



国际化的软件测试管理方法



2011. 4. 21
上海滔瑞信息技术有限公司
imbus Shanghai
熊晓虹



简介



熊晓虹



- 国际软件测试工程师认证协会(ISTQB)中国分会专家
- 通过ISTQB认证的基础级测试工程师，高级测试经理

- 复旦大学计算机科学系学士
- 德国Erlangen-Nuernberg 大学信息科学专业硕士

- 现任上海滔瑞公司（IMBUS Shanghai）技术总监

- 曾任西门子通讯事业部（德国，北京）测试工程师，项目经理，高级部门经理
- 曾任伟创力（中国）电子设备有限公司高级总监，负责伟创力无线测试实验室
- 曾任IMBUS 德国公司，测试工程师，项目经理

- 17年无线通讯行业的开发与测试工作经验，多年的跨国项目管理经验，多年的跨国集团测试团队管理经验。



■ 问题的提出

■ 测试团队的管理

■ 软件测试过程的管理

问题的提出 — 有关软件质量的一些数据



	1994	1996	1998	2000	2002	2004	2006	2009
Successful	16%	27%	26%	28%	34%	29%	35%	32%
Challenged	53%	33%	46%	49%	51%	53%	46%	44%
Failed	31%	40%	28%	23%	15%	18%	19%	24%

- 2009年在所有软件项目中，只有32% 的项目是成功的。
- 而44%的项目或者拖时，或者超支，或者不得不压缩功能，或者都兼有之！
- 另有 24% 的软件项目不得不终止或者从来没有投入使用。
这里收集了全世界约 8 万个软件项目的数据！

来源: *The Standish Group*

Jorge Dominguez

- 软件系统的多样性和复杂性
- 软件项目组织的多样性和复杂性
- 日益增长的时间和费用上的压力

问题的提出

- 对需求不清晰
- 过于乐观的成本预算
- 对新技术的困难估计不足
- 管理层过高的期望目标
- 项目进程和成本监控不到位
- 项目参与人员的技能
- 软件配置管理和事件管理
- 项目的规模和组织的复杂程度
- 开发与测试的关系
- 测试不系统
- 危机管理的缺失
- 质量体系和相关过程不完善
- 等等

应该记住

软件的质量取决于

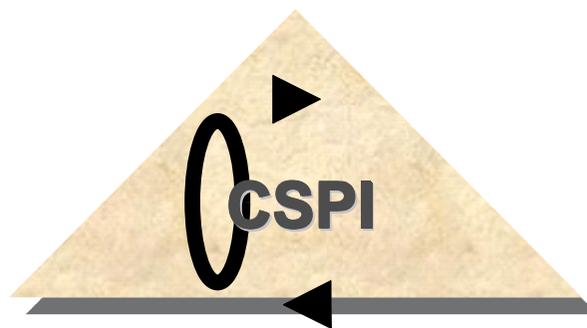
- 人员的素质
- 过程的质量
- 使用的技术

要取得好的软件质量
就需要这三个方位的
全面有效地管理

人员

过程

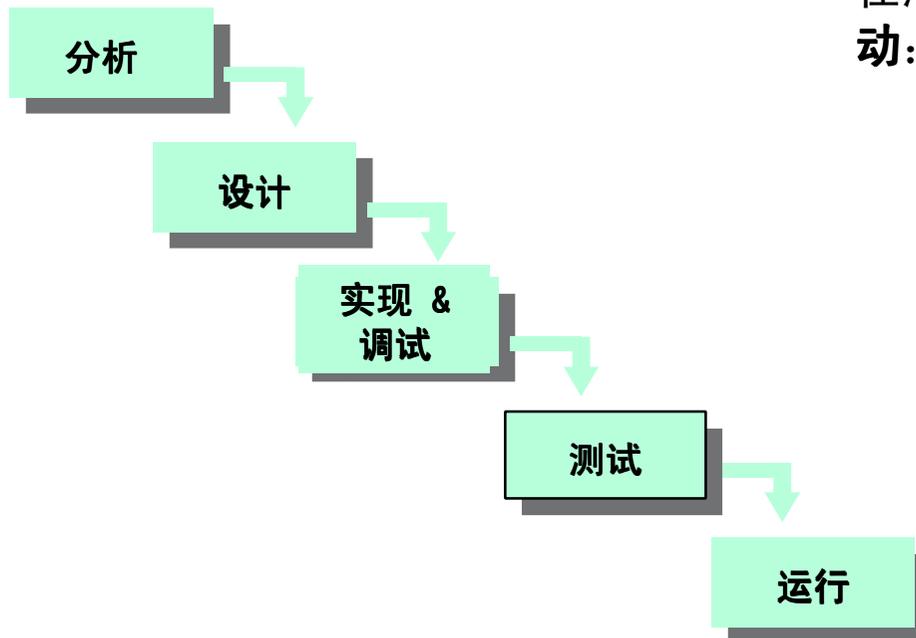
技术



持续软件过程改进 (Continuous SW Process Improvement)

测试在国际上
已经作为一门
独立的学科，
为软件质量保证
提供建设性和
分析性的措施

传统：瀑布模型

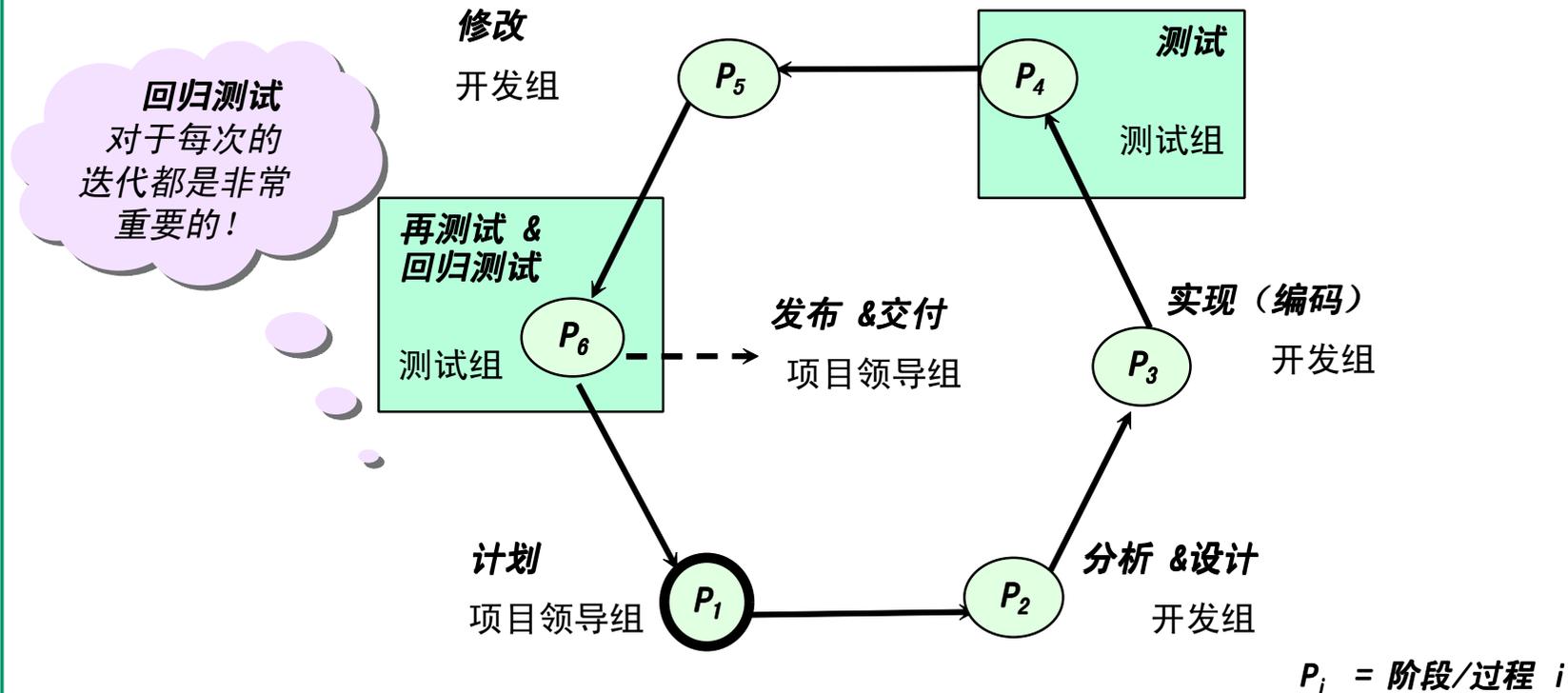


在瀑布模型中，测试作为“一次性”活动：

- 交给用户使用前的“产品检验”
- 软件项目结束的“最后检查”
- 在模型中即没有包括修改错误的过程，也没体现项目的延续

nach: Wallmüller Ernest, Software-Qualitätssicherung in der Praxis, Hanser Verlag, München, 1990.

