



软件测试步骤及相关测试技术的应用



北京大学



参考文献:

- (1) 蔡建平, 软件测试大学教程. 北京: 清华大学出版社, 2009. 9.
- (2) 刘竹林, 软件测试技术与案例实践教程. 北京: 北京师范大学出版社, 2011. 5.
- (3) 郑人杰, 马素霞, 殷人昆, 软件工程概论. 北京: 机械工业出版社, 2009. 11.
- (4) 吕云翔, 王洋, 肖咚, 软件测试案例教程. 北京: 机械工业出版社, 2010. 10.
- (5) 吕云翔, 刘浩, 王昕鹏, 周建, 软件工程课程设计. 北京: 机械工业出版社, 2009. 6.
- (6) 蔡建平, 软件测试实验指导教程. 北京: 清华大学出版社, 2009. 11.
- (7) Roger S. Pressman 著, 郑人杰等译. 软件工程-实践者的研究方法 (第七版). 北京: 机械工业出版社, 2011. 5.
- (8) 韩离凯等, 软件测试. 北京: 清华大学出版社, 2013. 8.
- (9) 朱少民等, 软件测试实验教程. 北京: 清华大学出版社, 2019. 6.





一、软件测试的步骤:

(1) 单元测试（往往采用白盒测试技术）：集中于每个独立的模块。该测试以详细设计文档为指导，测试模块内的重要控制路径。

(2) 集成测试：集中于模块的组装。其目标是发现与接口有关的错误，将经过单元测试的模块构成一个满足设计要求的软件结构。

(3) 确认测试（有效性测试）：目标是发现软件实现的功能与需求规格说明书不一致的错误。（通常采用黑盒测试技术）

(4) 系统测试：集中检验系统所有元素（包括硬件、软件）之间协作是否合适，整个系统的性能、功能是否达到。





单元测试，集中对源代码实现的每个程序单元进行测试，检查各个程序模块是否正确地实现了程序规定的功能。

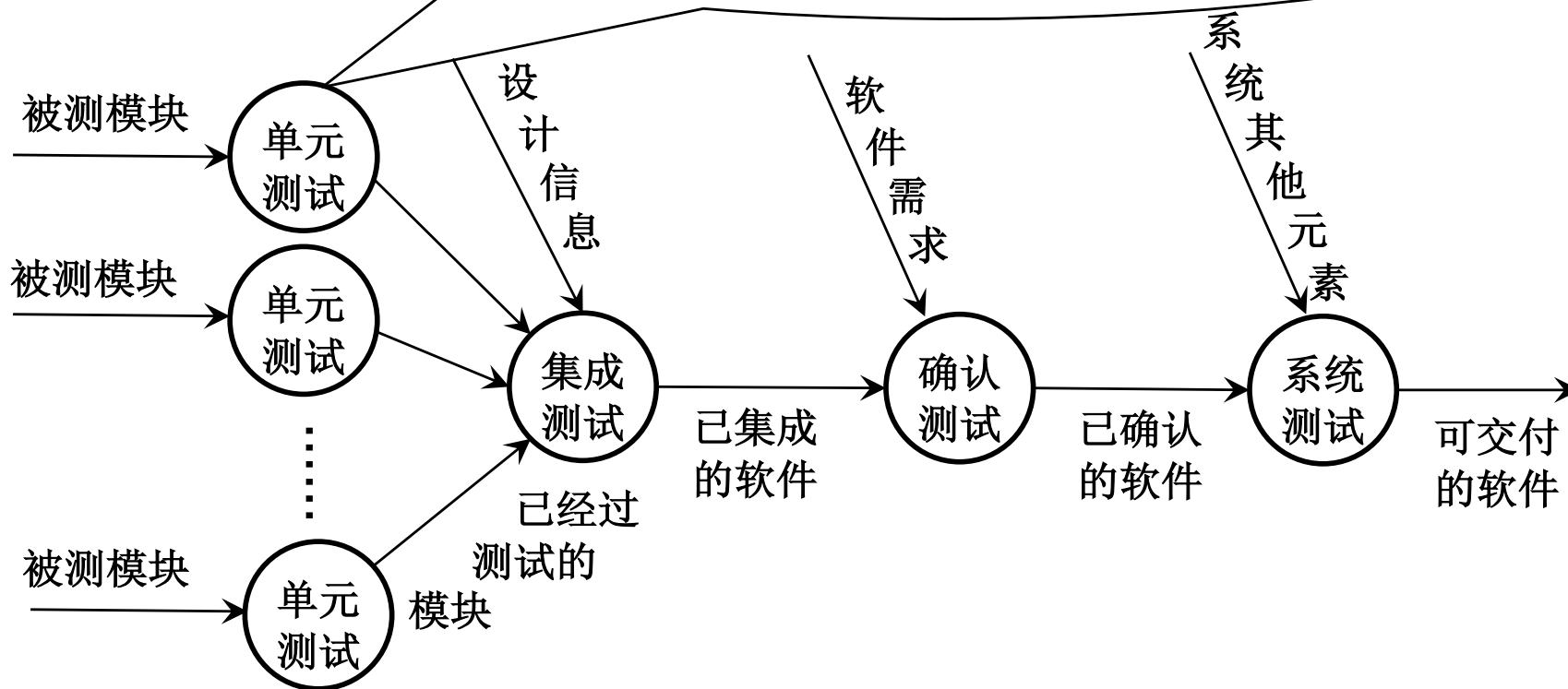


图1 软件测试的步骤



把已经测试过的程序模块组装起来，
