



Sponsors of Tomorrow.™

软件测试概述

程绍银

sycheng@ustc.edu.cn





引言

生活中有测试（质检）的例子？





引言

- ❑ 随着计算机技术的飞速发展，计算机系统的规模和复杂性急剧增加，其软件开发成本以及由于软件故障而造成的经济损失也正在增加，**软件质量问题**已成为人们共同关注的焦点
- ❑ **软件是信息系统的灵魂**，**软件安全是确保信息安全的前提**，**软件测试是保障软件安全的重要手段**
- ❑ 软件测试是对软件需求分析、设计规格说明和编码的最终复审，是软件质量保证的关键步骤
- ❑ 随着软件系统规模和复杂性的增加，进行专业化高效软件测试的要求越来越严格，软件测试职业的价值逐步得到了认可，软件测试技术已作为一门新兴产业而快速发展起来了



大纲

- ❑ 软件测试背景
- ❑ 软件缺陷与软件测试
- ❑ 软件测试过程
- ❑ 软件测试与软件开发的关系
- ❑ 软件测试工具



大纲

✓ 软件测试背景

- ❏ 软件缺陷与软件测试
- ❏ 软件测试过程
- ❏ 软件测试与软件开发的关系
- ❏ 软件测试工具



Bug这个词背后的故事

- ❏ 把缺陷称为bug?
- ❏ 故事发生在1945年9月的一天，一个炎热的下午，机房是一间第一次世界大战时建造的老建筑，没有空调，所有窗户都敞开着
- ❏ Hopper正领着她的研究小组夜以继日地工作，研制一台称为“MARK II”的计算机，它使用了大量的继电器（电子机械装置，那时还没有使用晶体管），一台不是纯粹的电子计算机
- ❏ 突然，MARK II死机了。。。



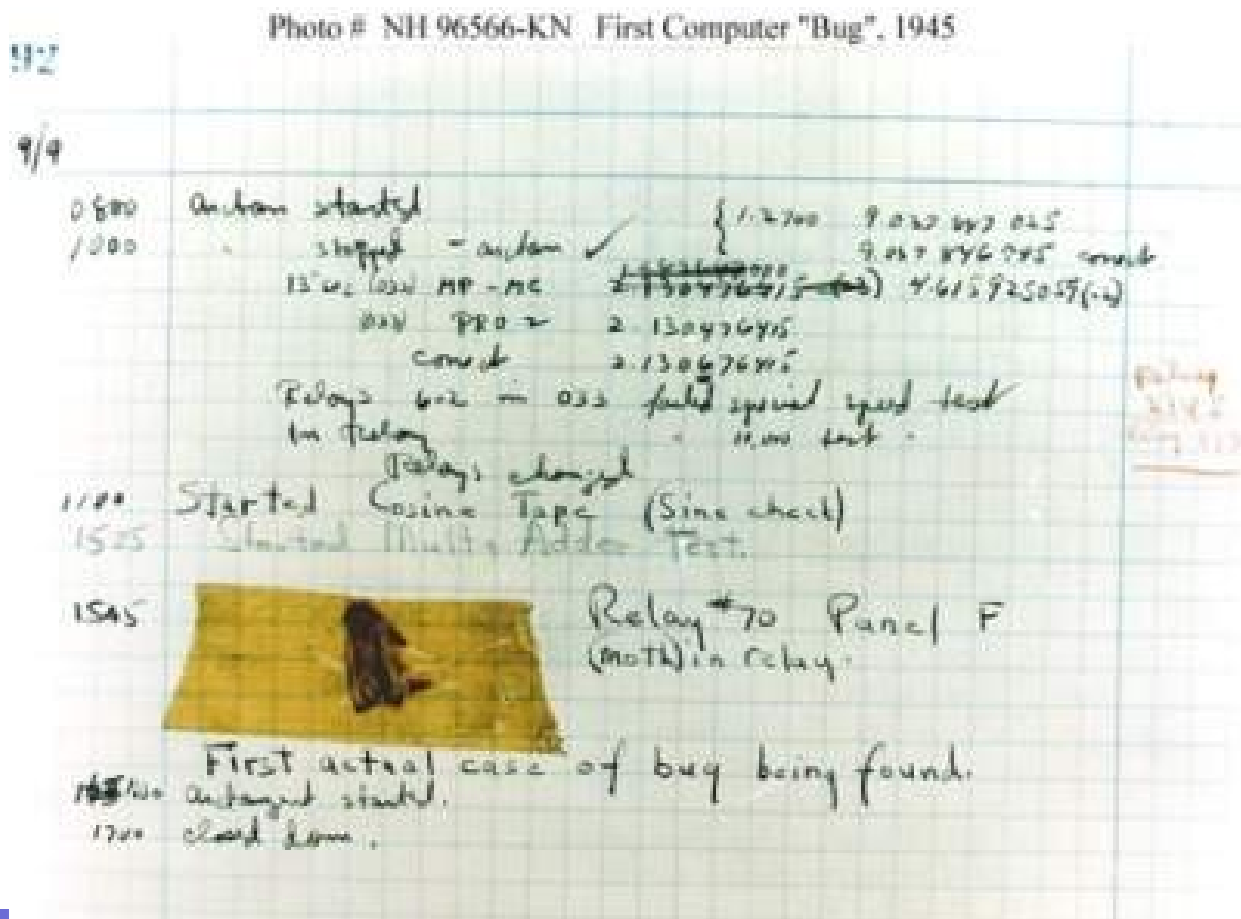
Bug这个词背后的故事

- ❏ 研究人员试了很多次还是启动不来，然后就开始用各种方法找问题，看问题究竟出现在哪里，最后定位到板子F第70号继电器出错
- ❏ Hopper观察这个出错的继电器，惊奇地发现一只飞蛾躺在中间，已经被继电器打死。她小心地用镊子将蛾子夹出来，用透明胶布帖到“事件记录本”中，并注明“第一个发现虫子的实例”，然后计算机又恢复了正常
- ❏ 从此以后，人们将计算机错误戏称为臭虫（Bug），而把找寻错误的工作称为“找臭虫”（Debug）



Bug 文物

- Grace Hopper的事件记录本，连同那个飞蛾，现在都陈列在美国历史博物馆中





奔腾处理器浮点除bug

- ❏ $(4195835 / 3145727) * 3145727 - 4195835 \neq 0$
- ❏ 测试工程师发现了这个bug，但是管理层认为这个bug不严重，不需要修复
- ❏ 一个用户于10/30/1994报告了这个bug，英特尔企图通过新闻发布和公开声明来掩饰这个问题的严重性。要求用户证明自己受到这个bug影响才能更换芯片
- ❏ 最后，英特尔不得不公开道歉并拿出4.5亿美元来更换所有问题芯片



