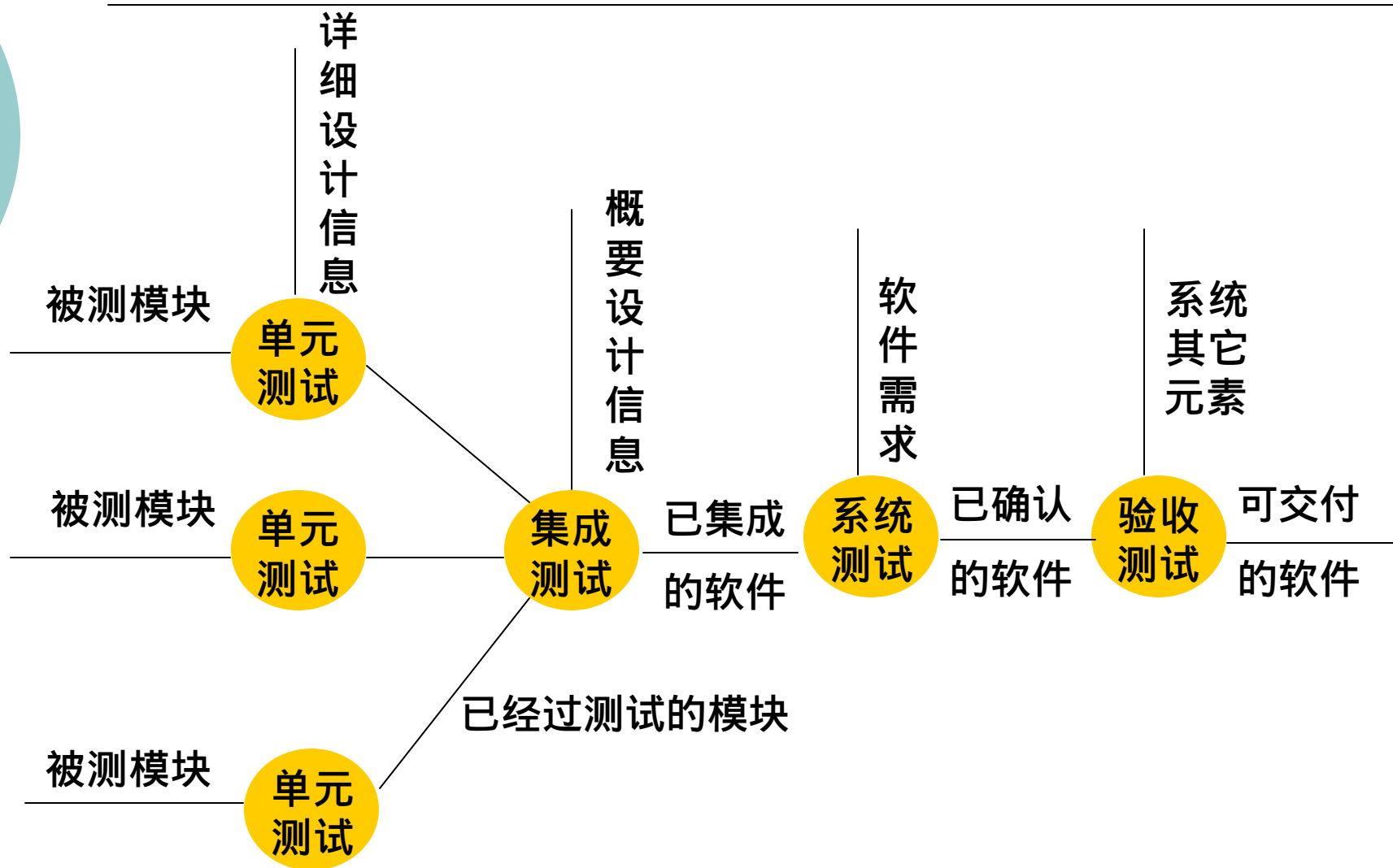
- 系统测试结束后，在项目组看来开发和测试工作已经全部完成，可以交付使用，并与用户一起进行测试，以验证是否符合与用户事先约定的验收标准。
- 测试人员
 - 产品经理或其他高级经理
 - 开发工程师
 - 测试工程师
 - 用户




2.2.5测试阶段与测试方法

测试级	目的	执行者	测试环境	测试方法
单元	从单个模块中发现逻辑、数据和运算缺陷	软件工程师	单独的；桩和支撑程序	白盒测试
集成	发现模块间接口缺陷	软件工程师	单独的和/或模拟；桩和支撑程序	白盒测试 <i>Top-down, bottom-up, 或outside-in</i>
系统	测定软件是否满足需求	软件质保组； 软件确认组	实际的环境（可能没有最终的硬件）	功能测试和ALAC
回归	确认软件经过一些小的变更或修改后是否仍满足所有的需求	软件质保组； 软件确认组	实际的环境（可能没有最终的硬件）	功能测试和ALAC
验收	确定软件是否满足客户的需求	客户，软件质保组和/或项目组	实际的环境（通常在客户方）	功能测试和ALAC（客户可能有自己的测试方法）

一般的软件测试过程





软件测试的基本过程

软件
开发
人员.

回归测试 终结回归测试