进行集成测试，主要对与设计相关的
软件体系结构的构造进行测试

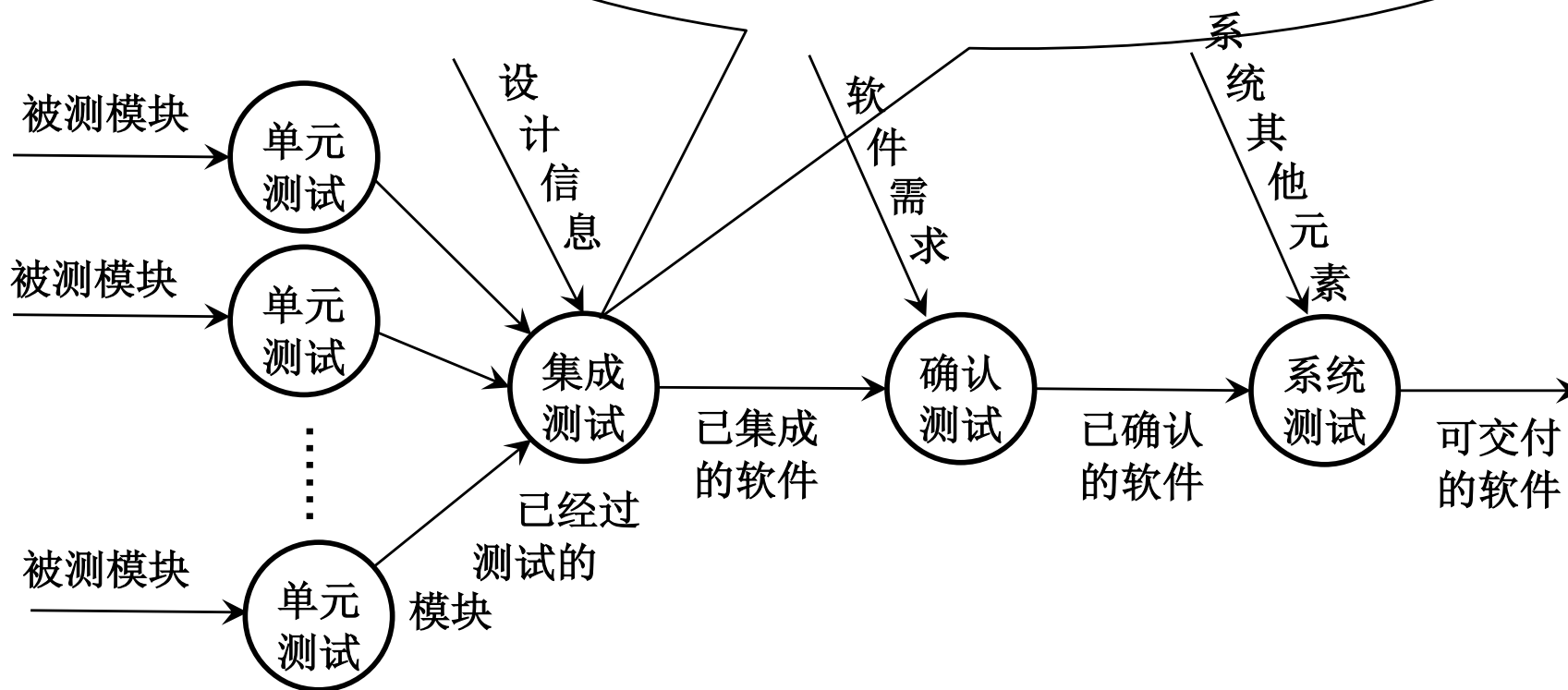


图1 软件测试的步骤





确认测试则要检查已实现的软件是否满足了软件需求规格说明书中确定的各种需求，以及软件配置是否完全、正确

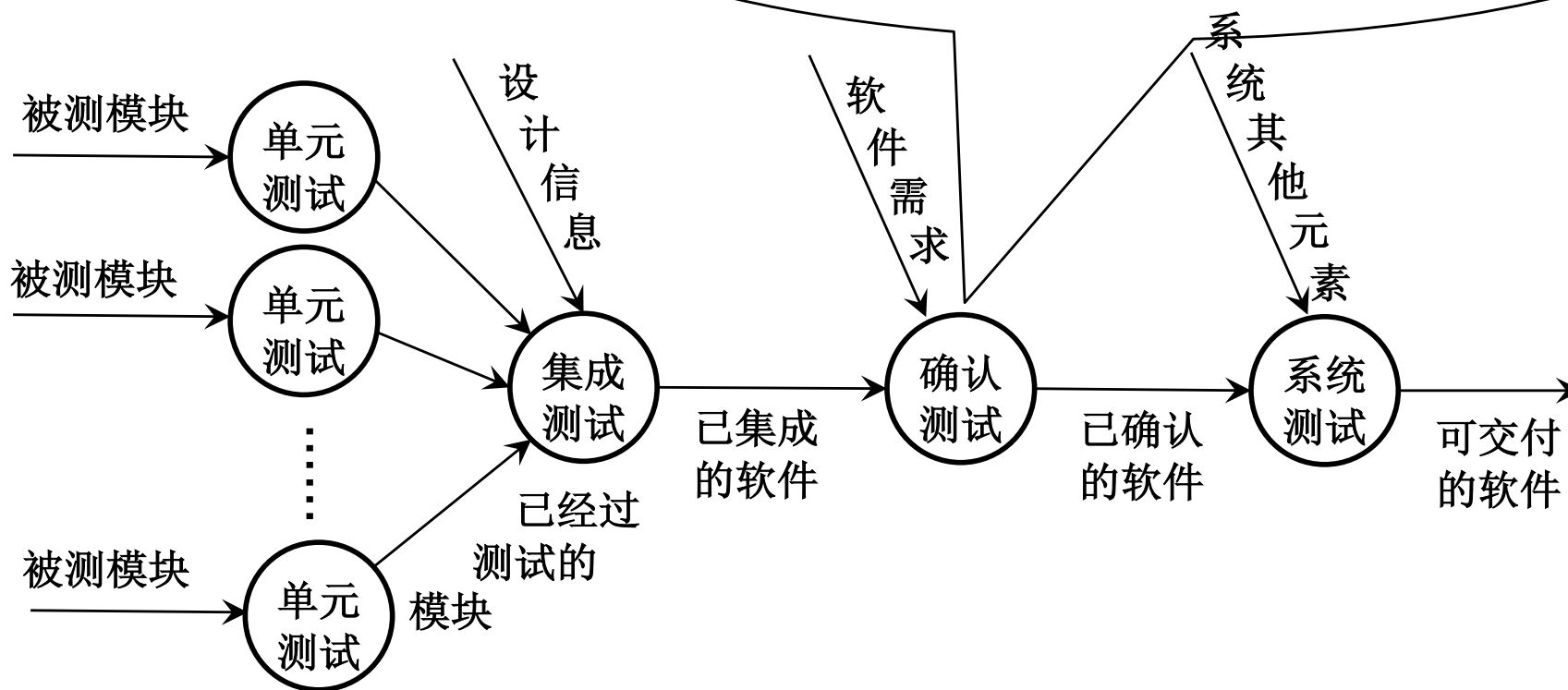


图1 软件测试的步骤





系统测试，则把已经经过确认的软件纳入到实际运行环境中，与系统中其他成分组合在一起进行测试

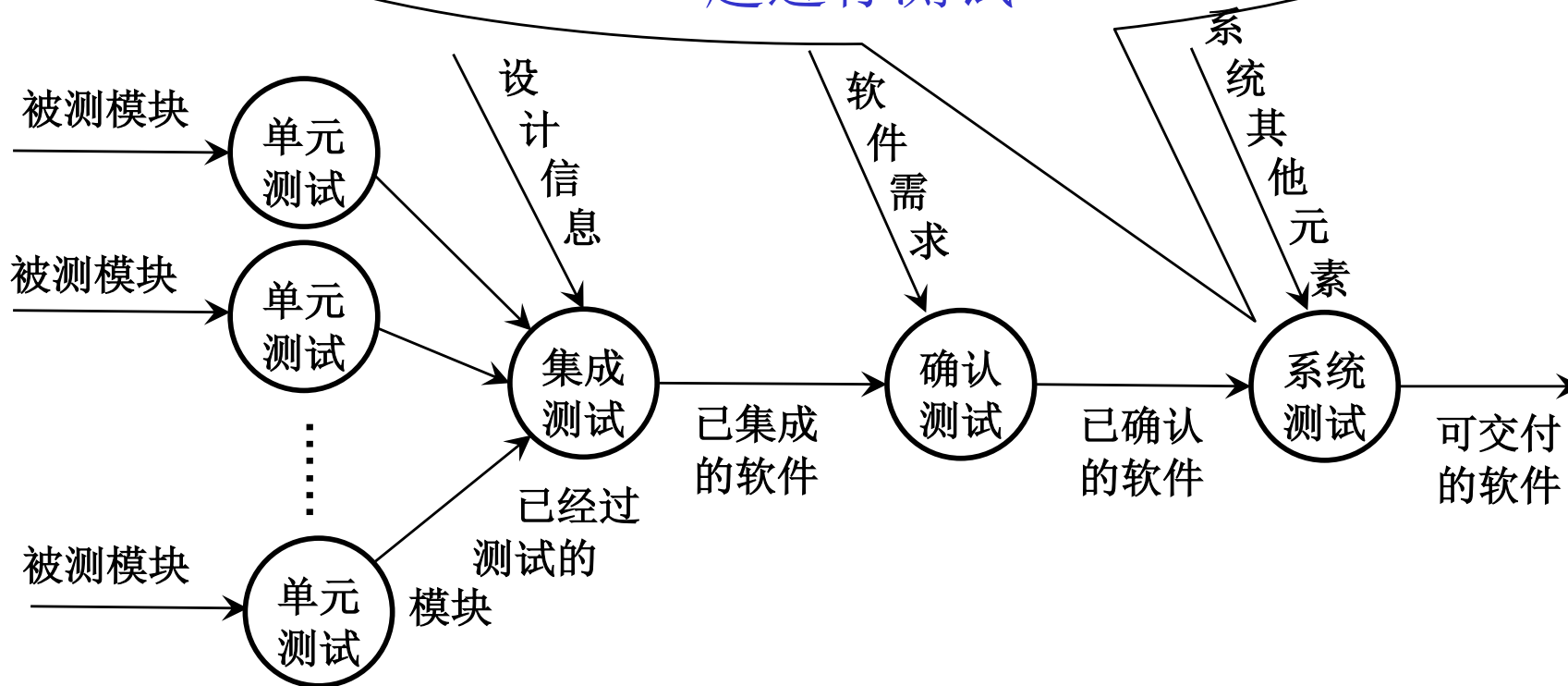
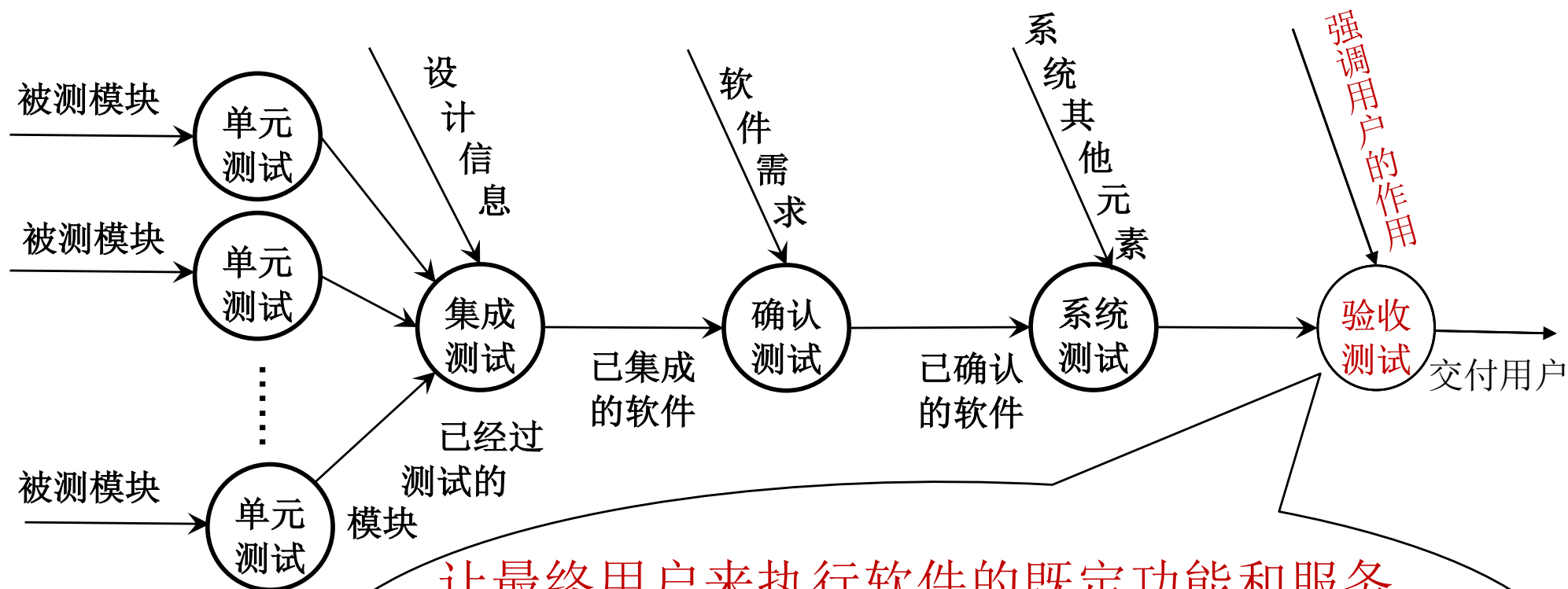


图1 软件测试的步骤





让最终用户来执行软件的既定功能和服务，检查软件是否满足软件需求说明书中的确认标准。作用是向用户表明系统能够按照预定要求那样工作，使系统最终可以正式发布或向用户提供服务