爱国者导弹防御系统





爱国者导弹防御系统

- ❑ 1991年，首次应用在海湾战中对抗伊拉克的飞毛腿导弹的防御战争中
- ❑ 它在几次对抗导弹战役中失利，其中一枚在沙特阿拉伯的多哈击毙28名美国士兵
- ❑ 症结在于一个软件缺陷，一个很小的系统时钟错误累积超过14小时后，就失去准确度。多哈袭击战中，系统持续运行了100多小时



美国航天局火星基地登陆

- ❑ 1999年，登陆飞船在试图登陆火星表面时失踪。原因在于为了省钱，使用廉价的触点开关代替昂贵的雷达装置
- ❑ 理论上，飞船离地面1800米时，将丢弃降落伞，点燃登陆推进器，缓缓降落地面，直到飞船脚上的触点开关着地后关闭推进器
- ❑ 错误修正委员会事后发现，机械震动也会触发着地开关，结果就是飞船关闭推进器下落，摔成碎片

美国航天局火星基地登陆

■ 背后原因很简单

- ▶ 计算机中设置了一个数据位来关闭燃料
- ▶ 一个小组测试飞船的脚落地过程，不去注意着地数据位是否置位
- ▶ 另一个小组测试此后的着陆过程，测试前总是会重置计算机，清除数据位
- ▶ 双方独立工作都很好，但从未联合起来





“千年虫”问题

- ❑ 20世纪70年代程序员通过只使用2位数字来表示年份来节约宝贵的内存和磁盘空间
- ❑ 他们从未想到他们的程序会运行到2000年以后
- ❑ 时间过去了，很多遗产软件仍然在使用，用来解决千年错误的费用已超过数亿美元



AT&T长途电话服务崩溃

- 千年虫是人们预料到的可能爆发的问题，却没有大规模发作；而1990年美国电话电报公司(AT&T)的软件故障却是没有人预料到的



AT&T长途电话服务崩溃

- ❑ AT&T旗下的MaBell公司拥有世界上最大、最知名的性能可靠的电话网络系统，即使飓风和地震都不能对它造成损坏
- ❑ 然而在1990年1月15日，AT&T的114个交换中心中，**有一个开关发生了很小的机械故障**，正是这点小故障却立刻导致了该中心的瘫痪
- ❑ 故障发生后，当工作人员**启动备份系统时，系统却发出了使其他交换中心掉线并重置的信号**，而且其他中心也随之向外发送类似的信号



AT&T长途电话服务崩溃

- ❑ 事后一位作家在他的文章中描述说，那天系统全部崩溃，“就像一百多个摔跤选手挤在一个小台子上一样”，每一个人站起来都要把别人拉到
- ❑ 美国航空公司估计他们在这次事故中损失了二十多万个预定电话，而哥伦比亚广播公司甚至都不能到其本地电话局查询损失的情况
- ❑ 最后调查证明，这次事故的罪魁祸首是一个复杂软件的升级版本中一行错误的代码，AT&T进行此次升级的目的是要加快电话拨叫的速度。



迪士尼的狮子王

■ 1994年上映；2011年3D版上映





迪斯尼的狮子王

- ❑ 1994年秋，迪斯尼发布了第一个面向儿童的多媒体光盘游戏Lion King Animated Storybook
- ❑ 因为首次进军这个市场，大力的宣传使得销售额非常客观，称为那个夏季“必买游戏”
- ❑ 很快，客服电话淹没在愤怒的家长 and 哭诉玩不成游戏的孩子们的狂潮中
- ❑ 后来证实，软件只在少数系统中工作正常（开发游戏的系统），但在大众系统中不行

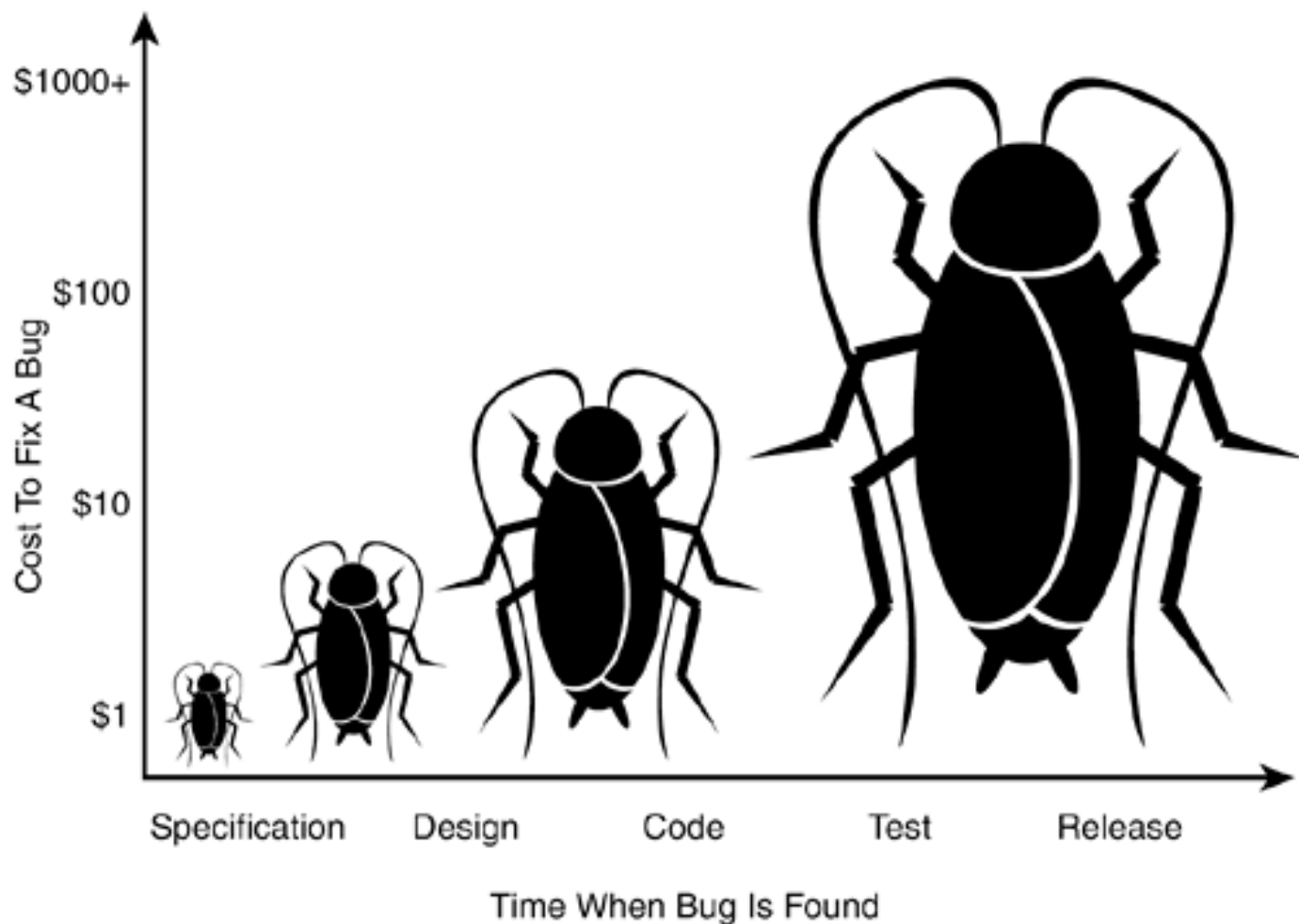


为什么要进行软件测试？

- ❑ 为什么要进行软件测试？就是因为软件缺陷的存在。因为**只有通过测试，才可以发现软件缺陷**。也只有发现了缺陷，才可以将软件缺陷从软件产品或软件系统中清理出去
- ❑ 软件中存在的缺陷给我们带来的损失是巨大的，这也说明了软件测试的必要性和重要性



