

# 软件质量与评测技术

## Software Quality & Evaluation Technology

计算机学院 单纯  
[sherryshan@bit.edu.cn](mailto:sherryshan@bit.edu.cn)  
2025年11月

# 软件测试实践

## Software Testing Practice

计算机学院 单纯

[sherryshan@bit.edu.cn](mailto:sherryshan@bit.edu.cn)

2025年11月26日

软件测试方法和技术最终要应用到实际工程项目中，通过项目实践来检验。测试员只有通过不断的项目实践来获取经验，才能提高自己的测试实战能力。

# 内容概览

- 1. 组建测试团队
- 2. 测试需求分析与测试计划
- 3. 设计和维护测试用例
- 4. 部署测试环境
- 5. 报告所发现的缺陷
- 6. 软件测试和质量分析报告
- 7. 软件测试项目管理



# 1. 组建测试队伍

- 1.1 测试团队的地位和责任
- 1.2 测试团队的构成
- 1.3 测试团队的管理和发展

# 1.1 测试团队的地位和责任

- 1.1.1 软件测试团队的任务
- 1.1.2 测试团队的规模



## 1.1.1 软件测试团队的任务

- 软件测试团队的责任
- 测试团队的地位及其与其他团队的关系

# 软件测试团队的责任（1）

## ■ 测试人员的基本责任

- 发现软件程序、系统或产品中所有的问题
- 尽早发现问题
- 督促开发人员尽快地解决程序中的缺陷

# 软件测试团队的责任（2）

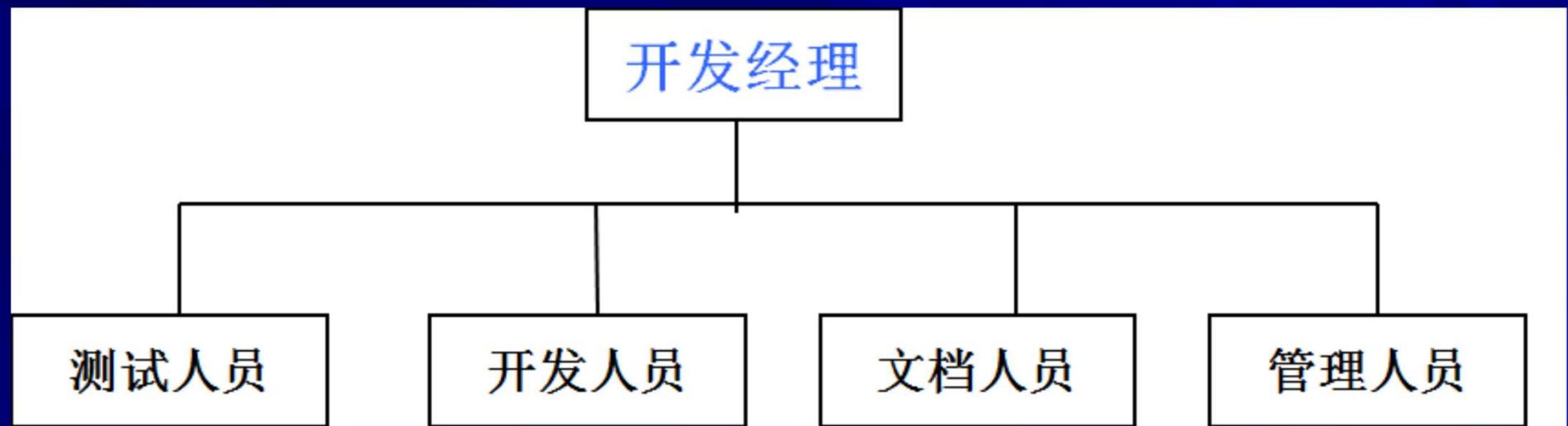
## ■ 软件测试团队的责任还包括

- 帮助项目管理人员制定合理的开发计划
- 对问题进行分析、分类总结和跟踪，以便让项目的管理者和相关的负责人能够对产品当前的质量情况一目了然
- 帮助改善开发流程、提高产品开发效率
- 督促代码编写具有更好的规范性、易读性、可维护性等

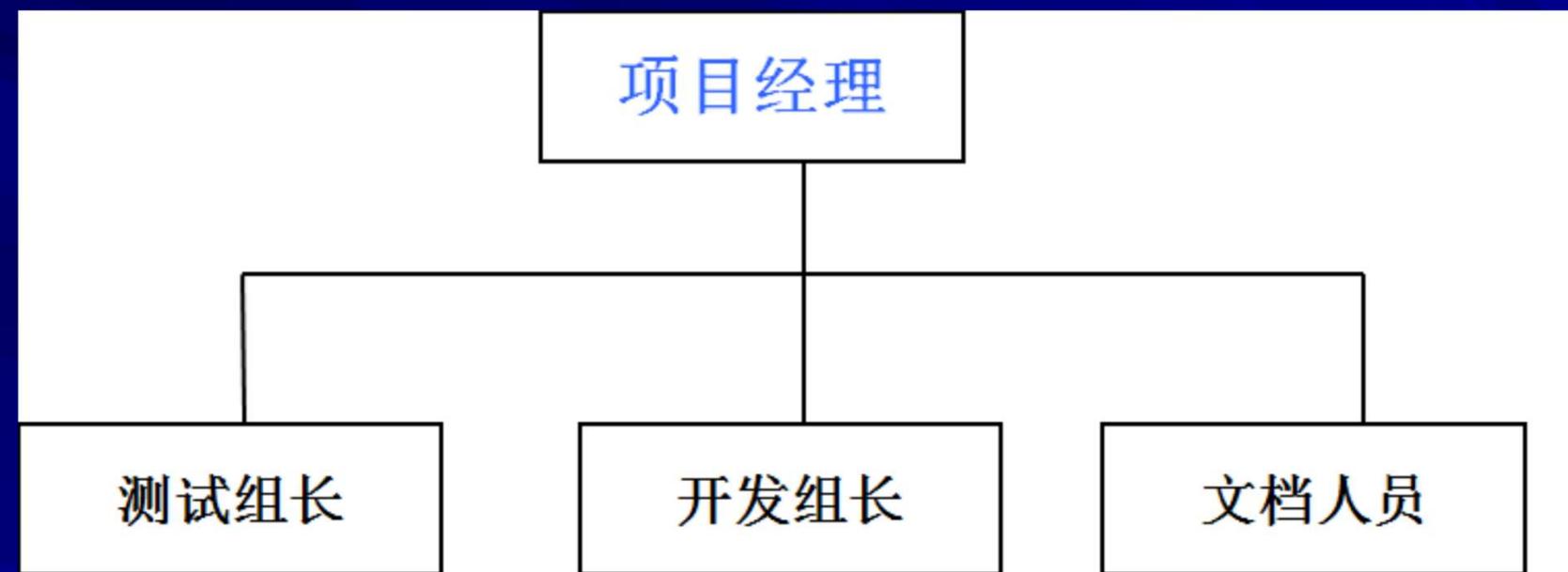
# 测试团队的地位及其与其他团队的关系

- 通过了解开发团队的构成，可以基本确定测试团队的地位。在不同的公司中，开发团队的模式存在较大的差异。可以概括为3类
  - 以开发为核心的组织模型
  - 以项目经理为核心的组织模型
  - 三国鼎立的组织模型

