

测试技术应用与实践



软件测试目的



我们先来讨论一个问题：

软件测试的目的是什么呢？

软件测试定义



- 什么是软件测试？
 - 程序测试是为了发现错误而执行程序的过程
 - 对软件系统中潜在的各种风险进行评估的活动
 - V&V, 验证 (Verification) 和确认 (Validation)
 - 静态测试和动态测试
- 两种观点
 - 验证软件是‘工作的’ (Dr.Bill Hetzel)
 - 证明软件是‘不工作的’ (G.J.Myers)

软件产品的质量需求



- 质量的功能需求
 - 功能性、易用性
- 质量的非功能需求
 - 性能、有效性、可靠性
- 质量的用户需求
 - 功能性、易用性、可用性、安全性
- 质量的企业需求
 - 可维护性、兼容性、可移植性、可扩展性

软件缺陷



- 软件缺陷定义：
 - 从内部看，软件缺陷是软件产品开发或维护过程中所存在的错误、毛病等各种问题；
 - 从外部看，软件缺陷是系统需要实现的某种功能的失效或违背。
- 软件缺陷表现形式：
 - 功能、特性没有实现或部分实现；
 - 设计不合理，功能特性不明确，逻辑不清晰或矛盾；
 - 产品实际结果和所期望的结果不一致；
 - 没有达到产品规格说明书所规定的特性、性能指标等；
 - 运行出错，包括运行中断、系统崩溃、界面混乱等；
 - 硬件或系统软件上存在的其他问题。

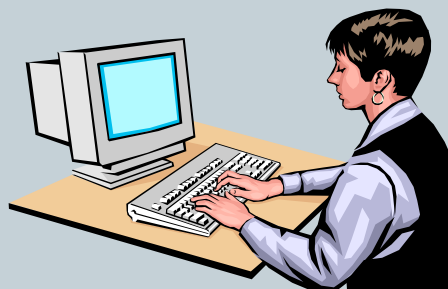
软件问题的产生



人的错误



错误未发现引入缺陷



执行软件



发现故障



无容错措施导致失效



分析原因找根源

如何指出问题



- 例一：

在今天的测试过程中，系统出现10次宕机现象

- 软件失效现象：发生宕机
- 软件故障情况：使用非法数据输入的情况下发生宕机
- 软件缺陷：软件中缺少合法性校验
- 软件错误：详细设计环节缺少合法性校验内容，且文档评审工作不到位
- 修改建议：增加合法性校验，加强文档评审工作

如何指出问题



- 例二：

软件在**500**用户并发情况下会发生大量交易失败

- 软件失效现象：不能承担**500**用户的并发交易
- 软件故障情况：在进行用户交纳月通话费的情况下交易失败
- 软件缺陷：服务器**CPU**占用率持续在**100%**
- 软件错误：系统设计环节未进行关键技术验证与仿真
- 修改建议：重新选择服务器（重点是**CPU**），加强对关键技术的验证与仿真工作

软件测试的目的



- 以最少的人力、物力和时间找出软件中潜在的各种错误和缺陷，通过修正种错误和缺陷提高软件质量，回避软件发布后由于潜在的软件缺陷和错误造成的隐患带来的商业风险。
- 测试是以评价一个程序或者系统属性为目标的一种活动，测试是对软件质量的度量与评估，以验证软件的质量满足用户的需求，为用户选择与接受软件提供有力的依据。
- 通过分析错误产生的原因还可以帮助发现当前开发工作所采用的软件过程的缺陷，以便进行软件过程改进。同时通过对测试结果的分析整理，还可以修正软件开发规则，并为软件可靠性分析提供依据。

