

# 测试过程管理

# 引言

---

- 课程的目标
  - 对项目管理的基本了解
  - 对测试过程的了解
  - 测试的标准流程和测试项目组织

# 项目管理

---

- 项目的概念：
  - 项目是为创造独特的产品、服务或成果而进行的临时性工作
- 特性
  - 具备明确的目标和任务
  - 具备明确的范围
  - 具备明确时效
  - 需要

## 项目的成果

---

- 项目可以创造：
  - 一种产品，既可以是其他产品的组成部分，也可以本身就是终端产品；
  - 一种能力（如支持生产或配送的业务职能），能用来提供某种服务；
  - 一种成果，例如结果或文件（如某研究项目所产生的知识，可据此判断某种趋势是否存在，或某个新过程是否有益于社会）。
- 项目的例子包括（但不限于）：
  - 开发一种新产品或服务；
  - 改变一个组织的结构、人员配备或风格；
  - 开发或购买一套新的或改良后的信息系统；
  - 建造一幢大楼或一项基础设施；
  - 实施一套新的业务流程或程序。

# 项目阶段过程

---

- 启动
- 规划
- 执行
- 监控
- 收尾

# 管理一个项目

---

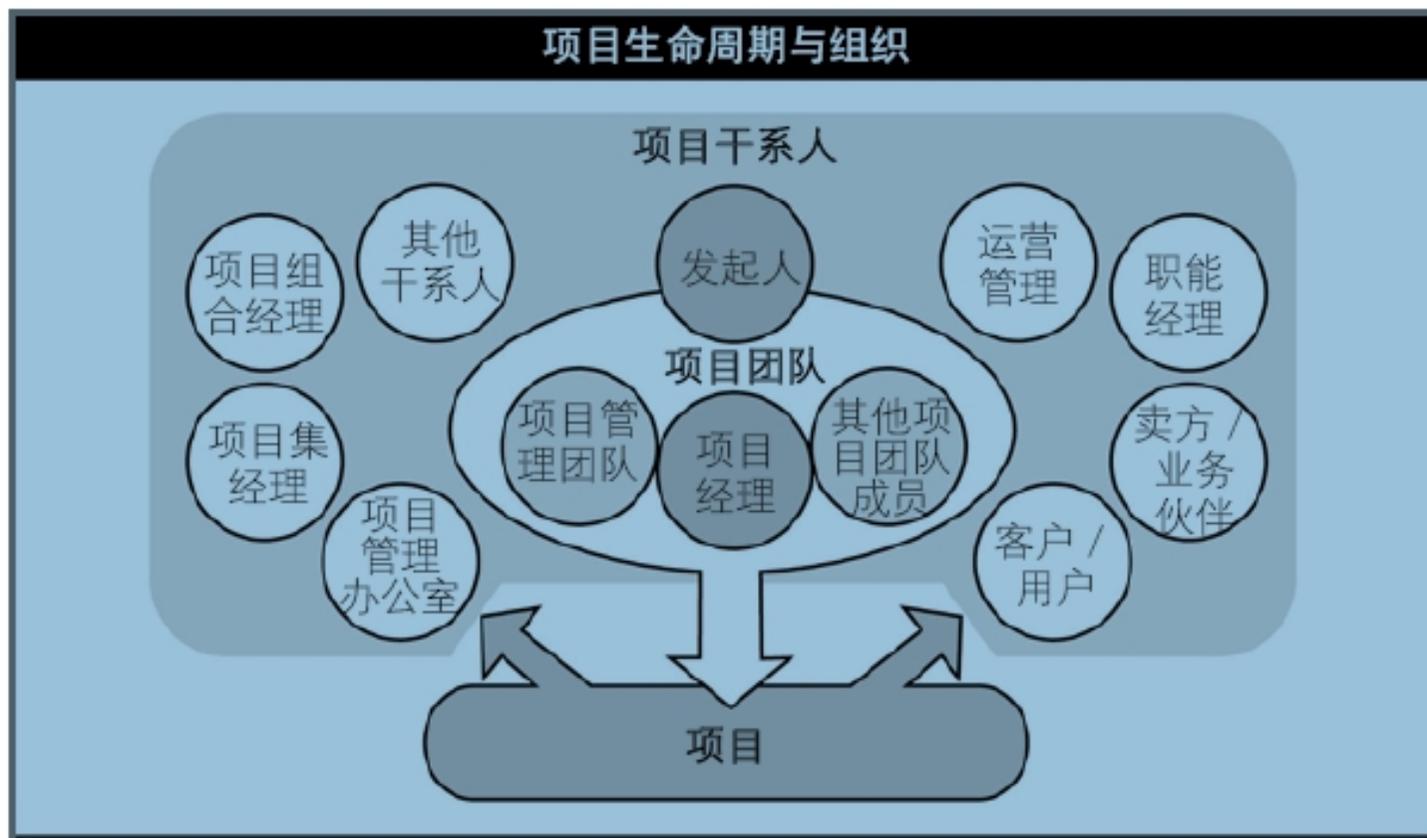
- 管理一个项目通常要：
  - 识别需求；
  - 在规划和执行项目时，处理干系人的各种需要、关注和期望；
  - 平衡相互竞争的项目制约因素，包括（但不限于）：
    - ○ 范围
    - ○ 质量
    - ○ 进度
    - ○ 预算
    - ○ 资源
    - ○ 风险

# 项目生命周期与组织

---

- 2.1 项目生命周期——概述
- 2.2 项目与运营
- 2.3 项目干系人
- 2.4 组织对项目的影响

# 项目干系人



# 组织结构对项目的影晌

项目特征 \ 组织结构	组织结构		矩阵型			项目型
	职能型		弱矩阵	平衡矩阵	强矩阵	
项目经理的职权	很少或没有		有限	小到中	中到大	大到几乎全权
可用的资源	很少或没有		有限	少到中	中到多	多到几乎全部
项目预算控制者	职能经理		职能经理	职能经理与项目经理	项目经理	项目经理
项目经理的角色	兼职		兼职	全职	全职	全职
项目管理行政人员	兼职		兼职	兼职	全职	全职

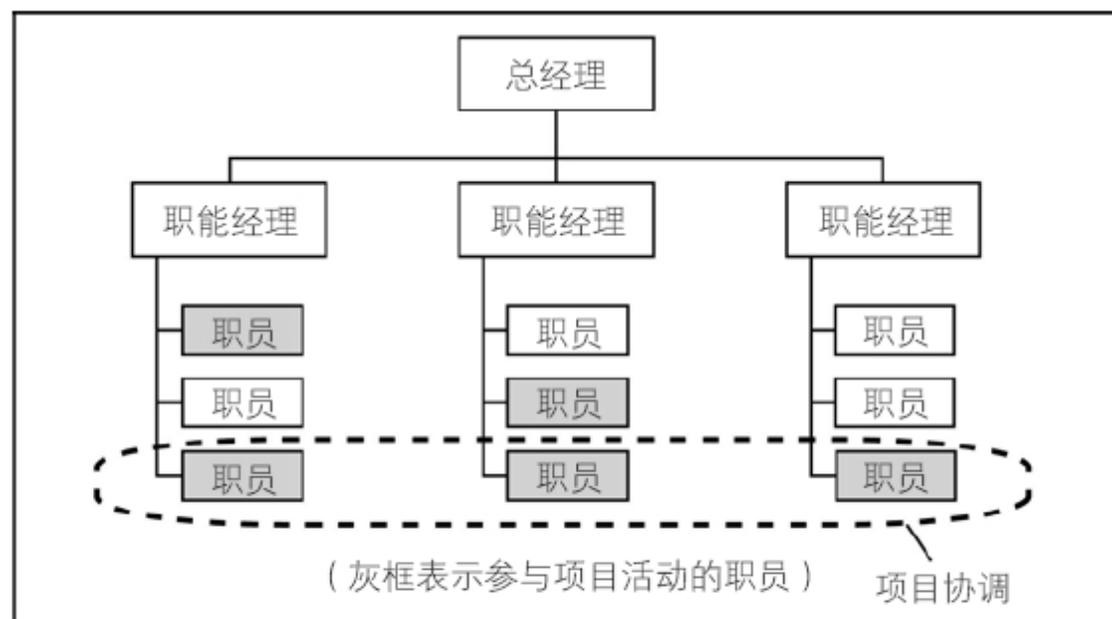
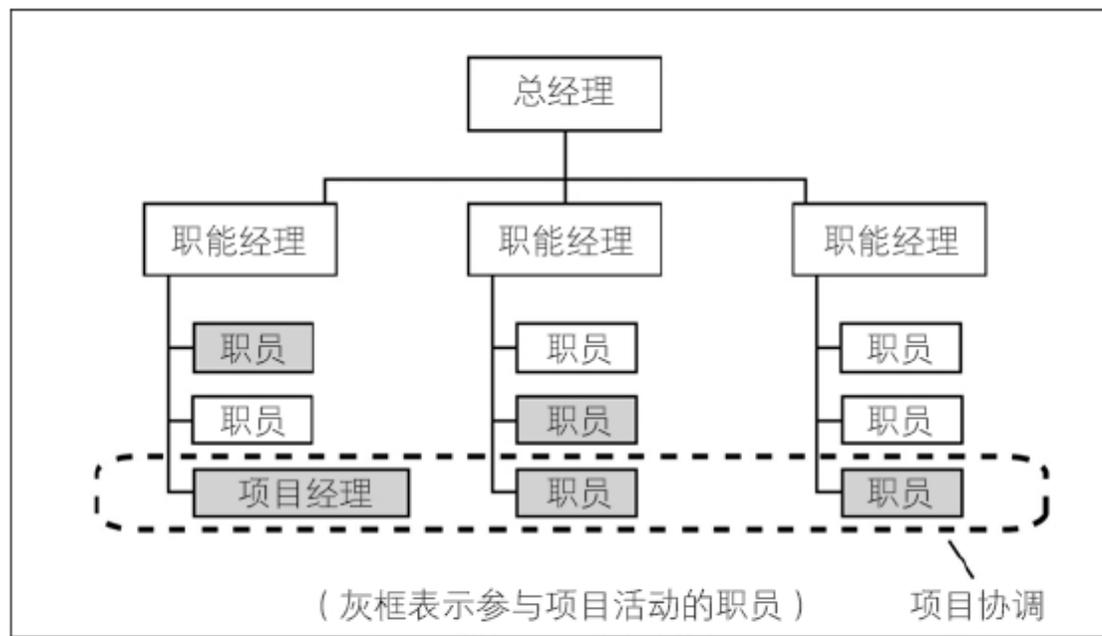