现代：迭代-增量开发模型 (*iterative-incremental development model*) (I)

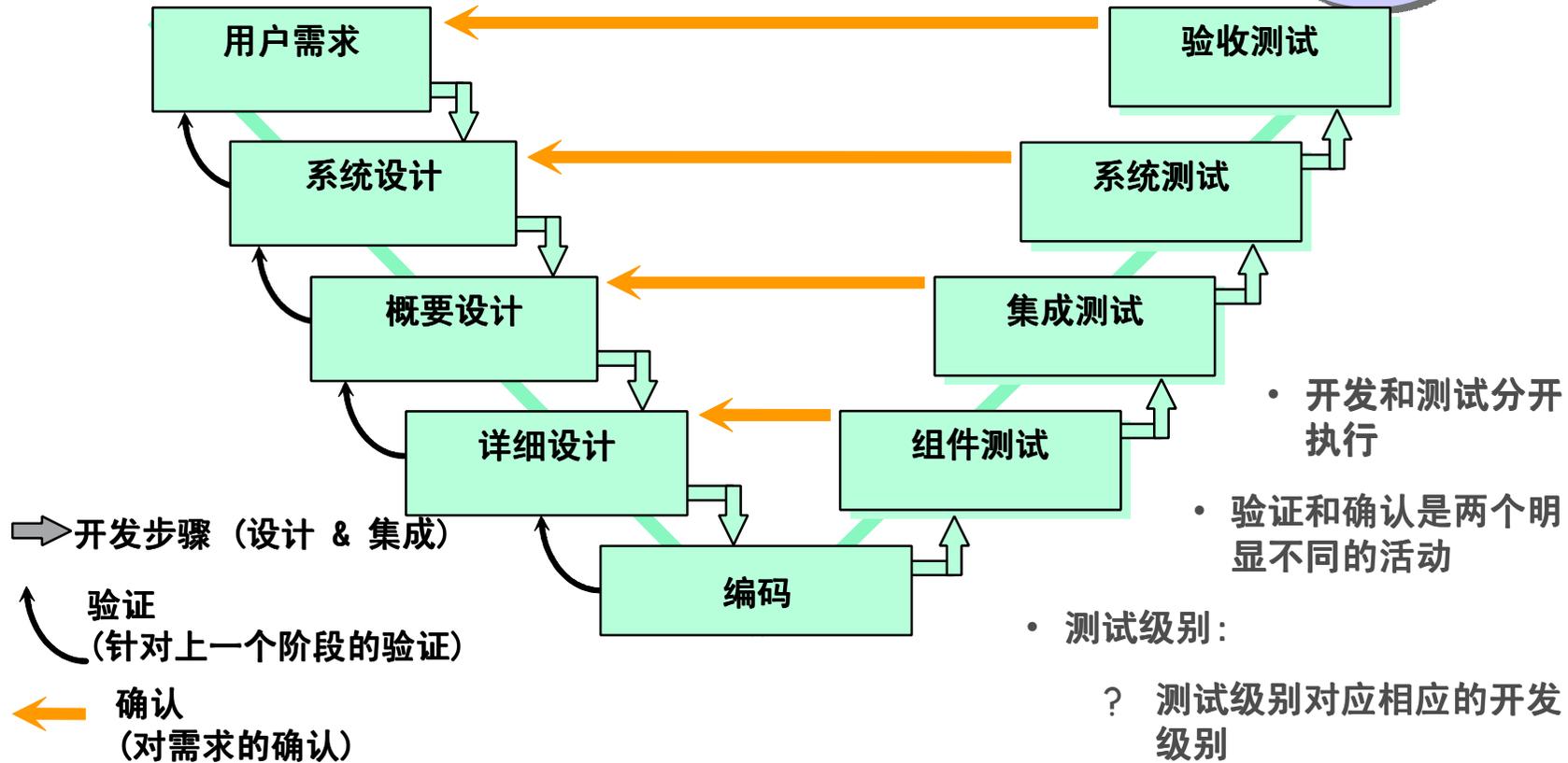


s.a: Balzert, Lehrbuch der SW-Technik, Bd. 2, Spektrum Akademischer Verlag, 1998, S.97ff.

迭代-增量开发模型的例子：原型开发 (Prototyping)、快速应用开发 (Rapid Application Development RAD)、RUP/统一软件开发过程 (Rational Unified Process) 和敏捷开发模型 (agile development) 等。

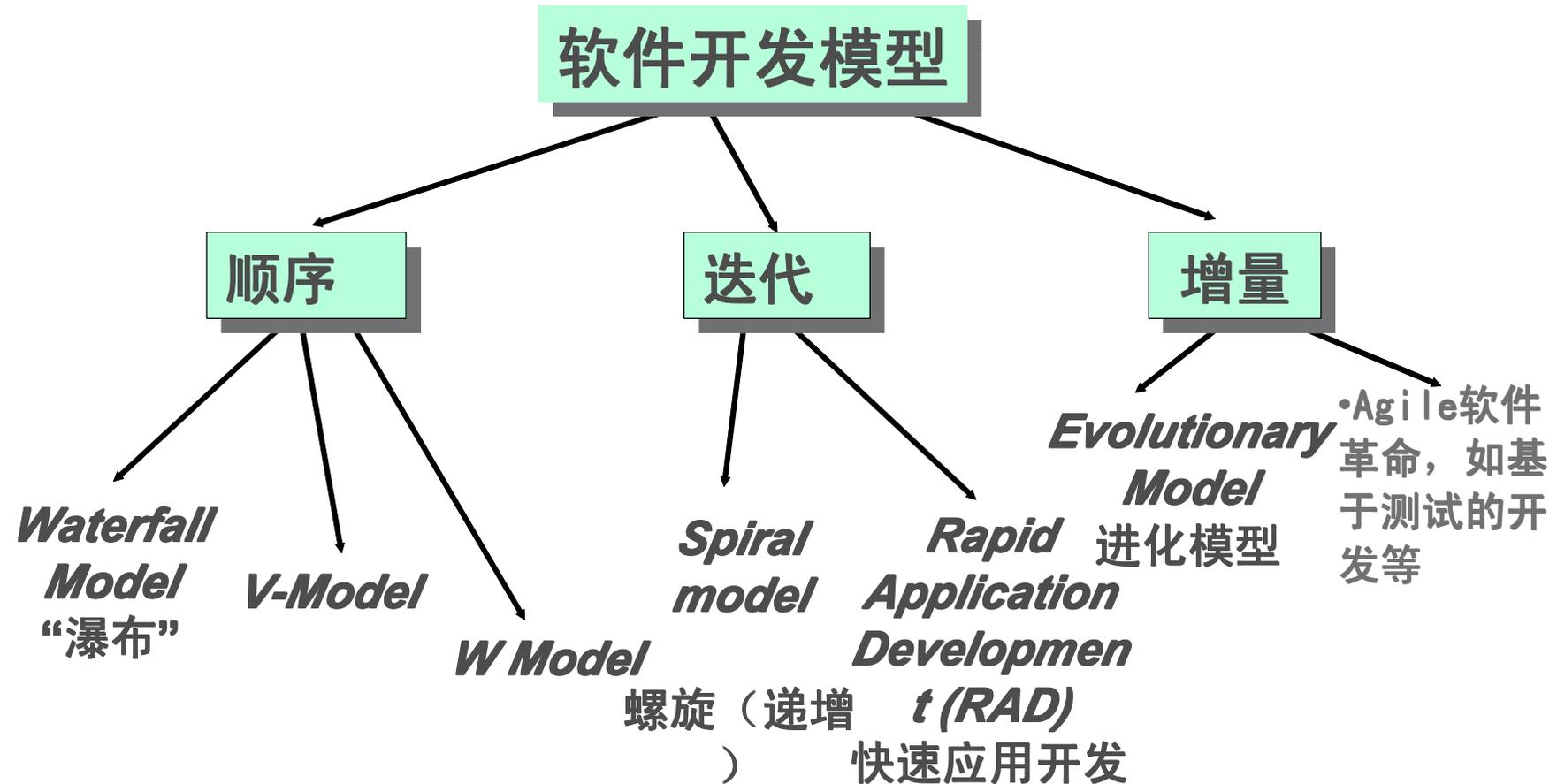
经典：V-模型 (V-model)