二、单元测试

1、单元测试的内容：

- 主要依据详细设计说明书和源代码清单
- 主要测试模块的I/O条件和模块的逻辑结构
- 主要采用白盒测试方法设计测试用例，辅以黑盒测试的测试用例，使之对任何合理的和不合理的输入都要鉴别和响应
- 要对所有的局部和全局数据结构、外部接口和程序代码的关键部分进行代码审查

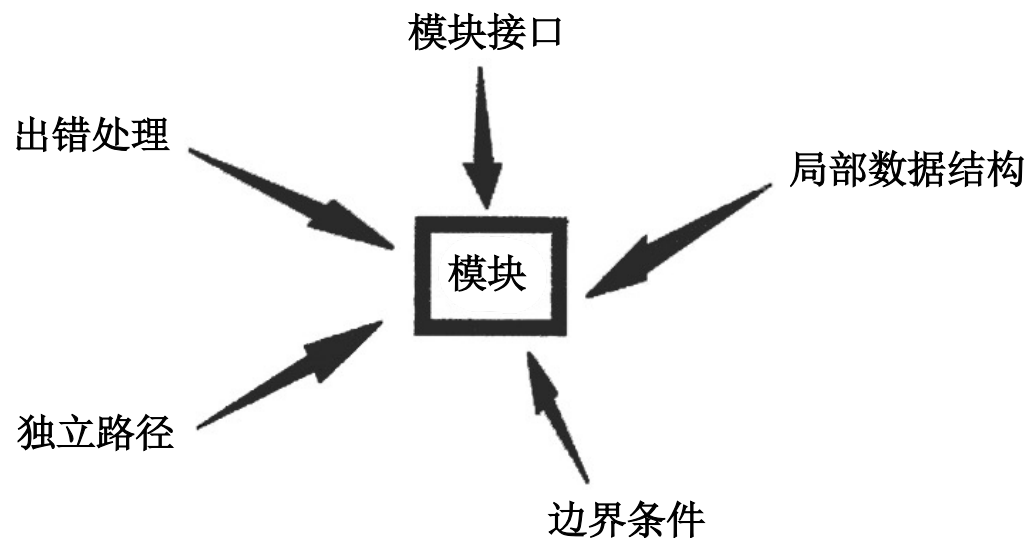


图2 单元测试的内容





2、单元测试的步骤

- (1) 在代码编写完成后完成单元测试计划。根据《软件需求规格说明书》和《软件详细设计说明书》做出《单元测试计划》，建立单元测试环境。

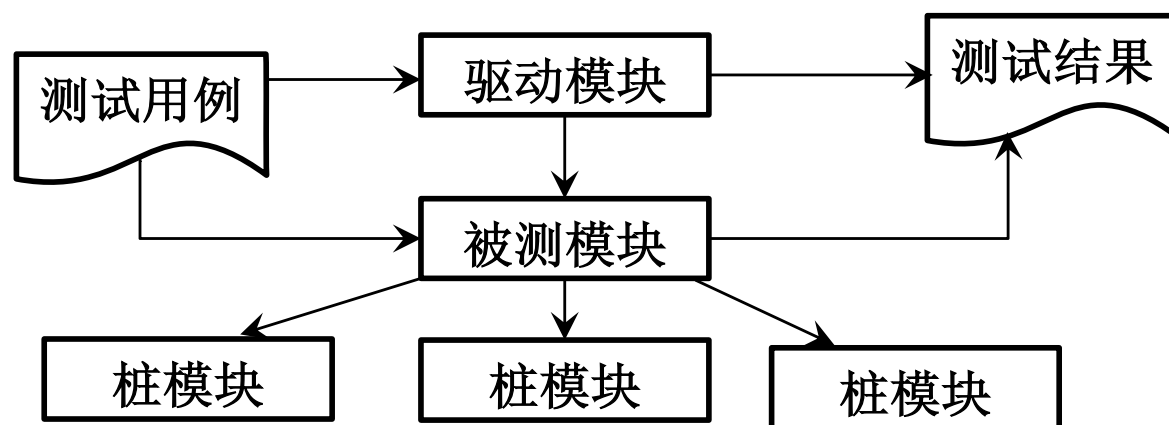


图3 单元测试的测试环境

- 驱动模块：相当于被测模块的主程序，它接收测试数据，把这些测试数据传递给被测模块，最后再输出实测结果。
- 桩模块：由被测模块调用，用以代替被测模块所调用的子模块。





(2) 代码审查

- 开发组、测试组和相关人员联合进行
- 开发人员讲解自己的代码，通过checklist(检查单)方式进行错误查找
- 一般有正式的计划、流程和结果报告

(3) 静态测试：代码走查

- 开发人员对编写的代码进行静态分析

表1 代码审查和代码走查

	代码走查	代码审查
准备	通读设计和编码	应准备好需求描述文档、程序设计文档、程序的源代码清单、代码编码标准和代码缺陷检查表
形式	非正式会议	正式会议
参加人员	开发人员为主	项目组成员包括测试人员
主要技术方法	无	缺陷检查表
注意事项	限时、不要现场修改代码	限时、不要现场修改代码
生成文档	会议记录	静态分析错误报告
目标	代码标准规范，无逻辑错误	代码标准规范，无逻辑错误



表2 单元测试检查表格式

1. 单元测试检查表

单元测试检查表如下：

案例：

单元测试检查表

单元名称_____系统名称_____

任务编号_____初次测试日期_____

检查目标：关键测试项是否已纠正

检查项：

- (1) 是否有输入参数没有使用的？是否有输出参数没有产生的测试项？
- (2) 是否有数据类型不正确或不一致的测试项？
- (3) 有无局部变量使用前未初始化的测试项？
- (4) 有无外部接口的调用语句、文件存取或数据库错误的测试项？
- (5) 有无逻辑路径错误的测试项？
- (6) 该单元是否有多个入口或多个正常的出口？
- (7) 代码中有任何偏离本项目标准的地方？
- (8) 代码中有任何对于用户来说不清楚的错误提示信息？
- (9) 如果该单元是设计为可重用的单元模块，代码中是有无妨碍可重用的地方？

2. 审查结果说明

(请用单独的一页或多页。每一项动作必须指出所引用的问题。)

审查结果

- (1) 如果上述9个问题的答案均为“否”，那么测试通过，请在此标记并且在最后签名。
- (2) 如果代码存在严重的问题，例如多个关键问题的答案为“是”，那么程序编制者纠正这些错误，并且必须重新安排一次单元测试。