软件测试的目的



验证用户需求



发现软件缺陷



改进开发过程

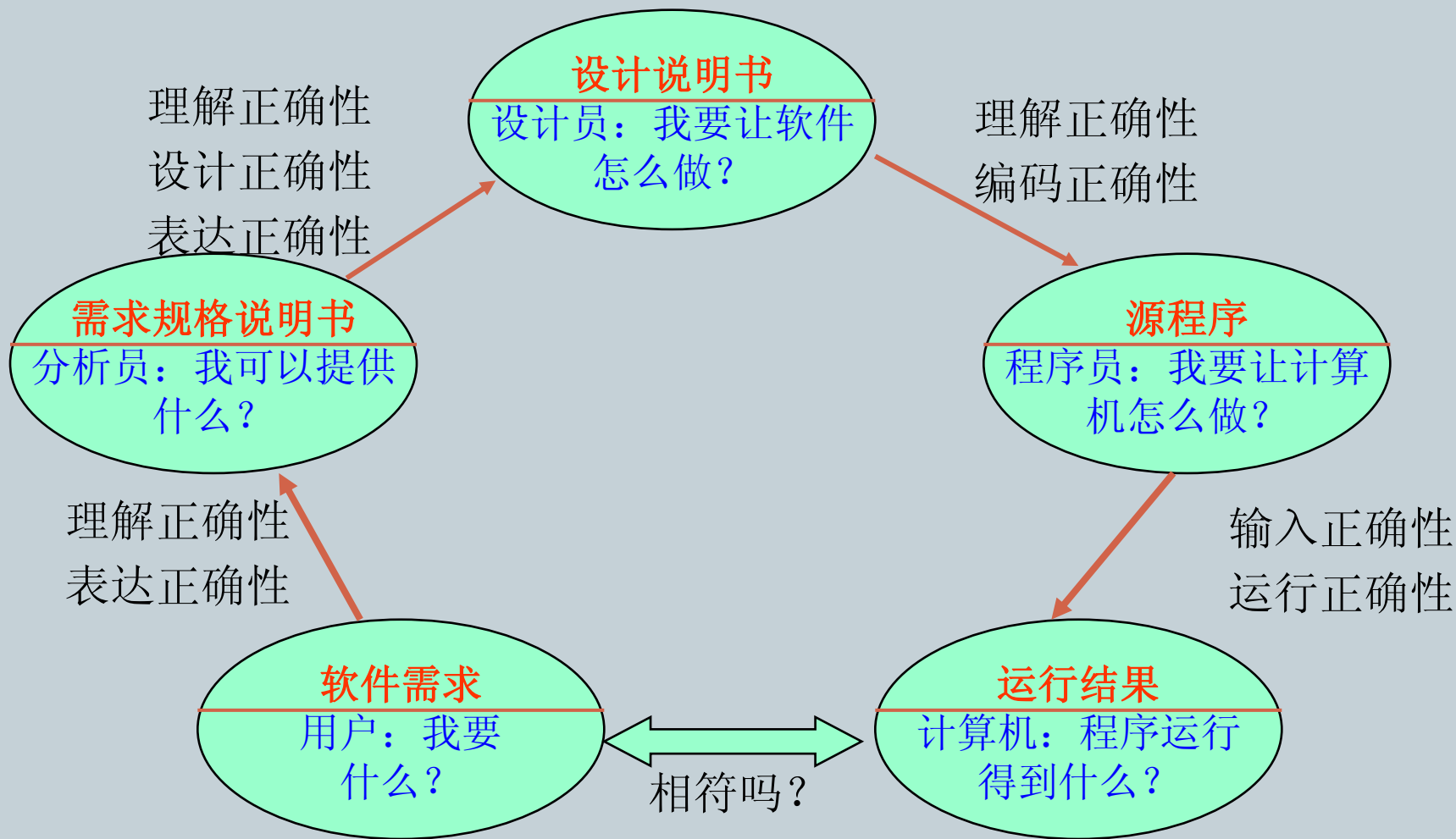
软件测试流程



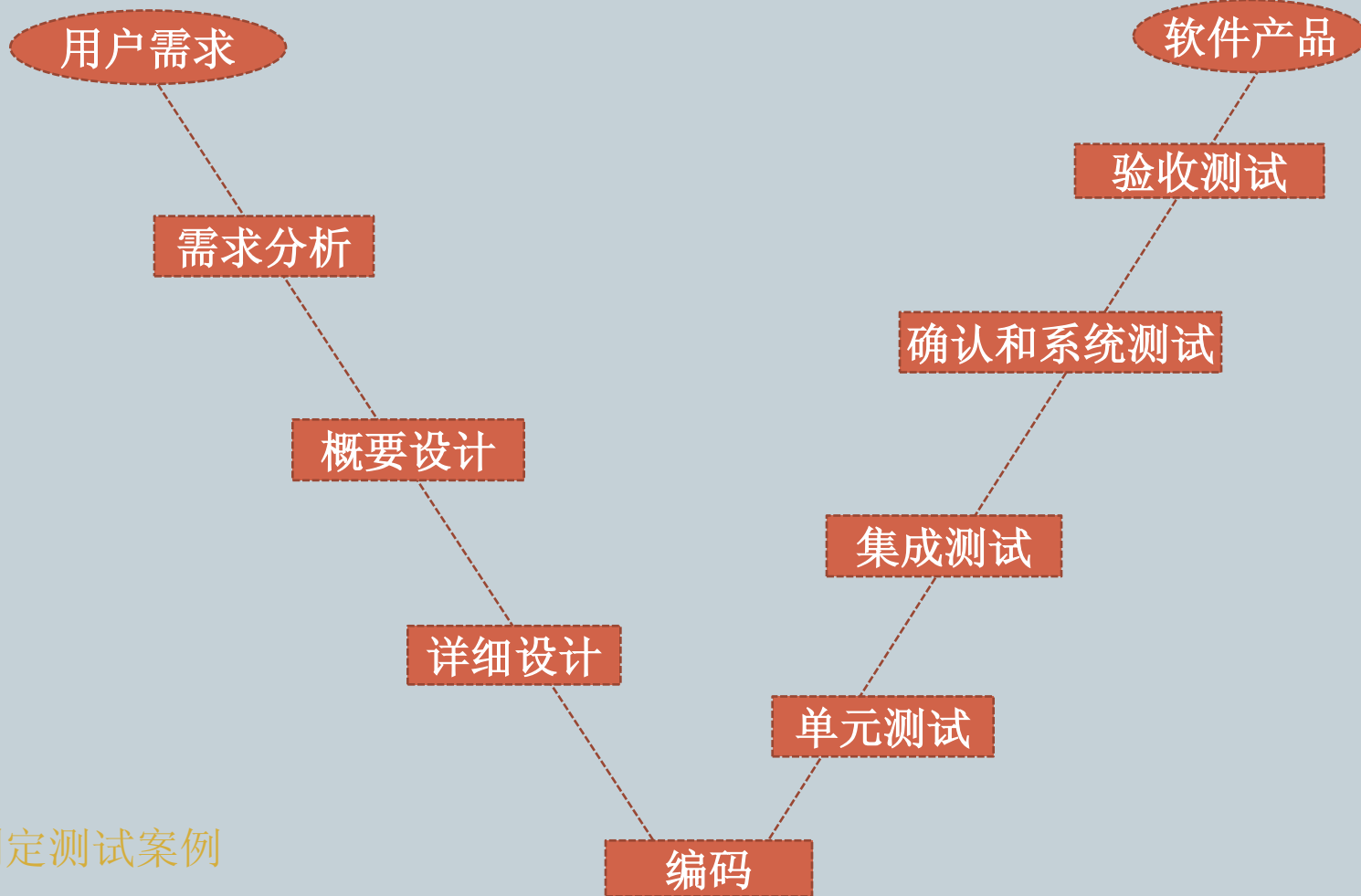
我们再来看看这个问题：

软件测试的流程是什么呢？

软件生存周期