在实际运用中，通常会对V-模型做相应调整（对测试级别进行合并或重新组合）！



sa: A.-P. Bröhl, Das V-Modell, Oldenbourg Verlag

其他软件开发模型



从不同的软件生命周期模型中我们可以得出结论：

- 软件测试和软件开发一样是软件生命周期中的一个不可缺少的组成部分
- 不同的软件生命周期模型决定了不同的软件测试管理模式

■ 什么是测试？

“测试” – I S T Q B 定义

测试 =

是一个过程，它包括了软件生命周期的所有活动，有**静态的**也有**动态的**。它涉及到**计划**、**准备**和对软件及其相关工作产品的**评估**，目的是

- **判定软件或软件的工作产品是否满足特定需求，**
- **证明它们是否符合目标，**
- **发现缺陷。**

Definition from glossary <http://www.istqb.org/downloads/glossary-current.pdf>

- ISTQB成立于2002年11月，是针对软件测试人员的国际上唯一的资质认证机构
 - 是基于实践的
 - 通过认证的不同级别提供了一种职业规划
 - 提升软件测试的专业化
 - 集合了许多专家的智慧，包括实践专家、咨询顾问、培训专家以及学术专家等
 - 对于考试而言，培训是可选的
- 能证实软件测试人员的专业能力

ISTQB成员国的地理位置



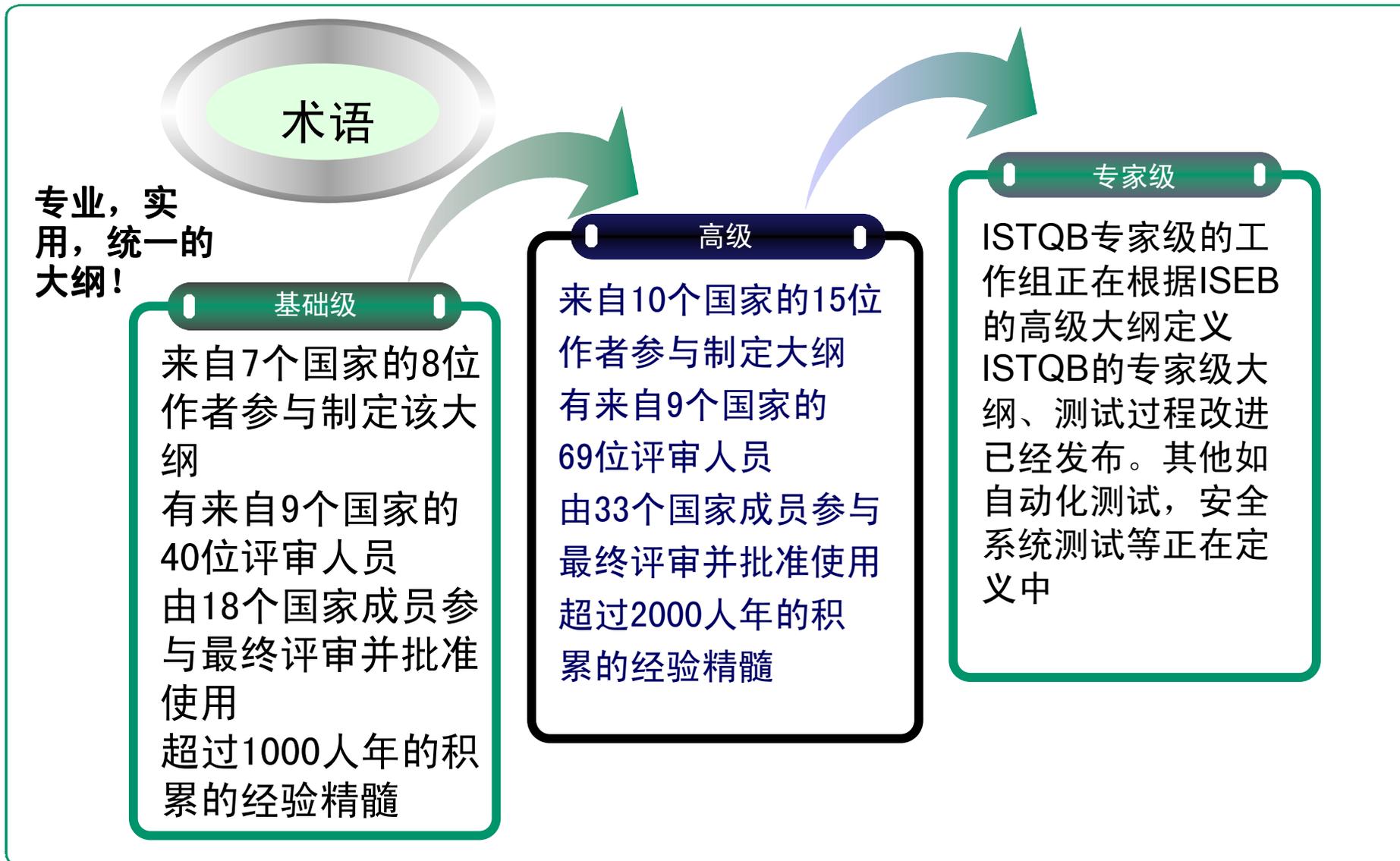
● 成员国

● 南美西班牙语系国家

● 保加利亚、塞尔维亚
罗马尼亚、黑山

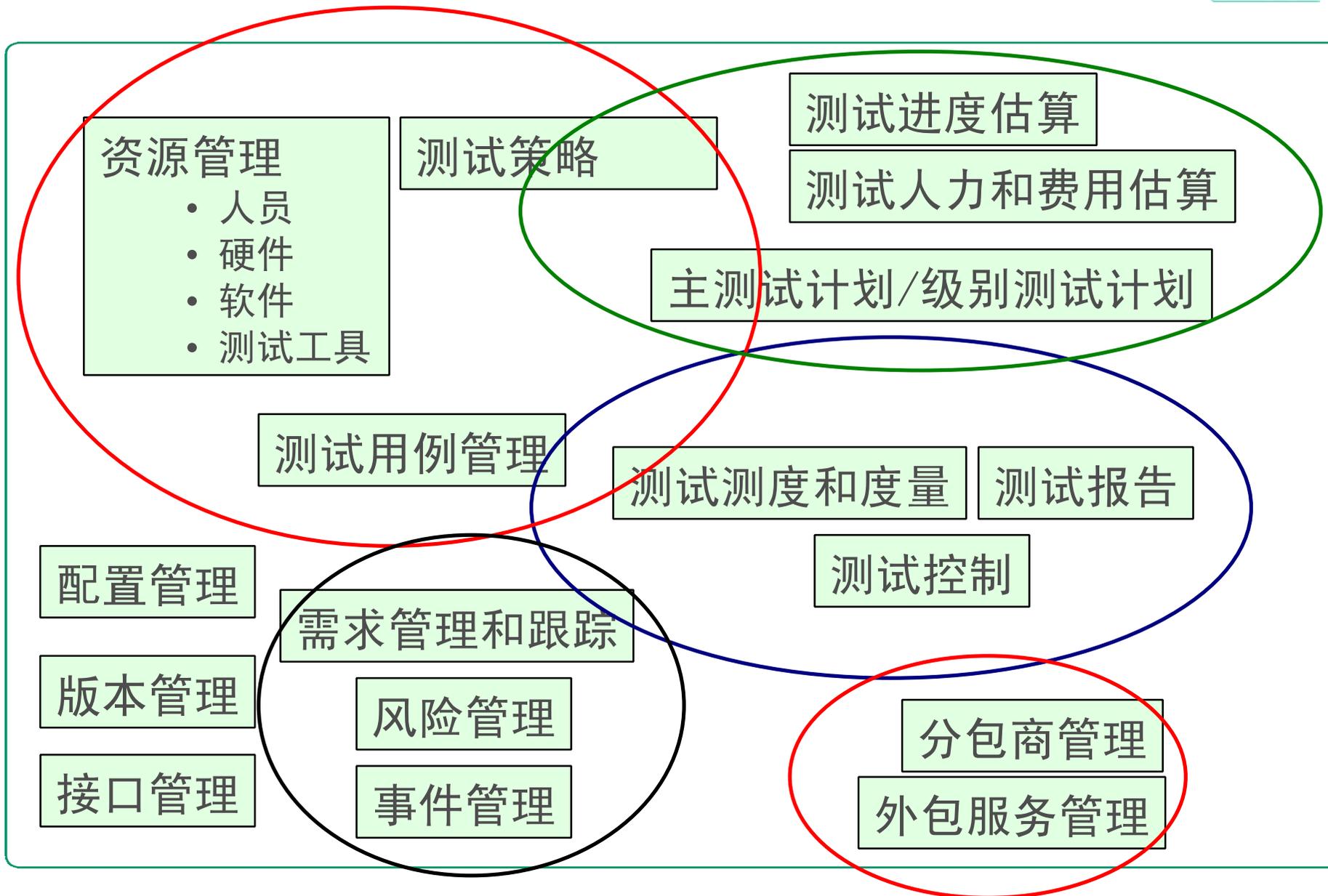
● 海湾国家（伊朗、伊
拉克、科威特等）

统一的术语表大大节约了沟通成本



- **国际性:** 是国际唯一的权威软件测试资质认证, 全世界已有超过16.5万的软件测试工程师通过了ISTQB的认证
- **统一性:** 全球统一的培训大纲
- **公正性:** 独立授权机构培训, 独立授权机构考试
- **可信度:** 被至少超过50个主要软件国家共同承认
- **唯一性:** 唯一国际性的软件测试专业/就业人员认证
- **独立性:** 不服务于任何政治团体和国家
- **国际最高水平:** 由世界软件测试权威们共同制订培训大纲和教材, 代表世界最高水平和最新研究成果。

软件测试管理的内容



■ 问题的提出

■ 测试团队的管理

■ 软件测试过程的管理

- 个人技能
- 团队的整体实力
- 组织的适合测试
- 团队的沟通和激励

如何才能成就一个测试人才?