图 2-8 弱矩阵型组织



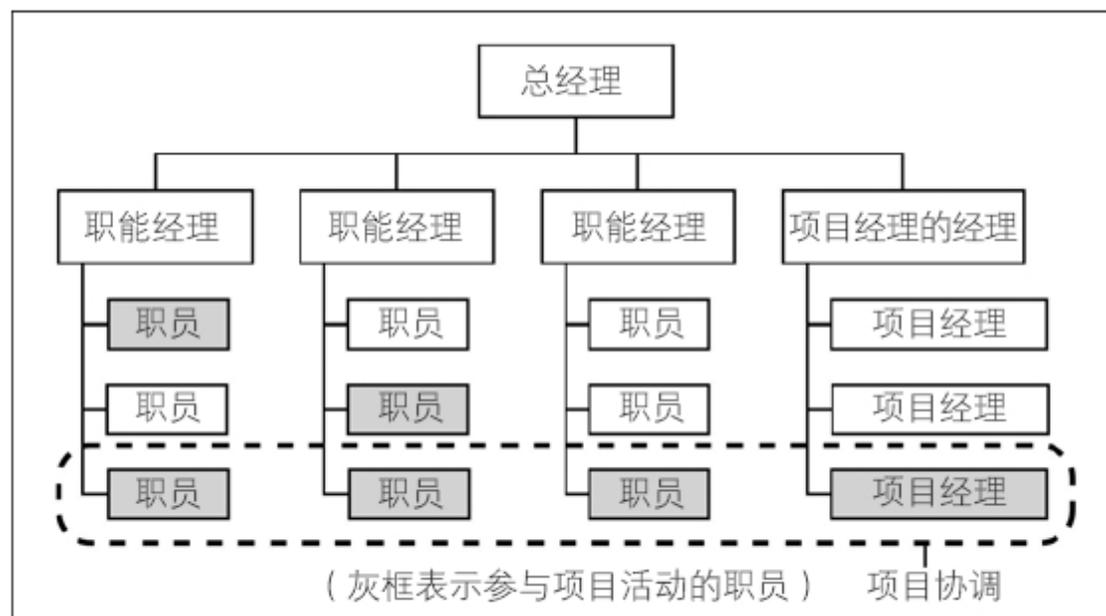


图 2-10 强矩阵型组织

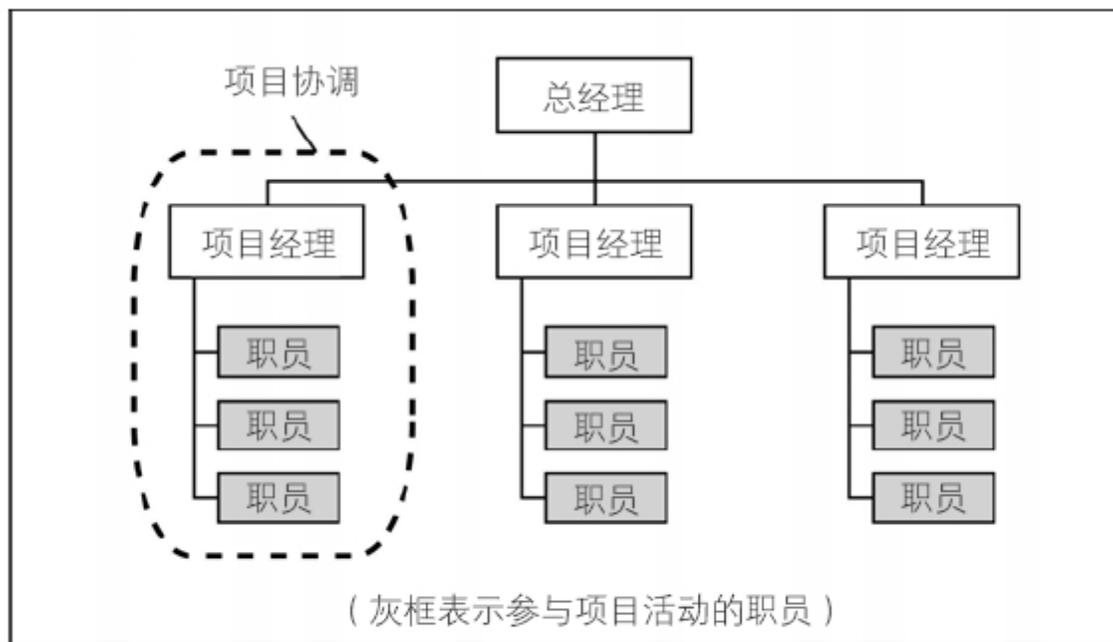
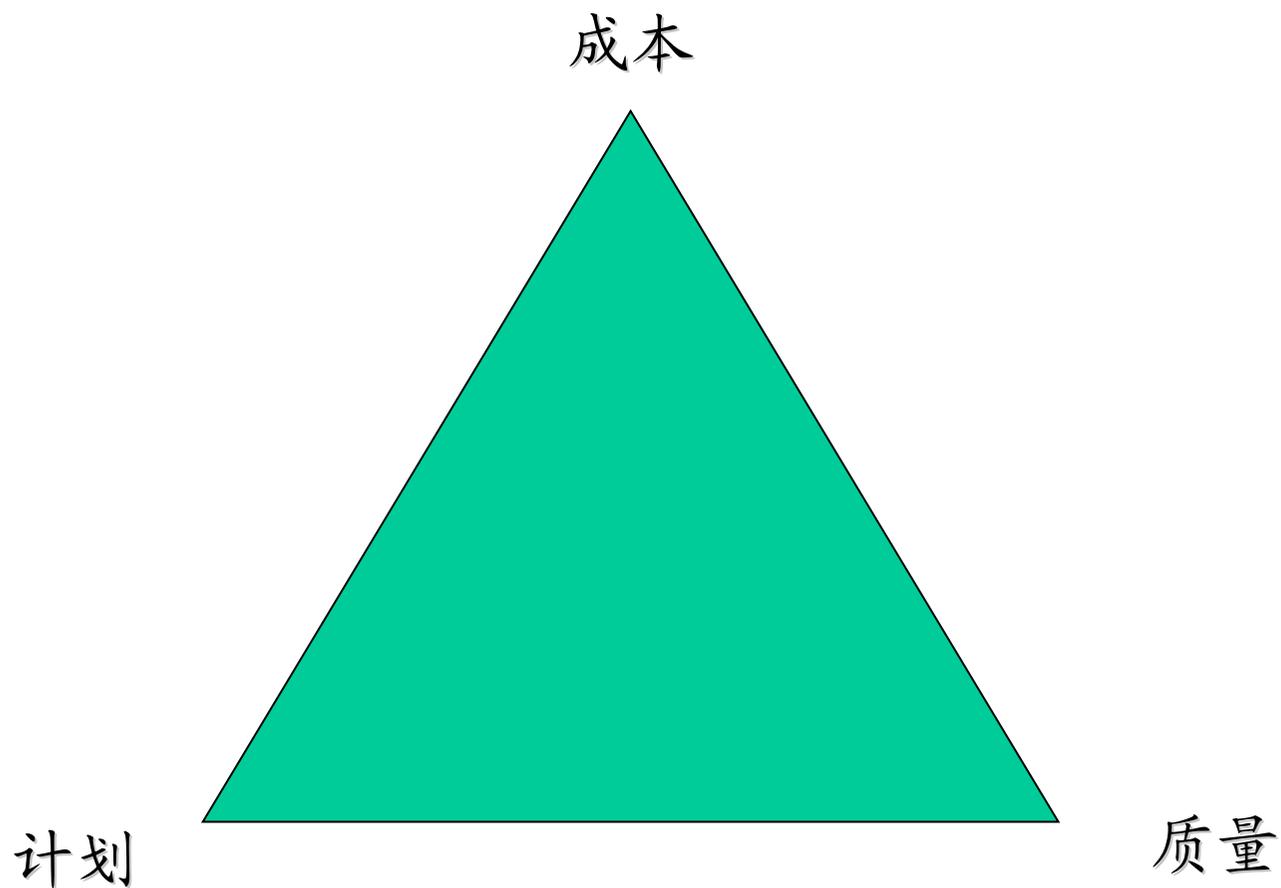