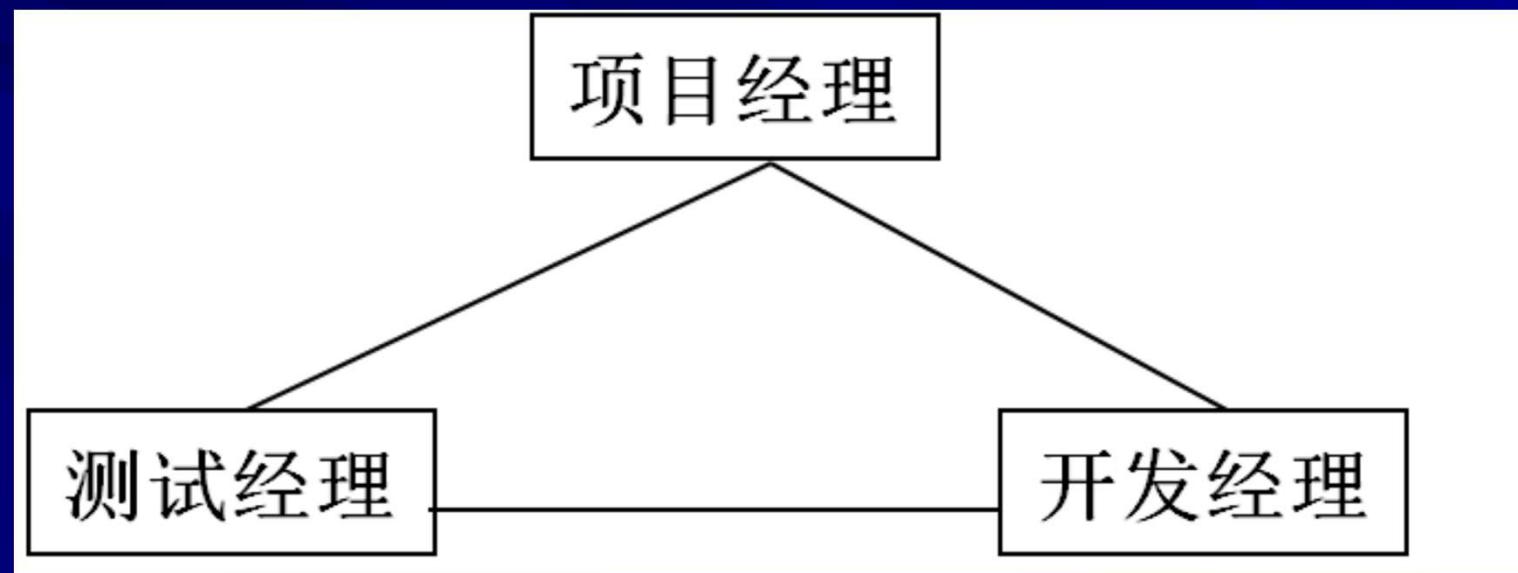
# 以开发为核心的组织模型



# 以项目经理为核心的组织模型



# 三国鼎立的组织模型



## 1.1.2 测试团队的规模（1）

- 针对一个项目建立测试小组，规模可以根据测试的范围来评估测试的工作量
- 对于长期存在的一个测试部门，其规模的确定要考虑研发部门或工程部门的预算、产品路线图（Product Roadmap）、多个项目并行、重叠的影响、项目延迟等各种情况。一般在考虑各种因素的情况下，还要加上10%~20%的富裕量（Buffer）

## 1.1.2 测试团队的规模 (2)

- 不同的应用，软件测试人员和软件开发人员的比例也是不同的，大致可以分为3类
  - 像操作系统一类的产品，对测试要求最高，测试人员与开发人员为2:1
  - 像应用平台、支撑系统一类的产品，对测试要求比较高，测试人员和开发人员的比例一般在1:1左右
  - 对于特定的应用类产品，由于用户对象清楚，范围小，甚至对应用平台或应用环境加以限制，测试人员与开发人员的比例一般在1:2~1:4

# 1.2 测试团队的构成

- 1.2.1 测试团队的基本构成
- 1.2.2 测试人员的责任
- 1.2.3 测试团队的组织模型



# 1.2.1 测试团队的基本构成（1）

- QA/测试经理

- 人员管理、资源配置、测试方法改进等

- 实验室管理人员

- 设置、配置和维护实验室的测试环境

- 内审员

- 审查流程、建立测试模板、跟踪缺陷测试报告的质量等

## 1.2.1 测试团队的基本构成（2）

- 测试组长

- 负责项目的管理、测试计划、测试用例、任务安排等

- 测试设计人员/资深测试工程师

- 产品设计规格说明书的审查、测试用例的设计、技术难题的解决、培训和指导、实际测试任务的执行

- 一般（初级）测试工程师

- 执行测试用例和相关的测试任务

## 1.2.2 测试人员的责任

- 初级测试工程师
- 测试工程师
- 资深测试工程师
- 测试实验室管理员
- 软件包构建或发布工程师
- 测试组长
- 测试经理

# 初级测试工程师（1）

- 了解和熟悉产品的功能、特性等
- 验证产品在功能、界面上是否和产品规格说明书一致
- 按照要求，执行测试用例，进行功能测试、验收测试等，并能发现所暴露的问题
- 清楚地描述所发现的缺陷

# 初级测试工程师（2）

- 使用简单的测试工具
- 努力学习新技术和软件工程方法，不断提高自己的专业水平
- 接受测试工程师的指导，执行主管所交代的其他工作

# 测试工程师（1）

- 熟悉产品的功能、特性，审查产品规格说明书
- 验证产品是否满足了规格说明书所描述的需求
- 根据需求文档或设计文档，可以设计功能方面的测试用例
- 根据测试用例，执行各种测试，发现所暴露的问题

# 测试工程师（2）

- 全面使用测试工具，包括开发测试脚本
- 安装、设置简单的测试环境
- 报告所发现的软件缺陷，审查和跟踪软件缺陷，直到缺陷关闭
- 编制测试报告
- 负责对初级测试工程师的指导，执行主管所交代的其他工作

# 资深测试工程师（1）

- 负责系统一个或多个模块的测试工作
- 制定某个模块或某个阶段的测试计划、测试策略
- 设计自动化测试框架或结构，开发测试脚本、必要的测试工具
- 设计测试环境所需的系统或网络结构，安装、设置复杂的系统测试环境

# 资深测试工程师（2）

- 熟悉产品的功能、特性，审查产品规格说明书，并提出改进意见
- 审查系统、程序设计说明书，提出对系统模块设计的改进要求
- 审查代码
- 验证产品是否满足了规格说明书所描述的需求

# 资深测试工程师（3）

- 根据需求文档或设计文档，设计复杂的测试用例，包括性能测试、故障转移测试、安全性测试等方面的测试用例
- 负责执行性能测试、故障转移测试、安全测试等
- 负责对测试工程师的指导，执行主管所交代的其他工作

# 测试实验室管理员（1）

- 负责测试环境所需的网络规划和建设，维护网络的正常运行
- 建立、设置和维护测试环境所需的应用服务器或软件平台
- 对实验室 的硬件、软件资源进行登记、分配和管理
- 申请所需求 的、新的硬件资源、软件资源，协助有关部门进行采购、验收

# 测试实验室管理员（2）

- 对使用实验室的硬件、软件资源的权限进行设计和设置，保证其安全性
- 安装新的测试平台、被测试的系统等
- 优化测试环境，提高测试环境中网络、服务器和其他设备运行的性能

# 软件包构建或发布工程师（1）

- 负责源程序代码管理系统（如CVS、SourceSafe等）的建立、管理和维护
- 制定check in/check out等相关的源代码控制规则
- 文件名定义规范，建立合理的程序文件结构和存储目录结构
- 为程序的编译、链接等软件包构造工作建立自动处理文件

# 软件包构建或发布工程师（2）

- 检查被测试的软件包及其文件版本是否正确、有效
- 负责日常的软件包构建（Daily Build），而且确保软件包不含病毒、不缺少任何文件等
- 软件包的接收、发送、存储和备份等

# 测试组长（1）

- 负责一个独立的测试项目，制定整个项目的测试计划、测试策略，包括风险评估、日程表安排等
- 测试小组的管理或参与测试团队的管理，负责测试项目内部的资源和任务安排
- 熟悉产品的功能、特性，审查产品需求定义和功能规格说明书，并提出修改意见

# 测试组长（2）

- 实施软件测试，验证产品是否满足了规格说明书所描述的需求，并对软件问题进行跟踪分析和报告，推动测试中发现问题及时合理地解决
- 编写项目的整体测试报告，保证产品质量
- 对竞争者的产品进行分析，提出对软件的进一步改进的要求并且评估改进方案是否合理