相关领域的业务知识

业务实力
hard skill

实践经验

培训
(e.g. Cert. Tester)

系统的使用

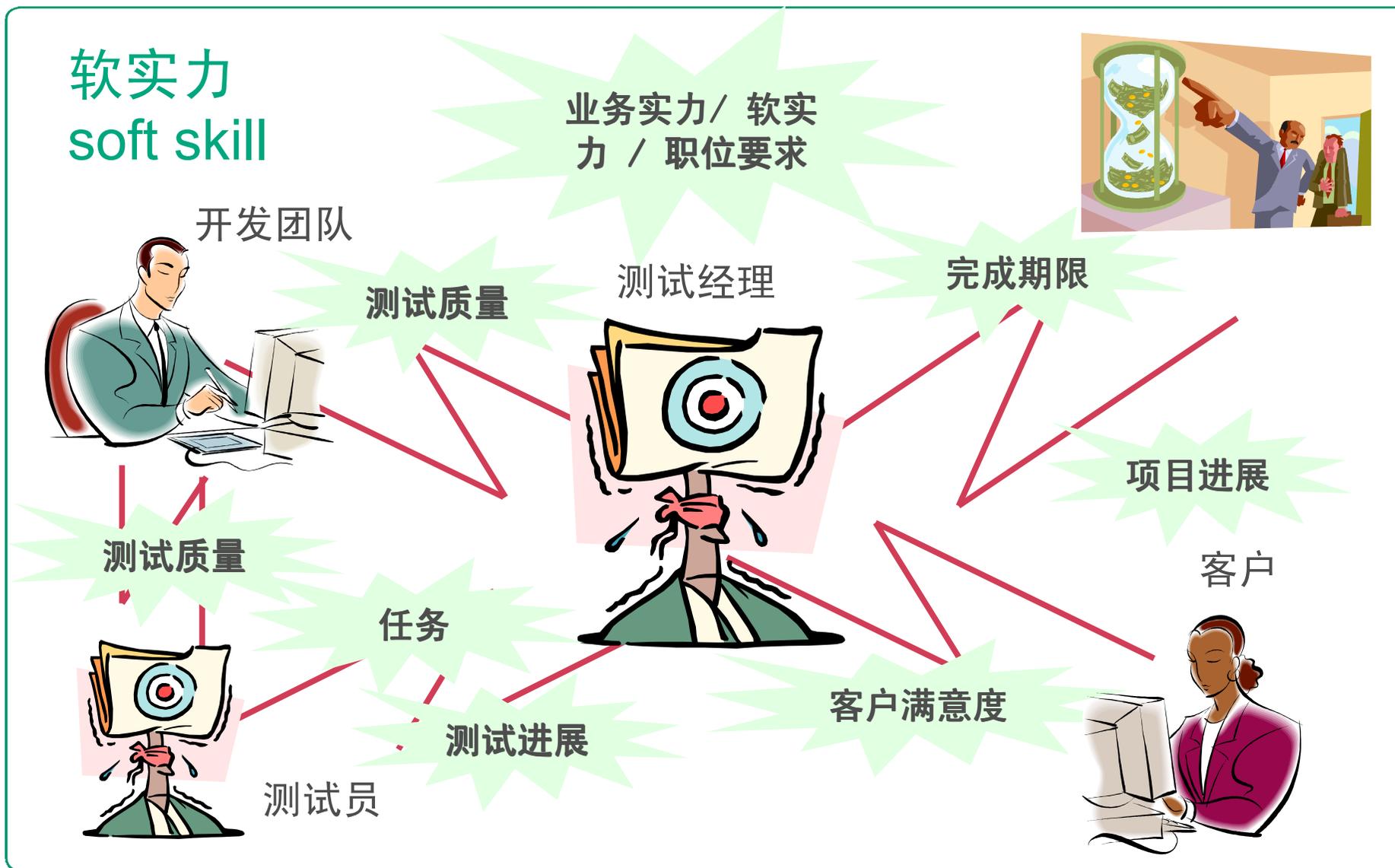
系统的开发

系统的测试

测试员

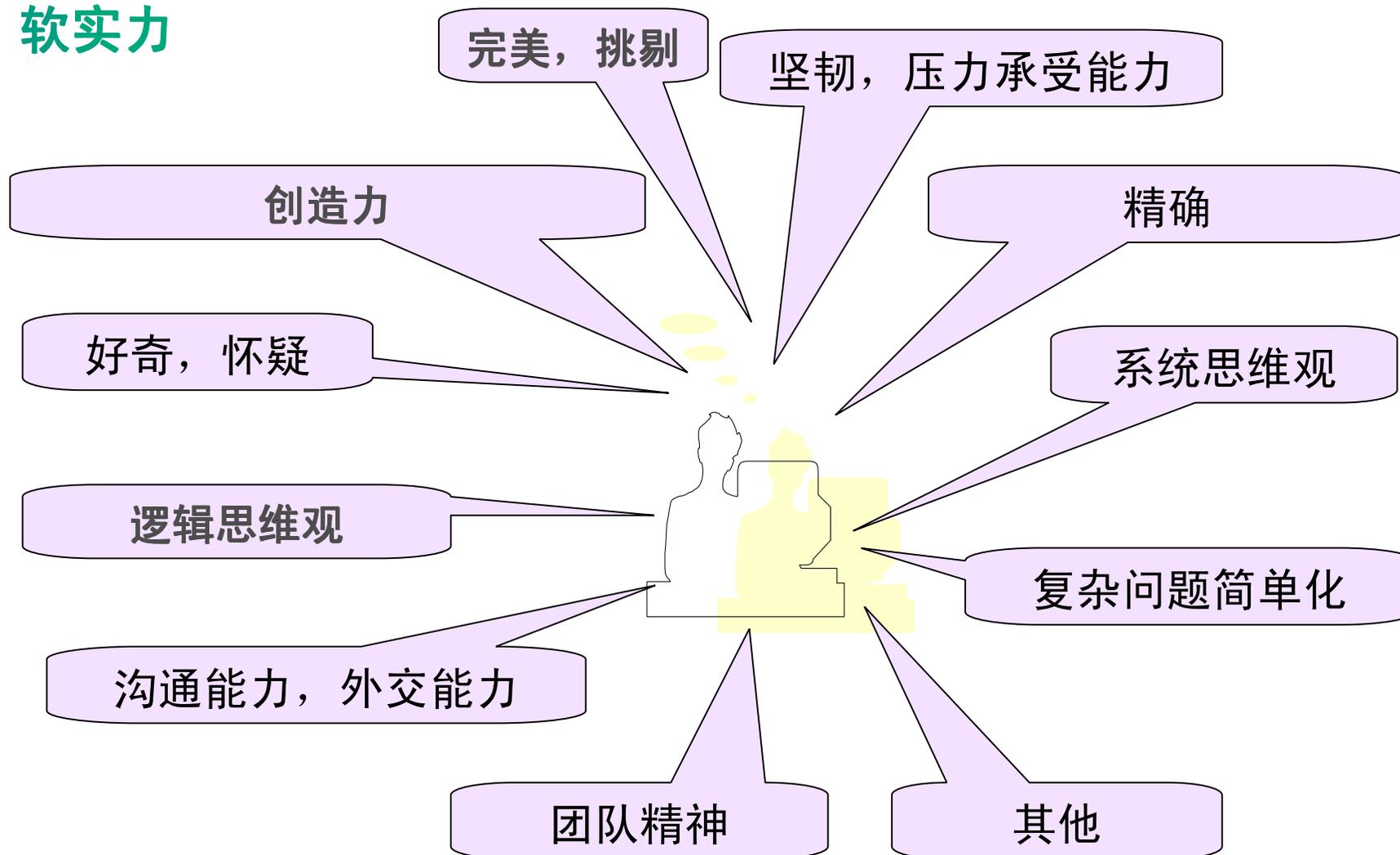


如何才能成就一个测试人才?