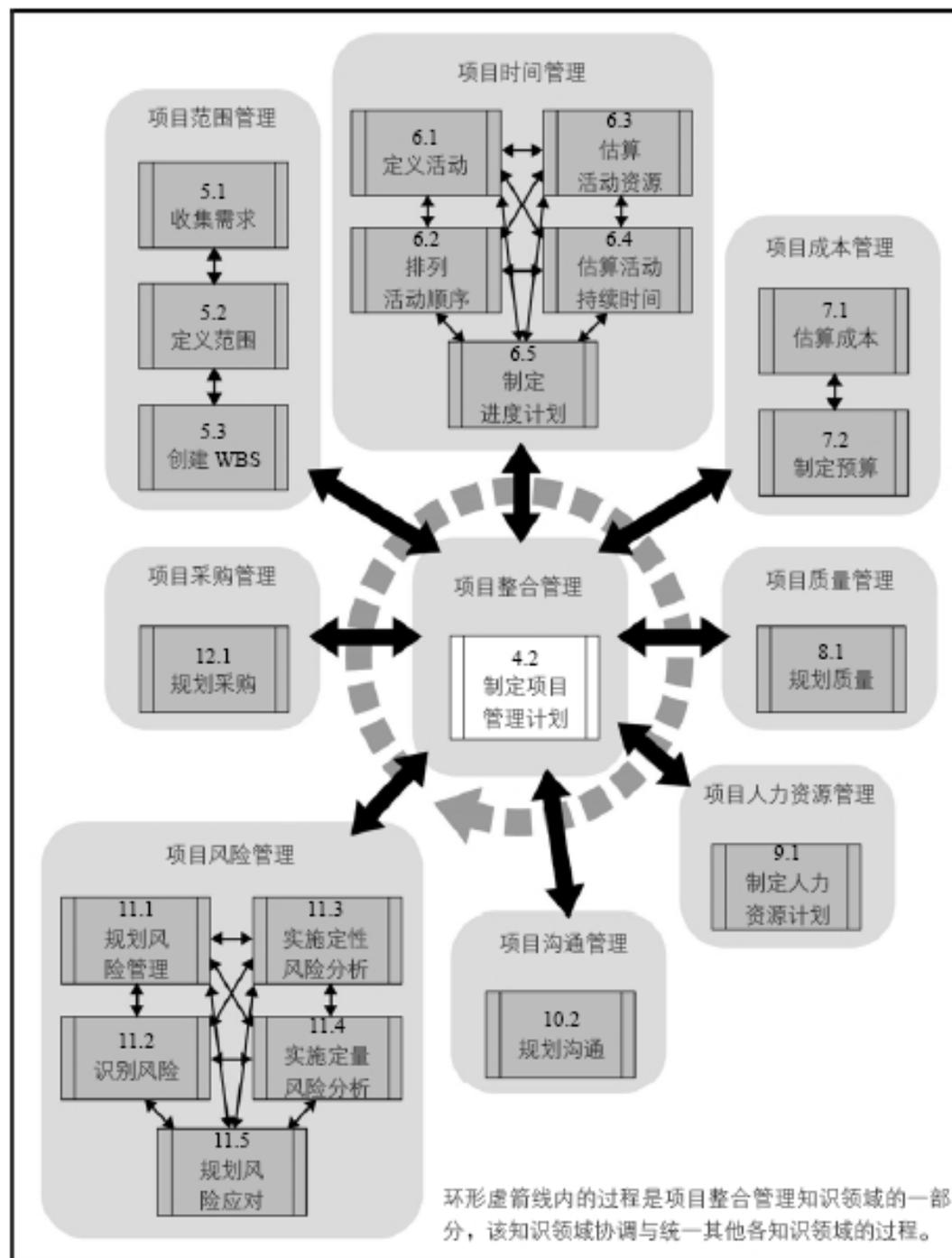
图 2-11 项目型组织

# 项目三要素

---



- 项目规划



# 测试过程管理

---

- 测试工作的目的
- 测试过程规划和实施

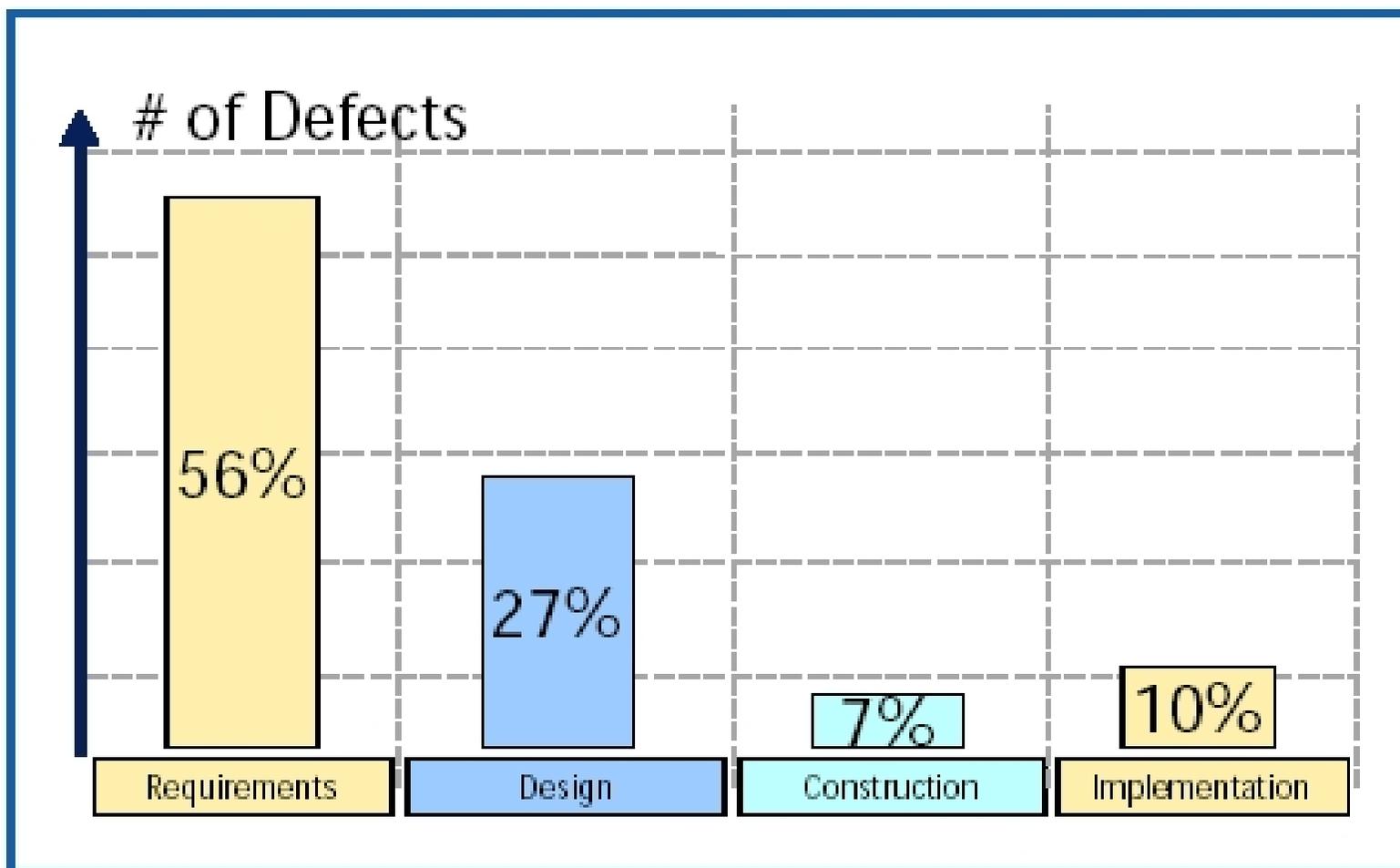
## 测试工作的目的

---

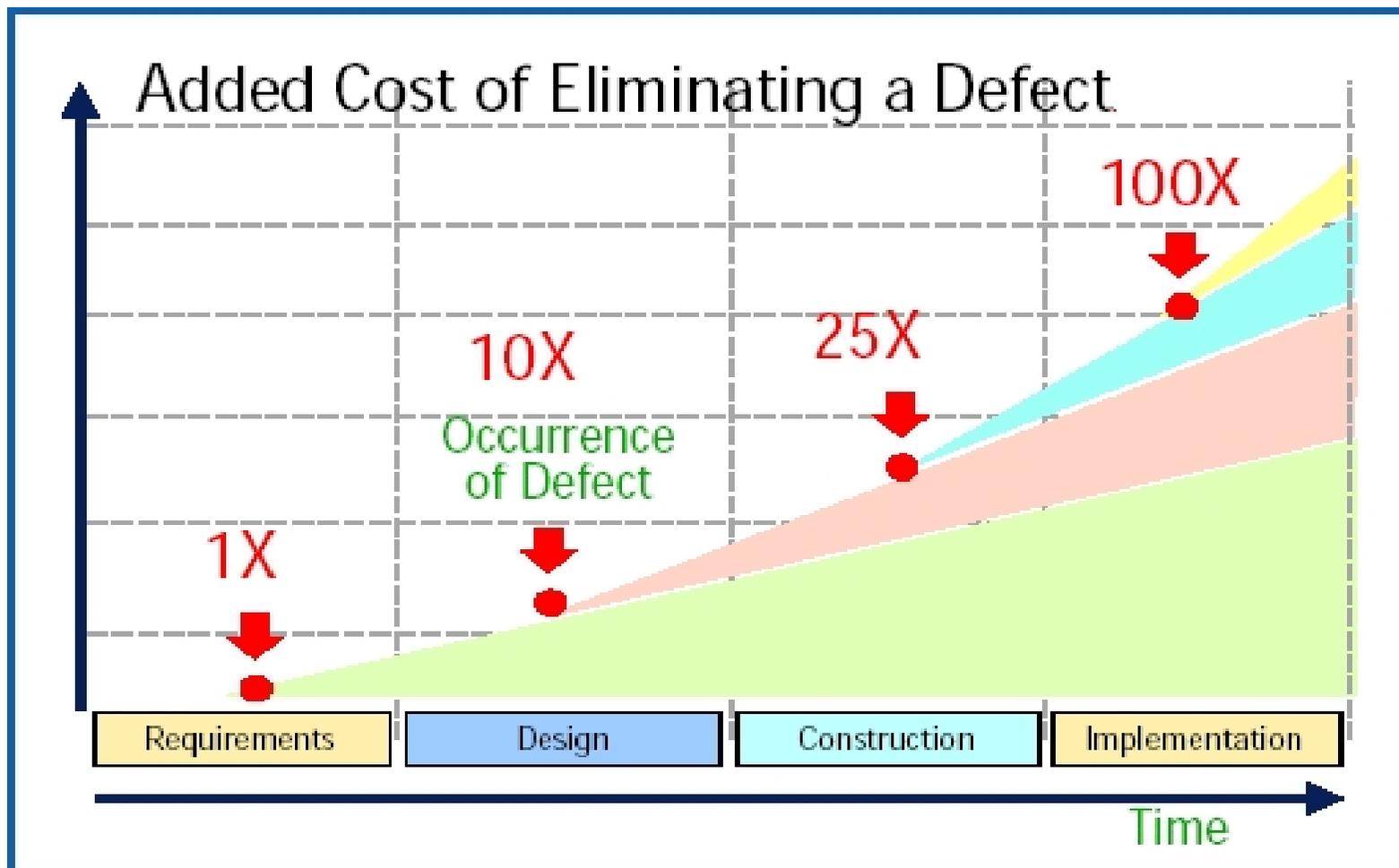
在测试技术兴起的早期，测试的目的主要有两个：证明系统可用，满足了需求；发现系统中的错误。