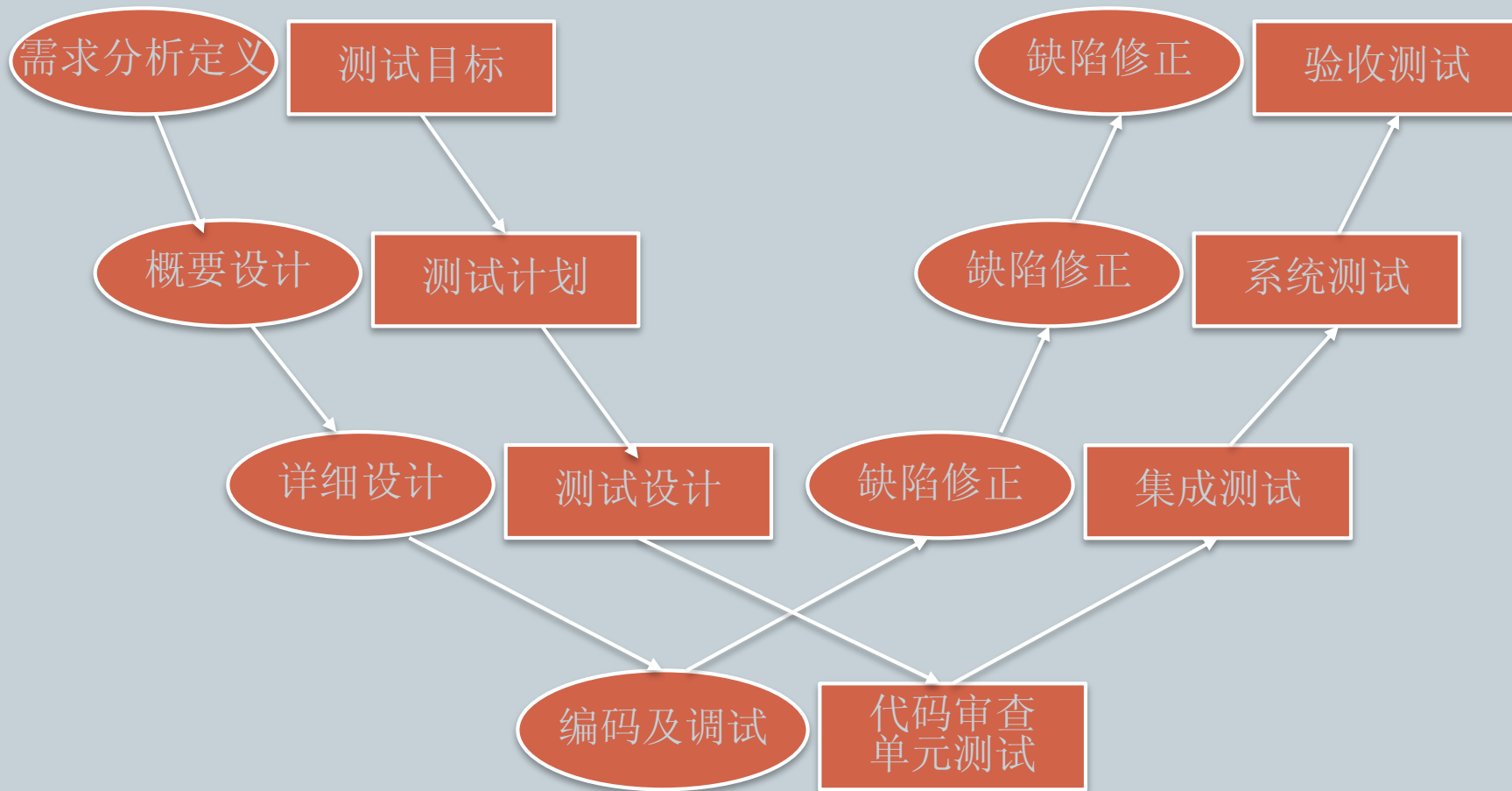


V模型

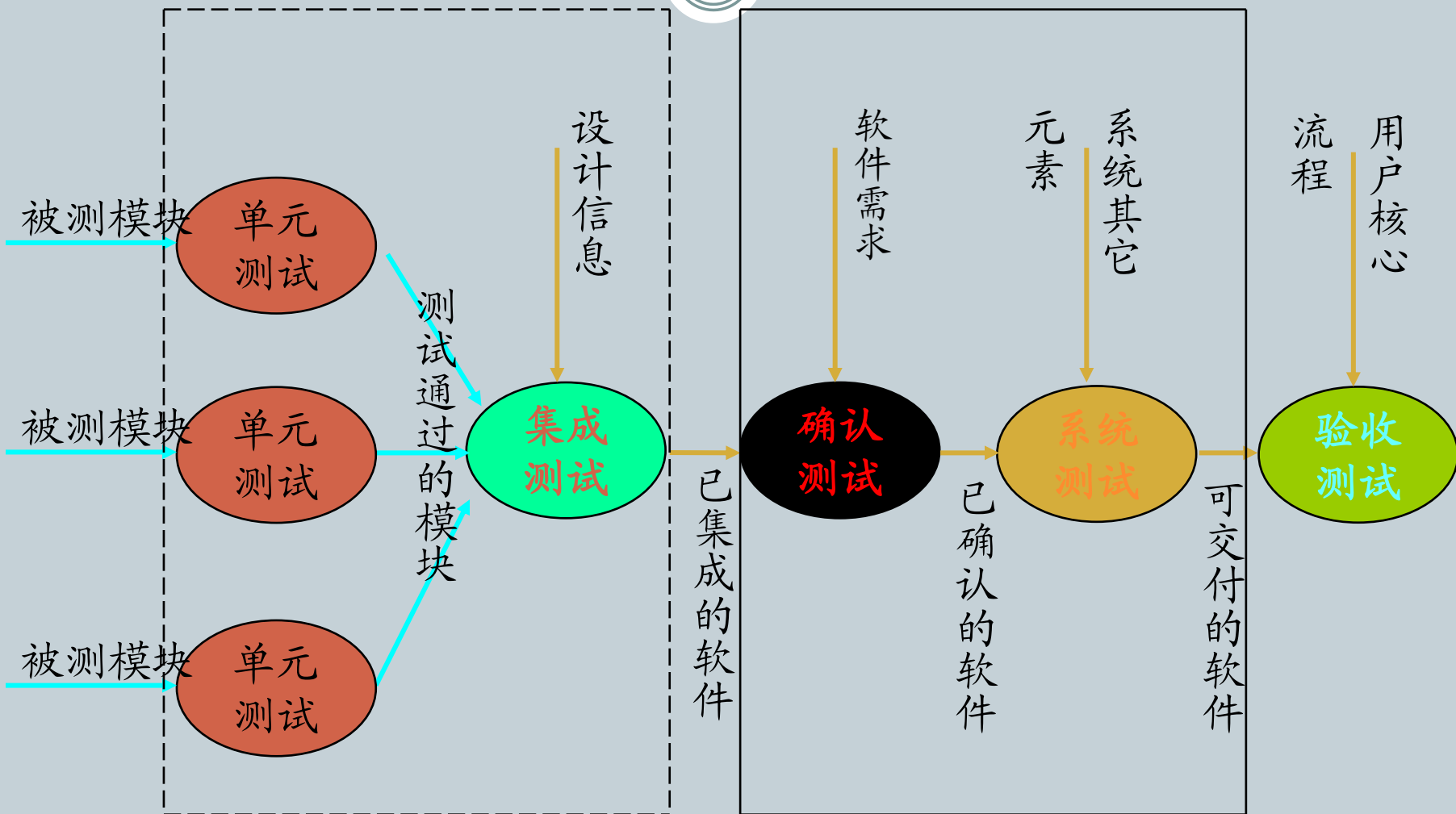


——制定测试案例

W模型



经典测试阶段



测试策略



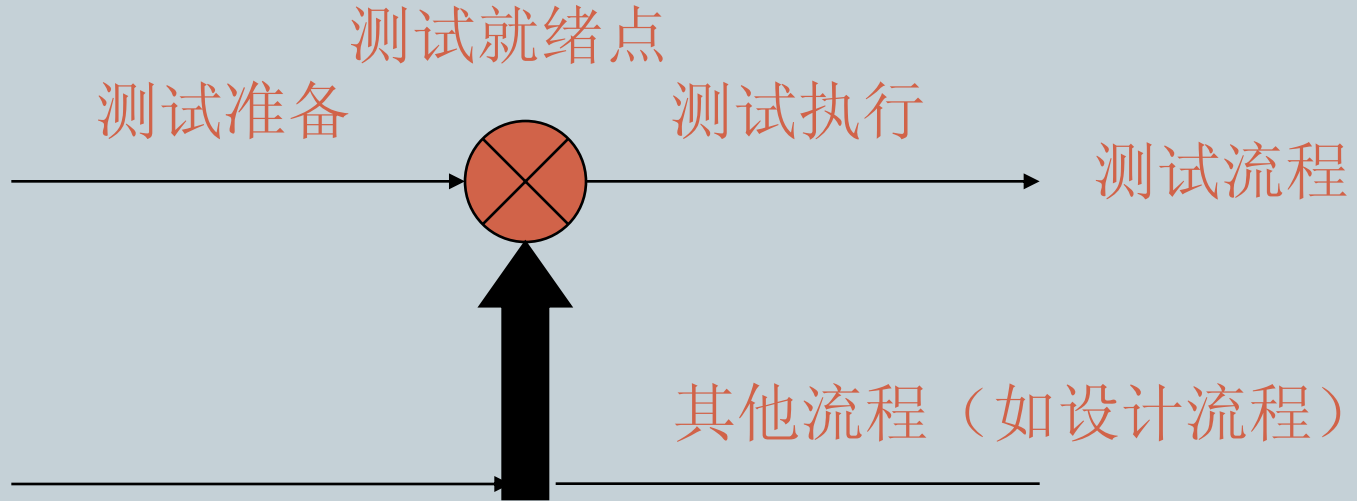
测试类型	对象	目的	依据	方法
单元测试	模块内部程序错误	消除局部模块的逻辑和功能上的错误和缺陷	详细设计	白盒为主，黑盒为辅
集成测试	模块间的集成和调用关系	找出与软件设计相关的程序结构，模块调用关系，模块间接口方面的问题	概要设计	白盒与黑盒结合
确认与系统测试	整个系统中的软硬件	对整个系统进行一系列的整体、有效性测试	系统设计说明书	黑盒为主
验收测试	整个系统中的软硬件	对整个系统进行一系列的整体、有效性测试	需求规格说明书	黑盒