# 测试组长（3）

- 监督测试流程的执行，并将执行过程中所发现的问题反馈给测试经理或项目经理
- 为团队成员提供技术指导，协助测试经理工作

# 测试经理（1）

- 负责整个测试团队或部门的管理，包括测试岗位的定义、组织团队结构的建立和优化、团队的建设和发展、培训活动的组织、员工的激励等
- 负责一个完整产品的软件测试和质量保证等工作，包括项目组长的指定、项目资源的安排、项目进度的跟踪、项目审查和总结等

# 测试经理（2）

- 测试部门年度/季度计划、预算的编写、实施和评估
- 促进质量文化的普及，促进开发团队的每位成员建立正确的质量观
- 协助人力资源部门做好测试人员的招聘、考核等方面的工作

# 测试经理（3）

- 定义、实施软件测试流程或整个开发周期流程，并收集、处理流程实施中所存在问题，最终不断改进流程
- 审查项目的测试计划、测试策略等，包括资源调度和平衡、风险评估等
- 和其他部门协调，参加多方会议审查产品需求定义和功能规格说明书，解决其中的问题

# 测试经理（4）

- 指导测试项目组实施软件测试，并对软件问题进行跟踪分析和报告，推动测试中发现问题及时合理地解决
- 审查项目的测试报告，组织产品质量的分析，提交质量分析报告
- 对竞争者的产品进行深度分析，提出改进软件产品的建议，或评估改进方案是否合理等

# 一个微软测试工程师的一天

## (1)

- 产品构建完成之后，每日凌晨，测试编译自动开始
- 如果测试编译成功，BVT（Basic Verification Test，基本验证测试）自动开始
- 测试工程师早晨来上班，第一件事就是检查测试包（Test Build）与BVT结果的E-mail

# 一个微软测试工程师的一天 (2)

- 如果有BVT错误，在第一时间里分析原因，隔离错误代码并报告最高级别的缺陷（Pri 0 Bug，开发团队应于当日之内修正这类Bug）
- 测试工程师接着在缺陷管理系统中检查Bug情况，验证分配给自己的、开发人员已修改的Bug

# 一个微软测试工程师的一天

## (3)

- 关闭Bug，并针对此Bug修正所影响的范围，执行回归测试
- 验证最近开发的测试脚本执行的结果。如果有新的错误，报告Bug并进行调试，解决脚本中的问题
- 开发新的测试规范或新的测试脚本
- 使用个人所建的任务，验证自己新开发的测试脚本

# 一个微软测试工程师的一天

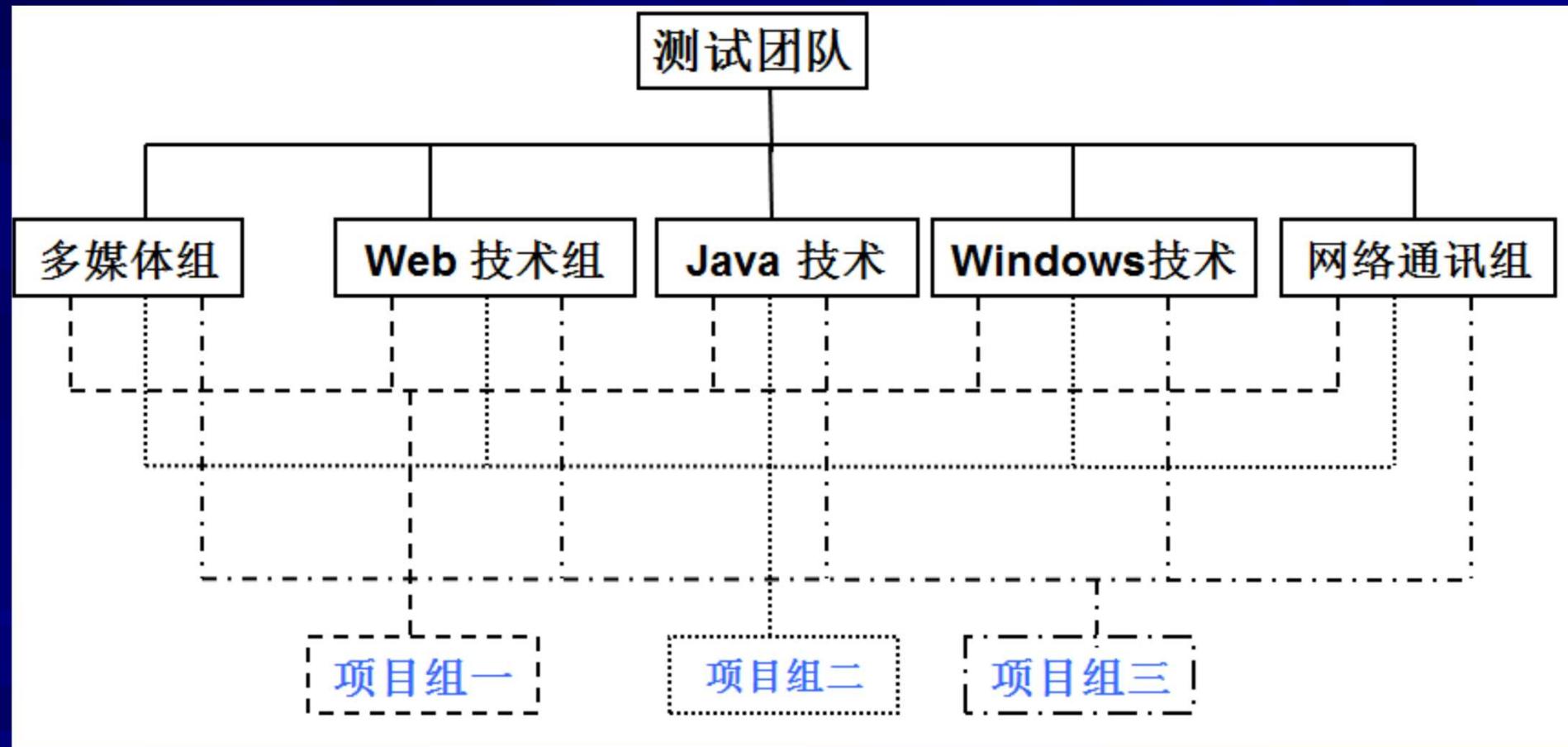
## (4)

- 用已通过的脚本来验证所对应开发人员的、新版本的程序，尽量发现任何严重的问题
- 改进与提高自动化测试系统的功能
- 参与产品规格说明书、测试用例的评审会议
- 复审测试同伴写的脚本和相关文档
- 回答项目相关的其他各种问题