为什么要进行软件测试？





大纲

- ❏ 软件测试背景
- ✓ 软件缺陷与软件测试
- ❏ 软件测试过程
- ❏ 软件测试与软件开发的关系
- ❏ 软件测试工具



什么是软件缺陷

■ IEEE国际标准729给出了软件缺陷的定义

- ▶ 软件缺陷就是软件产品中所存在的问题，最终表现为用户所需要的功能没有完全实现，不能满足或不能全部满足用户的需求

任何程序、系统中的问题，和产品设计书的
的不一致性， 不能满足用户的需求



软件缺陷的表现

- ❑ 根据**软件缺陷**的定义，可以从两方面考虑
 - ▶ **产品内部**：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
 - ▶ **外部**：系统所需要实现的**某种功能的失效或违背**
- ❑ 软件缺陷的主要类型/现象
 - ▶ 功能、特性没有实现或部分实现
 - ▶ 设计不合理，存在缺陷
 - ▶ **实际结果和预期结果不一致**
 - ▶ **运行出错**，包括运行中断、系统崩溃、界面混乱
 - ▶ 数据结果不正确、精度不够
 - ▶ 用户不能接受的其他问题，如存取时间过长、界面不美观



软件缺陷的产生

❏ 技术问题

- ▶ 算法错误，语法错误，计算和精度问题，接口参数传递不匹配

❏ 团队工作

- ▶ 误解、沟通不充分

❏ 软件本身

- ▶ 文档错误、用户使用场合
- ▶ 时间上不协调、或不一致性所带来的问题
- ▶ 系统的自我恢复或数据的异地备份、灾难性恢复等问题



缺陷 - bug

- | | |
|----------------|--------------------|
| ❑ 缺点 (defect) | 偏差 (variance) |
| ❑ 故障 (fault) | 失败 (failure) |
| ❑ 问题 (problem) | 矛盾 (inconsistency) |
| ❑ 错误 (error) | 事件 (incident) |
| ❑ 异常 (anomaly) | 特殊 (feature) |





软件错误、缺陷、故障与失效

- ❑ 软件**错误**：软件生存期内的**人为错误**，导致软件缺陷产生。是人为过程，相对于软件本身是外部行为
- ❑ 软件**缺陷**：存在于**软件**（文档、数据、程序）**中的偏差**，导致软件在某个特定条件下出现故障，这时称软件缺陷被激活
- ❑ 软件**故障**：软件运行过程中出现的**不希望或不可接收的内部状态**。是动态行为
- ❑ 软件**失效**：软件运行时产生的**不希望或不可接受的外部行为结果**



软件错误、缺陷、故障与失效

- ❑ 软件错误是一种人为错误。一个软件错误必定产生一个或多个软件缺陷
- ❑ 当一个软件缺陷被激活时，便产生一个软件故障；同一个软件缺陷在不同条件下被激活，可能产生不同的软件故障
- ❑ 软件故障如果没有及时的容错措施加以处理，便不可避免地导致软件失效；同一个软件故障在不同条件下可能产生不同的软件失效



漏洞

- ❏ 漏洞是在**硬件、软件、协议的具体实现或系统安全策略上存在的缺陷**，从而可以使攻击者能够在**未授权的情况下访问或破坏系统**
- ❏ 具体举例来说，比如在Intel Pentium芯片中存在的逻辑错误，在Sendmail早期版本中的编程错误，在NFS协议中认证方式上的弱点，在Unix系统管理员设置匿名Ftp服务时配置不当的问题都可能被攻击者使用，威胁到系统的安全。因而这些都可以认为是系统中存在的**安全漏洞**
- ❏ 漏洞又叫**脆弱性(Vulnerability)**



软件测试学科的发展

- ❑ 软件测试是伴随着软件的产生而产生的
- ❑ 早期的软件开发过程中，**测试**的含义比较狭窄，将测试等同于“**调试**”，目的是纠正软件中已经知道的故障，**常常由开发人员自己完成这部分的工作**。对测试的投入极少，测试介入也晚，常常是等到形成代码，产品已经基本完成时才进行测试
- ❑ 直到**1957年**，**软件测试才开始与调试区别开来**，作为一种发现软件缺陷 的活动



软件测试学科的发展

■ 从测试的思想导向来划分为4个阶段:

- ▶ 1957~1978年, 以**功能验证**为导向, 测试是证明软件是**正确的** (正向思维)
- ▶ 1978~1983年, 以**破坏性**为为导向, 测试是为了找到软件中的**错误** (逆向思维)
- ▶ 1983~1987年, 以**质量评估**为导向, 测试是提供产品的评估和质量度量
- ▶ 1988年起, 以**缺陷预防**为导向, 测试是为了展示软件符合设计要求, 发现缺陷、预防缺陷



更好的阶段划分

■ 分为3个阶段——初期阶段、发展阶段和成熟阶段

- ▶ 初级阶段（1957～1971）测试通常被认为是对产品进行**事后检验**，缺乏有效的测试方法
- ▶ 发展阶段（1972～1982），1972年第一次关于软件测试的**正式会议**，促进了软件测试的发展
- ▶ 成熟阶段（1983到现在），国际标准Std 829-1983，形成一门**独立的学科和专业**，成为软件工程学科中的一个重要组成部分



软件工程成为一级学科

■ 学位[2011]11号：关于印发《学位授予和人才培养学科目录（2011年）》的通知

■ 08 工学

▶ 0835 软件工程

新增

- ▮ 083501 软件工程理论与方法
- ▮ 083502 软件工程技术
- ▮ 083503 软件服务工程
- ▮ 083504 领域软件工程



软件工程的二级学科

- ❑ 083501 **软件工程理论与方法**：在计算机科学和数学等基本原理的基础上，研究大型复杂软件开发、运行和维护的**理论和方法**，以及**形式化方法**在软件工程中的应用，主要包括软件语言、形式化方法、软件自动生成与演化、**软件建模与分析**、软件智能化理论与方法等内容
- ❑ 083502 **软件工程技术**：研究大型复杂软件开发、运行与维护的原则、方法、技术及相应的**支撑工具、平台与环境**，主要包括软件需求工程、软件设计方法、软件体系结构、模型驱动开发、**软件分析与测试**、软件维护与演化、软件工程管理以及软件工程支撑工具、平台与环境等内容
- ❑ 083503 **软件服务工程**：研究软件服务工程原理、方法和技术，构建**支持软件服务系统的基础设施和平台**，主要包括软件服务系统体系结构、软件服务业务过程、软件服务工程方法、软件服务运行支撑等内容
- ❑ 083504 **领域软件工程**：研究软件工程在具体领域中的应用，并在此基础上形成**面向领域**的软件工程理论、方法与技术，主要包括领域分析、领域设计、领域实现、应用工程等内容