H模型

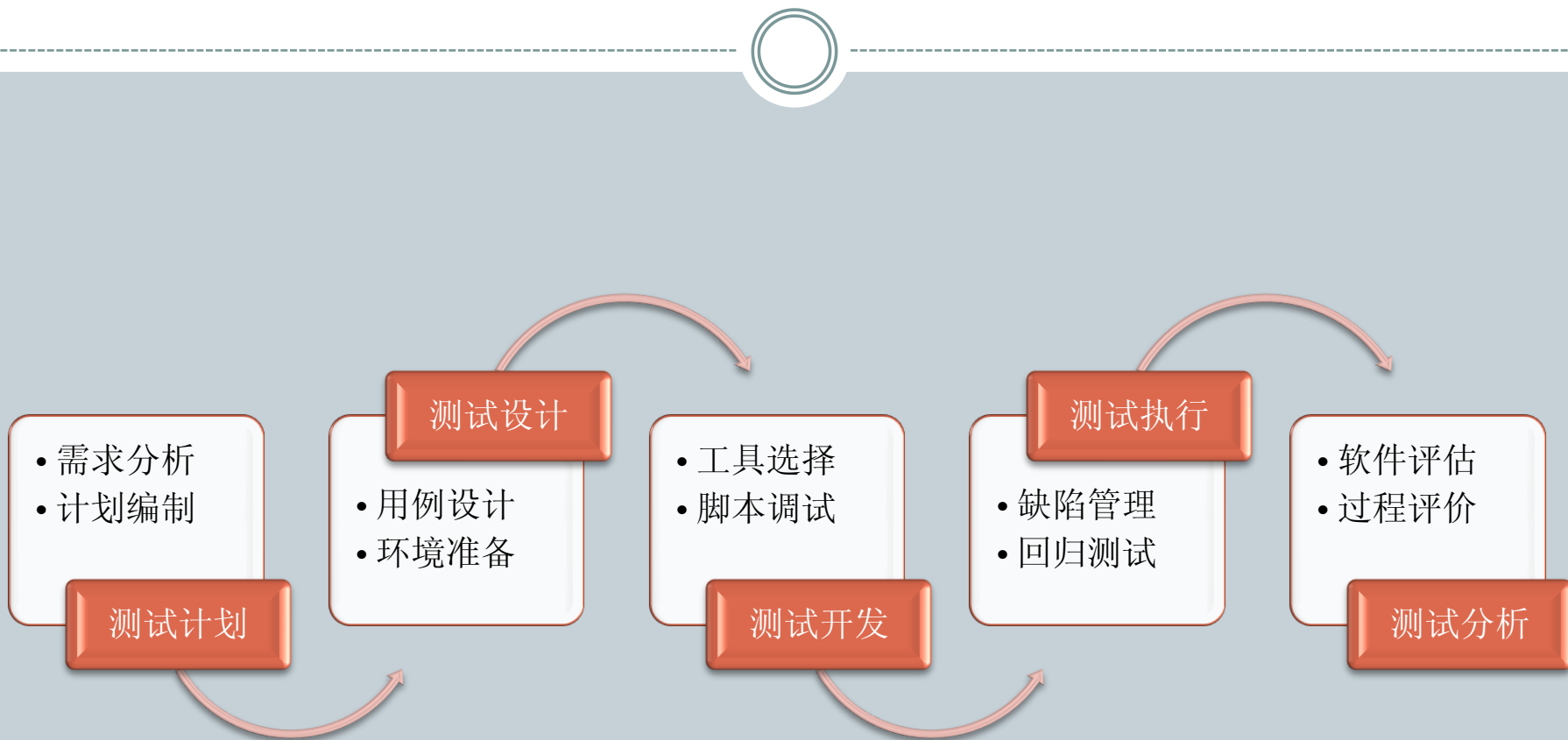


- 测试流程：

- 测试准备活动：测试计划、测试设计、测试开发。
- 测试执行活动：测试运行、测试评估。



测试流程图



测试计划

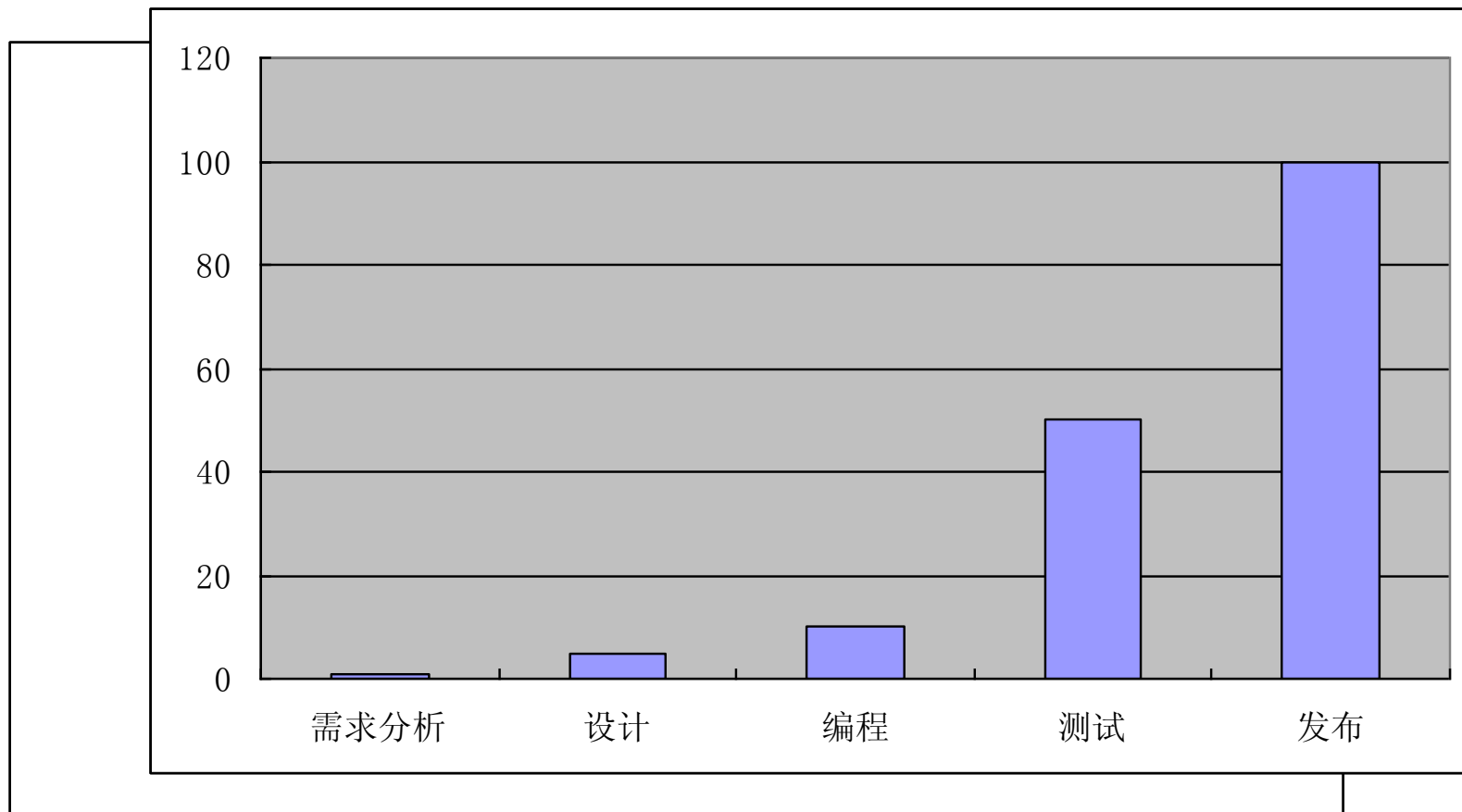


1. 产品需求文档审查和评审
2. 项目的测试需求和任务
3. 测试范围分析和工作量估计
4. 测试资源需求
5. 测试里程碑和进度安排
6. 测试风险分析
7. 制定测试策略
8. 测试计划书

1. 产品需求文档审查和评审



- 需求评审的重要性



1. 产品需求文档审查和评审



- 需求评审表现在
 - 发现需求定义中的问题，尽早将缺陷发现出来，降低成本；
 - 保证软件需求的可测试性；
 - 通过评审，使大家认识理解一致；
 - 通过评审，更好地理解产品的功能性和非功能性需求，为制定测试计划和方案打下基础；
 - 通过评审，明确测试目标和范围，有效控制测试变更，降低测试风险

1. 产品需求文档审查和评审



- 需求评审的质量因素
 - 正确性
 - 完备性
 - 易理解性
 - 一致性
 - 可行性
 - 易修改性
 - 易测试性
 - 易追溯性
- 附件：需求文档审查

2. 项目的测试需求和任务



- 软件功能测试需求，如：
 - 程序安装、启动正常，有相应的提示框、错误提示等
 - 每项功能符合实际要求
 - 每项功能正常运行，输出结果正确
 - 能处理各种不正常的操作，对异常数据可以进行提示、容错处理等
 - 系统的界面清晰、美观
 - 菜单、按钮操作正常、灵活，能处理一些异常操作
 - 能接受正确的数据输入，如测试最大输入的文字数、特殊符号等
 - 数据的输出结果准确、格式清晰，可以保存和读取
 - 功能逻辑清楚，符合使用者习惯
 - 系统的各种状态按照业务流程而变化，并保持稳定
 - 支持各种应用的环境
 - 能配合多种硬件周边设备
 - 软件升级后，能支持旧版本的数据
 - 与外部应用系统的接口有效

2. 项目的测试需求和任务



- 非功能性的系统测试需求

- 单机版系统

- B/S应用系统

- C/S应用系统

- 复杂应用系统

- ✦ 以功能为中心

- 可扩充性、可维护性

- ✦ 以数据库为中心

- 数据处理的性能、正确性和有效性

- ✦ 以流程为中心

- 灵活、可配置性、接口规范

2. 项目的测试需求和任务



- 非功能测试需求的例子：
 - 性能要求
 - ✦ 通过页面进行搜索，查询时间控制在5秒以内；
 - ✦ 支持50人同时登录。
 - 容错处理
 - ✦ 业务处理过程中中断时，系统能保存当前状态，在某时刻可以恢复操作；
 - 兼容性要求
 - ✦ 支持浏览器包括IE、FireFox、Mozilla等；
 - 安全性要求
 - ✦ **Cookie**和**Session**的有效期验证；
 - ✦ 数据访问权限设置；
 - ✦ 数据备份和恢复等。

2. 项目的测试需求和任务



- 软件即服务的测试需求

系统质量	说明
性能	指按用户负载条件对响应时间和吞吐量所作的度量
可用性	指对系统资源和服务可供最终用户使用的程度度量，通常以系统的正常运行时间来表示
可伸缩性	指随时间推移为部署系统增加容量（和用户）的能力。可伸缩性通常涉及向系统增加资源，但不应要求对部署体系结构进行更改
安全性	指对系统及其用户的完整性进行说明的复杂因素组合。安全性包括用户的验证和授权、数据的安全以及对已部署系统的安全访问
容量	指在不增加资源的情况下，系统处理异常峰值负载的能力。潜在容量是可用性、性能和可伸缩性特性中的一个因素
可维护性	指对已部署系统进行维护的难易程度，其中包括监视系统、修复出现的故障以及升级硬件和软件组件等任务