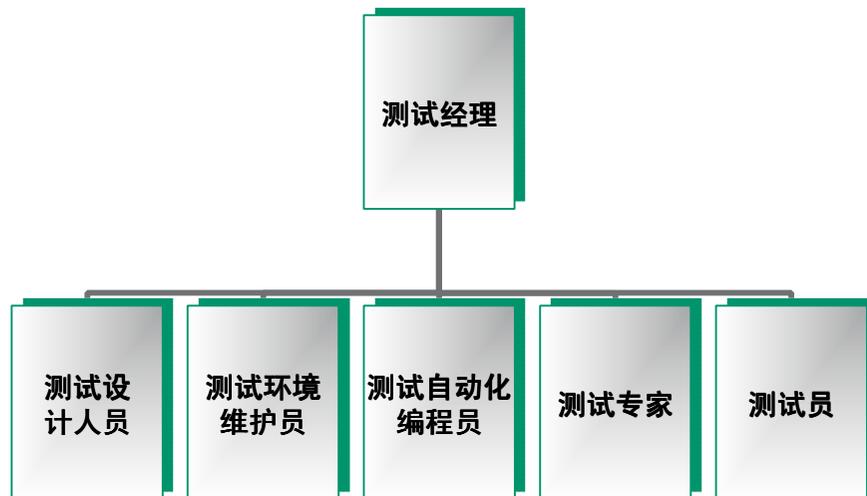


如何才能成就一个测试人才？

软实力



专业角色



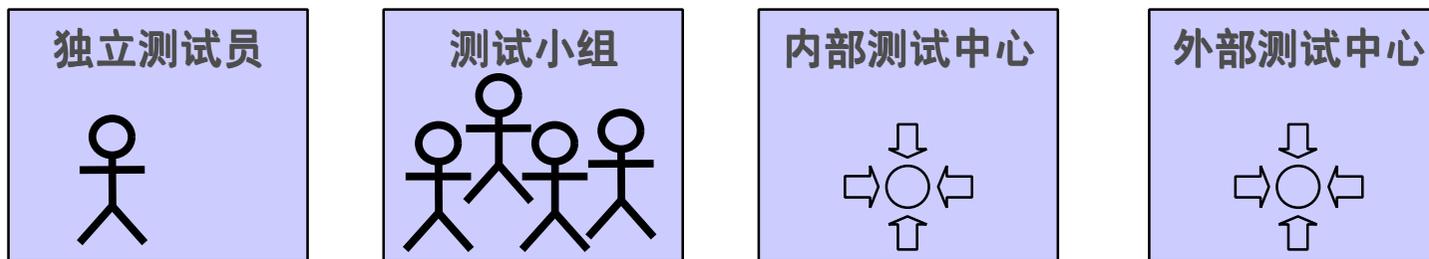
团队角色

Team-Role Contribution

-  **PLANT (PL)**
Creative, solves difficult problems
-  **RESOURCE INVESTIGATOR (RI)**
Enterprising, quick to explore opportunities
-  **CO-ORDINATOR (CO)**
Makes good use of group activities
-  **SHAPER (SH)**
Thrusting and challenging
-  **MONITOR EVALUATOR (ME)**
Discerning and objective
-  **TEAMWORKER (TW)**
Co-operative, averts friction
-  **IMPLEMENTER (IMP)**
Organised, efficient, practical
-  **COMPLETER FINISHER (CF)**
Painstaking, conscientious
-  **SPECIALIST (SP)**
Single-minded, professionally dedicated

组织对测试的适应

例子



- + 测试知识
- + 服务性
- + 独立性
- + 清晰的接口 (测试件)
- + 受尊重度
- + 费用控制
- 组织复杂度
- 组织可能与开发隔离
- 开发人员可能对质量不够负责

- + 专项知识
- + 和研发人员的紧密联系
- + 被研发人员接受
- 隔离
- “隧道效应”

- 组织对测试团队的重视，开发团队与测试团队的相互尊重，认可，理解，协作

■ 激励测试团队成员的十诀

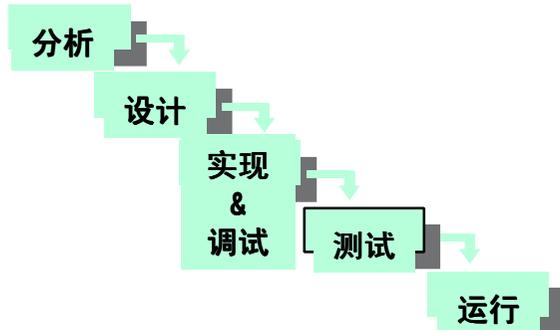
- 明确目标
- 个别交流
- 给予空间
- 绩效月评
- 正面鼓励
- 及时表彰
- 团队活动
- 适当放假
- 投其所好
- 庆祝成功