信息安全呢？

▣ 学位[2011]11号：关于印发《学位授予和人才培养学科目录（2011年）》的通知

▣ 08 工学

▶ 0837 安全科学与工程

新增

▶ 0812 计算机科学与技术（可授工学、理学学位）

▮ 081204 计算机网络与信息安全

- 研究计算机网络设计与实现和保障网络环境下信息系统安全的学科
- 主要研究计算机网络体系结构；研究计算机网络传输、交换和路由技术；研究计算机网络管理与优化技术；研究以计算机网络为平台的计算技术；研究计算机网络环境下信息的保密性、完整性、可用性和可追溯性
- 主要目标是合理地将网络设备、安全设备、计算机系统、应用系统组成计算机网络，配以安全管理系统，满足应用对网络性能、可靠性和安全性的要求



什么是软件测试

- ❏ 定义1： 1983年IEEE提出的软件工程标准术语中给软件测试下的定义是：“使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别”
- ❏ 定义2： 软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例，并利用这些测试用例去执行程序，以发现软件故障的过程。该定义强调寻找故障是测试的目的
- ❏ 定义3： 软件测试是一种软件质量保证活动，其动机是通过一些经济有效的方法，发现软件中存在的缺陷，从而保证软件质量



软件测试的正面性

Bill Hetzel博士（正向思维的代表）：

- ❑ 软件测试就是为程序能够按预期设想那样运行而建立足够的信心
- ❑ “软件测试是一系列活动以评价一个程序或系统的特性或能力并确定是否达到预期的结果”
- ❑ 测试是为了验证软件是否符合用户需求，即验证软件产品是否能正常工作



软件测试的反面性

Glenford J. Myers （反向思维的代表）：

- ❑ 测试是为了证明程序有错，而不是证明程序无错误
- ❑ 一个好的测试用例是在于它能发现至今未发现的错误
- ❑ 一个成功的测试是发现了至今未发现的错误的测试



软件测试的两面性

软件测试

正向思维—
验证软件正常工作

评价一个程序或系统的特性或能力并确定是否达到预期的结果

在设计规定的环境下运行软件的所有功能，直至全部通过

逆向思维—
假定软件有错误

测试是为发现错误而针对某个程序或系统的执行过程

寻找容易犯错误的地方和系统的薄弱环节，试图破坏系统，直至找不出问题



验证和确认 (V & V)

- Verification: Are we building the product right?
- 是否正确地构造了软件？即是否正确地做事，验证开发过程是否遵守已定义好的内容。验证产品满足规格设计说明书的一致性
- Validation: Are we building the right product?
- 是否构造了正是用户所需要的软件？即是否正在做正确的事。验证产品所实现的功能是否满足用户的需求