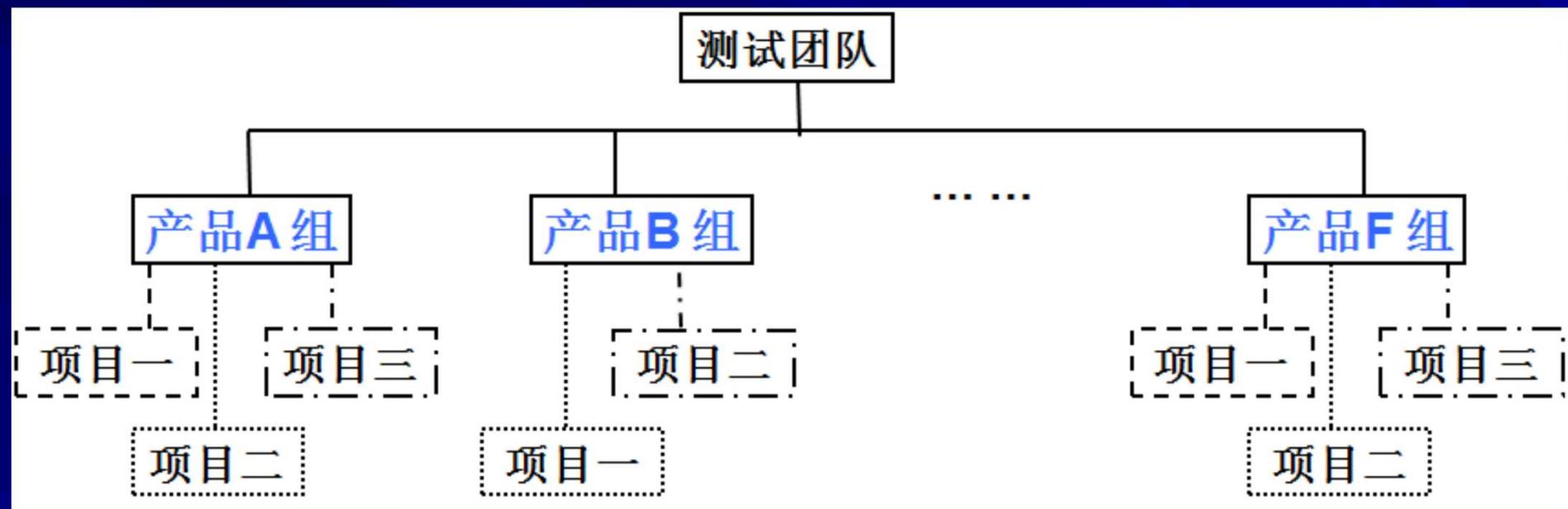
### 1.2.3 测试团队的组织模型

- 测试团队的组织直接关系到测试团队的生产效率和生产力，其组织的方式由测试团队规模和具体任务、技术等决定，对于小的测试团队所要求的结构很简单，一般以项目来组织就可以了。
- 对于大型测试团队的组织，需要构造多层结构，可以归纳为两种基本类型
  - 按技术领域来组建团队
  - 按产品线来组建团队