■ 问题的提出

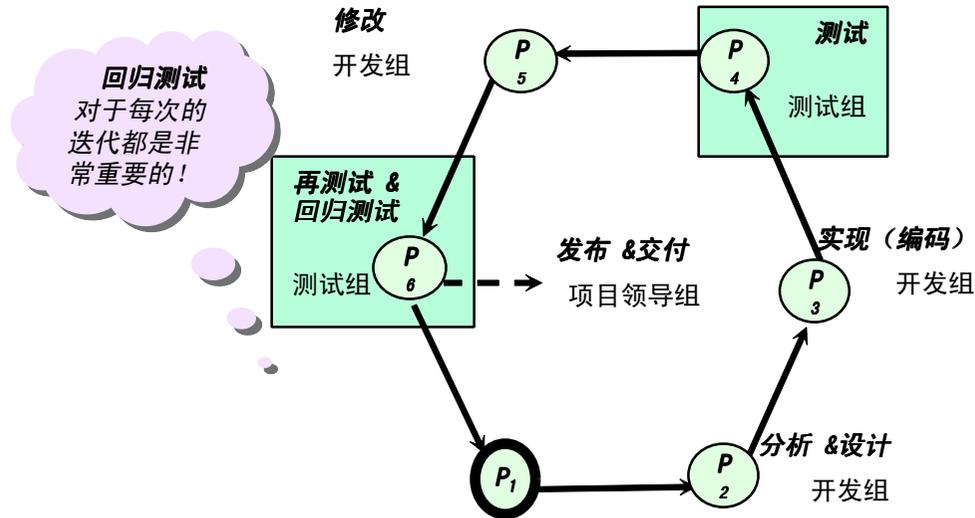
■ 测试团队的管理

■ 软件测试过程的管理

传统：瀑布模型



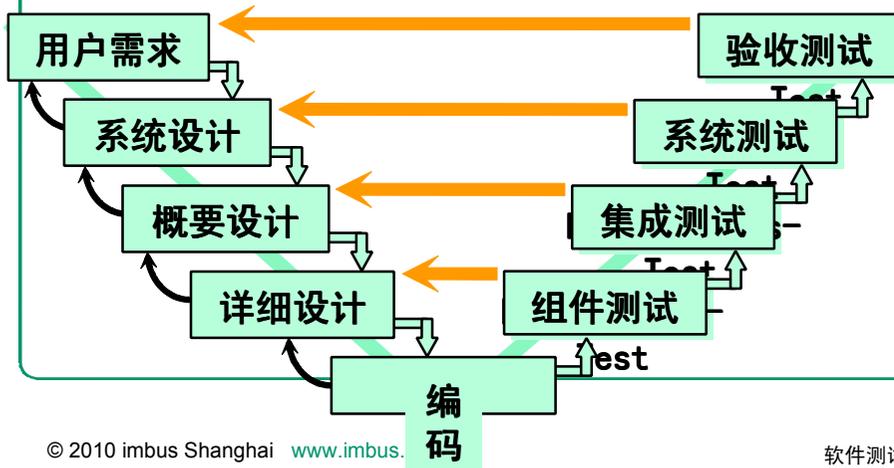
现代：迭代-增量开发模型



$P_i = \text{阶段/过程 } i$

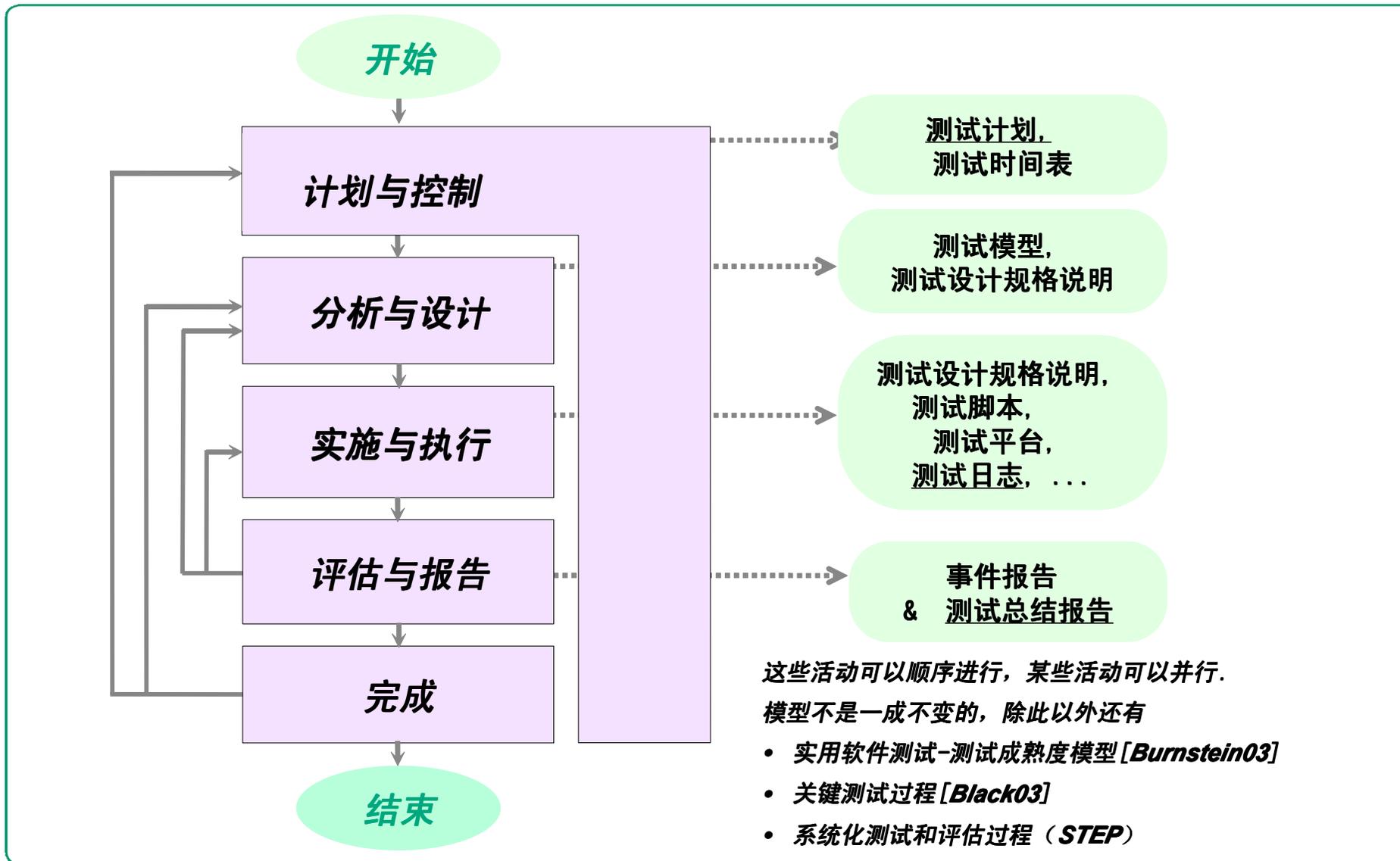
s.a: Balzent, Lehrbuch der SW-Technik, Bd. 2, Spektrum Akademischer Verlag, 1998, S.97ff.