下一次单元测试的日期：_____

- (3) 如果代码存在小的缺陷，那么程序编制者纠正这些错误，并且审查组必须安排一次跟踪会议。

跟踪会议的日期：_____

测试人签名：_____日期：_____

代码审查的形式：

- 首先给出单元测试检查表
- 其次给出审查后要做的工作





(4) 动态测试：设计单元测试用例。

- a. 根据《单元测试计划》和《软件详细设计说明书》编写出《单元测试用例》
- b. 着重测试模块的4个方面：模块接口、局部数据结构、重要的执行路径、错误执行路径以及与这4个方面相关的边界条件

➤ **模块接口**：即对通过被测试的数据流进行测试，检查进出模块的数据是否正确。为此，必须对全部的模块接口，包括参数表、调用子模块的参数、全局数据、文件输入/输出操作等进行测试

➤ **测试项目包括以下内容**

● 调用其他模块时

- 所传送的实际参数个数与被调用模块的形式参数的个数是否相同；
- 所传送的实际参数类型与被调用模块的形式参数的类型是否相同；
- 所传送的实际参数单位与被调用模块的形式参数的单位是否一致





- 调用内部函数时，参数个数、属性和次序是否正确。
- 在模块有多个入口的情况下，是否引用与当前入口无关的参数。
- 是否修改了只读型参数。
- 全局变量是否在所有引用它们的模块中都有相同的定义。

➤ 如果模块内包括外部I/O，还应考虑以下因素

- 文件属性是否正确
- OPEN与CLOSE语句是否正确
- 缓冲区容量与记录长度是否匹配
- 在进行读/写操作之前是否打开了文件
- 在结束文件处理时是否关闭了文件
- 正文书写/输入有无错误
- I/O错误是否检查并做了处理





➤局部数据结构测试

➤测试项目包括以下内容

- 不一致或不正确的数据类型说明
- 使用尚未赋值或初始化的变量
- 错误的初始值或默认值
- 变量名拼写或书写错误
- 上溢、下溢或者地址错误

➤在模块工作过程中，必须测试模块内部的数据能否保持完整性，包括内部数据的内容、形式及相互关系不发生错误。

➤路径测试：对模块中重要的执行路径进行测试

➤路径测试的测试用例必须能发现由于计算错误而产生的错误，常用于发现以下错误





➤ 路径测试的测试用例必须能发现由于计算错误而产生的错误，常用于发现以下错误

- 误解的或不正确的算数优先级
- 混合模式的运算错误
- 错误的初始化
- 精确度不够
- 表达式的不正确符号表示

➤ 还必须能发现不正确的判定或不正常的控制流而产生的错误，常见错误如下

- 不同数据类型的比较错误
- 不正确的判定或不正确的变量
- 不正确的或不存在的循环终止
- 当遇到分支循环时不能退出
- 不适当地修改循环变量





➤ **错误处理测试**：检查模块的错误处理功能是否包括错误或缺陷，处理设施是否有效

- 是否可拒绝不合理的输入
- 出错的描述是否难以理解
- 出错的描述是否不能够对错误定位
- 显示的错误与实际的错误是否相符
- 对错误条件的处理是否正确
- 在对错误处理之前错误条件是否已经引起系统的干预等

➤ **边界条件测试**：必须采用边界值分析方法来设计测试用例，注意数据流、控制流中刚好等于、大于或小于确定的边界值时，模块是否能够正常工作

- 在n次循环的第0次、1次、n次是否有错误
- 运算或判断中取最大值、最小值时是否有错误
- 数据流、控制流中刚好等于、大于、小于确定的比较值时是否出现错误





c. 编写测试用例注意以下事项

- 对于代码比较复杂的结构，首先画出控制流程图，然后尽量做到分支覆盖，核心代码要尽量做到路径覆盖
- 在测试用例的输入数据时：尽量包括真实数据、边界数据
- 测试语句和测试数据尽量简单，容易理解
- 为避免对其他代码过多的依赖，可以实现简单的桩模块

(5) 执行单元测试。

(6) 书写《缺陷跟踪报告》，然后进行程序修改并进行回归测试（回归测试：是指修改旧代码后，重新进行测试以确认修改没有引入新的错误）。

(7) 书写《测试总结报告》。如果达到测试通过标准（这个指标在《测试计划》中有明确规定），则停止测试，编写《单元测试报告》



北京大学



三、集成测试

1、集成测试的任务

- (1) 把各个模块连接起来的时候，穿越模块接口的数据是否丢失；
- (2) 一个模块的功能是否对另一个模块的功能产生不利的影响；
- (3) 各个子功能组合起来，能否达到预期要求的父功能；
- (4) 全局数据结构是否有问题；
- (5) 单个模块的误差累加起来，是否达到不可接受的程度。

2、集成测试的方法

- (1) 一次性组装方式：首先对每个模块分别进行单元测试，然后把所有模块组装在一起进行测试
- (2) 增量式组装方式：首先对每个模块分别单元测试，然后逐步组装成较大的系统，在组装的过程中边连接边测试。





3、冒烟测试

是一种常用的集成测试方法，是时间关键性的步进机制，让软件项目团队频繁地对项目进行评估。冒烟测试包括以下活动：

- 将已经转换为代码的软件构件集成到构建（build）中。一个构建包括所有的数据文件、库、可复用的模块以及实现一个或多个产品功能所需的工程化构件
- 设计一系列测试以暴露影响构建正常完成其功能的错误。其目的是为了发现极有可能造成项目延迟的业务阻塞（show stopper）错误
- 每天将该构建与其他构建及整个软件产品集成起来进行冒烟测试





三、确认测试

1、确认测试的任务

又称为有效性测试，验证软件的有效性，即验证软件的功能和性能及其他特性是否与用户的要求一致。

2、确认测试的步骤

- 有效性测试是在模拟的环境（可能是开发环境）下，运用黑盒测试的方法，验证被测软件是否满足需求规格说明书列出的需求。
- 软件配置审查是保证软件配置的所有成分齐全，各方面的质量符合要求，具有维护阶段所必须的细节，并且已编排好分类的目录。