软件测试的两个重要问题

■ 测试(例)生成

- ▶ 如何设计测试输入和相应的测试输出？

■ 测试充分性评价

- ▶ 已做的测试是否充分？测试何时终止？

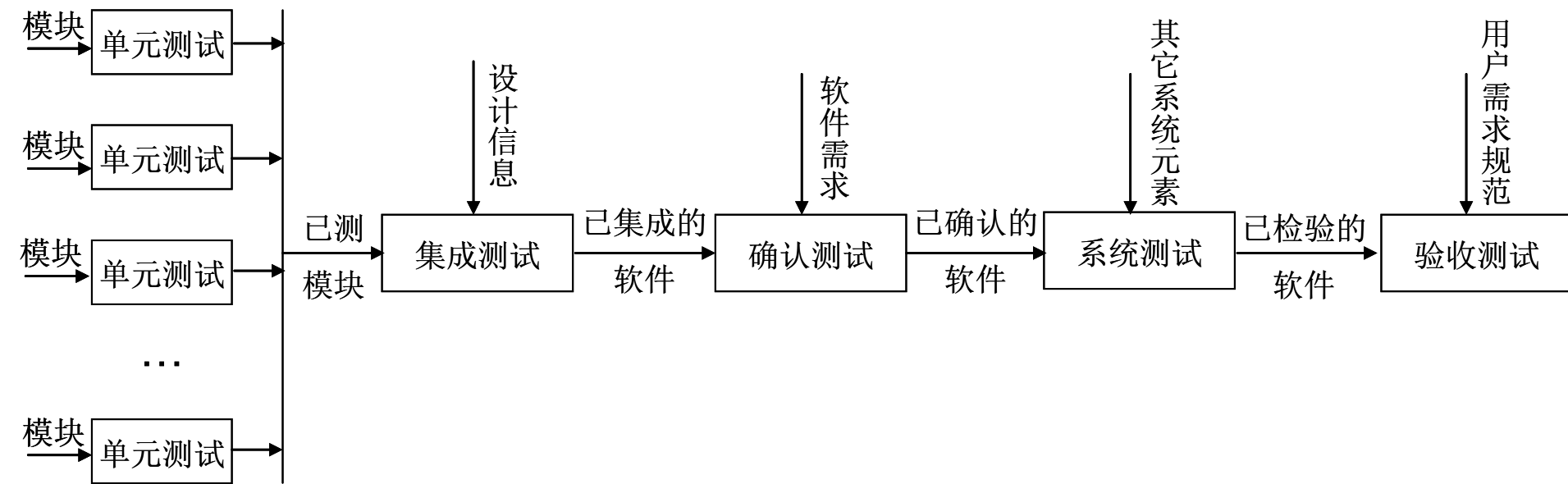


大纲

- ❑ 软件测试背景
- ❑ 软件缺陷与软件测试
- ✓ 软件测试过程
- ❑ 软件测试与软件开发的关系
- ❑ 软件测试工具

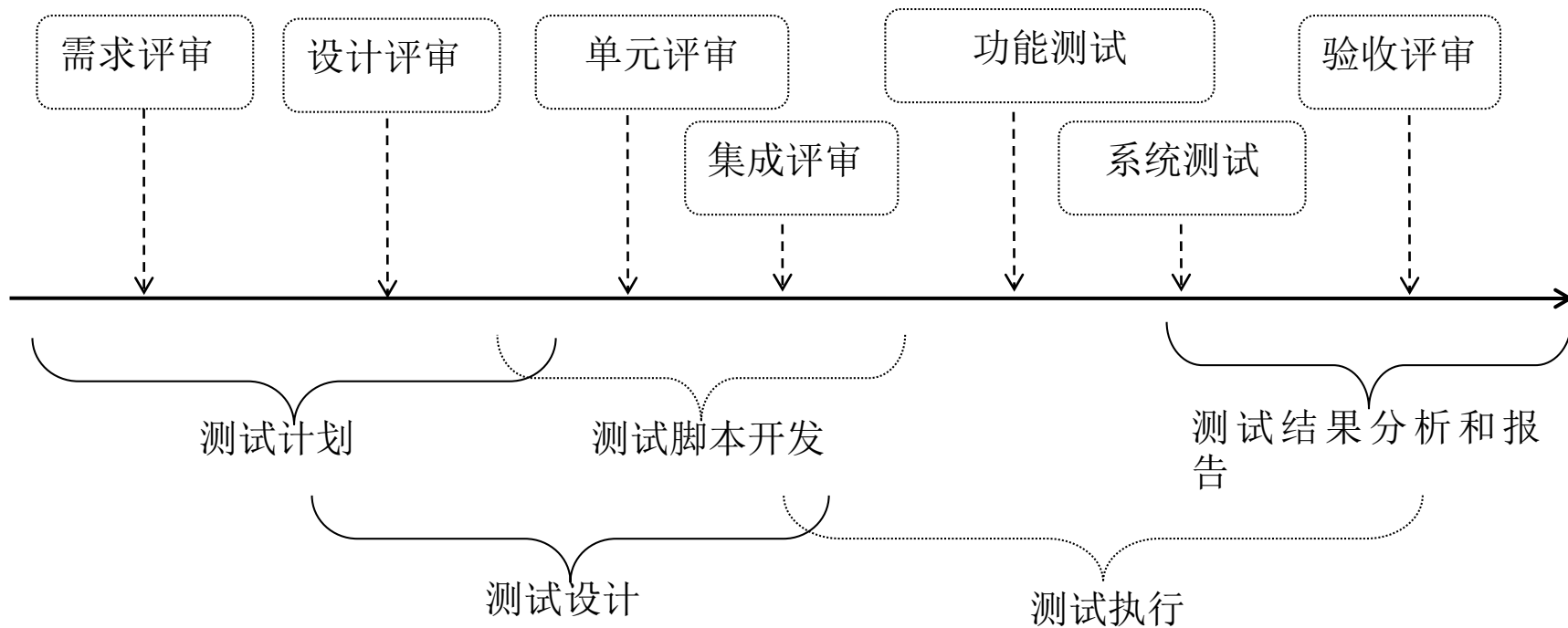


软件测试过程





软件测试过程





需求和设计评审

- **需求评审和设计评审**是验证软件产品的需求定义和设计实现，验证所定义的**产品特性是否符合客户的期望、系统的设计是否合理、是否具有可测试性以及满足非功能质量特性的要求**
- 这个阶段主要通过对需求文档、设计文档等阅读、讨论，从中发现软件需求工程和系统设计中所存在的问题



单元测试

- ❑ 单元测试的对象是**程序系统中的最小单元---模块或组件**上，在编码阶段进行，针对每个模块进行测试，主要通过**白盒测试**方法，从程序的内部结构出发设计测试用例，检查程序模块或组件的已实现的功能与定义的功能是否一致、以及编码中是否存在错误
- ❑ 多个模块可以平行地、对立地测试，通常要编写**驱动模块**和**桩模块**
- ❑ 单元测试一般由**编程人员和测试人员**共同完成



集成测试

- 集成测试，也称组装测试、联合测试、子系统测试，在单元测试的基础上，**将模块按照设计要求组装起来同时进行测试**，主要目标是发现与接口有关的模块之间问题
- 两种集成方式：非增式集成方式和增式集成方式
 - ▶ **非增式集成测试法**
 - 对每一个模块进行单元测试
 - 在此基础上按程序结构图将各模块连接起来，把连接后的程序当作一个整体增式进行测试
 - ▶ **增式集成测试法**
 - 不断地把待测模块连接到已测模块集(或其子集)上，对待测模块进行测试，直到最后一个模块测试完毕



功能测试 / 确认测试

- ❑ 功能测试一般须在完成集成测试后进行，而且是针对应用系统进行测试
- ❑ 功能测试是基于产品功能说明书，是在已知产品所应具有的功能，**从用户角度**来进行**功能验证**，以确认每个功能是否都能正常使用



系统测试

- ❑ 系统测试是将软件放在整个计算机环境下，包括软硬件平台、某些支持软件、数据和人员等，在实际运行环境下进行一系列的测试，包括恢复测试、安全测试、强度测试和性能测试等
- ❑ 测试人员要求：
 - ▶ 系统开发人员不能进行系统测试
 - ▶ 系统开发组织不能负责系统测试



验收测试 & 安装测试

- ❑ **验收**测试的目的是向未来的用户表明系统能够像预定要求那样工作，验证软件的功能和性能如同用户所合理期待的那样
- ❑ **安装**测试是指按照软件产品安装手册或相应的文档，在一个和用户使用该产品完全一样的环境中或相当于用户使用环境中，进行一步一步的安装操作性的测试



验收测试主要任务

- ❑ 明确规定验收测试通过的标准
- ❑ 确定验收测试方法
- ❑ 确定验收测试的组织和可利用的资源
- ❑ 确定测试结果的分析方法
- ❑ 制定验收测试计划并进行评审
- ❑ 设计验收测试的测试用例
- ❑ 审查验收测试的准备工作
- ❑ 执行验收测试
- ❑ 分析测试结果，决定是否通过验收



软件测试阶段

阶 段	输 入	输 出
需求分析	需求定义, 市场分析文档, 相关技术文档	市场需求分析会议记要, 功能设计, 技术设计
设计审查	市场需求文档, 技术设计文档	测试计划, 测试用例
功能验证	代码完成文件包, 功能详细设计说明书, 最终技术文档	完整测试用例, 完备的测试计划, 缺陷报告, 功能验证测试报告
系统测试	代码修改后的文件包 完整测试用例, 完备的测试计划	缺陷报告 缺陷状态报告 项目阶段报告
确认测试	代码冻结文件包 确认测试用例	缺陷状态报告 缺陷报告审查 版本审查
版本发布	代码发布文件包 测试计划检查清单	当前版本已知问题的清单 版本发布报告



大纲

- ❏ 软件测试背景
- ❏ 软件缺陷与软件测试
- ❏ 软件测试过程
- ✓ 软件测试与软件开发的关系
- ❏ 软件测试工具



软件开发过程

- ❑ 正规的软件开发过程一般包括六个阶段，即
 - ▶ 第一阶段 计划
 - ▶ 第二阶段 需求分析
 - ▶ 第三阶段 设计
 - ▶ 第四阶段 程序编写
 - ▶ 第五阶段 测试
 - ▶ 第六阶段 运行和/维护
- ❑ 这六个阶段构成了软件的生存周期

