

# 软件测试技术 — 软件测试管理

古乐

2004-06-27

交大慧谷软件测试沙龙四期主题演讲

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 什么是管理？

- 管理的任务就是保证项目能成功地完成。项目管理从定义工作和制定计划开始，项目管理的基本的原则是：
  - 每个项目都有一个基于承诺的计划
  - 管理制度解决了项目之间和组织结构之间的固有冲突
  - 利用监控和评审系统对照计划审计和跟踪过程

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 改进软件测试的质量

- 软件测试管理是软件项目管理的一个子集或分支，与传统的软件项目管理在核心上没有本质的区别，但由于管理对象的特殊性，因此在具体执行上具有较大的不同。
- 为改进软件测试的质量，组织必须按照下面的六个步骤进行：
  - 理解当前的测试过程的状态
  - 开发希望的过程
  - 按照优先级列出需要改进的活动
  - 为这些改进活动制定计划
  - 承诺执行活动所需的资源
  - 重新开始步骤1

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 定义软件测试过程

- 定义软件测试过程可以指导员工去选择合适的途径。通过定义软件测试过程，他们能更好地理解他们应做什么，应期望同事们做什么，这使他们能关注于他们的工作。定义软件测试过程还为组织提供了一种工作框架，并且允许每个人对其进行调整以适应他们的需要。

# 一些要考虑的地方

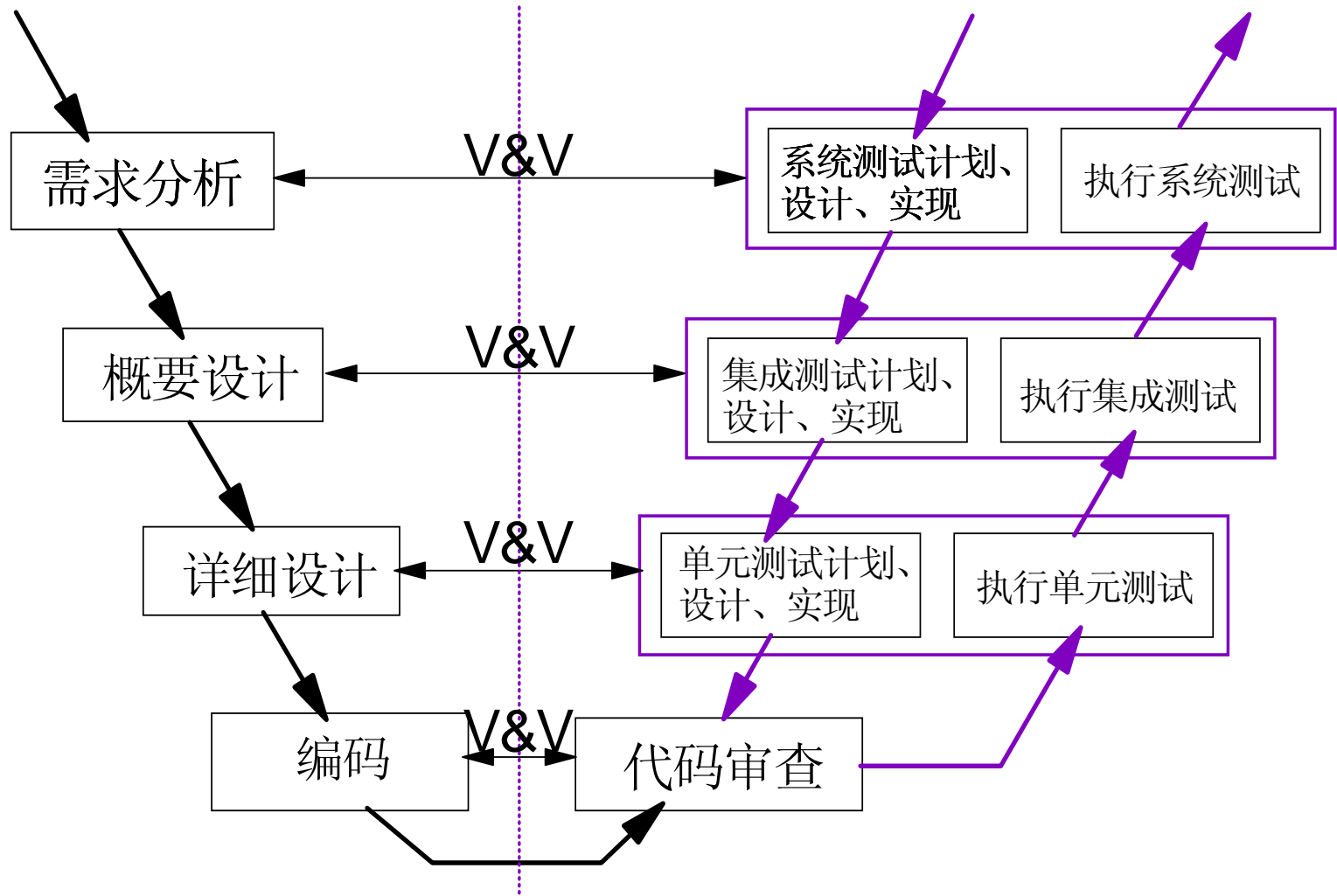
- 因为软件项目各不相同，它们的软件工程过程也必然会有区别
- 如果没有统一的软件工程过程，组织和项目组就只能各自定义适合它们需要的软件测试过程
- 为项目组指定软件测试过程必须考虑成员的经验水平，当前产品的状态以及可用的工具与设备