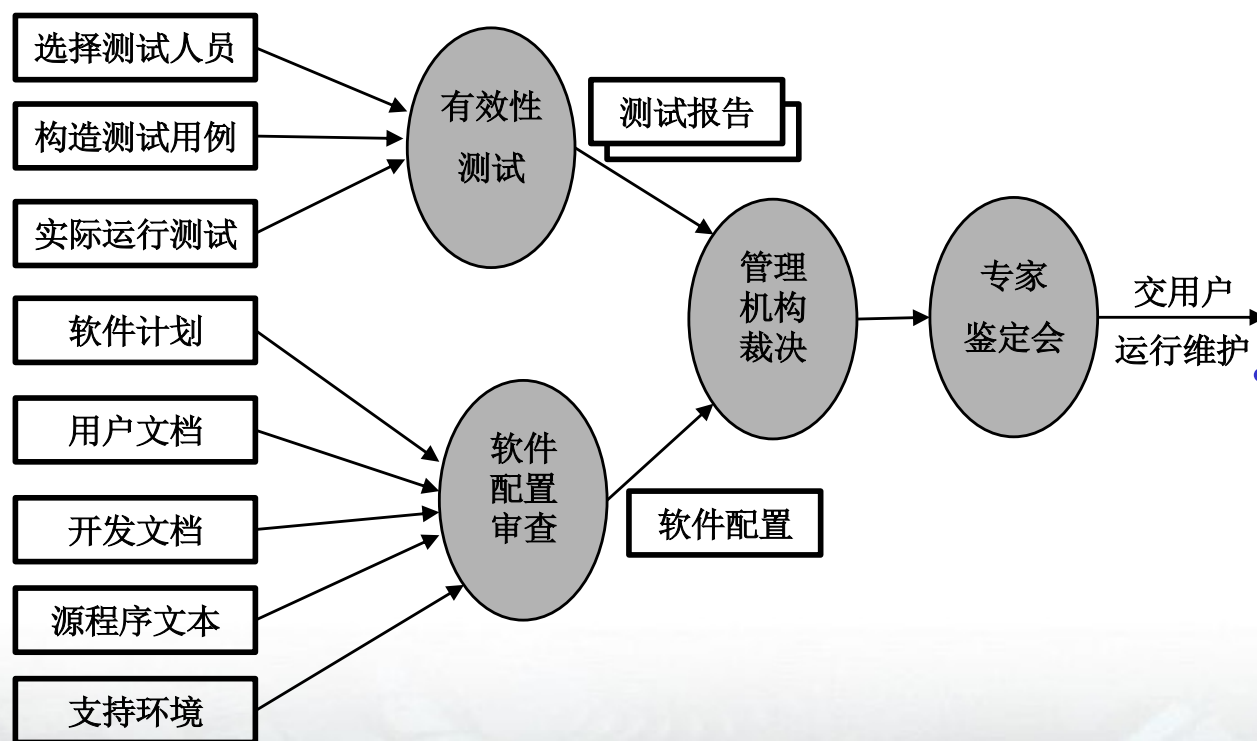


图4 确认测试流程图





四、系统测试

集中检验系统所有元素（包括硬件、信息等）之间协作是否合适，整个系统的性能、功能是否达到。

系统测试实际上是一系列不同的测试，以下是用于系统测试的几种典型软件系统测试：

- **功能测试：** 在规定的时间内运行软件系统的所有功能，以验证这个软件系统有无严重错误。
- **恢复测试：** 是一种系统测试，它指采取各种人工干预方式强制性地使软件出错，使其不能正常工作，进而检验系统的恢复能力。
 - 如果恢复是自动的，则重新初始化、检测点设置、数据恢复以及重新启动等都是对其正确性的评价；
 - 如果恢复需要人员的干预，则要估算出修复的平均时间，以及确定它是否能在可接受的限制范围内。





•**安全性测试**：就是试图去验证建立在系统内的预防机制，以防止来自非正常的侵入：

- 充当任何角色，测试者可以通过外部书写的方式得到口令；
- 可以用用户设计的软件去破坏已构造好的任何防御的袭击系统；
- 可以破坏系统，使系统不能为他人服务；
- 可以跳过侵入的恢复过程，而故意使系统出错；
- 也可以跳过找出系统的入口钥匙，而放过看到的不安全数据等；

•**强度测试**：是在非正常数量、频率或容量资源方式下运行一个系统。目的是检查在系统运行环境不正常乃至发生故障的情况下，系统可以运行到何种程度的测试。确定本系统正常工作的最差工作环境，也可能是用于验证在标准工作压力下的各种资源的最下限指标。

•**性能测试**：测试软件在被组装进系统的环境下运行时的性能：

- 性能测试应覆盖测试过程的每一步，即使在单元层，单个模块的性能也可以通过白盒测试来评价；
- 性能测试有时与强度测试联系在一起，常常需要硬件和软件的测试设备。





❖ 常见的性能指标:

- 系统/事务平均响应时间
- 事务/交易的处理效率
- 页面浏览量: 用户向服务器发请求, 服务器处理一次真实的请求
- 连接时间
- 发送时间
- 处理时间
- 页面下载时间
- 吞吐率: 即每秒服务器处理的HTTP申请数
- 每秒点击次数
- 每秒的SSL连接数
- 内存和CPU使用率
-





- **可用性测试：**从使用的合理性、方便性等角度对软件系统进行检验，以发现人为因素或使用上的问题。
- **部署测试（配置测试）：**软件必须在多种平台及操作系统环境中运行。有时将部署测试称为配置测试，是在软件将要在其中运行的每一种环境中测试软件。另外，部署测试检查客户将要使用的所有安装程序及专业安装软件，并检查用于向最终用户介绍软件的所有文档。





五、测试驱动开发

测试驱动开发是极限编程的重要特点。即：

在开发功能代码之前先编写测试代码，然后编写相关的代码满足这些测试用例。然后循环进行添加其他功能，直到完成全部功能的开发。

测试驱动的原则：

- (1) 测试隔离。不同代码的测试应相互隔离。
- (2) 测试列表。在任何阶段如果需求功能改变，添加功能时，首先把相关功能点加到测试列表中，然后继续手头工作。
- (3) 先写断言。编写测试代码时，首先编写对功能代码判断使用的断言，然后编写相应的功能代码。





- (4) 可测试性。功能代码的设计和编写时应该具有较强的可测试性。
- (5) 及时重构。无论是功能代码还是测试代码，对结构不合理，重复的代码等情况，在测试通过后，及时进行重构。





六、面向对象软件的测试策略

1、面向对象环境中的单元测试

- 面向对象软件的类测试等同于传统软件的单元测试，类包含的操作是最小的可测试单元
- 不独立地对单个操作进行测试，而是将其作为类的一部分测试，如超类的操作必须在每个子类的环境中测试
- 面向对象软件的类测试是由封装在该类中的操作和类的状态行为驱动的。