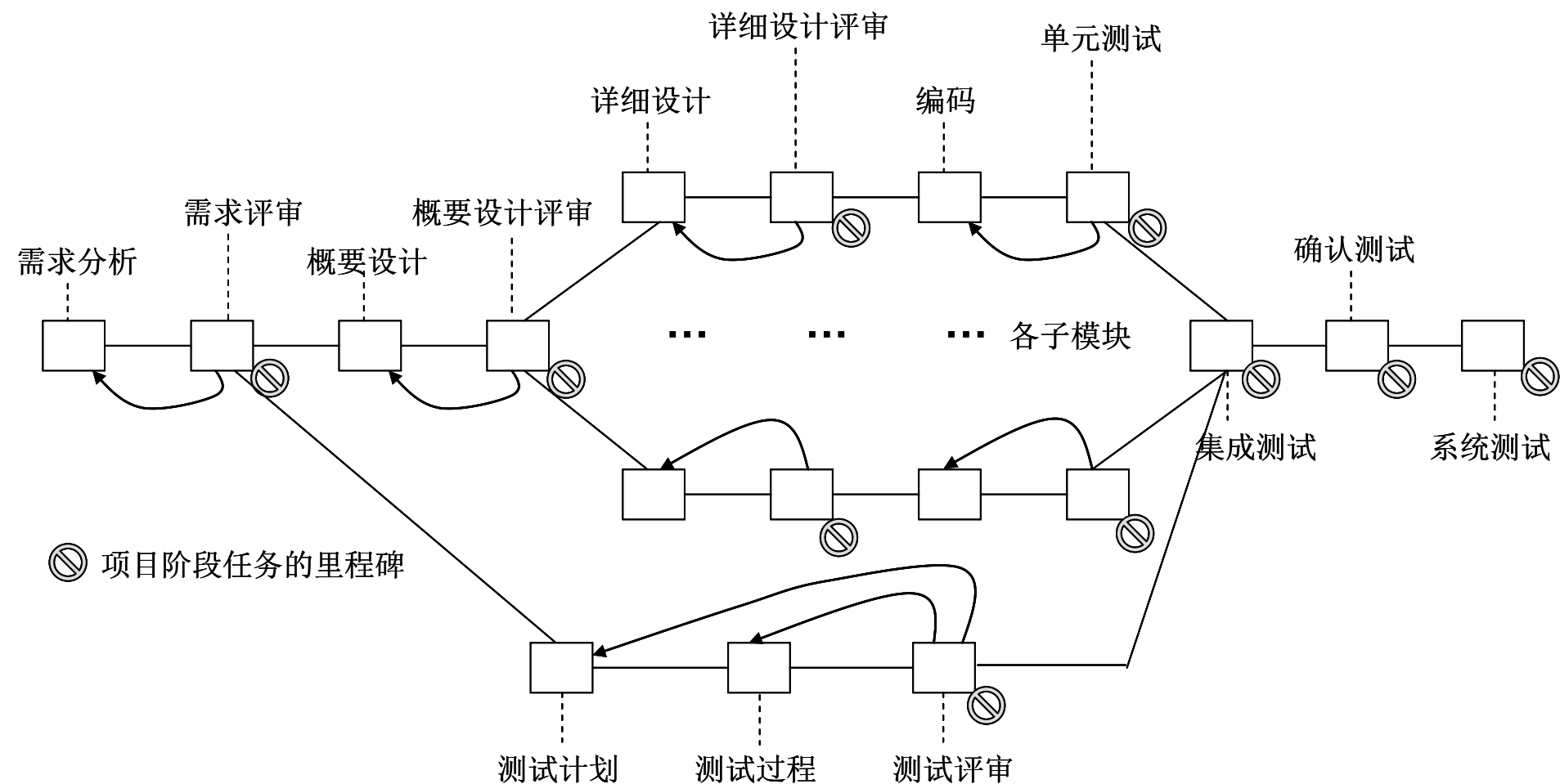


软件测试在软件开发中的作用

- ❏ 项目规划阶段：负责整个测试阶段的监控
- ❏ 需求分析阶段：确定测试需求分析，制定系统测试计划。测试需求分析是指产品生存周期中测试所需的资源、配置、各阶段评审通过的标准等
- ❏ 概要设计和详细设计阶段：制定集成测试计划和单元测试计划
- ❏ 编码阶段：开发相应的测试代码或测试脚本
- ❏ 测试阶段：实施测试，并提交相应的测试报告