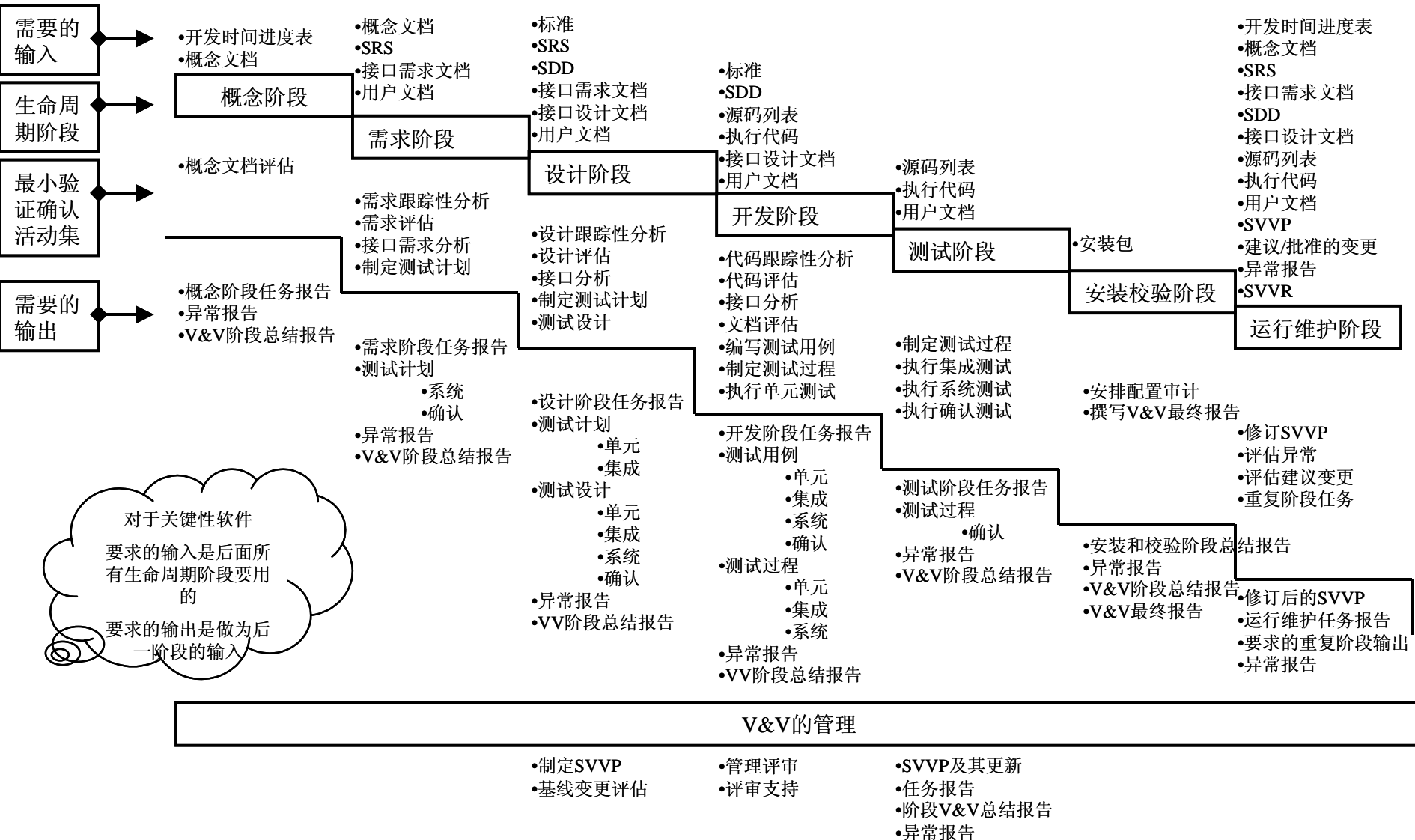
# 制定过程标准

- 尽管有很明显的理由为项目制定项目特定的软件测试过程，但是也同样存在相反的理由要求在组织内实现过程的标准化的，因为：
  - 标准的过程可以减少组织在培训、评审、工具支持等方面遇到的问题
  - 如果有标准的方法，则每个项目的经验可以供整个组织所共享
  - 过程标准为过程与质量度量提供了标准
  - 因为定义过程要花费工作和时间，因此为每个项目组定义新的过程是行不通的