2. 项目的测试需求和任务



阶段	输入和要求	输出
需求分析审查	市场/产品需求定义、分析文档和相关技术文档。 要求：需求定义要准确、完整和一致，真正理解客户的需求	需求定义中的问题列表，批准的需求分析文档。测试计划书的起草
设计审查	产品规格设计说明、系统架构和技术设计文档、测试计划和测试用例。 要求：系统结构的合理性、处理过程的正确性、数据库的规范性、模块的独立性等。清楚定义测试计划的策略、范围、资源和风险，测试用例的有效性和完备性	设计问题列表。批准的各类设计文档、系统和功能的测试计划和测试用例。测试环境的准备
单元测试	源程序、编程规范、产品规格设计说明书和详细的程序设计文档。 要求：遵守规范、模块的高内聚性、功能实现的一致性和正确性	缺陷报告、跟踪报告，完善的测试用例、测试计划，对系统功能及其实现等了解清楚
集成测试	通过单元测试的模块或组件、编程规范、集成测试规格说明和程序设计文档、系统设计文档。 要求：遵守规范、模块的高内聚性、功能实现的一致性和正确性	缺陷报告、跟踪报告，完善的测试用例、测试计划，对系统功能及其实现等了解清楚
功能验证	代码软件包（含文档），功能详细设计说明书，测试计划和用例。 要求：模块集成功能的正确性、适用性	缺陷报告、代码完成状态报告、功能验证测试报告
系统测试	修改后的软件包、测试环境、系统测试用例和测试计划。 要求：系统能正常地、有效地运行，包括性能、可靠性、安全性、兼容性等	缺陷报告、系统性能分析报告、缺陷状态报告、阶段性测试报告
验收测试	产品规格设计说明、预发布的软件包、确认测试用例。 要求：向用户表明系统能够按照预定的要求那样工作，使系统最终可以正式发布或向用户提供服务。用户要参与验收测试，包括α测试（内部用户测试）β（外部用户测试）	用户验收报告、缺陷报告审查、版本审查、最终测试报告
版本发布	软件发布包、软件发布检查表（清单）	当前版本已知问题的清单、版本发布报告
维护	变更的需求、修改的软件包、测试用例和计划 要求：新的或增强的功能正常、原有的功能正常，不能出现的回归缺陷	缺陷报告、更改跟踪报告、测试报告

3. 测试范围分析和工作量估计



- 功能测试范围的分析
 - 功能子模块的划分
 - 业务流程的划分
 - 如：一般性的Web测试项目
 - ✦ 用户登录
 - ✦ 站点地图和导航条
 - ✦ 链接
 - ✦ 表单
 - ✦ 数据校验
 - ✦ Cookie
 - ✦ Session
 - ✦ SSL、防火墙等的测试
 - ✦ 接口测试

3. 测试范围分析和工作量估计



- 系统测试范围的分析
 - 容错处理
 - ✦ 对错误数据和异常情况的处理能力
 - 兼容性要求
 - ✦ 软件、硬件、数据等兼容
 - 配置要求
 - ✦ 不同配置参数的适应力
 - 性能要求
 - ✦ 负载测试、压力测试、疲劳测试
 - 安全性要求
 - ✦ 数据保护、权限限制、抗攻击能力
 - 可靠性要求
 - ✦ 长时间稳定运行

3. 测试范围分析和工作量估计



- 定义测试任务
 - 工作分解结构表方法（WBS）
 - ✦ 列出项目需要完成的各项任务，如测试计划、需求和设计评审、测试设计、脚本开发、测试执行等
 - ✦ 对每个任务进一步细化和分解
 - ✦ 根据任务的层次给任务进行编号

3. 测试范围分析和工作量估计



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

3. 测试范围分析和工作量估计



- 工作量估算因素
 - 效率
 - ✦ 人员效率、自动化水平
 - 测试动作
 - ✦ 测试动作数、测试用例时间
 - 阶段
 - ✦ 测试周期不同阶段（测试设计、测试开发、测试执行）
 - 复杂度
 - ✦ 测试需求的维数
 - 风险
 - ✦ 一般设定为工作量的10%—20%

3. 测试范围分析和工作量估计



- 测试工作量的估计

$$W = W_0 + W_0 \times R_1 + W_0 \times R_2 + W_0 \times R_3$$

- ✦ W 为总工作量， W_0 为一轮测试的工作量
- ✦ R_1 、 R_2 、 R_3 为每轮的递减系数

代码质量差， R_1 、 R_2 、 R_3 值为80%、60%、40%

代码质量好， R_1 、 R_2 、 R_3 值为60%、30%、0

3. 测试范围分析和工作量估计



- 工作量估算分解
 - 每个测试用例设计时间20—30分钟
 - 每日开发自动化测试脚本8—10个
 - 每个工程师每天运行50—60个测试用例
 - 每个测试用例执行5—8分钟

测试工作量举例分析

3. 测试范围分析和工作量估计

任务	工作量 (人日)
测试过程总计	67
1. 测试计划	8
1.1. 需求评审	3
1.2. 定义测试需求	5
1.3. 工作量分析	1
1.4. 确定资源	1
1.5. 编写测试计划文档	3
2. 设计测试用例	18
2.1. 分析测试用例结构	1
2.2. 测试用例设计方法讨论	2
2.3. 编写功能测试用例	10
2.4. 编写系统测试用例	3
2.5. 测试用例评审	2
3. 开发自动化测试脚本	12
3.1. 录制和编写测试脚本	6

任务	工作量 (人日)
3.2. 调试和修改测试脚本	3
3.3. 测试数据开发	3
4. 测试执行	25
4.1. 设置测试环境	1
4.2. 功能测试 (手工测试部分)	10
4.3. 功能测试 (自动化测试部分)	3
4.4. 系统测试	4
4.5. 性能测试	4
4.6. 缺陷记录、验证与跟踪	3
5. 测试分析与报告	4
5.1. 测试需求覆盖评估	1
5.2. 编写测试报告	2
5.3. 评审测试报告	1
5.4. 项目总结分析	1