经典：V-模型

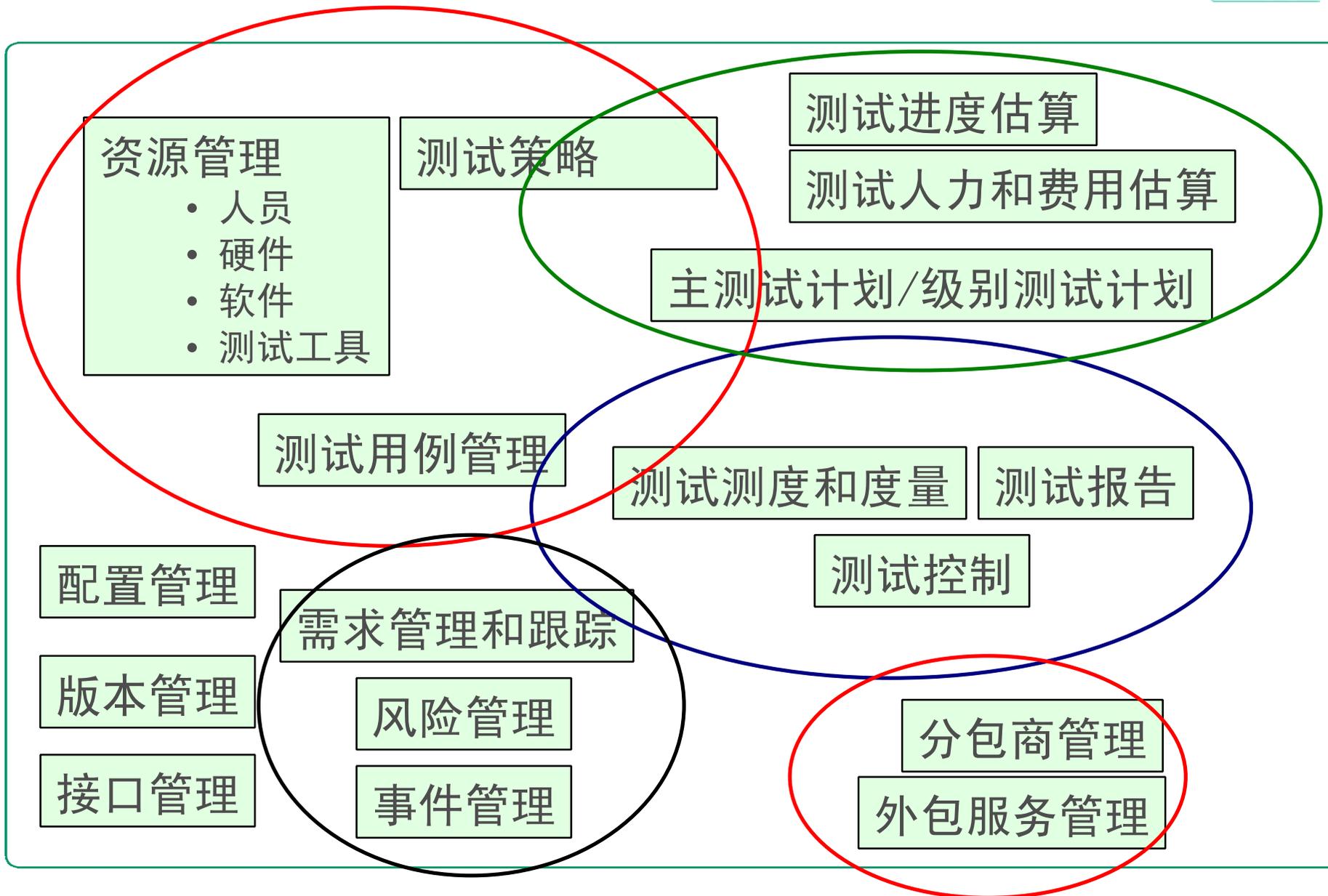


- 软件测试和软件开发一样是软件生命周期中的一个不可缺少的组成部分
- 不同的软件生命周期模型决定了不同的软件测试管理模式

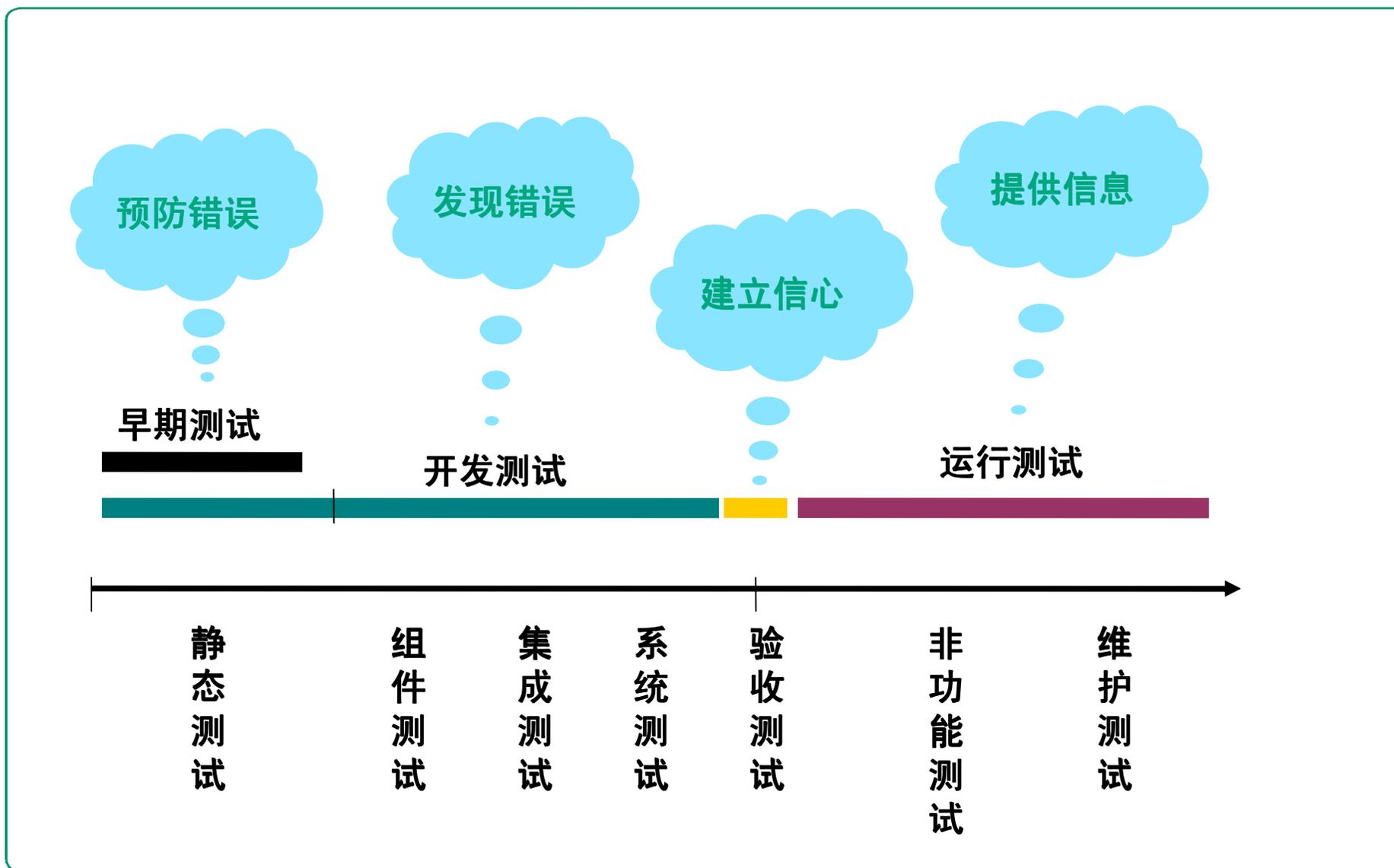
通用测试过程模型 (ISTQB)