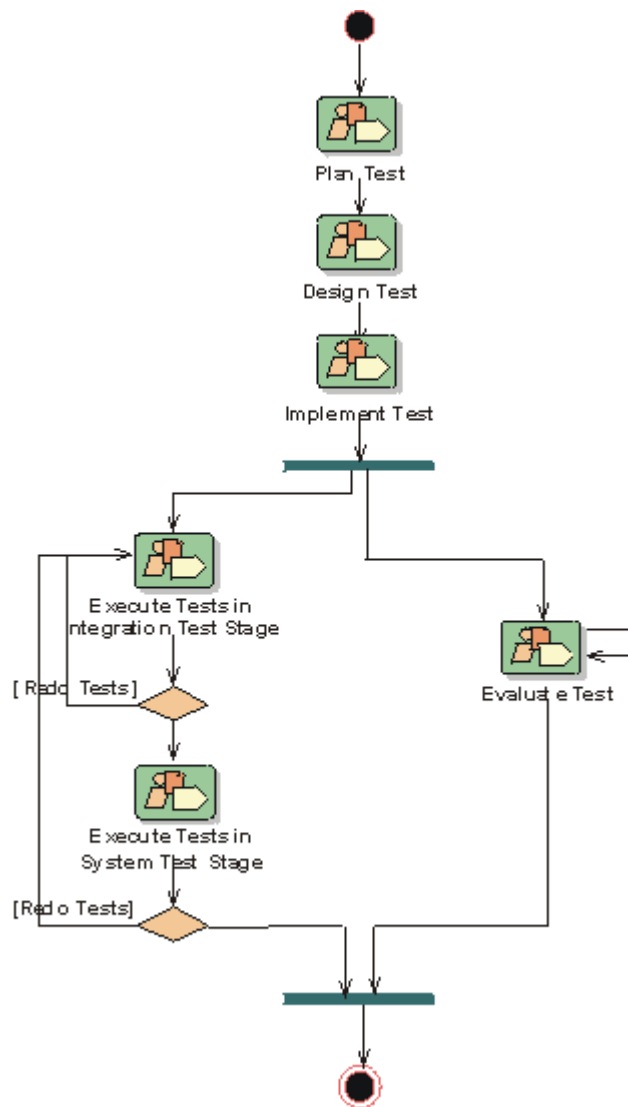
# 传统V模型



# 验证与确认模型



# RUP测试过程模型



# XP的测试过程

- 在XP的过程中，测试处于一个比较特殊的地位，XP要求“先写测试，后编码”。失败的测试用例驱动编码和设计，可以减少不必要的开发量。开发人员必须保证单元测试和集成测试视作用运行无误，甚至现场的客户代表也要能够编写功能测试程序。这是所有软件工程方法都一致推荐的做法。无论怎么强调测试的重要性都不为过。