软件测试与软件开发的关系

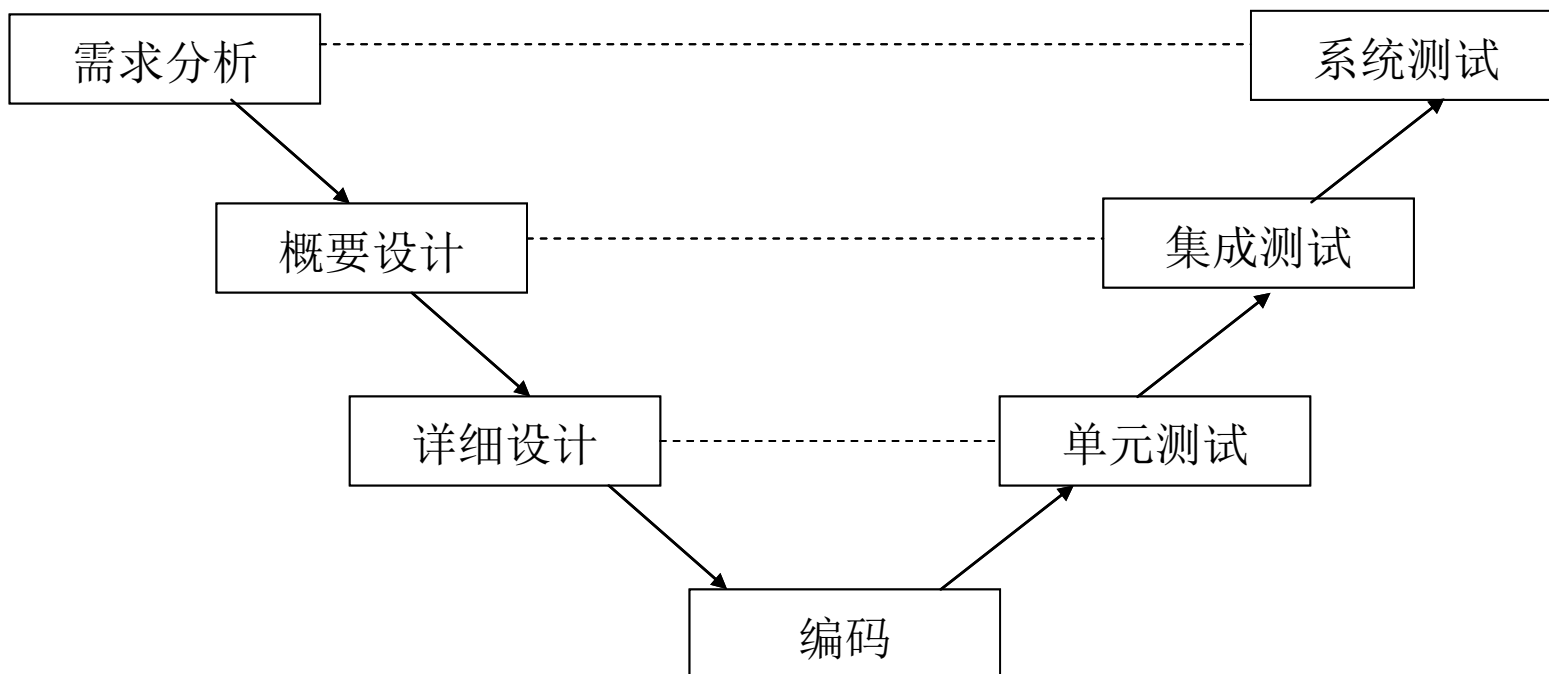




软件测试过程模型

❑ 软件测试过程V模型

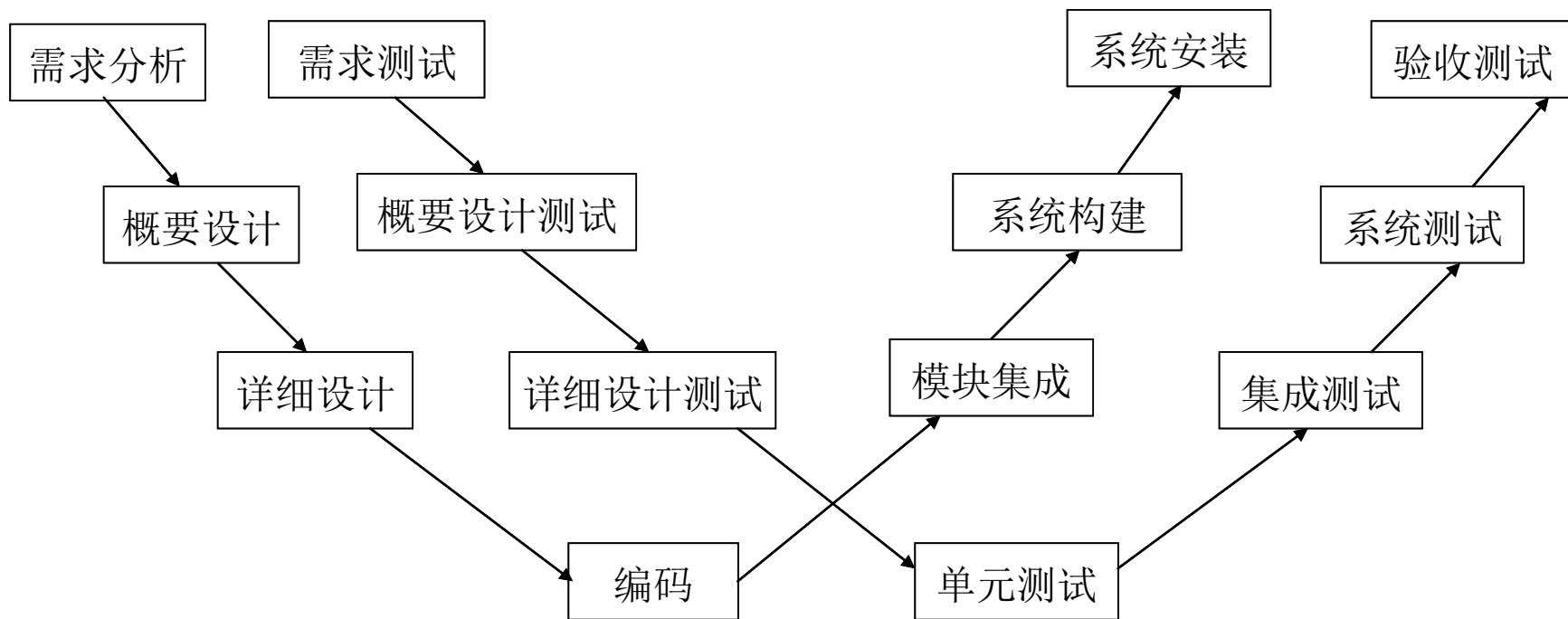
- ❑ 特点：非常明确地表明了测试的不同级别，清晰地展示了软件测试与开发之间的关系