随着软件技术的成熟，人们发现大量错误的根源在于需求和设计，因此现代测试的目的在于：证明系统可用，满足了需求；发现系统中的错误；避免错误的发生。

# 缺陷产生的原因



# 测试工作的目的



# 区别测试与调试

---

- 调试与测试
  - 调试(debug)
    - 调试是定位缺陷发现的原因,然后修改软件更正缺陷的过程
  - 测试
    - 以预先定义的标准为依据发现软件缺陷的过程.

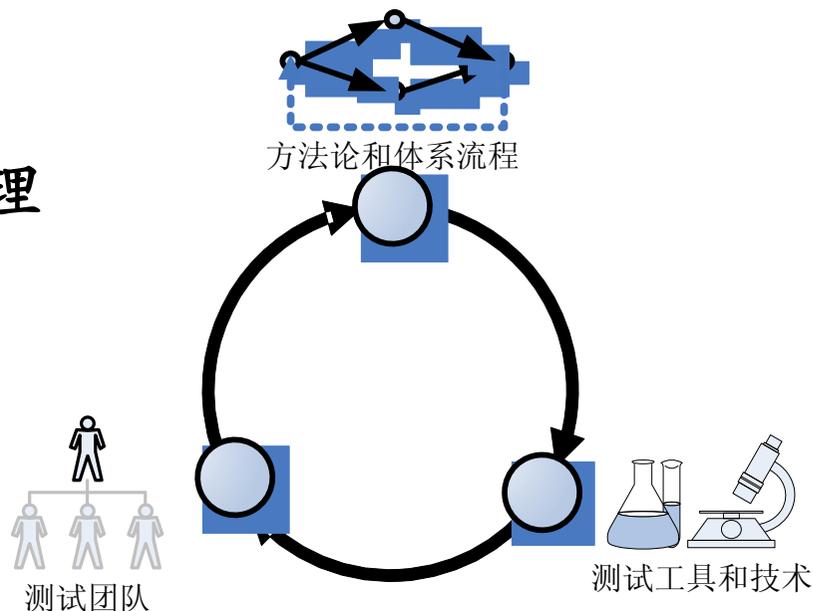
# 测试过程管理

---

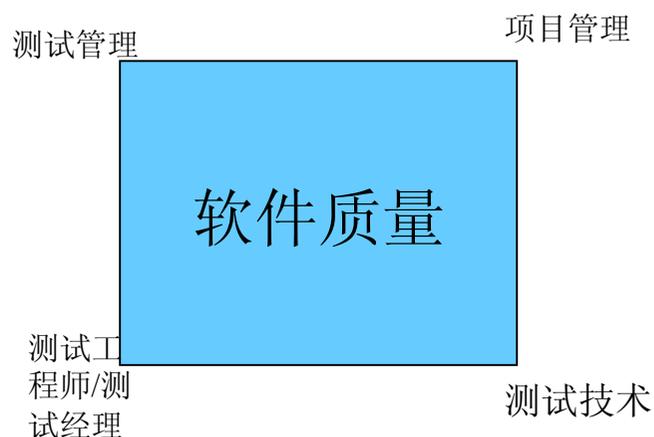
- 测试工作的目的
- 测试过程规划和实施

# 测试工具方法论

- 成功测试的要素：
  - 方法论、体系流程
  - 测试团队：
    - 具有技能的测试经理
    - 自动测试工程师
  - 测试工具和技术



# 支持测试工作的要素



- 良好的项目管理支持测试的实施
- 有效的测试管理使测试活动集成在项目活动中
- 专业的测试工程师和测试经理实施测试
- 应用有效的测试技术实施测试

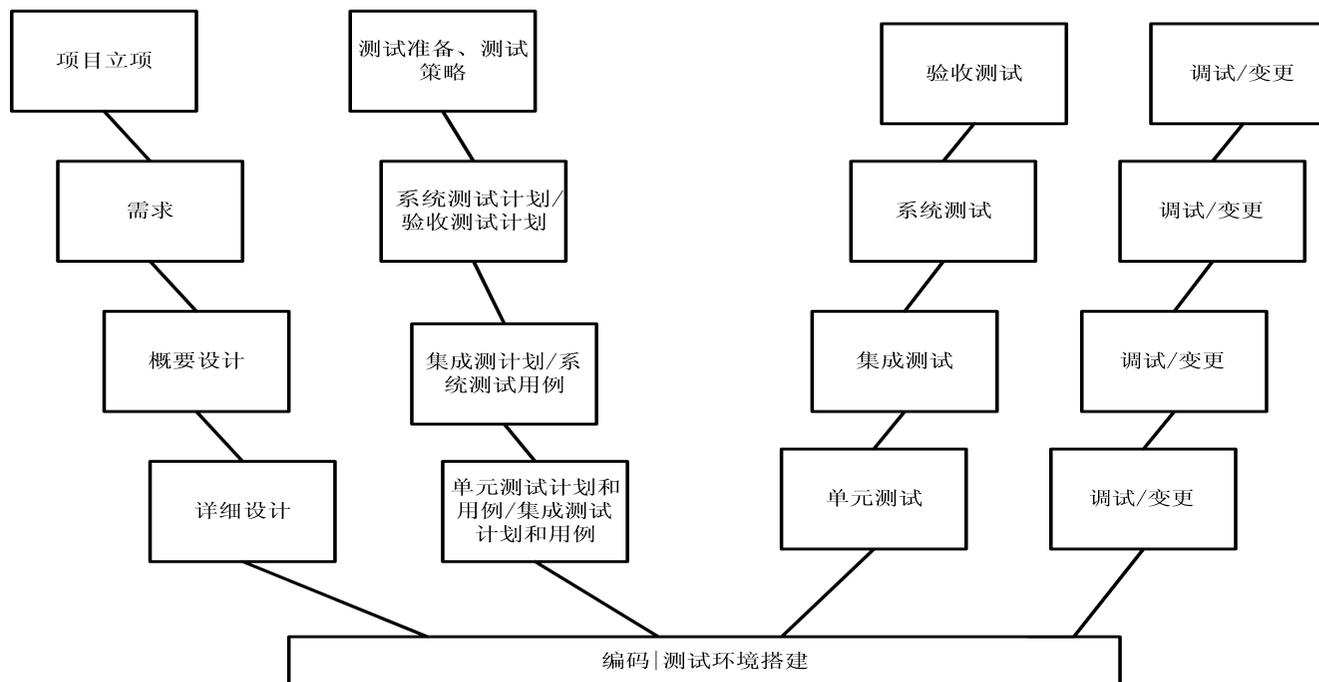
# 测试过程设计

- 项目过程流图



# 测试过程模型

- 双“V”模型（“W”模型）的项目运做模式



## 测试过程模型

---

- 双“V”模型可以使测试工作能够尽可能早的开展，利于项目的进度控制。
- 测试组能够尽早了解系统特性，使测试工作能够尽快深入开展。
- 适应可能发生的频繁变更问题。

# 测试计划和实施的原则(一)

---

- 穷尽测试是不可能的
  - 无论你做多少测试，都不能百分之百确保系统中没有问题。
- 测试工作具有创造性和挑战性
  - 你必须理解系统要实现什么。
  - 你必须能分析出系统的极限和约束。
  - 你必须对系统的应用领域有深刻的理解。
- 测试是有风险的
  - 不能明确所有的风险
  - 风险估计对测试的重点和程度有重要影响。
- 测试分析、计划和设计是非常重要的
  - 测试的目的必须明确并在项目中达成共识。测试必须经过系统的规划和设计，否则将陷入混乱。

## 测试计划和实施的原则(二)

---

- 测试者的工作态度非常重要
  - 如果你工作不认真严谨，你的工作将不会很有效。
  - 你必须有愿望要发现问题，并努力破坏系统。
- 时间和资源非常重要
  - 如果没有足够的时间和资源，测试无法完成。
- 测试准备和时间表的重要性
  - 早期的测试准备有助于对系统的理解。例如技能培训。
  - 测试开始的越早，问题发现的越早，给项目带来的益处就越大。
- 必须度量和跟踪测试覆盖度
  - 你必须知道有哪些需求、设计和代码测试没有覆盖到。
  - 复杂的系统如果不进行严格的测试度量，很难保证全面覆盖到。

# 如何为项目规划测试

---

测试作为一个子项目来规划。

# 如何规划测试

---

- 测试作为一个子项目来规划：
  - 测试对象？
  - 测试工作范围？
  - 限制？
  - 测试工作的策略

 测试计划

# 如何规划测试-测试对象

- 被测对象以此为基础

测试需求?