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 测试计划模板

1. 介绍
  1. 1目的
  1. 2测试策略
  1. 3范围
  1. 4参考文献
  1. 5术语
2. 测试项目
3. 被测特性
4. 不被测特性
5. 测试方法
6. 测试通过/失败标准
  6. 1测试结果审批过程
7. 测试挂起/恢复的条件
8. 系统测试交付物
9. 测试任务
10. 环境需求
  10. 1硬件需求
  10. 2软件需求
  10. 3测试工具
  10. 4其它

# 测试计划模板（续）

11. 角色和职责
12. 人员及培训
13. 系统测试进度
14. 风险和应急计划
15. 审批



# 测试计划的跟踪目标

- 对照计划跟踪实际执行的结果；
- 当出现偏差的时候，采取纠正活动并进行管理；
- 对变更进行承诺

# 测试计划跟踪政策

- 使用和维护软件测试计划文档，作为软件测试跟踪的基础。
- 测试经理应保持对软件测试任务状态和问题的了解。
- 当软件测试计划不能完成时，通过调整执行或调整计划，采取纠正行动。
- 在相关小组参与和同意的情况下，作出计划承诺的更改。
- 上级管理部门评审所有对组织外个人和团体的承诺更改以及新的软件承诺约定。

# 计划跟踪的内容

- 跟踪进度的偏差情况；
- 跟踪资源的使用情况，包括人力资源、环境资源；
- 跟踪风险，风险的管理包括风险的识别、风险的规避、风险应急计划；
- 跟踪工作量的偏差；
- 跟踪规模的偏差。

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 测试用例模板

测试项编号	ST_FUN_TD_001_001		
优先级			
测试项描述			
预置条件			
用例序号	输入	执行步骤	期望结果
<b>001</b>		1、	
<b>002</b>			
<b>003</b>			

# 测试用例样例

测试项编号	UT_TC_002_001_002		
优先级	中		
测试项描述	测试目录参数合法，但检索层次非法情况		
预置条件	无		
用例序号	输入	执行步骤	期望结果
001	dir="Dir1" lev=-2 g_DirRoot=NULL		返回FALSE; 反馈目录检索层次设置 错误; 全局变量 g_DirRoot 内 容不变;
002	dir="Dir1" lev=81 g_DirRoot=NULL		返回FALSE; 反馈目录检索层次设置 错误; 全局变量 g_DirRoot 内 容不变;

# 测试用例的跟踪

- 用例的规模；（用例的总数、各个优先级用例数、某个特性的测试用例数、某个版本计划执行的用例数等）
- 用例的覆盖情况；（包括代码覆盖率、特性覆盖率等）
- 用例的执行频率；（用例在整个生命周期中被执行的次数）
- 用例的有效性；（用例发现缺陷的有效性）