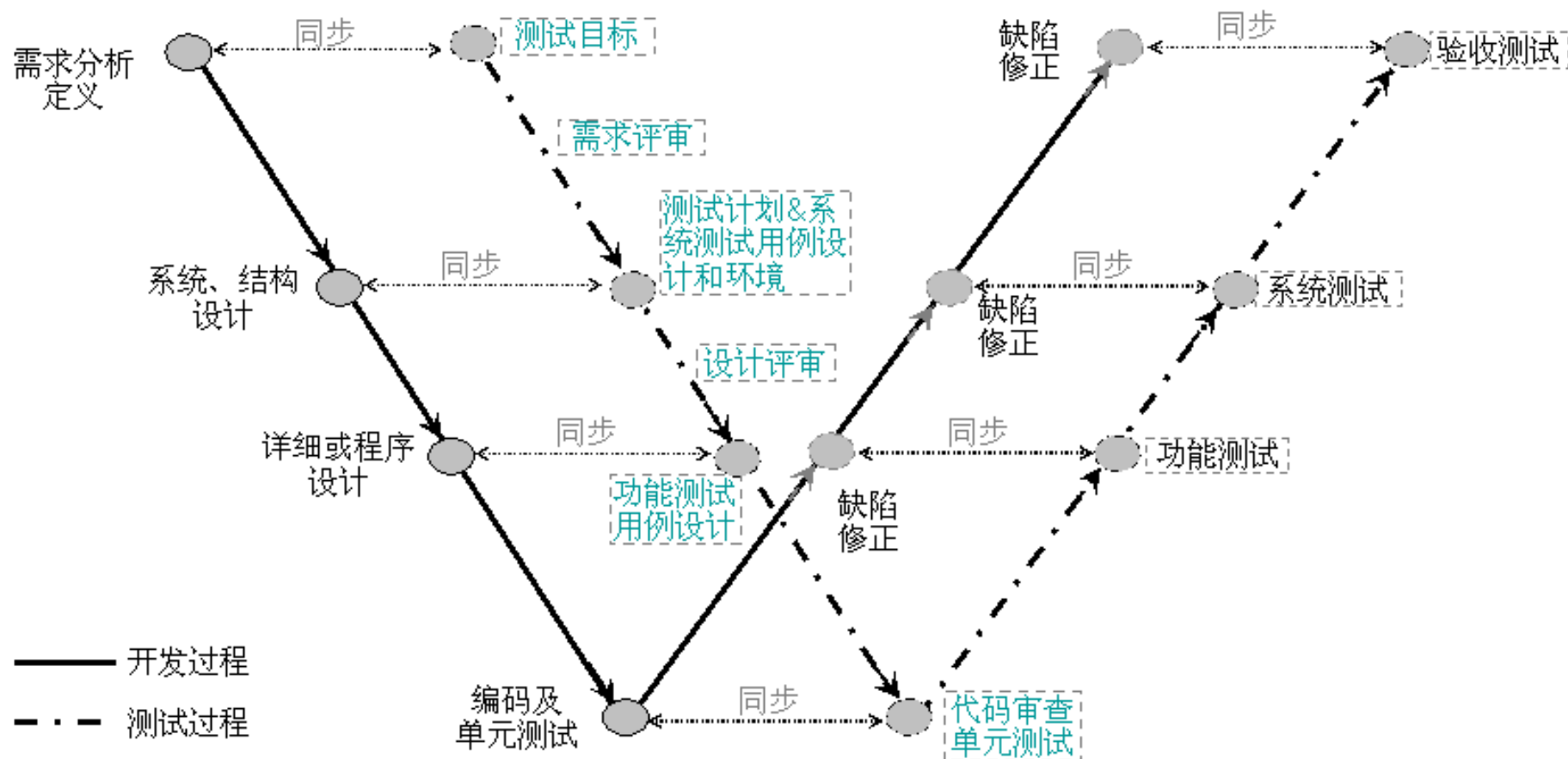
软件测试过程模型

■ 软件测试过程W模型





软件测试与软件开发的关系





软件测试环境的搭建

- 测试环境是指用来运行软件的环境
- 测试环境
 - ▶ 硬件 + 软件 + 网络 + 数据准备 + 测试工具



软件测试环境的搭建

■ 硬件环境：

- ▶ 主要是指PC机、笔记本电脑、服务器、各种PDA终端等
- ▶ 例如现要测试微软的word 2003这一款软件，那么是在PC机上测试还是在笔记本电脑上测试？如果在PC机上测试，那么CPU是奔腾2.4G，还是赛扬1.7G？内存是DDR512M，还是SD128M？
- ▶ 不同的机器类型，不同的机器配置，必然会导致不同的反应速度，因此测试一款软件时一定要考虑硬件配置



软件测试环境的搭建

■ 软件环境：

- ▶ 主要是软件运行的操作系统
- ▶ 比如word 2003是在Window 2003下检测，还是在Window XP下检测，这里可能会有兼容性的问题



软件测试环境的搭建

■ 网络环境：

- ▶ 主要指的是C/S结构还是B/S结构
- ▶ 例如要测试微软的outlook 2003这一款软件，那么是在局域网里测，还是在互联网里测？如果在局域网中测，那么是在10M的局域网里测，还是在100M的局域网里测？
- ▶ 不同的网络类型，不同的传输速度，必然会导致不同的收发速度，因此测试一款软件时也不能忽视网络的因素



软件测试环境的搭建

▣ 数据准备：

- ▶ 主要指的是测试数据的准备
- ▶ 测试数据的准备应考虑数据量和真实性，即尽可能获取大量的真实的数据，包括正确和错误的数据
- ▶ 当无法取得真实数据时应尽可能模拟出大量的数据



软件测试环境的搭建

■ 测试工具：

- ▶ 目前市场上的测试工具很多，可分为静态测试工具、动态测试工具、黑盒测试工具、白盒测试工具、测试执行评估工具、测试管理工具等
- ▶ 对测试工具的选择是一个比较重要的问题，应根据测试需求和实际条件来选择已有的测试工具，或购买、自行开发相应的测试工具



软件测试环境的搭建

▣ 搭建软件测试环境还应注意以下几点：

- ▶ 尽量模拟用户的真实使用环境
- ▶ 测试环境中尽量不要安装其它与被测软件无关的软件，但最好安装杀毒软件，以确保系统没有病毒
- ▶ 测试环境应与开发环境独立



大纲

- ❑ 软件测试背景
- ❑ 软件缺陷与软件测试
- ❑ 软件测试过程
- ❑ 软件测试与软件开发的关系
- ✓ 软件测试工具



软件测试工具

■ 白盒测试工具

▶ 静态测试工具

- ▮ 职能：主要集中在需求文档、设计文档以及程序结构上，可以进行类型分析、接口分析、输入输出规格说明分析等
- ▮ 工具：McCabe & Associates 公司开发的McCabe Visual Quality ToolSet分析工具；ViewLog公司开发的LogiScope分析工具；Software Research公司开发的TestWork/Advisor分析工具及Software Emancipation公司开发的Discover分析工具，北京邮电大学开发的DTS缺陷测试工具等



软件测试工具

■ 白盒测试工具

▶ 动态测试工具

- ▮ 职能：功能确认与接口测试、覆盖率分析、性能分析、内存分析等
- ▮ 工具：Compuware公司开发的DevPartner软件、Rational公司研制的Purify系列等

▶ 动静结合

- ▮ Coverity
- ▮ Fortify



软件测试工具

■ 黑盒测试工具

- ▶ 工具：Rational公司的TeamTest， Compuware公司的QACenter
- ▶ 分类
 - 功能测试工具
 - 性能测试工具



软件测试工具

■ 测试设计和开发工具

- ▶ 测试设计：说明测试被测软件特征或特征组合的方法，确定并选择相关测试用例的过程
- ▶ 测试开发：将测试设计转换成具体的测试用例的过程
- ▶ 工具类型：
 - 测试数据生成器
代表工具：Bender & Associates公司提供的功能测试数据生成工具SoftTest；Parasoft公司提供的C/C++单元测试工具Parasoft C++test等
 - 基于需求的测试设计工具
 - 捕获/回放工具和覆盖分析工具

.....



软件测试工具

■ 测试执行和评估工具

- ▶ 测试执行和评估：执行测试用例并对结果进行评估的过程，包括选择用于执行的测试用例、设置测试环境、运行所选择的测试、记录测试执行活动、分析潜在的软件故障并测量测试工作的有效性
- ▶ 工具类型
 - 捕获 / 回放
 - 覆盖分析
 - 存储器测试



软件测试工具

■ 测试管理工具

- ▶ 测试管理：帮助完成测试计划，跟踪测试运行结果等
- ▶ 用途：
 - 测试用例管理
 - 缺陷跟踪管理：BugZilla, BugFree
 - 配置管理：SubVersion(SVN)、CVS
- ▶ 代表工具：

Rational公司的Test Manager、Compuware公司的TrackRecord等软件、TestDirector



软件测试工具

■ MI公司产品

- ▶ LoadRunner、WinRunner、TestDirector、QT

■ IBM Rational公司产品

- ▶ Rational Robot (功能/性能测试工具)
- ▶ Rational Purify (白盒测试工具)
- ▶ Rational Test manager (测试管理工具)
- ▶ Rational ClearQuest (缺陷/变更管理工具)

■ Compuware公司产品

- ▶ 自动黑盒测试工具QACenter
- ▶ 自动白盒测试工具DevPartner
- ▶ Vantage应用级网络性能监控管理软件



小结

- ❏ 软件测试背景
- ❏ 软件缺陷与软件测试
- ❏ 软件测试过程
- ❏ 软件测试与软件开发的关系
- ❏ 软件测试工具