质量指



# 如何规划测试-测试工作范围

---

- 确定测试工作范围：明确测试需要的资源；测试进度安排；技术和工具。
- 单元测试？
- 集成测试？
  - 哪种方式的集成？子系统、子系统集成
- 系统测试？
  - 功能验证
  - 性能（容量、稳定性、容错性）
  - 安全性
  - 易用性

## 如何项目测试-限制

---

- 项目进度限制
- 项目的成本限制
- 资源的限制

项目中存在的各种限制会影响对项目测试工作范围的定义。

# 如何规划测试-制定测试策略

---

- 技术方法和工具
- 测试资源的估计和获取（人力、设备、软件工具）
- 测试进度安排
- 测试人员分工
- 测试启动准则
- 测试结束准则
- 与其他组的协同

## 测试策略-技术方法和工具

---

- 测试环境：模拟环境？生产环境？
- 测试方法：手工测试？自动测试？
- 测试程序：自行开发？成品工具？调试的程序？
- 测试例设计方法：功能点？等价类？边界？正交矩阵？

# 如何规划测试-测试资源和获取

---

- 测试资源：
  - 人力
  - 设备
  - 软件工具
- 资源获取方式：
  - 人力：项目工程部？开发部？外包？
  - 设备和软件工具：购买？租用？自行开发？

## 测试策略-测试进度安排

---

- 测试工作阶段的划分:

- 计划

- 设计

- 实现

- 执行

- 报告分析

- 如何保证测试计划、设计、实现与开发任务的并行?

# 测试策略-测试人员的分工

---

- 测试任务分类
  - 测试管理
  - 测试设计
  - 测试实现
  - 测试执行
  - 测试报告

# 测试策略-测试的启动准则

---

- 何时可以开始测试？
  - 在测试计划阶段就必须明确，而且是在整个项目组达成一致。
  - 可参考的准则：
    - 完成单元测试，单元测试符合结束准则
    - 建立代码基线
    - 发布了测试版本
    - 通过版本基本验证项

## 测试策略-测试结束准则

---

- 项目有预先定义的结束时间，因此测试工作也不可能无休止的进行。
- 可参考的测试结束准则：
  - 所有需求都进行了验证
  - 通过所有功能验证
  - 无致命和严重的问题

# 测试策略-与其他组的协同

---

- 与开发组的协同
- 与配置组的协同
- 与质量经理的协同
- 与项目各级管理者的协同



# 组间协同-与开发组



- 测试组与开发组共同确认测试的范围
- 开发组为测试方案提供建议
- 测试组借鉴开发组的程序开发测试程序
- 测试组与开发组共同确认测试开始和启动准则
- 测试组向开发组通报测试结果和分析
- 测试组和开发组之前的版本传递经配置组完成。

## 组间协同-配置组

---



- 所有测试记录提交配置组管理
- 所有测试版本由配置组负责合成和发布

## 组间协同-质量经理

---



- 与质量经理一起制定测试流程和规范
- 质量经理审计测试的进行状况

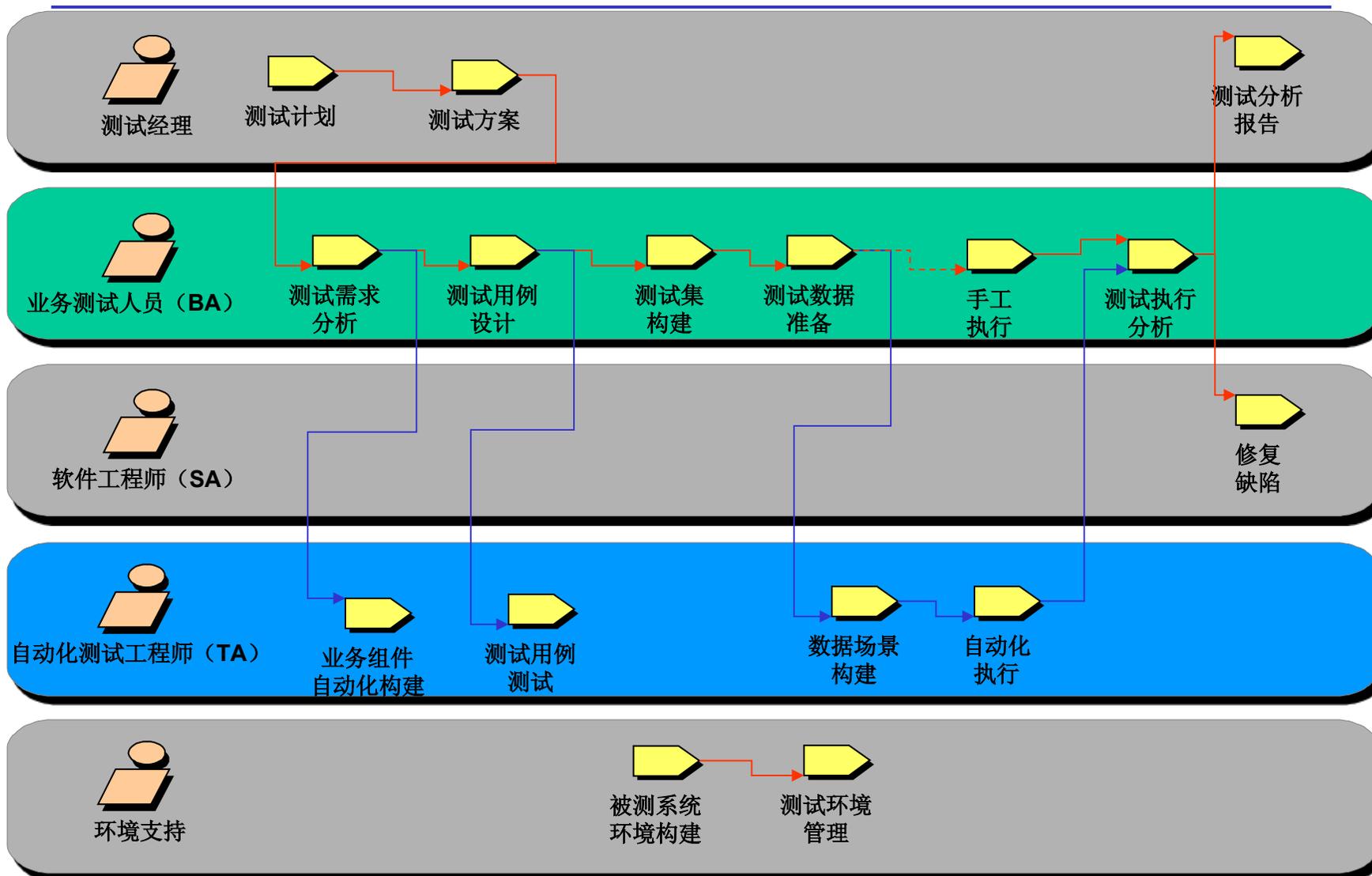
## 组间协同-各级管理者



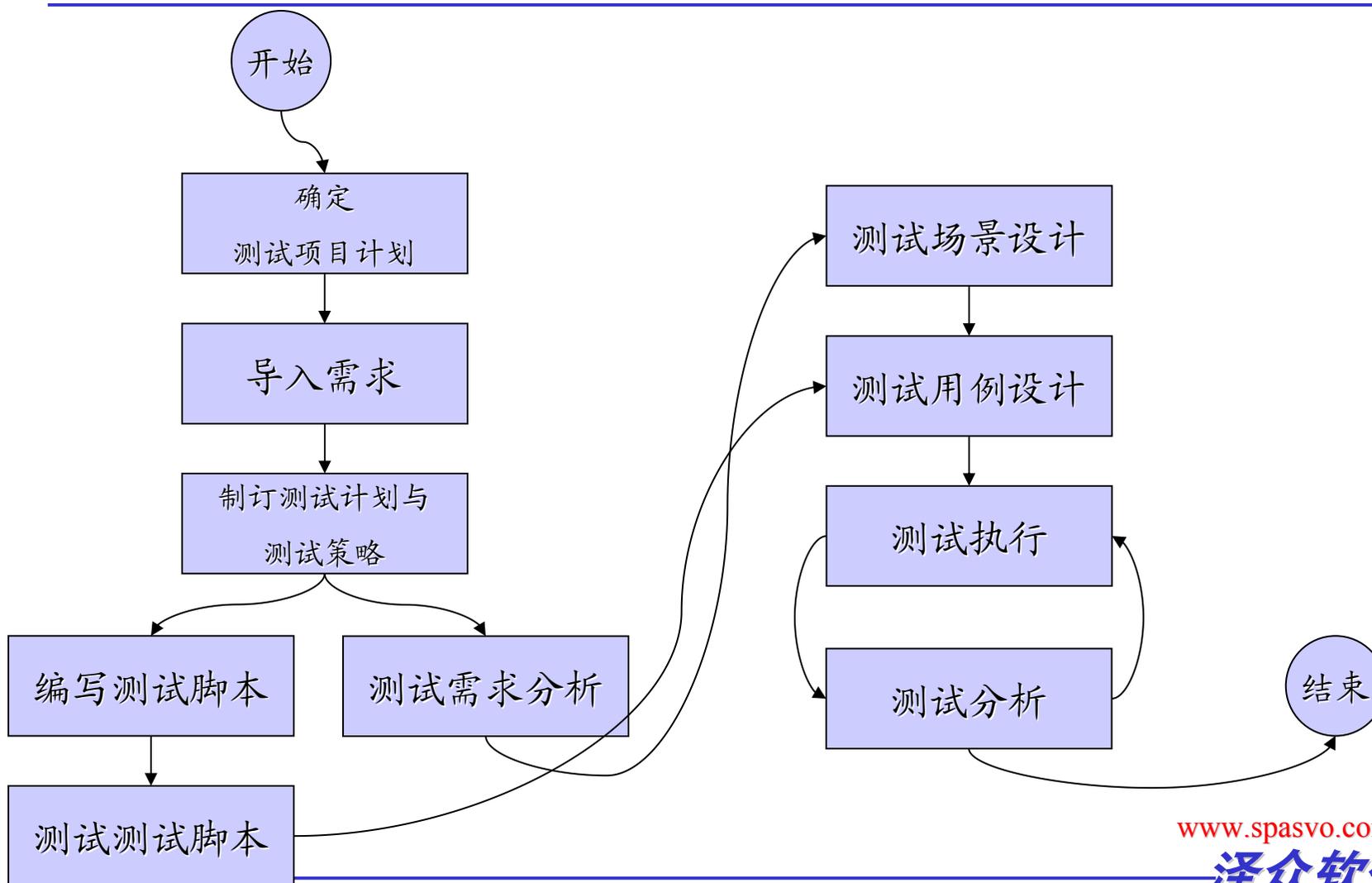
- 测试组与各级管理者共同确认测试的范围
- 各级管理者为测试方案提供建议
- 测试组与各级管理者共同确认测试开始和启动准则
- 测试组向各级管理者通报测试结果和分析
- 各级管理者监控测试执行进度和结果