2、面向对象环境中的集成测试

- 由于面向对象软件没有明显的层次控制结构，因此，传统的自顶向下和自底向上集成策略没有太大意义
- 由于类的成分间直接或间接地相互作用，每次将一个操作集成到类中（传统的增量集成方法）往往也是不可能的
- 面向对象软件的集成测试有两种策略
 - 基于线程的测试（thread-based testing）：对响应系统的一个输入或事件所需的一组类进行集成，每个线程单独地集成和测试，应用回归测试以确保没有产生副作用。
 - 基于使用的测试（use-based testing）：通过测试很少使用服务类的那些类（称之为独立类）开始构造系统，独立类测试完后，利用独立类测试下一层次的类（称之为依赖类）。继续依赖类的测试直到完成整个系统。





- 面向对象软件的集成测试中，驱动程序和桩程序（承接模块）的使用也发生了变化
 - 驱动程序可用于测试低层中的操作和整组类的测试
 - 驱动程序也可用于代替用户界面以便在界面实现之前可以进行系统功能的测试
 - 桩程序（承接模块）可用于需要类间的协作但其中一个或多个协作类仍未完全实现的情况下
- 簇测试（cluster testing）是面向对象软件集成测试中的一步
 - 利用试图发现协作中的错误的测试用例来测试协作的类簇





七、Web应用的测试策略

- Web应用测试策略采用了所有软件测试的基本原理，并使用面向对象系统所使用的测试策略
- Web应用的测试步骤
 - 1、对Web应用的内容模型进行评审，以发现错误。
 - 2、对接口模型进行评审，保证适合所有的用例。
 - 3、评审Web应用的设计模型，发现导航错误。
 - 4、测试用户界面，发现表现机制和导航机制中的错误。
 - 5、对功能构件进行单元测试
 - 6、对贯穿体系结构的导航进行测试
 - 7、在各种不同的配置环境下，实现Web应用，并测试Web应用对每一种配置的兼容性
 - 8、进行安全性测试，试图攻击Web应用或其所处环境的弱点
 - 9、进行性能测试
 - 10、通过可监控的最终用户群对Web应用进行测试，对他们与系统的交互结果进行以下方面的评估，包括内容和导航错误、可用性、兼容性、安全性、可靠性及性能等方面的评估





表3 界面测试指标

指标	检查项	测试人员评价
合适性和正确性	用户界面是否与软件的功能相融洽	
	是否所有界面元素的文字和状态都正确无误	
容易理解	对于常用的功能，用户是否不必阅读手册就能使用	
	是否所有界面元素提供(例如图标)都不会让人误解	
	是否所有界面元素都提供了充分而必要的提示	
	界面结构是否能够清晰地反映工作流程	
	用户是否容易知道自己在界面中的位置，不会迷失方向	
	是否有联机帮助	
及时反馈信息	是否提供进度条、动画等反应正在进行的比较耗时间的过程	
	是否为重要的操作返回必要的结果信息	
出错处理	是否对重要的输入数据进行校验	
	执行有风险的操作时，是否有“确认”、“放弃”等提示	
	是否根据用户的权限自动屏蔽某些功能	
	是否提供 Undo 功能用以撤销不期望的操作	





表4 界面测试指标(续)

指 标	检查项	测试人员评价
风格一致	同类的界面元素是否有相同的视感和相同的操作方式	
	字体是否一致	
	是否符合广大用户使用同类软件的习惯	
适应各种水平的用户	所有界面元素都具备充分必要的键盘操作和鼠标操作	
	初学者和专家是否都有合适的方式操作这个界面	
	色盲或者色弱的用户是否能正常使用该界面	
国际化	是否使用国际通行的图标和语言	
	度量单位、日期格式、人的名字等是否符合国际惯例	
合理布局和谐色彩	界面的布局是否符合软件的功能逻辑	
	界面元素是否在水平或者垂直方向对齐	
	界面元素的尺寸是否合理？行、列的间距是否保持一致	
	是否恰当地利用窗体和空间的空白，以及分割线条	
	窗口切换、移动、改变大小时，是否界面正常	
	界面的色调是否让人感到和谐、满意	
	重要的对象是否用醒目的色彩表示	
	色彩使用是否符合行业的习惯	
个性化	是否具有与众不同的、让用户记忆深刻的界面设计	
	是否在具备必要的“一致性”的前提下突出“个性化”设计	





❖ Web页面测试工具

- **HttpUnit**: 是在Junit之上构建的测试框架, 它支持Web应用的黑盒测试和in-container容器内测试

www.httpunit.org

- **JwebUnit**是在HttpUnit上创建的一个辅助工具包, 它减少了测试Web程序所需要编写的代码。

<http://Jwebunit.sourceforge.net>

- **StrutsTestCase**: 是为测试Struts应用而在Junit基础上创建的测试框架。

<http://Strutstestcase.sourceforge.net>

- **HtmlUnit**: 是一个具有浏览器基本功能的Java组件。Html是Junit的扩展测试框架之一, 它支持Html文件, 并提供了一些API, 允许访问网页、填写表格、点击链接等。

- **Mocha(JS的测试框架)**和**Chai(JS的断言库)**

<http://htmlunit.sourceforge.net>



北京大学

Android测试工具

❖ JUnit

- 单元测试

❖ Espresso

- UI测试
- AS新建的项目中自动集成了Espresso
- <https://developer.android.com/training/testing/espresso/index.html>

❖ Monkey Test

- 随机UI测试
- 脚本录制回放
- 鲁棒性测试



unittest

❖ Python已经集成测试工具unittest，只需要import

```
import unittest

class User(object):
    def __init__(self, input_name, input_pwd):
        self.username = input_name
        self.pwd = input_pwd

class TestUser(unittest.TestCase):
    def setUp(self):
        self.user = User('XiaoMing', 'Password')
    def testUsernameNotNull(self):
        # 断言句: 判断是否为真
        self.assertTrue(self.user.username != None)
    def testUsernameEqual(self):
        # 断言句: 判断是否相等
        self.assertEqual(self.user.username, 'XiaoMing')

if __name__ == '__main__':
    unittest.main()
```

```
34659@DESKTOP-0QKCHS4 D:\Desktop
> python .\pytest.py
..
-----
Ran 2 tests in 0.001s

OK
```