# 目录

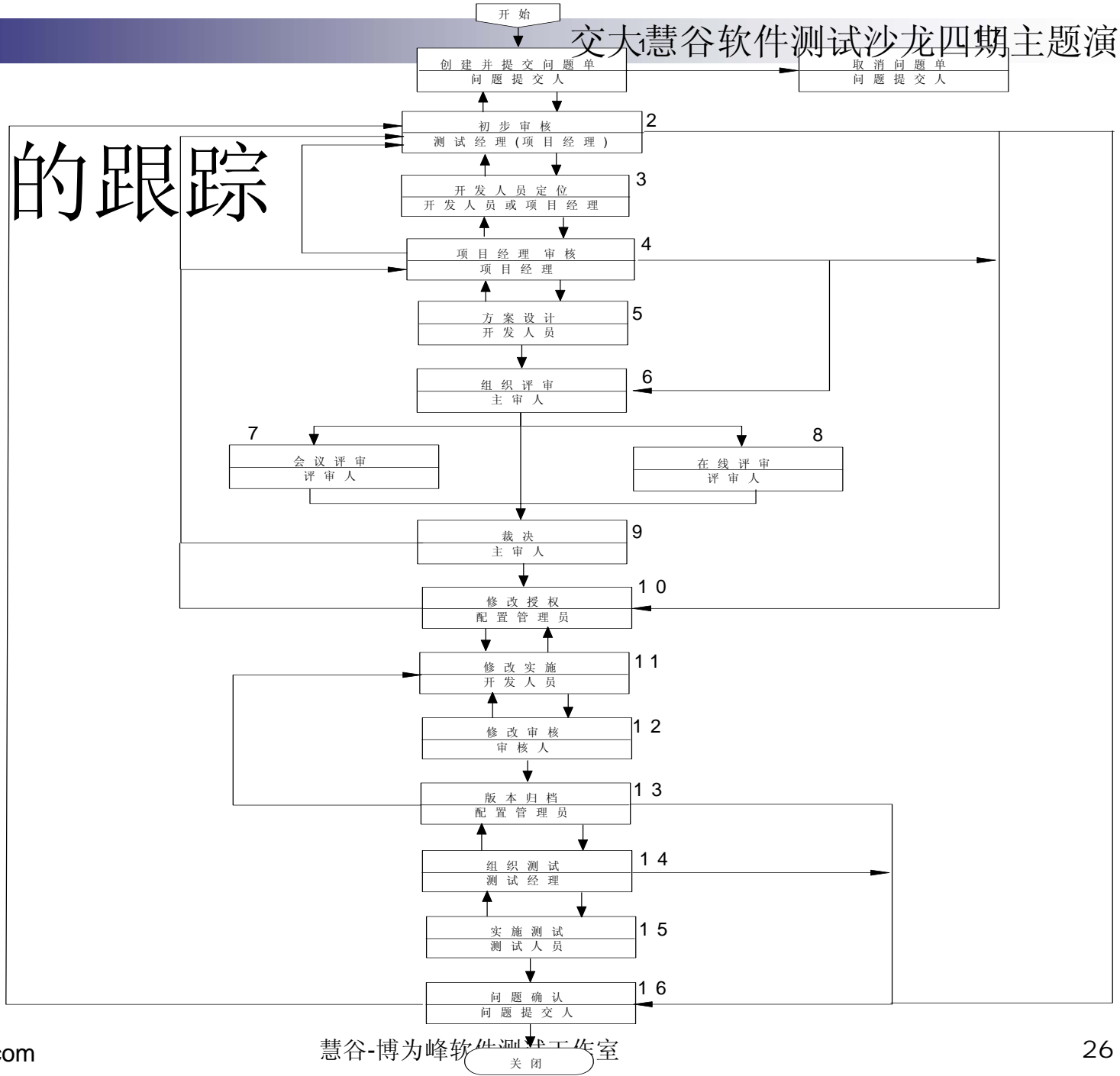
- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程



# 缺陷的属性定义

- 基本属性：产品、版本、特性、发现缺陷的活动、触发因素、结果影响、发现问题阶段、严重程度、紧急程度、可否重现、（重现条件）等；
- 扩展属性（缺陷定位后确认）：子系统、模块、问题产生根源、缺陷类型、责任人、缺陷年龄等；