# 按技术领域来组建团队

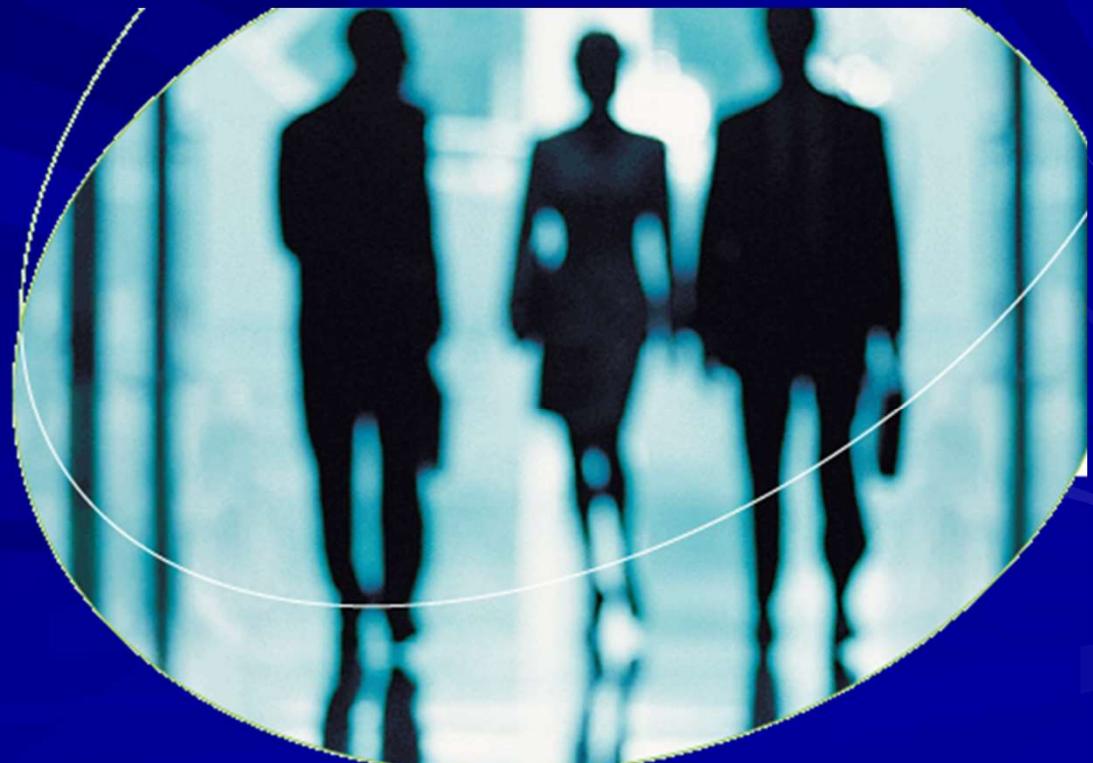


# 按产品线来组建团队



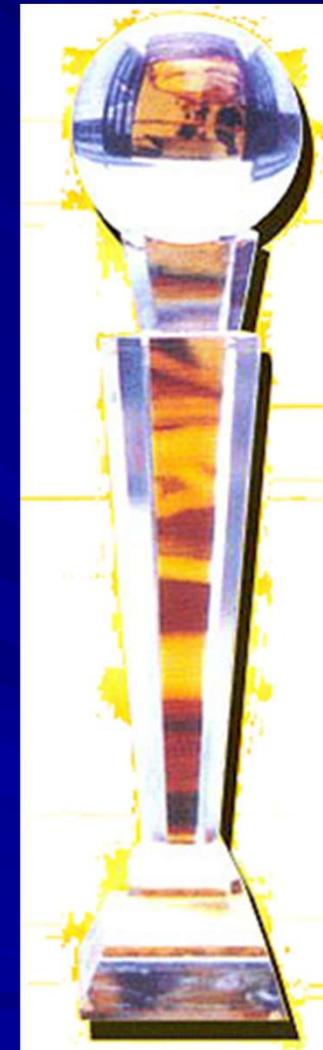
# 1.3 测试团队的管理和发展

- 1.3.1 测试团队的激励方法
- 1.3.2 知识共享和在岗培训

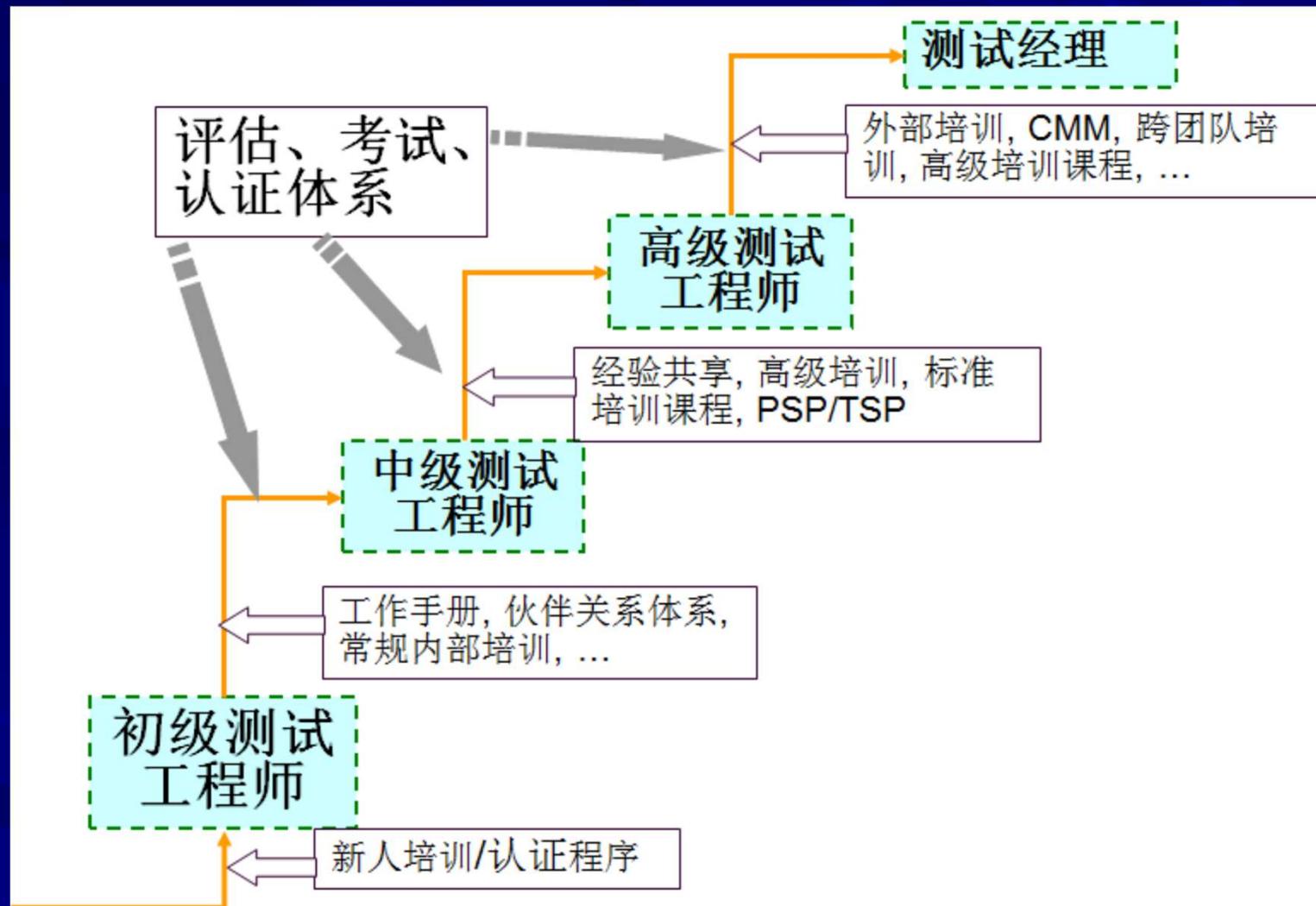


## 1.3.1 测试团队的激励方法

- 表扬和奖励
- 站在测试小组一边
- 提高士气
- 支持合理的工作方式



## 1.3.2 知识共享和在岗培训



## 2. 测试需求分析与测试计划

- 2.1 测试的目标和准则
- 2.2 测试需求分析
- 2.3 测试项目的估算与进度安排
- 2.4 测试风险和测试策略
- 2.5 测试计划的内容与编制

# 2.1 测试的目标和准则

- 2.1.1 确定测试目标
- 2.1.2 软件测试项目的出入准则
- 2.1.3 测试项目管理的原则

## 2.1.1 确定测试目标

- 明确测试目标是测试需求分析和计划测试的前提



# 哪个是测试目标？

■ 能作为测试的目标吗？有什么问题吗？

指标	缺省取值	本项目取值	指标说明
测试执行率	95%	[95%]	实际执行的测试用例数/计划执行的测试用例数
执行通过率	90%	[90%]	执行结果为“通过”的测试用例数/实际执行的测试用例总数
严重缺陷修复率	100%	[100%]	已修复并关闭的严重缺陷数/QC 测试发现的严重缺陷总数
缺陷解决率	100%	[100%]	已解决的缺陷数/QC 测试发现的缺陷总数

# 测试目标

## ■ 设计和执行测试的原因或目的

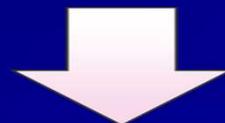
- 向风险管理活动提供信息
- 提供软件系统质量有关信息
- 评估软件产品是否满足相关利益者的期望
- 评估缺陷修正（清除）而不带来负面效应
- 评估软件变更实施而不带来负面效应
- 评估软件是否完全符合合规性要求

# 项目的具体测试目标

- 提供哪些质量风险信息
- 新改动的业务是否正确实现，对已有业务是否有负面影响
- 是否满足功能性要求和非功能性要求
- 在测试覆盖率、测试效率上的具体要求

# 如何确定测试目标？

- 哪些业务改动，会影响哪些已有业务？
- 系统改动会影响哪些系统功能和非功能特性？
- 测试覆盖率：新业务/功能？已有业务/功能呢？
- 如何最大程度提高测试效率？



- 产品质量要求、业务功能关系分析、测试范围分析、测试策略和方法选择

## 2.1.2 软件测试项目的出入准则

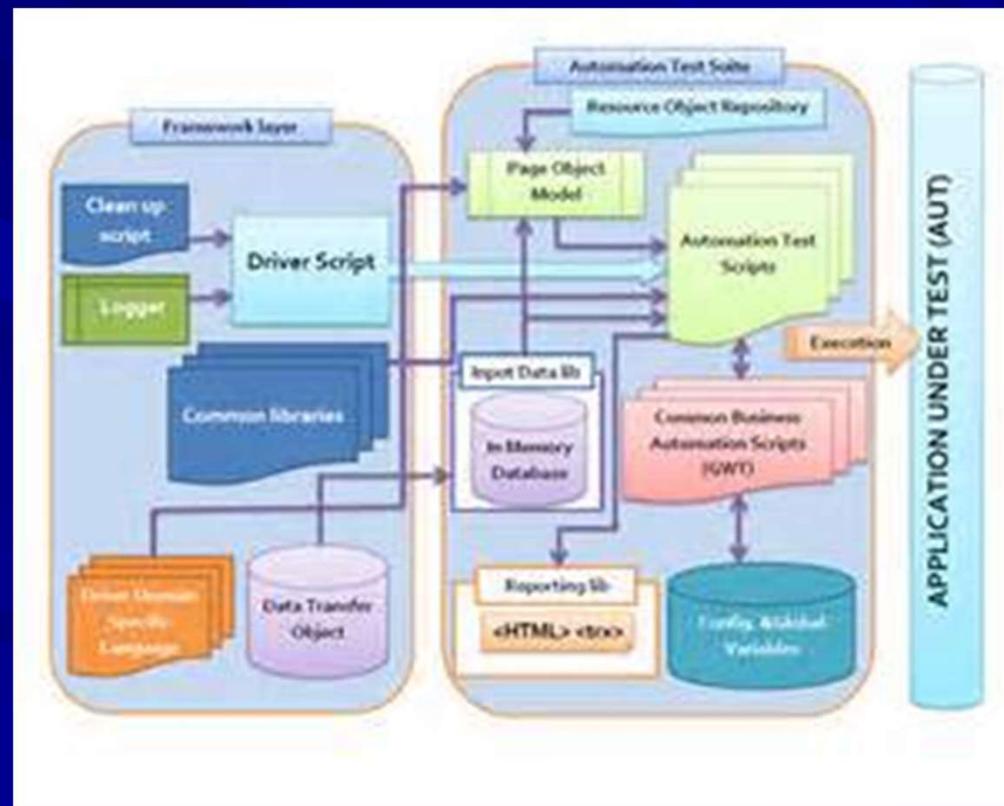
- 清楚了解项目的整体计划框架
- 完成需求规格说明书评审
- 技术知识或业务知识的储备
- 标准环境
- 技术设计文档
- 足够的资源
- 人员组织结构及其责任已确定