4. 测试资源需求



主要测试资源

人力资源

- 测试管理人员
- 测试设计人员
- 测试开发人员
- 测试执行人员
- 测试配置人员

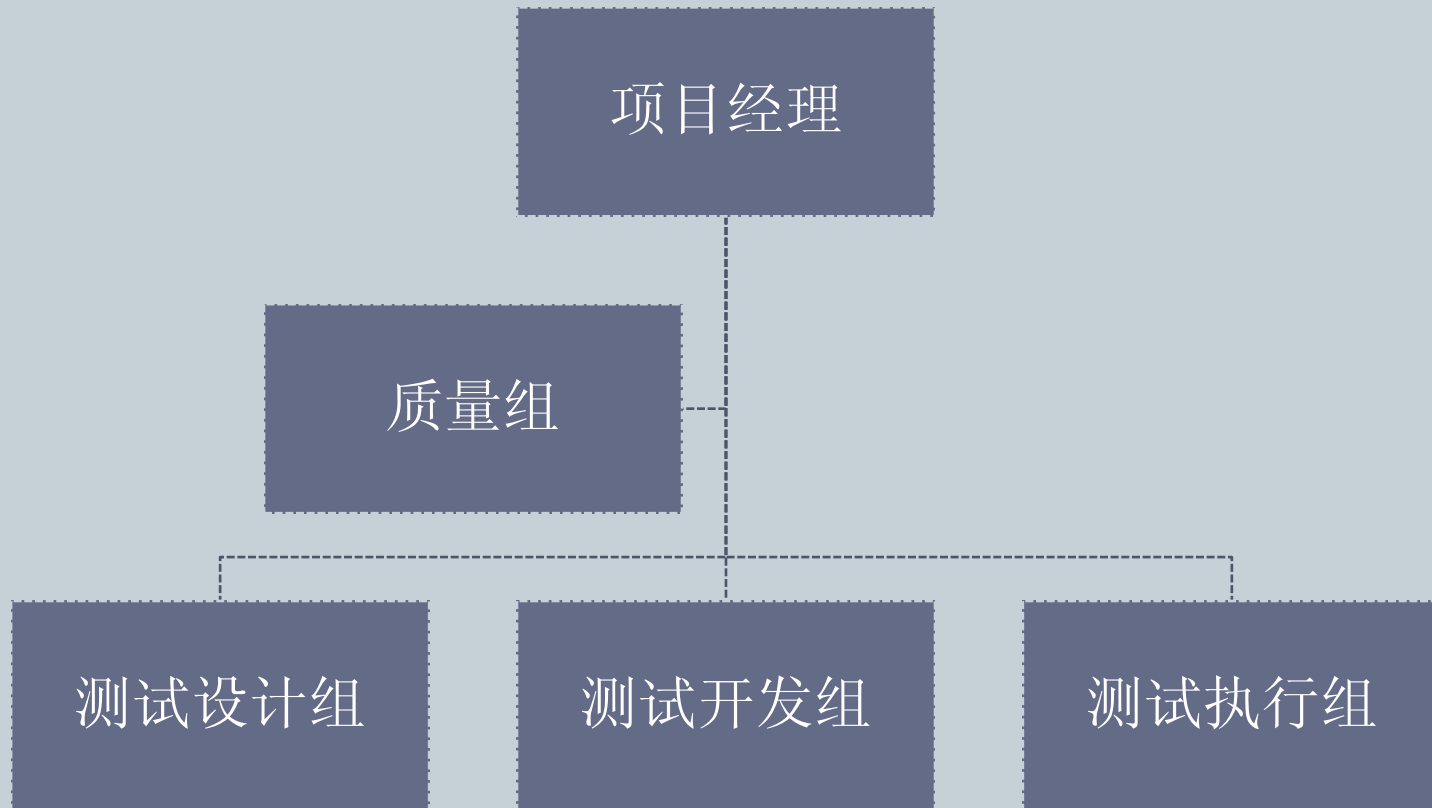
硬件资源

- 网络设备
- 主机设备
- 其他设备

软件资源

- 系统软件
- 第三方软件
- 测试工具

4. 测试资源需求



4. 测试资源需求



- 人员培训

- 帮助整个项目组建立对所有问题的共识，包括对测试需求的一致理解
- 每一个测试人员达到合格的水平和能力，保证测试工作的质量，包括牢记缺陷报告的具体要求
- 提高部分测试人员的测试设计和开发能力，保证测试用例、测试脚本的质量
- 培训就是一个知识传递、知识共享的过程，可及时更新大家的知识

培训的考核是必要环节，培训不和考核结合起来，不能起到相应的效果，也很难检查培训的效果

5. 测试里程碑和进度安排



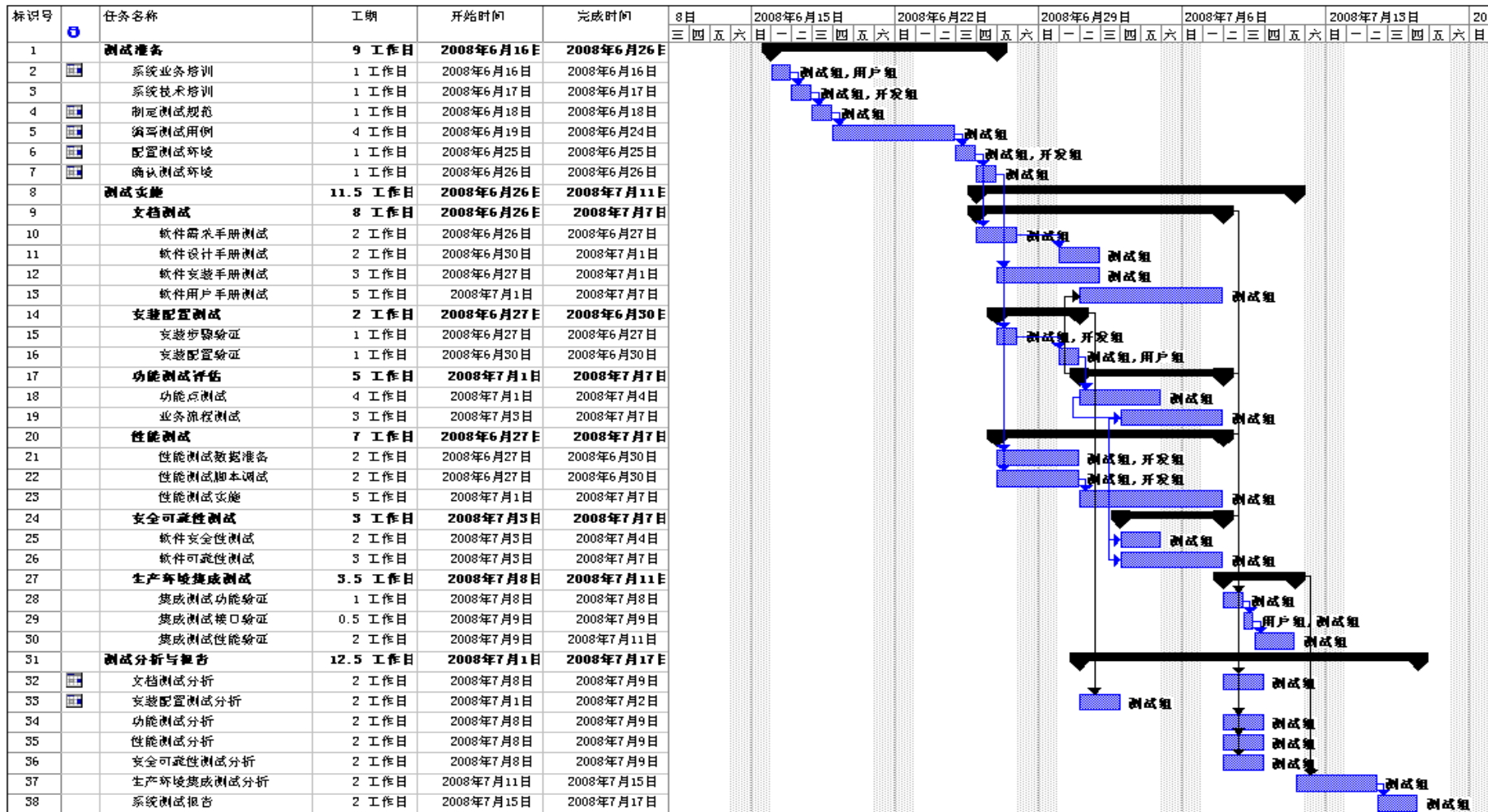
- 测试进度
 - 取决于测试目标和测试资源
- 测试里程碑
 - 项目中完成阶段性工作的标志，一个里程碑标志着上一个阶段结束、下一个阶段开始，也就是定义当前阶段完成的标准和新阶段启动的条件或前提
 - 里程碑也是有层次的，在一个父里程碑的下一个层次中定义子里程碑
 - 不同类型的项目，里程碑可能不同
 - 不同规模项目的里程碑，其数量的多少不一样，里程碑可以合并或分解