# 缺陷的跟踪



# Gompertz分析方法

## ■ 优点:

- 数据收集方式简单
- 具有广泛的业界应用实践基础
- 适用于大部分公司的缺陷发现模式

## ■ 应用条件:

- 测试过程的稳定性
- 测试执行的随机性
- 测试覆盖的全面性

# 四象限分析方法

## ■ 含义:

- 最好情况 — 高工作量/低遗留缺陷
- 正常情况 — 高工作量/高遗留缺陷
- 最坏情况 — 低工作量/高遗留缺陷
- 不确定情况 — 低工作量/低遗留缺陷

# Rayleigh分析方法

## ■ 基本假设:

- 开发过程中的缺陷发现率与现场的缺陷发现率成正比 —— **Myers**定律;
- 在缺陷数一定的情况下, 如果越多的缺陷被越早发现和清除, 那在后面阶段遗留的缺陷就越少——现场缺陷越少。

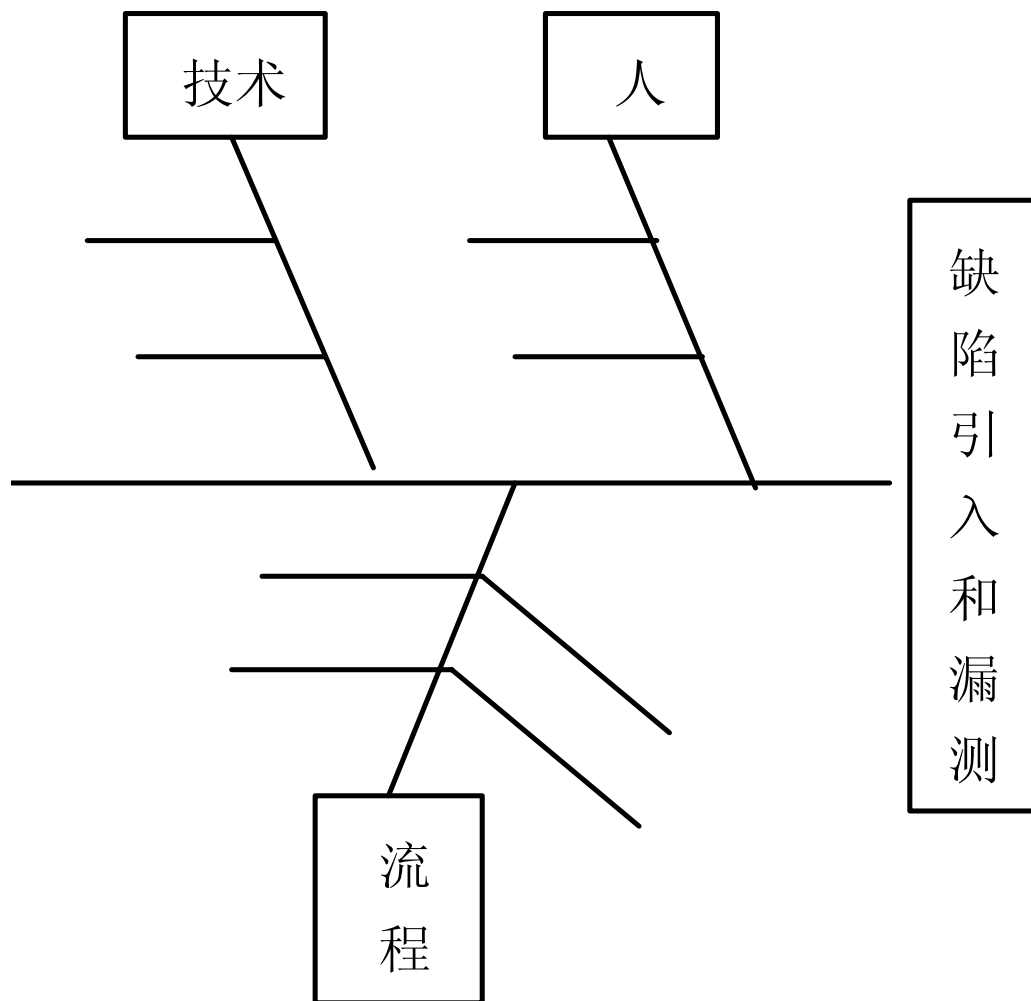
## ■ 应用

- 量化评估和预测
  - 评估产品质量
  - 预测现场缺陷 (通过调节因子提高预测有效性)
- 开发过程质量管理
  - 缺陷率的峰值越早出现越好;
  - 尽可能降低**Rayleigh**曲线。