## 2.1.3 测试项目管理的原则

- 可靠的需求
- 能够适应开发过程模型
- 充分测试和尽早开始测试
- 合理的时间表
- 充分沟通
- 基于数据库的测试管理系统

## 2.2 测试需求分析

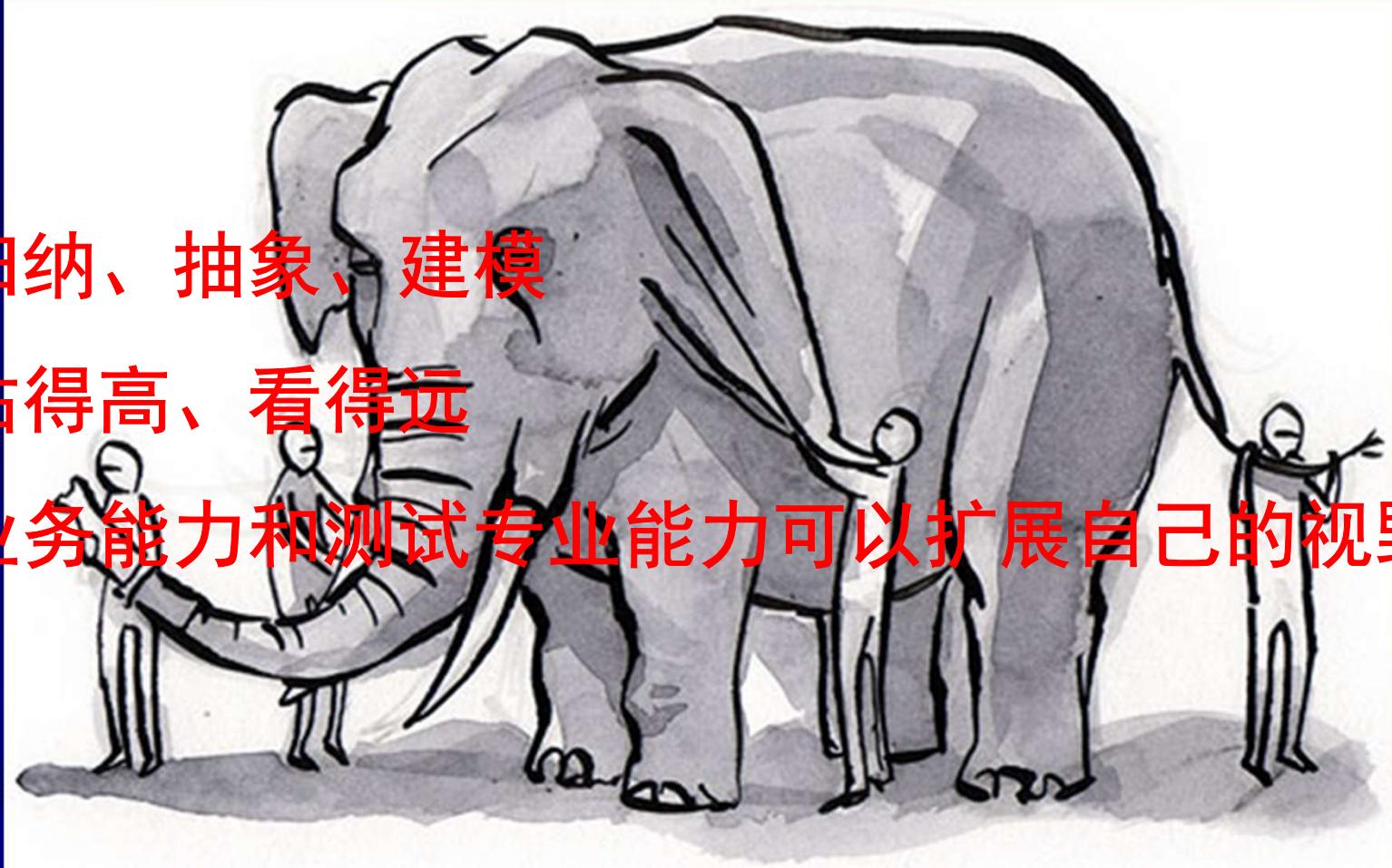
- 确定测试范围
- 测试项和测试子项
- 测试优先级
- 测试风险



# 测试需求分析过程

- 了解项目的背景、产品价值，解决什么业务问题 → 分析业务需求，确定测试目标
- 了解用户是谁，用户所关心的问题 → 分析用户需求
- 确定待测软件的功能特性，可以从整体到局部，从上到下，逐层分解，形成待测试的功能列表
- 确定待测软件的非功能特性，基于本系统的特点而需特别关注的质量属性
- 确定测试项的优先级

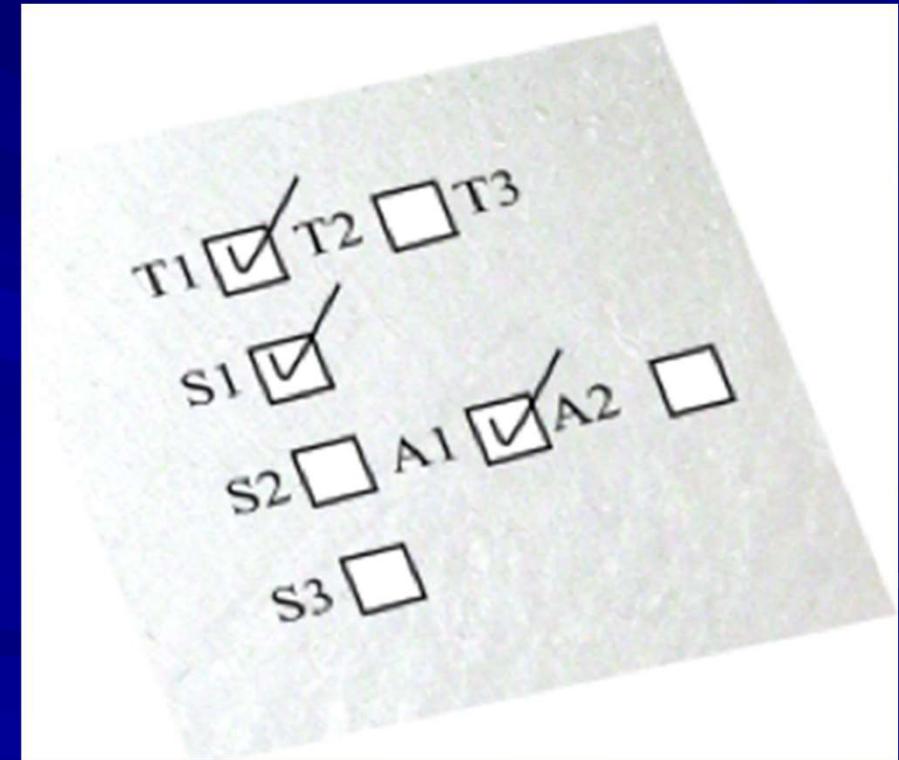
# 全面地进行分析



归纳、抽象、建模  
站得高、看得远  
业务能力和测试专业能力可以扩展自己的视野

# 测试需求分析项

- 项目背景
- 业务（流程、用户角色）
- 用例、场景
- 支撑业务的功能
- 功能优先级
- 非功能特性



# 各种软件需求



# 示例：业务需求还是功能需求？

- 2. 业务需求总体描述
- 2. 1. 主要功能模块

序号	功能模块列表
1	待办待阅
2	信贷系统待办待阅
3	日程安排和任务管理
4	系统公告
5	新闻动态
6	系统链接
7	工具箱
8	生日提醒
9	通讯录
10	IT 服务电话