# 测试阶段

# 管理测试过程



# 自动化测试过程

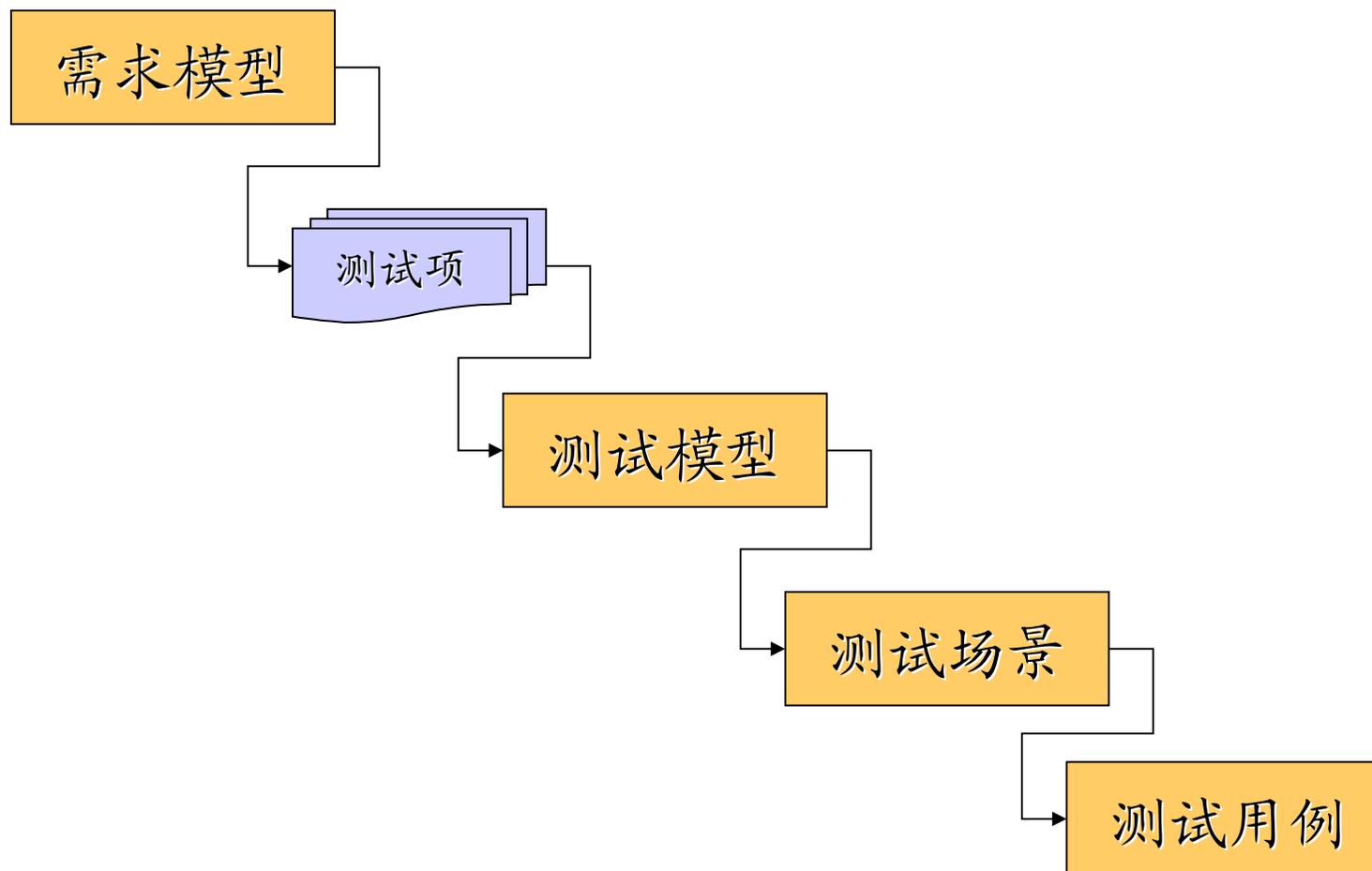


# 创建和管理测试需求

- 测试需求导入
  - Word/excel导入
- 测试需求的要素
  - 需求描述
  - 需求的UML
  - 测试项：需要测试的项目
- 需求的架构：
  - 需求树
- 测试需求覆盖率是测试的关键指标
- 不能用用例通过率代替需求覆盖率