# 根本原因分析方法

- 根本原因分析是一种通过识别和消除产生缺陷的关键原因，从而实现持续改进的方法论。
- 根本原因分析是针对单个缺陷或改进点进行的深入的分析活动。目的是避免缺陷的再次产生。
- 根本原因分析方法可以借用各种工具，例如鱼骨图、柏拉图等。

# 根本原因分析方法（续）



# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程



# 环境搭建

- 针对不同的测试目的构造不同的测试环境；
- 测试环境的构造应最大程度上有利于自动化；
- 测试环境应能够很好的接受测试的输入；
- 测试环境应能够把测试执行的结果反馈给测试人员；

# 用例执行

- 配置输入条件；
- 按用例执行步骤执行用例；
- 仔细观察每个可能的输出结果，与期望结果比较，记录差异点；
- 发现可能的缺陷；（由于用例不可能遍历每个可能的输出，因此不同的人在执行同一个测试用例的时候，可能会得到不同的结果，这是一个经验的积累）
- 避免用例之间的干扰，排除人为产生的错误；
- 隔离缺陷，协助开发人员定位问题；
- 如实的记录每个缺陷，缺陷信息应当详尽，避免歧义，并利于问题的重现；

# 执行日志

- 在测试执行过程中，每天都应当记录测试执行日志，一般测试执行日志应当包含下列内容：
  - 执行了哪些用例，谁执行的，是否通过，发现了哪些缺陷，总体的测试进展情况，可能的风险，遇到的问题等。

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 报告的内容

- 1 介绍
- 2 测试时间、地点和人员
- 3 测试环境描述
- 4 测试数据度量
  - 4.1 测试用例执行度量

被测对象	用例总数	执行总数	OK项	POK项	NG项	NT项	发现缺陷数	规模

# 报告的内容（续一）

## 4.2 测试进度和工作量度量

### 4.2.1 进度度量

任务	计划开始时间	计划结束时间	实际开始时间	实际结束时间

### 4.2.2 工作量度量

执行任务	开始时间	结束时间	工作量 (人时)

# 报告的内容（续二）

## 4.3 缺陷数据度量

被测对象	总数	致命	严重	一般	提示	设计错误	赋值错误	算法错误	接口错误	功能错误	其它

## 4.4 覆盖率数据度量

标识符	名称	PPP

# 报告的内容（续三）

## 4.5 综合数据分析

- 计划进度偏差 =  $(\text{实际进度} - \text{计划进度}) / \text{计划进度} \times 100\%$
- 用例执行效率 =  $\text{执行用例总数} / \text{执行总时间 (小时)}$
- 用例密度 =  $\text{用例总数} / \text{接口规模} \times 100$
- 缺陷密度 =  $\text{缺陷总数} / \text{接口规模} \times 1000$
- 用例质量 =  $\text{缺陷总数} / \text{用例总数} \times 100$
  