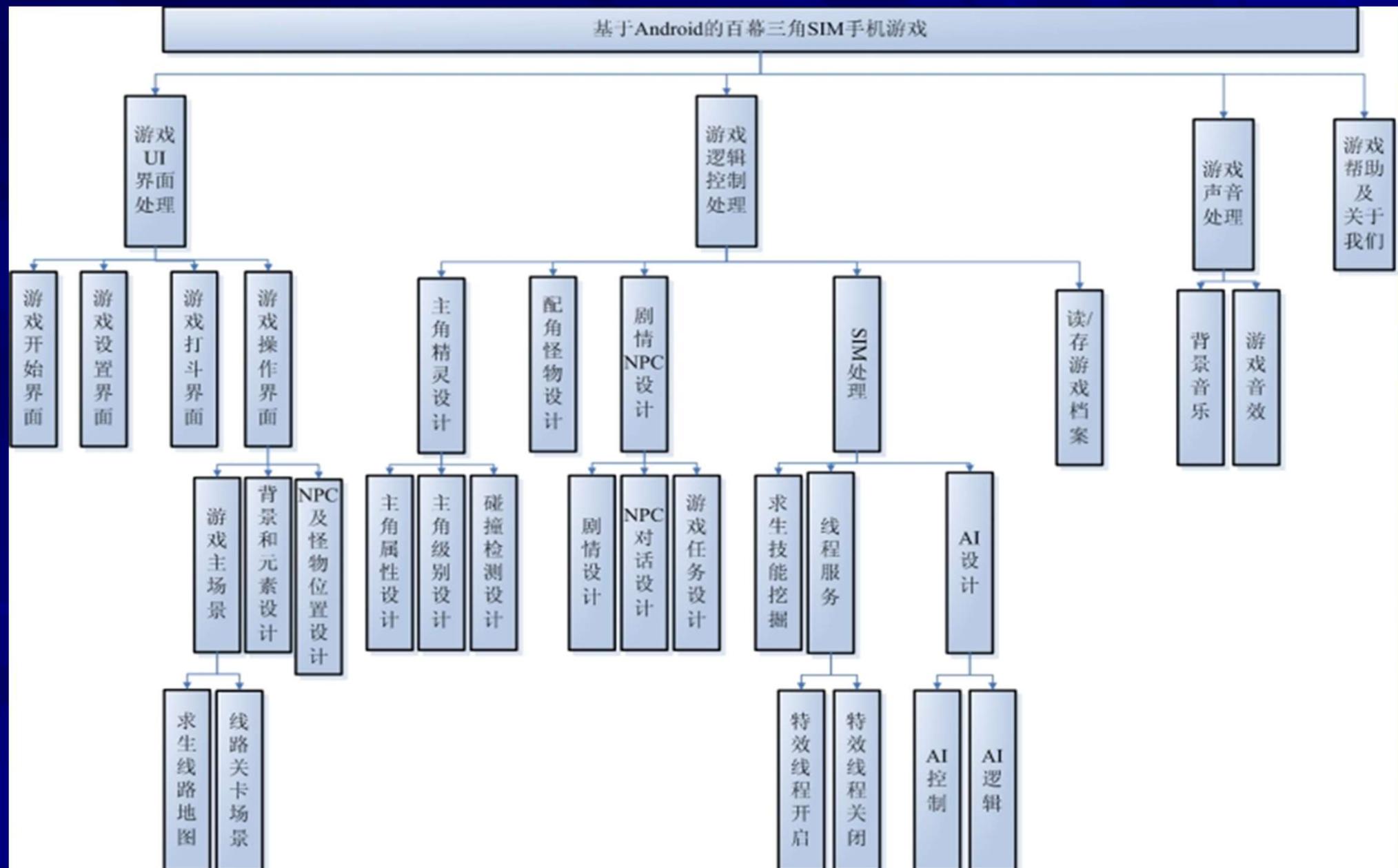
# 软件需求的层次

- 业务需求反映组织机构或客户对系统、产品的概括性要求，包括所要达到的业务目标，有项目视图与范围文档说明
- 用户角色需求描述用户使用系统而要完成的各种任务，有用例（use case）文档或方案脚本说明
  - 用户角色需求可以扩展到涉众（干系人）需求
- 功能需求定义开发人员必须实现的软件功能，它源于用户需求，是软件需求说明书 中重要的组成部分

# 业务需求

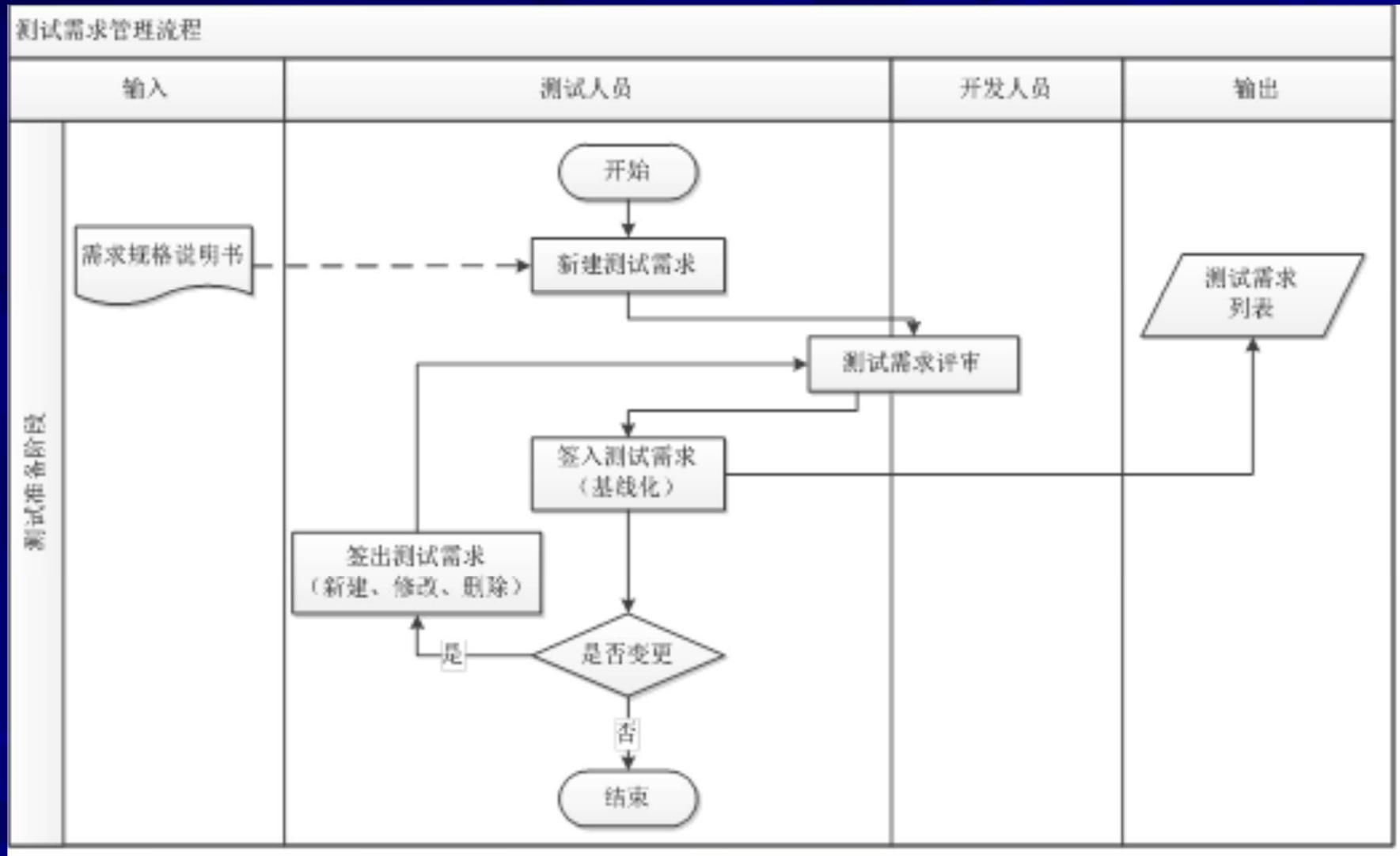
- 客户
- 业务流程
- 业务规则
- 业务操作
- 业务数据
- 业务安全性
- 业务可管理
- 业务发展

# 功能：层次分析



# 非功能性：性能需求

- 只有具备了清楚而量化的性能指标，性能测试才能开始实施。
  - 最终用户的体验，如2-5-10原则
  - 商业需求，如“比竞争对手的产品好”
  - 技术需求，如CPU使用率不超过70%
  - 标准要求
- 响应时间是用户的关注点，容量和数据吞吐量是（产品市场团队）业务处理方面的关注点，而系统资源占用率是开发团队的技术关注点





Screenshot of the "Test Management System" application window.

Top menu bar: 文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

Toolbar icons: 后退(B) 前进(F) 搜索(S) 收藏夹(C) 分析(A)

Address bar: 地址(D): http://qa.wor.or/ideas/explorer.jsp

User info: 项目: 贵阳电子商务管理信息系统, 用户: 王云鹏

Left sidebar:

- 系统管理
- 测试需求管理
- 测试计划管理
- 公用用例管理
- 测试执行管理
- 测试缺陷管理

Central area:

名称 (Name): DISW-01-02 企业用户注册

需求详细信息 (Requirement Details):

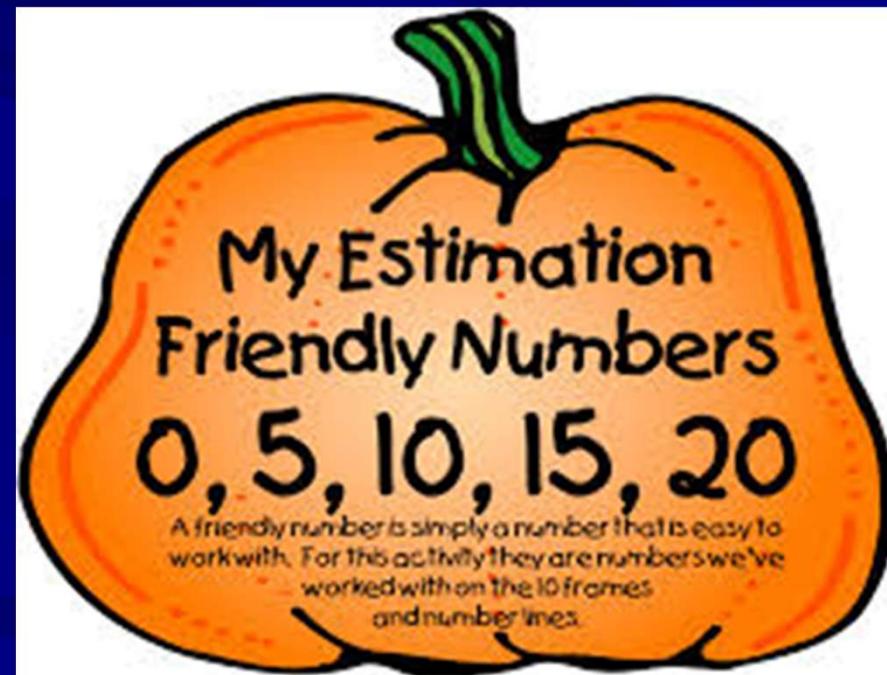
名称:	DISW-01-02 企业用户注册
版本号:	2
创建日期:	2016-11-15
目标发行版:	
修改日期:	2017-2-20 上午 10:52:57
已审阅:	未审阅
直接覆盖状态:	Failed