# 设计测试用例



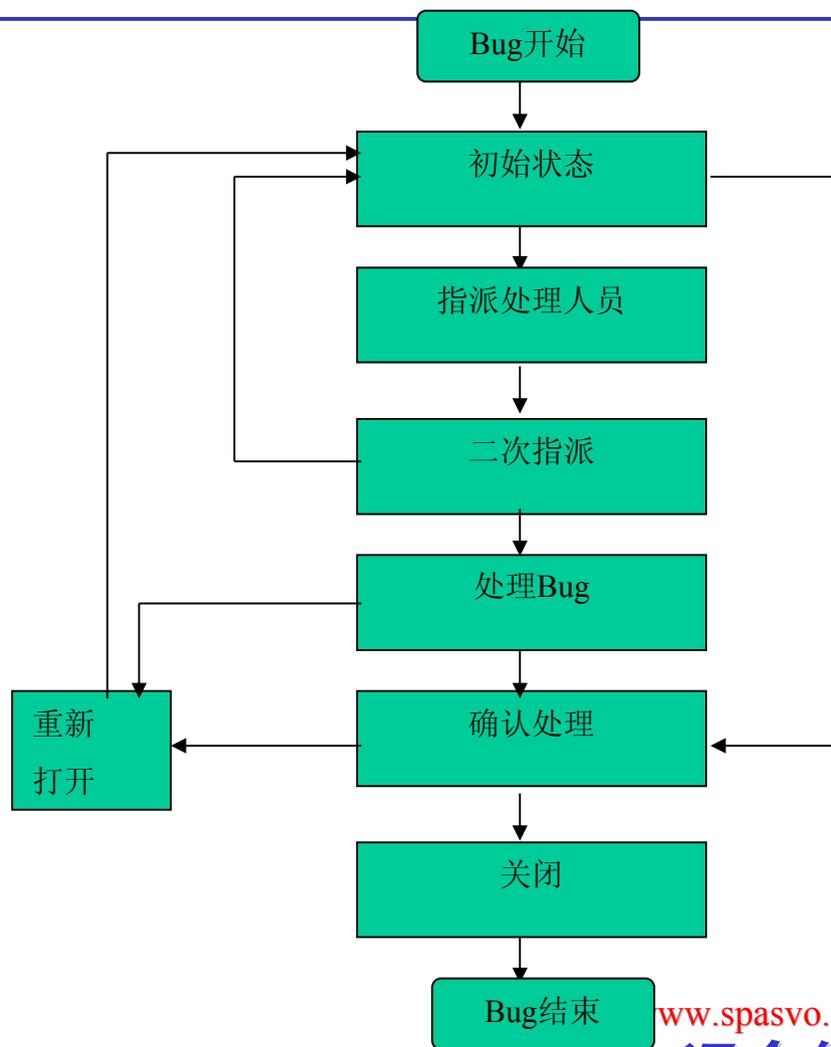
# 测试报告与分析

- 统计
- 覆盖率分析
- 自定义测试报告



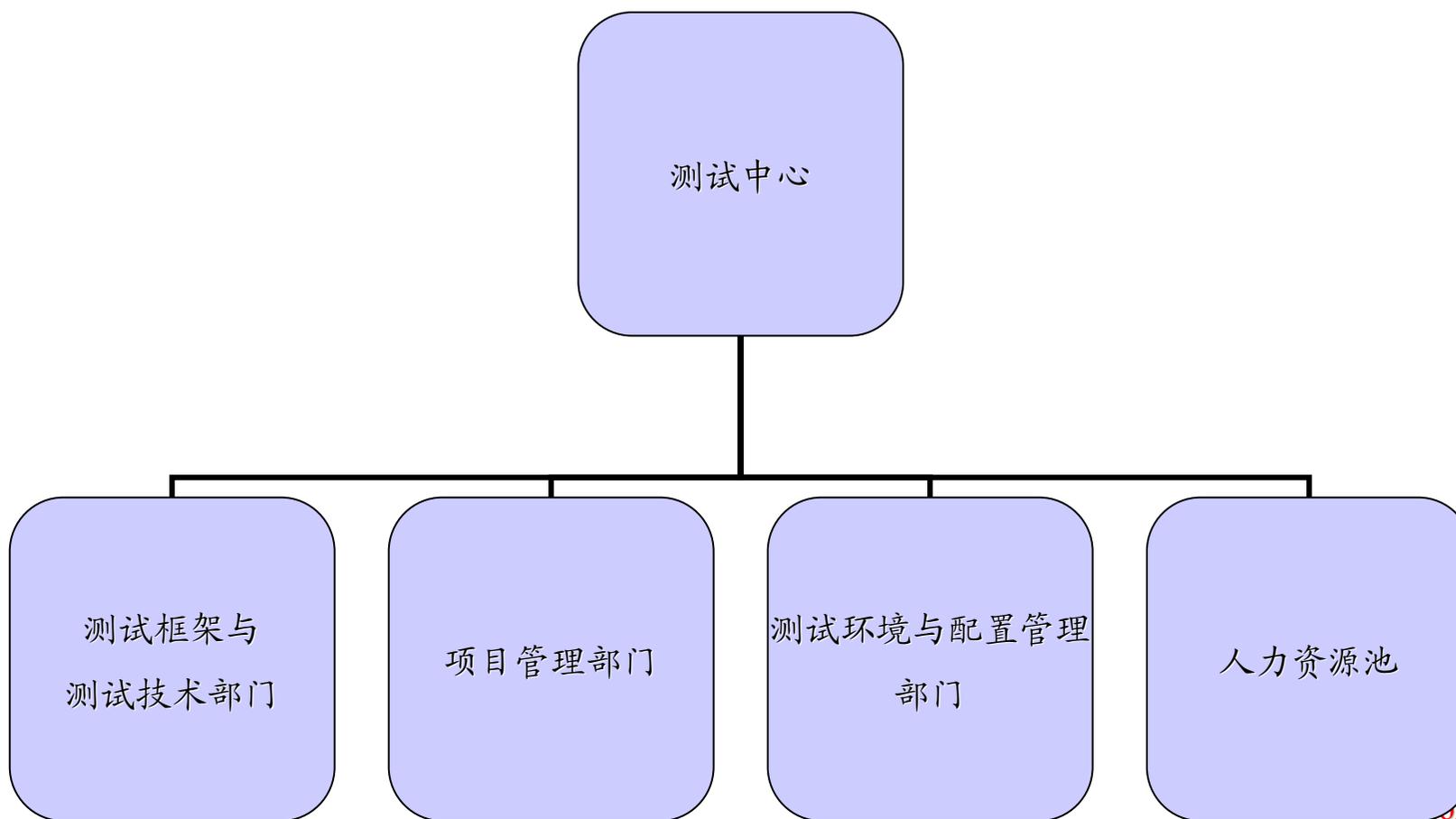
# 缺陷管理流程

- 自定义过滤器
- 与测试用例关联的缺陷
- 与需求关联的缺陷管理

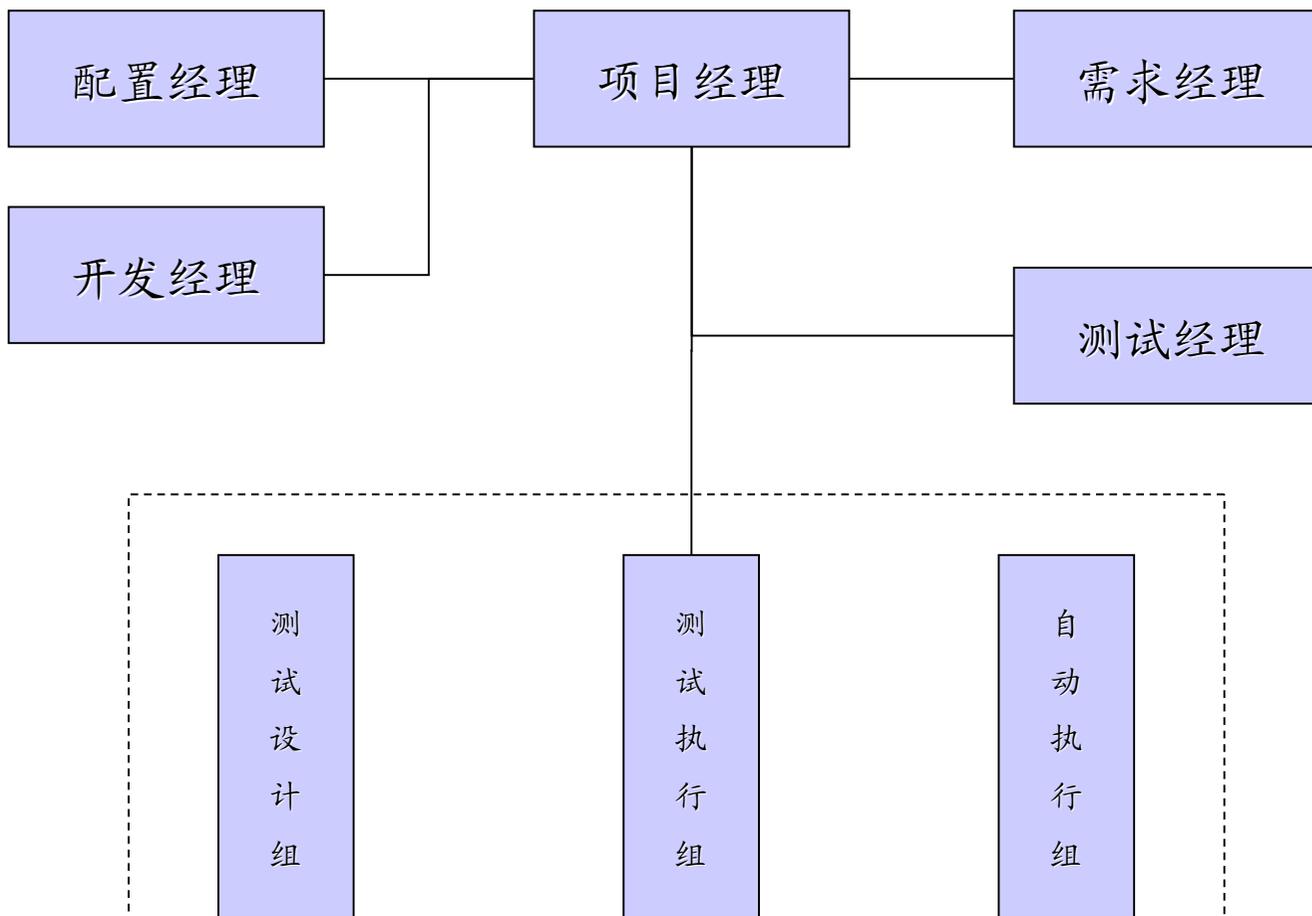


# 测试组织与关键过程

# 示例：测试组织



# 测试项目组织



# 测试关键过程

---

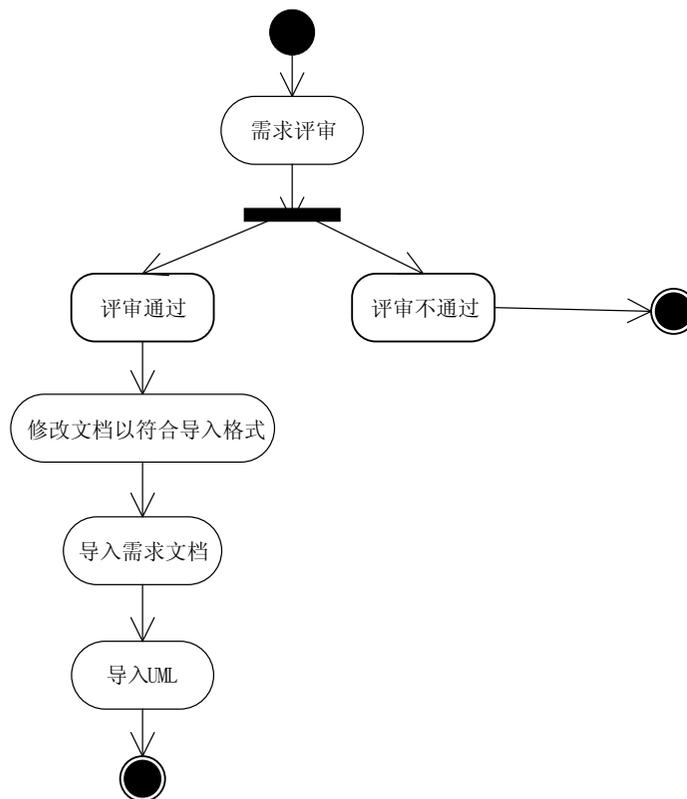
- 功能测试需求导入
- 功能测试活动
- 持续测试活动
- 缺陷管理活动

# 需求导入

---

- 前提：
  - 具备完整的需求描述文档
  - 具备完整UML需求描述文档
  - 测试管理工具具备导入功能
  - 测试管理工具提供导入文档的模板
  - 创建测试项目

# 功能测试需求导入



# 需求导入

---

- 角色:
  - 测试经理
  - 需求经理

# 需求导入

---

- 需求评审：
  - 评审需求整体是否完善
  - 评审需求的各个功能点是否按照要求进行描述
  - 评审需求点的角色
  - 评审需求点的前置条件是否清晰、明确
  - 评审需求点的处理流程是否清楚
  - 评审需求点的异常处理流程是否清楚
  - 评审需求点的验证方法