软件测试管理的内容



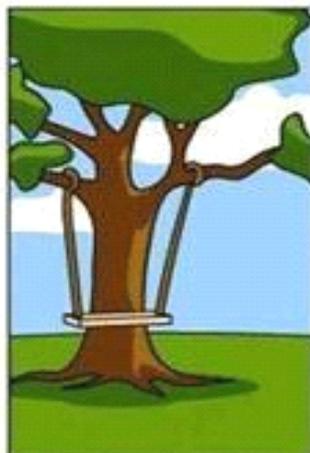
不同开发阶段软件测试的目的



从用户需求到实际产品



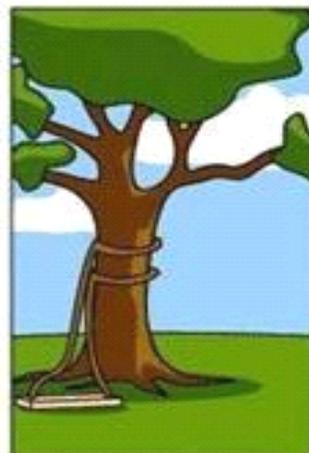
用户所描述的



项目组长理解的



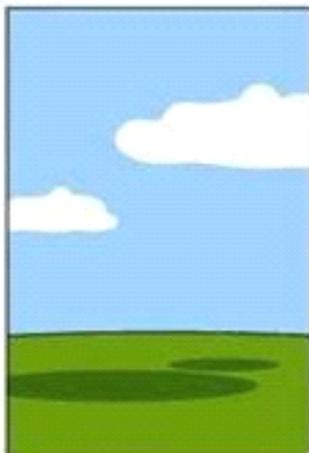
分析员的设计



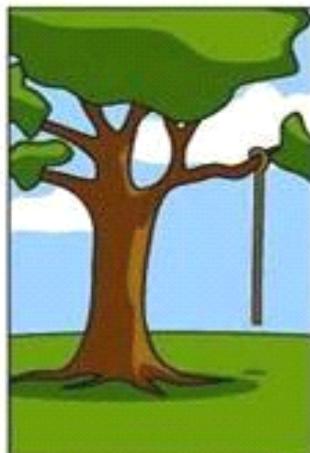
程序员编写的



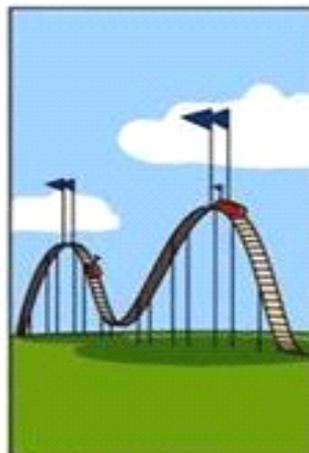
杏询员定义的



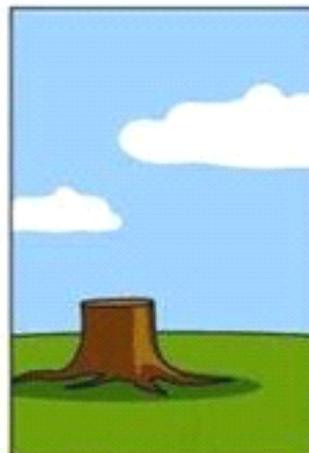
项目的文档



被安装的



在用户账单上的

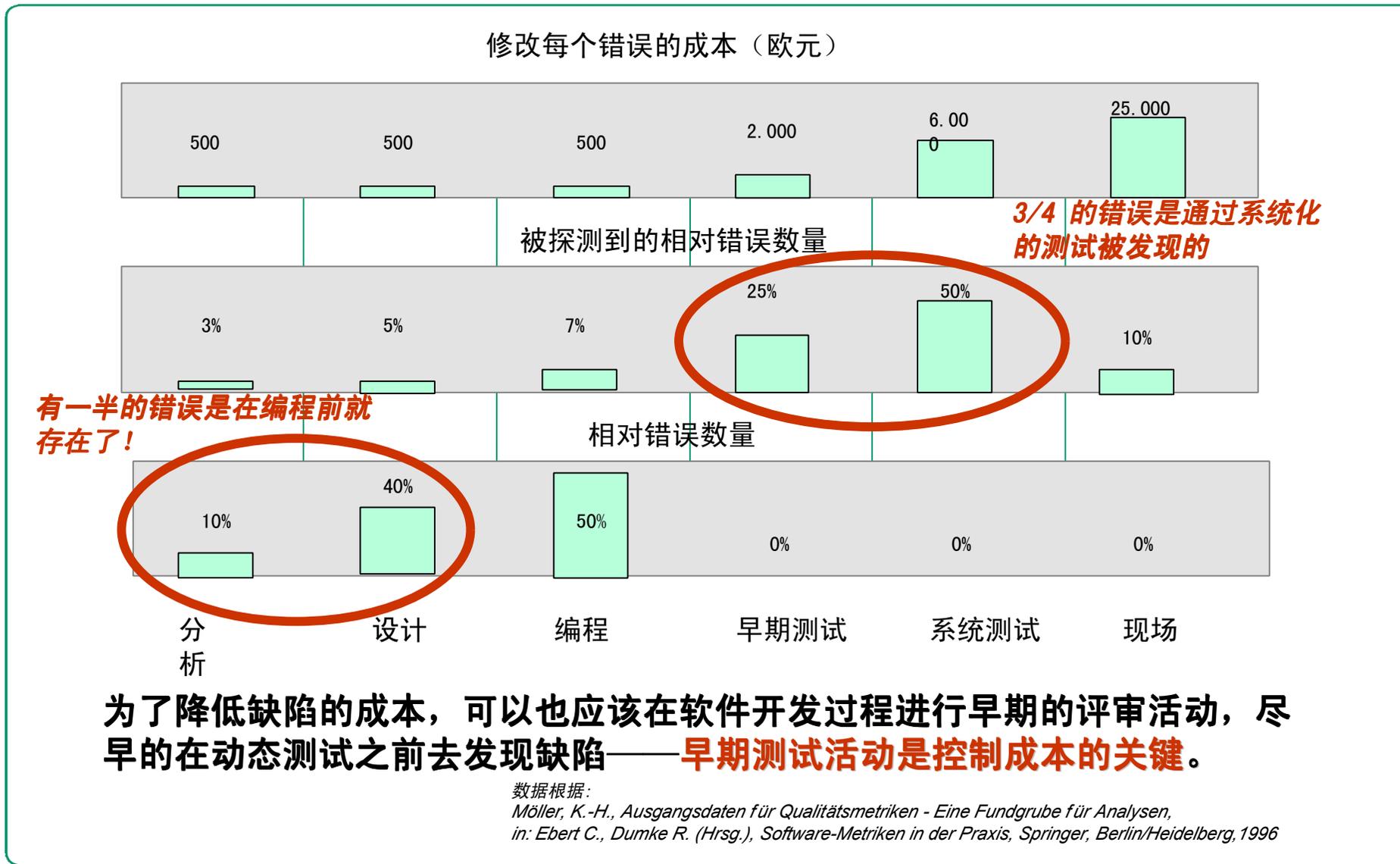


用户最后等到的



用户实际需要的

不同开发阶段错误的代价



外部失效成本

用户的成本:

- 软件产品的缺陷&错误 (数据丢失, 软件的错误反应, 硬件设备的损坏)

软件开发方的成本:

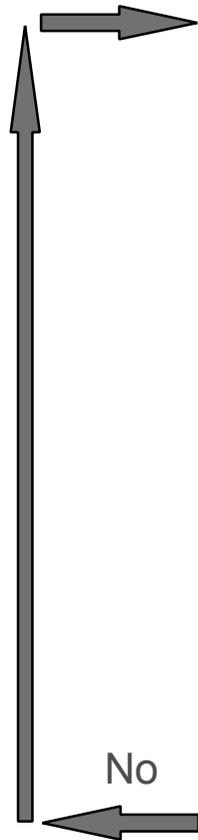
- 用户的不满意 (形象)
- 错误更改费用
- 对损害的赔偿费用

测试的成本

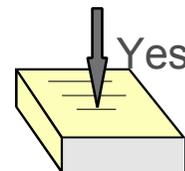
影响测试费用的因素有:

- 软件开发过程的成熟度
- 软件 (需求) 是否经常变更
- 软件测试过程成熟度
- 测试环境 (*test environment*) 的质量 & 可用性
- 软件的质量和软件的可测性
- 员工的素质
- 所遵循质量目标
- 是否按照测试策略

→ **测试的成本和内部修复的成本必须明显小于外部失效的成本!**



- 测试的目的是什么？
- 测试的主题是什么？
- 测试件是什么？有多少版本？版本交付测试的时间？哪些需要测试？哪些不属于测试范围？
- 采用什么测试技术？
- 需要测哪些测试级别？测试的颗粒度要多细？这样做是不是经济？
- 测试估算（工作量和时间表）
- 如果这样做，有哪些风险，是接受风险还是必须规避？
- 我有足够的资源吗？

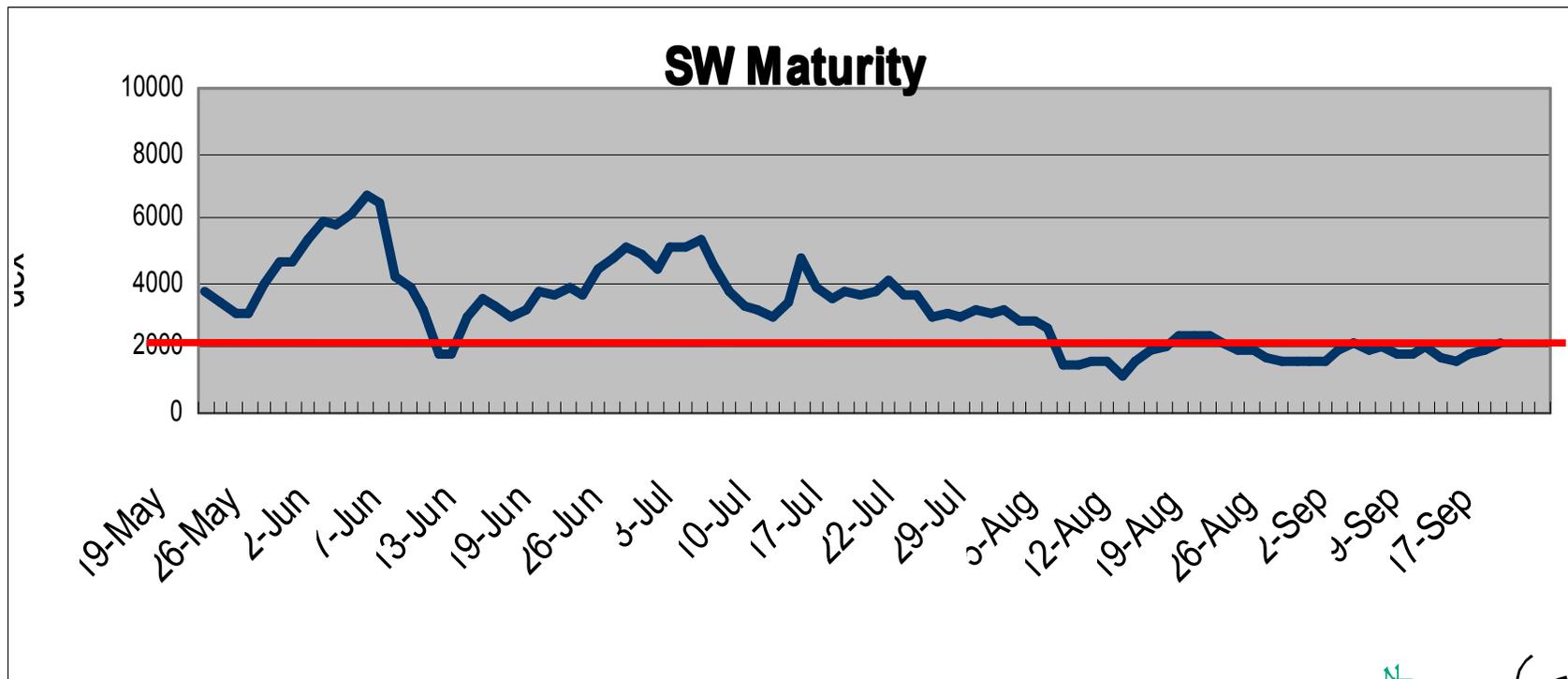


采用度量和度量元帮助管理者迅速理解所获得的信息，做出趋势判断和决策。

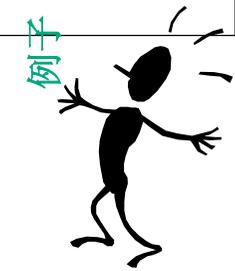
哪些数据需要度量？

- 已计划的进度、测试覆盖度及其随时间的演变
- 需求及其变更、以及对一系列进度、资源和任务等的影响
- 工作量和资源使用情况及其随时间的变化
- 里程碑和范围及其随时间的变化
- 完成任务所需的实际的和所计划成本
- 风险及减轻风险的措施，及其随时间的演变。
- 已发现的缺陷、已修正的缺陷及其修正缺陷所需的时间
- 根据风险和已发现的缺陷制定的软件成熟度模型

软件成熟度指数



— *SW maturity target*



根据测试结果，调整项目测试计划：

- 修定质量风险分析、测试优先级和/或测试计划
- 增减测试范围
- 增减资源或增减测试的工作量
- 推迟或提早发布时间
- 放松或加强测试退出准则

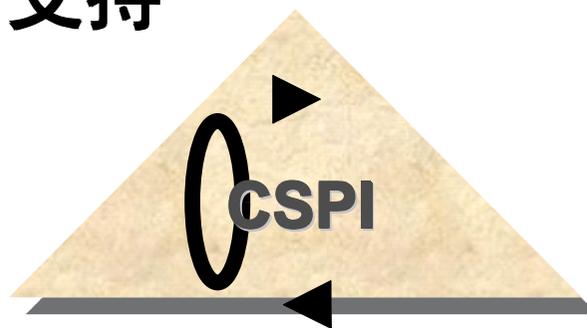
采用以上的方案，通常要求项目或业务的利益相关者认识一致，并且得到项目或业务经理的许可。

软件测试管理成功与否取决于

- 人员的有效管理
- 过程的有效管理
- 管理工具的支持

人员

管理层对测试的支持



过程

技术

持续软件过程改进 (Continuous SW Process Improvement)