右侧显示:

- 需求类型: 功能
- 产品:
- 创建时间: 22:59:18
- 目标周期:
- 需求 ID: 174
- 优先级:
- 作者: 王云鹏

## 2.3 测试项目的估算与进度安排

- 2.3.1 测试工作量估算
- 2.3.2 工作分解结构表方法



## 2.3.1 测试工作量估算（1）

- 测试工作量是根据测试范围、测试任务和开发阶段来确定的，测试范围和测试任务是测试工作量估算的主要依据
- 测试任务由质量需求、测试目标决定
- 测试范围由产品（新）功能特性或测试任务决定
- 代码质量越低，测试工作量越大，如回归测试次数与频率加大

## 2.3.1 测试工作量估算（2）

- 处在不同的开发阶段测试工作量不同
- 自动化程度高，测试工作量就越低
- 针对不同的应用领域、技术、编程语言，其估算方法不同

## 2.3.2 工作分解结构表方法（1）

- 测试工作量的估算依赖于测试任务的细化，对每项测试任务进行分解，然后根据分解的子任务进行估算。通常分解粒度越小，估算精度越高

## 2.3.2 工作分解结构表方法（2）

- 列出本项目需要完成的各项任务
- 对每个任务进一步细分，可进行多层次的细分，直到不能细分为止
- 根据任务的层次进行编号，形成工作分解结构表

# 示例：工作分解结构表

1 测试计划	4 测试执行
1.1 确定测试目标	4.1 第 1 轮新功能测试
1.2 确定测试范围	4.2 性能测试
1.3 确定测试资源和进度	4.3 安全性测试
1.4 测试计划写作	4.4 安装测试
1.5 测试计划评审	4.5 第 2 轮回归新测试
2 需求和设计评审	4.6 升级和迁移测试
2.1 阅读文档以了解系统需求	4.7 最后一轮回归测试
2.2 需求规格说明书评审	5 测试环境建立和维护
2.3 编写/修改测试需求	5.1 软硬件购买
2.4 设计讨论	5.2 测试环境建立
2.5 设计文档评审	5.3 日常维护
3 测试设计和脚本开发	6 测试结果分析和报告
3.1 确定测试点	6.1 缺陷跟踪和分析
3.2 设计测试用例	6.2 性能测试结果分析
3.3 评审和修改测试用例	6.3 编写测试报告
3.4 设计测试脚本结构	7 测试管理工作
3.5 编写测试脚本基础函数	7.1 测试人员培训
3.6 录制测试脚本	7.2 项目会议
3.7 调试和修改测试脚本	7.3 日常管理
3.8 测试数据准备	.....

## 2.4 测试风险和测试策略

- 2.4.1 测试风险管理计划
- 2.4.2 测试策略制定

## 2.4.1 测试风险管理计划

- 软件测试总是存在较高的风险，测试风险管理就是设法降低或缓解测试过程中的风险



# 风险识别（1）

- 风险识别的有效方法就是建立风险项目检查表
- 历史资料、Brainstorming等帮助建立项目检查表
- 风险识别并确定其程度，给出预防或处理措施

# 风险识别 (2)

## ■ 风险项目检查表

类别	内容	示例
人员风险	测试人员的状态、责任感、行为规范等。	个人工作疏忽而漏掉缺陷；某个人生病、离职等，造成资源不足，使测试不够充分、或缩小测试范围。
环境风险	多数情况下，测试环境是一个模拟环境，很难和实际运行环境一致。	用户数据量、运行环境的垃圾数据、十几年的运行时间等。
测试范围（广度）	很难完成 100% 的测试覆盖率，有些边界范围容易被忽视。	很难覆盖模块之间接口参数传递、成千上万的操作组合等测试。
测试深度	对系统容量、可靠性等测试深度不够。	象互联网上的应用，操作行为和习俗等研究不够、测试时达不到实际的用户数
回归测试	回归测试，一般都不会运行所有的测试用例，而是运行部分的测试用例。越到测试后期，回归测试执行的用例数越少。	修正一个缺陷后，除了验证这个缺陷之外，测试人员往往根据自己经验来确定回归测试范围，而且这个范围很小。
需求变更	软件需求变化相对较多，有时在后期还发生需求变更，从而影响设计、代码，最后反映到测试中来。	需求变更后，文档不一致，测试用例没有及时更新、回归测试不足等。
用户期望	测试人员不是用户，很难百分之百地把握用户的期望，这种差异也会带来风险。	测试的适用性测试就是一个典型的例子，用户界面喜好、操作习惯，都会因不同的用户存在或多或少的差异。
测试技术	借助一些测试技术完成测试任务，可能有些测试技术不够完善，有些测试技术存在一定的假定，都会带来风险。	如正交实验法在软件测试中应用时，很难达到其规定的条件。

# 风险常用控制方法

- 采取措施避免可以避免的风险
- 高风险转移为低风险
- 设法降低不可避免的风险
- 做好风险管理计划
- 制定处理风险的应急、有效方案
- 计划时，对于估算资源、时间、预算留有余地
- 制定文档标准，建立机制，保证文档及时产生

## 2.4.2 测试策略制定（1）

- 测试策略描述当前测试项目的目标和所采用的测试方法，描述不同测试阶段的测试对象、范围和方法以及每个阶段内所要进行的测试类型。

## 2.4.2 测试策略制定（2）

- 针对风险（工作量、时间等压力）采取对策，包括遵照的标准取舍、测试任务的优先级等
- 如何更好地执行测试用例以及后续的回归测试
- 选定使用测试技术和工具
- 考虑影响资源分配的特殊情况

# 测试策略影响因素

- 测试方式（静态/动态，探索式方式）
- 测试方法（黑盒/白盒）
- 测试层次（单元、集成、系统）
- 测试人员（责任、能力、独立性）
- 测试用例选择（如用例是否有优先级）
- 测试环境（设置是否简单、自动部署）
- 测试工具（能不能用测试工具、使用简单与否）
- 质量标准（国内标准或其他）

## 2.5 测试计划内容与编制

- 2.5.1 测试计划内容
- 2.5.2 测试项目的计划过程
- 2.5.3 制定有效的测试计划

## 2.5.1 测试计划内容

- 软件测试计划是指导测试过程的纲领性文件，描述测试活动的范围、方法、策略、资源、任务安排和进度等，并确定测试项、哪些功能特性将被测试、哪些功能特性将无需测试，识别测试过程中的风险
- 内容主要集中在测试目标和需求说明、测试工作量估算、测试策略、测试资源配置、进度表、测试风险等

# GBT 9386-2008 (1)

## ■ 测试计划应有如下结构

- 测试计划标识符
- 引言
- 测试项
- 要测试的特征
- 不要测试的特征
- 方法
- 测试项通过准则

# GBT 9386-2008 (2)

## ■ 测试计划应有如下结构（续）

- 暂停准则和恢复要求
- 测试交付项
- 测试任务
- 环境要求
- 职责
- 人员配备和培训要求

# GBT 9386-2008 (3)

- 测试计划应有如下结构（续）
  - 进度
  - 风险和应急
  - 批准

*Thank You*