- 缺陷严重程度分布饼图：
- 缺陷类型分布饼图：



# 报告的内容（续四）

## 5 测试评估

### 5.1 测试任务评估

评估结论：本次测试执行准备充足，完成了既定目标。

### 5.2 测试对象评估

评估结论：测试对象符合集成测试阶段质量要求，可以进入到系统测试执行阶段。

## 6 遗留缺陷分析

## 7 审批报告

提交人签字：

日期：

开发经理签字：

日期：

产品经理签字：

日期：

# 报告的内容（续五）

## 8 附件

附件1 测试用例执行表

附件2 测试覆盖率报告

附件3 缺陷分析报告

缺陷编号	简要描述	产生原因	预防措施	对测试的建议

# 报告的内容（续六）

## 附件4 缺陷漏测分析

缺陷编号	简要描述	单元测试有用例，但未测试	单元测试应当发现，但没有用例	单元测试缺陷修改引入的问题	新增需求引入的问题	开发拒绝修改的问题

漏测缺陷编号	缺陷描述	漏测原因分析	预防措施	责任人	时间

# 报告的管理

- 每轮正式的版本测试，都应当有一个完整的测试报告。

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 需求管理的目的

- 控制需求是使软件过程稳定、成功地达到可重复级的至关重要的因素。需求管理着重点是管理和控制需求，而不是需求工程。

# 测试需求管理

- 保证所有的测试需求都有相应的测试用例；
- 保证所有的测试用例都能回溯到测试需求；
- 保证所有的用例至少被执行过一次；
- 用于帮助确定回归测试的范围。

# 测试跟踪矩阵

## ■ 用例——规格对应矩阵

	测试规格1	测试规格2	测试规格3	。 。 。	测试规格n
测试用例1					
测试用例2					
测试用例3					
。 。 。					
测试用例n					



# 测试跟踪矩阵（续）

## ■ 用例——版本对应矩阵

	版本1	版本2	版本3	。 。 。	版本n
测试用例1	(OK, NOK, POK, NT)				
测试用例2					
测试用例3					
。 。 。					
测试用例n					

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 目标

- 目标1: 计划SCM 活动
- 目标2: 相应的软件部件(work products)被标识、控制、并是可用的
- 目标3: 对软件部件更改是受控的
- 目标4: 软件基线的状态和内容通知到受影响的组织和个人

# 承诺执行

- 承诺1：每个项目的**SCM**责任明确
- 承诺2： **SCM**贯穿项目的整个生命周期
- 承诺3： **SCM**应用于所有的软件部件，包括支持工具
- 承诺4： 建立基线库(**Baseline**)系统
- 承诺5： 定期评审基线库内容和**SCM**活动

# 执行能力

- 能力1：成立**CCB**管理基线的建立和变更
- 能力2：相应的组织结构协调**SCM**的实施
- 能力3：足够的资源和预算支持**SCM**的实施
- 能力4：相关人员对目标、工具、方法论的培训

# 活动

- 活动1： 在成文的流程的指导下制定**SCMP**
- 活动2： 成文及经过评审的**SCMP**是实施**SCM**的基础
- 活动3： 建立数据库系统管理基线
- 活动4： 标识受控的软件内容（**CIs**）
- 活动5： 有成文的流程指导管理更改申请和问题报告

# 活动（续）

- 活动6：有成文的流程指导对基线变更的管理
- 活动7：有成文的流程指导对产品的发布
- 活动8：有成文的流程指导对CIs状态的记录与跟踪
- 活动9：有相应的各种报告产生和发布
- 活动10：有成文的流程指导对基线评审

# 度量与分析

- 度量1：跟踪并分析各SCM活动的实施状况，比如：
  - 单位时间变更申请的处理数量
  - 对照SCMP的执行情况
  - 人员、费用的增加等



# 验证

- 验证1、高级领导的定期检查
- 验证2、项目经理的定期和事件驱动式检查
- 验证3、配置管理小组定期检查基线内容与文档的一致性
- 验证4、SQA机构对配置管理的检查

# 需要纳入配置管理的项

- （软件测试需求）；
- 软件测试计划；
- 软件测试方案；
- 软件测试用例；
- 软件测试报告；

# 目录

- 什么是管理
- 软件测试管理
  - 定义软件测试过程
  - 测试计划管理
  - 测试用例管理
  - 软件缺陷管理
  - 软件测试执行
  - 软件测试报告
  - 软件测试需求管理
  - 软件测试配置管理
  - 自动化软件测试过程

# 测试自动化需求

- 我们试图自动化软件测试过程是由于我们需要改进我们工作的质量和产量所驱动的。当我们能减少一个有秩序的过程的任务并且使之机械化，那么我们不仅仅节省了劳动力，还阻止了人为的错误。

# 自动化什么？

- 计划管理工具
- 用例管理工具
- 缺陷管理工具
- 需求管理工具
- 静态分析工具
- 自动化执行工具
- 报告自动生成工具
- 度量分析工具
- 接口模拟器

谢谢

Thank You!