5. 测试里程碑和进度安排



- 测试里程碑
 - 测试计划
 - ✦ 测试需求分析
 - ✦ 测试计划制定
 - 测试设计
 - ✦ 测试用例设计
 - ✦ 测试脚本开发
 - 测试执行
 - ✦ 环境准备
 - ✦ 测试记录
 - 测试分析
 - ✦ 缺陷分析
 - ✦ 测试报告

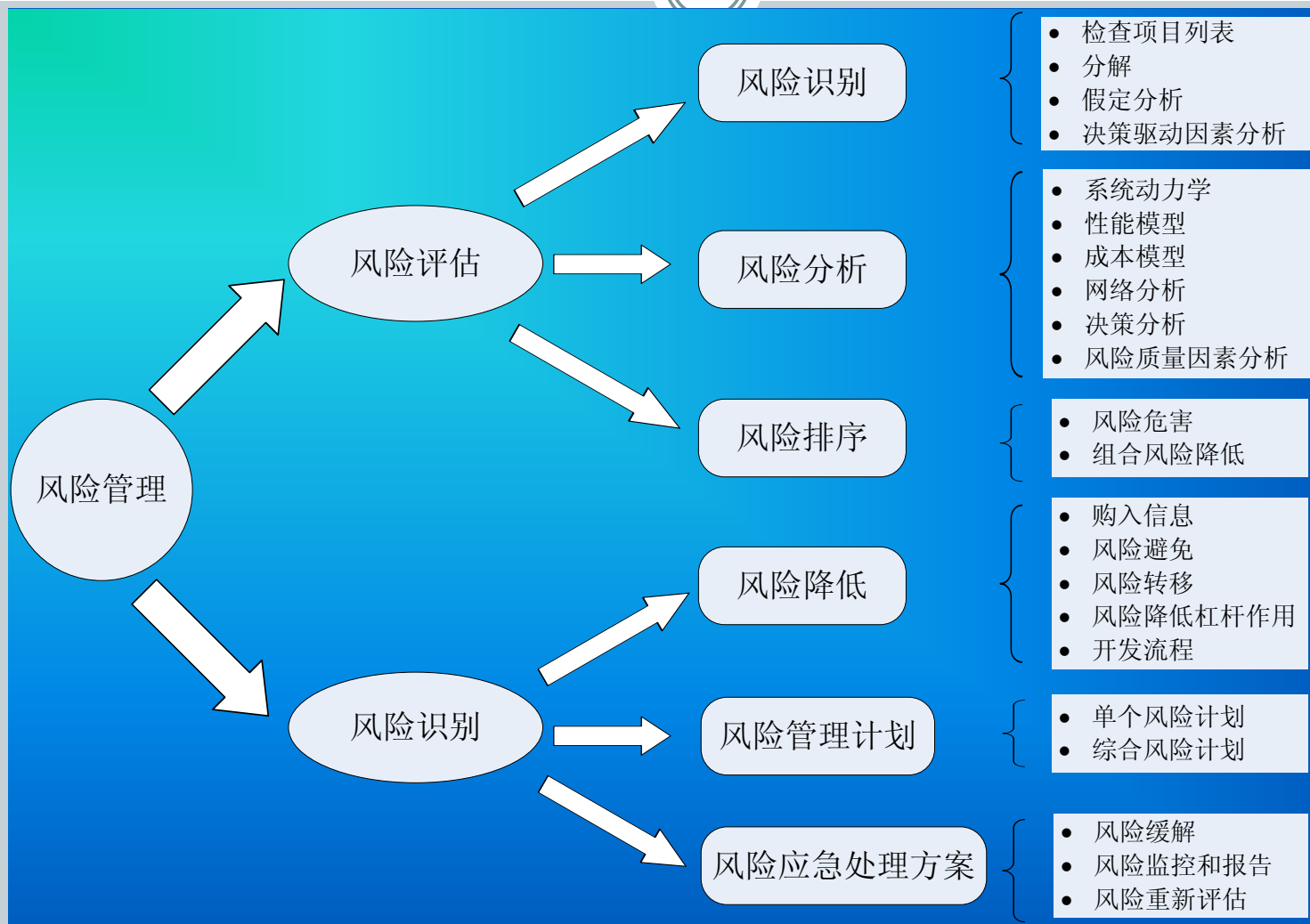
5. 测试里程碑和进度安排



项目：测试进度计划.mpp
日期：2008年6月15日

任务 进度 摘要 外部任务 期限
 拆分 里程碑 项目摘要 外部里程碑

6. 测试风险分析



6. 测试风险分析



- 软件测试风险类别
 - 人员风险
 - 环境风险
 - 测试广度
 - 测试深度
 - 回归测试
 - 需求变更
 - 用户期望
 - 测试技术
 - 测试工具

7. 制定测试策略



- 测试策略
 - 描述当前测试项目的目标和所采用的测试方法
- 试试回答如下问题
 - 如何利用可重复性的测试？
 - 回归测试的范围如何确定？
 - 如何收集衡量测试结果的指标？
 - 如何加强静态测试，如文档的审查？
 - 如何选择合理的、有效的测试数据集？
 - 如何处理单元测试和集成测试的关系？
 - 如何处理手工测试和自动化测试之间的平衡？
 - 如何建立稳定的、模拟系统实际运行的测试环境？

7. 制定测试策略



- 软件测试策略制定有**3**项基本要素
 - 输入，作为制定测试策略的依据，包括限制条件和已具有的资源
 - ✦ 所要求的软硬件的详细说明，包括测试环境、测试工具等
 - ✦ 人力资源和测试进度的约束，包括测试组成员的角色和职责说明
 - ✦ 测试方法和衡量测试是否通过的标准
 - ✦ 被测软件组件或系统的功能性和技术性需求文档，以及变更请求的控制流程
 - ✦ 软件系统所受的其他限制
 - 输出，制定策略的成果，即最终对所制定策略的定义或说明
 - ✦ 通过/失败的准则和测试风险评估的结果
 - ✦ 已批准和签署的测试策略文档
 - ✦ 和测试策略相对应的测试计划、测试用例的设计思想和思路
 - 制定策略的过程

7. 制定测试策略



- 测试策略制定原则
 - 双重视角、网状覆盖
 - 抓住核心、主次分明
 - 专家组合、协同测试
 - 分门别类、强调分工
 - 适时回归、螺旋上升

8. 测试计划书



- 目标和范围
 - 产品特性、质量目标、各阶段测试对象、目标、范围、限制
- 项目估算
 - 测试工作量、所需资源
- 风险计划
 - 风险识别、风险回避、监控和管理
- 进度安排
 - 分解项目工作结构、采用甘特图等制定时间、资源表
- 资源配置
 - 人员、硬件、软件等资源的组织和分配
- 跟踪和控制机制
 - 质量保证和控制、配置管理

8. 测试计划书



- GB8567-88 测试计划
- CSTC 测试计划