# 功能测试活动

---

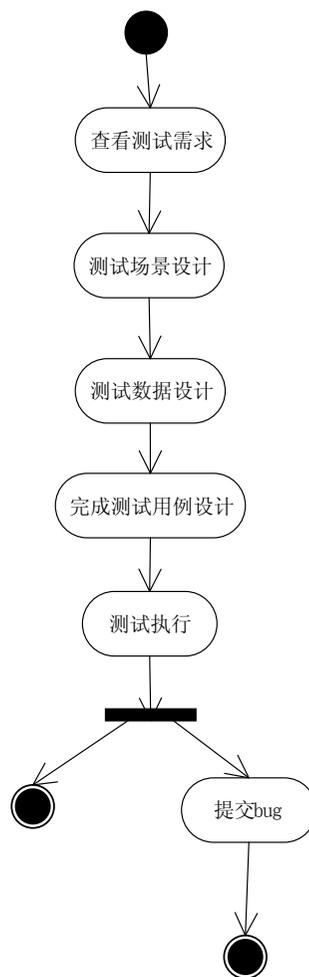
- 前提：
  - 程序员编程完成
  - 程序员自测完成
  - 提交版本进行功能测试
  - 具备测试版本、测试数据

# 功能测试活动

---

- 目标: 通过完整的测试, 发现缺陷
- 方法:
  - 手工测试与自动测试结合

# 功能测试活动：手工执行

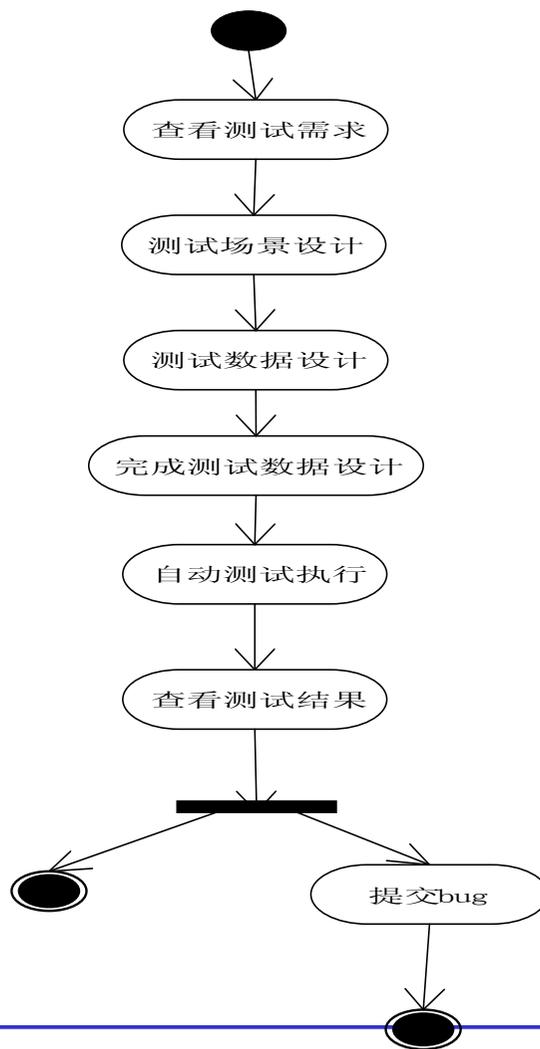


# 功能测试活动：手工执行

---

- 手工执行的优势：
  - 灵活性强
  - 能够发现测试用例之外的问题
- 手工执行缺点：
  - 测试用例无法上很大的数量
  - 手工执行的速度慢
  - 覆盖率比较低，只适合主流程测试
- 手工执行的适用：
  - 测试用例数量少的主流程测试

# 功能测试活动：自动执行

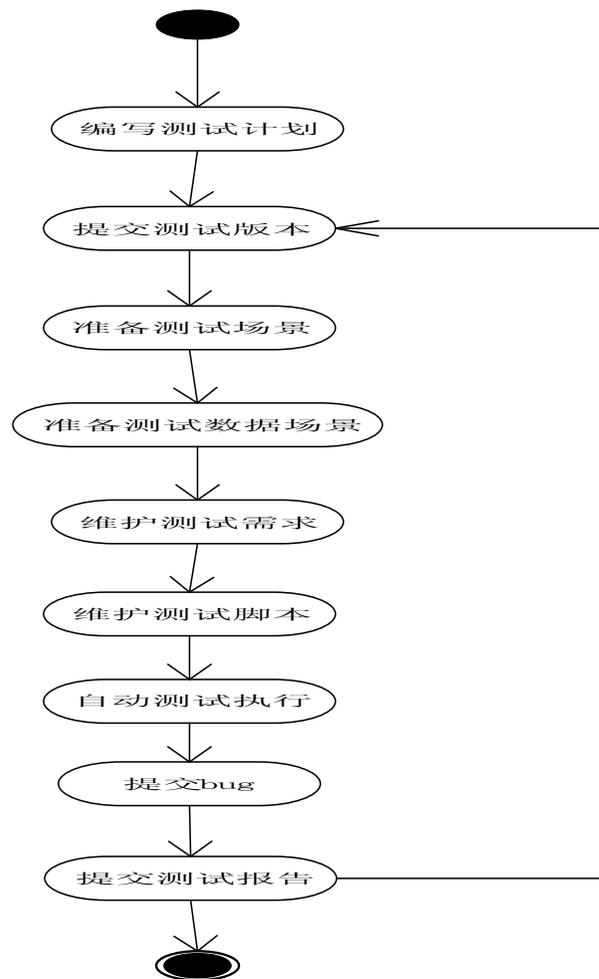


# 功能测试活动：自动执行

---

- 自动执行的优势：
  - 覆盖率高
  - 执行速度快
- 自动执行缺点：
  - 需要大量的测试数据为基础
  - 需要测试用例辅助设计工具
  - 需要自动测试工具
- 自动执行的适用：
  - 手工测试执行通过之后的测试

# 持续测试活动



# 缺陷管理活动

---

- 缺陷分析
  - 缺陷的趋势
    - 根据趋势来预测后期计划与工作量
  - 功能模块的缺陷个数比对
    - 衡量工作质量
  - 分析反复出现的缺陷
  - 分析“高发区”

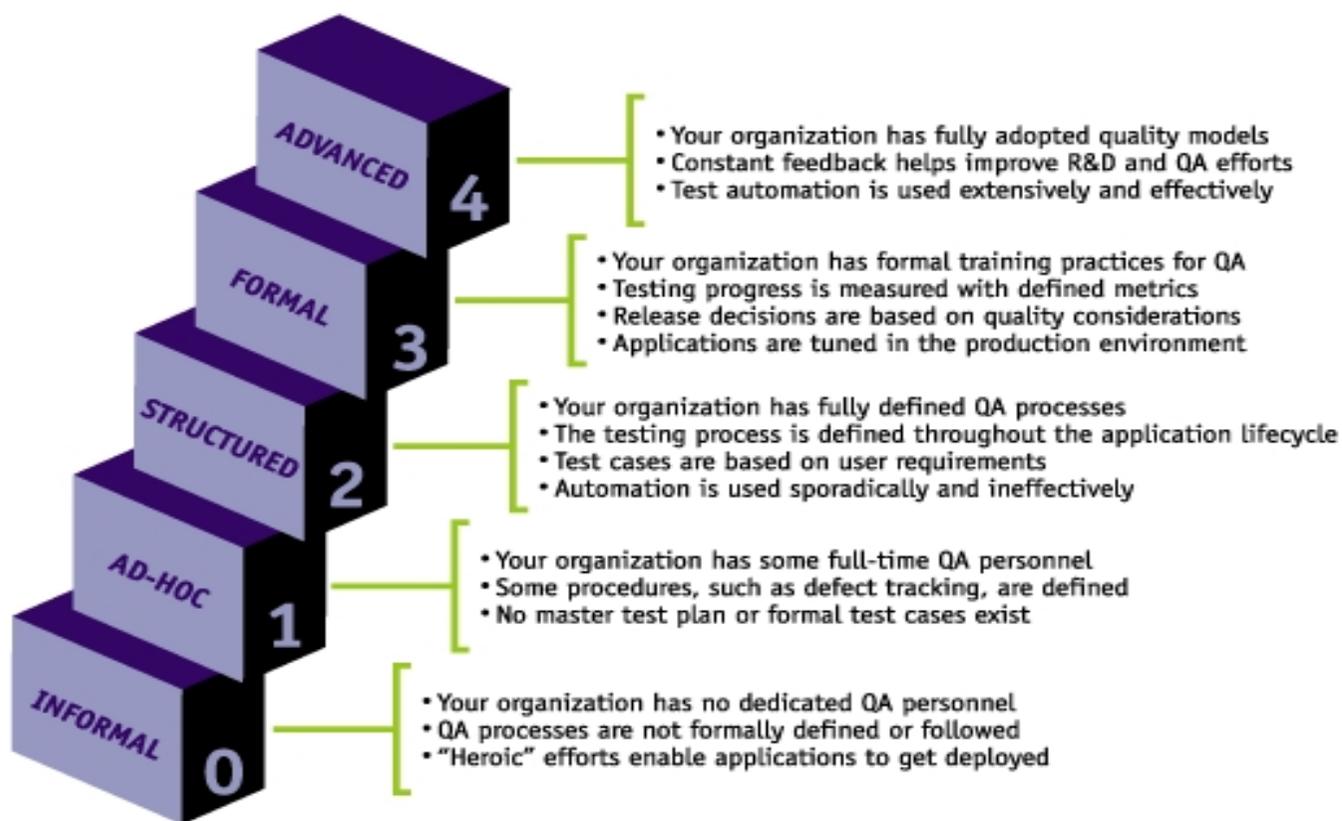
# 验收测试活动

---

- 输入：合同/协议
- 确定验收标准总则
- 确定验收标准
- 编写验收计划
- 配制项审核
- 可执行程序测试
  - 功能测试
  - 负载测试
  - 安全测试
  - 可用性测试
  - 其他测试
- 评审可执行测试是否通过

# 专业测试工作——TMM模型

## Application Risk Management Model (ARMM)



谢谢