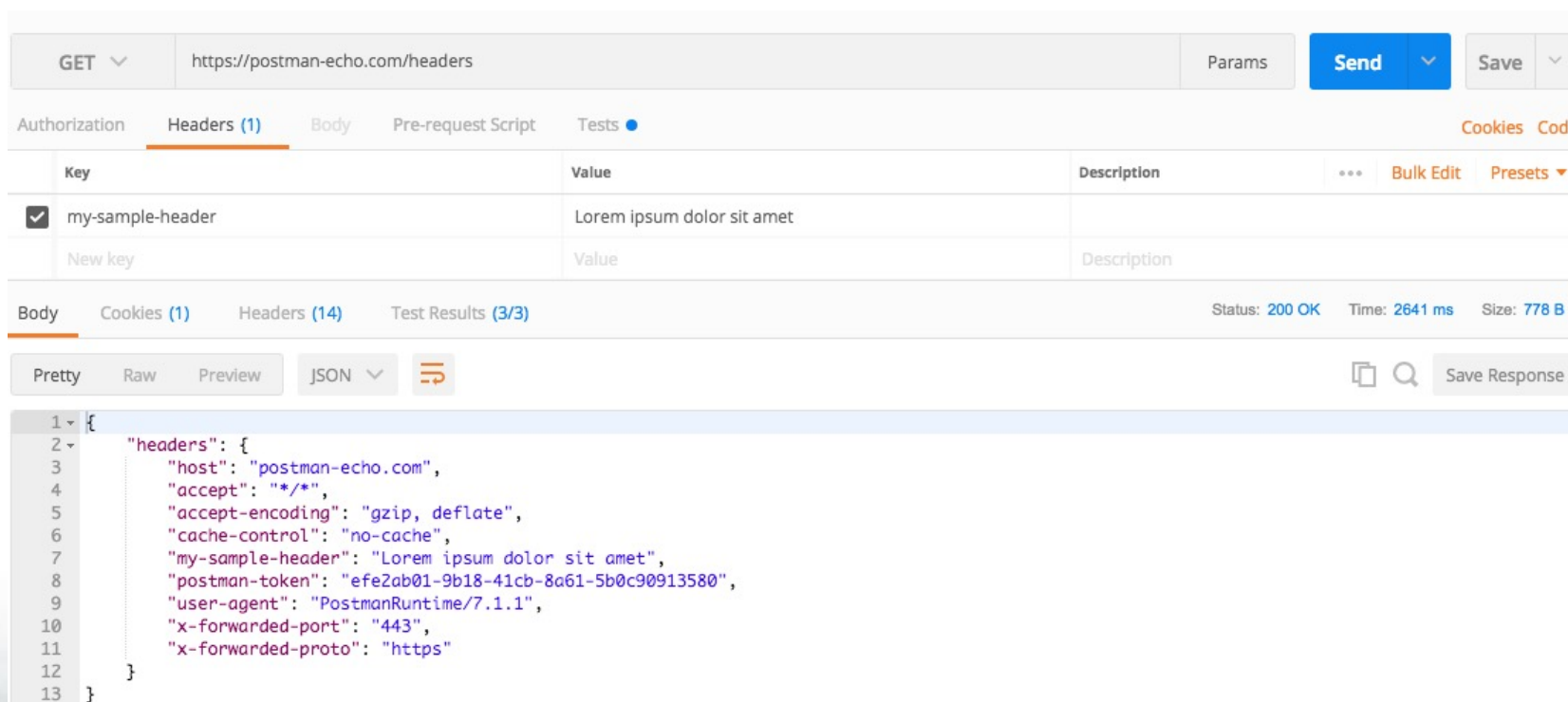
北京大学



接口测试

❖ 推荐使用postman进行测试

- 支持自定义HTTP Request
- 支持Pretty Response
- 支持Collection(分类收藏+共享)



北京大学



八、软件测试的具体实施过程

- ❖ 在完成了设计阶段的工作后，准备编码的同时，测试的准备工作也应该开始。
- ❖ 测试应该尽早地进行，及时地对编写好的单元进行测试，可以避免在编码后期进行测试时所面临的系统庞大、测试工作没有头绪的麻烦。例如，在一个会议管理系统设计工作完成后，可以进行分工，一部分人着手编码，另一部分人着手测试。
- ❖ 将要采用的测试策略可以通过讨论来确定。例如，针对一个会议管理系统，可以宏观上采用黑盒测试，微观上采用白盒测试。大的功能模块采用黑盒测试，小的构件采用白盒测试。
 - 在开发会议管理系统的过程中，每增加一个构件都采用白盒测试。
 - 通过白盒测试后，就把该构件集成到系统中。
 - 当所有的构件都集成完毕后，再采用黑盒测试。通过黑盒测试后，就可以发布该系统了。



北京大学



8.1 第一阶段--测试准备阶段

- ❖ 在这个阶段，要确定黑盒测试和白盒测试将采用的方法或工具。
- ❖ 如果要采用手工测试的方法，就应根据《系统需求规格说明书》和《系统设计说明书》编写测试用例。
- ❖ 如果采用测试工具，那么就应该熟悉特定测试工具的用法。





8.1.1 对功能设计测试用例--根据软件需求规格说明书

例如，下面是一个会议管理系统的顶层用况图

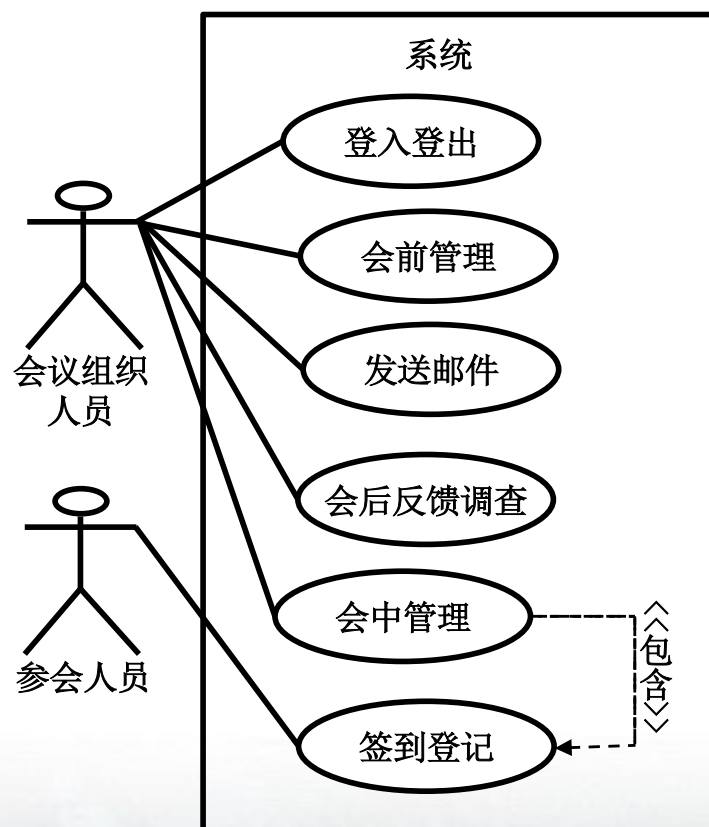
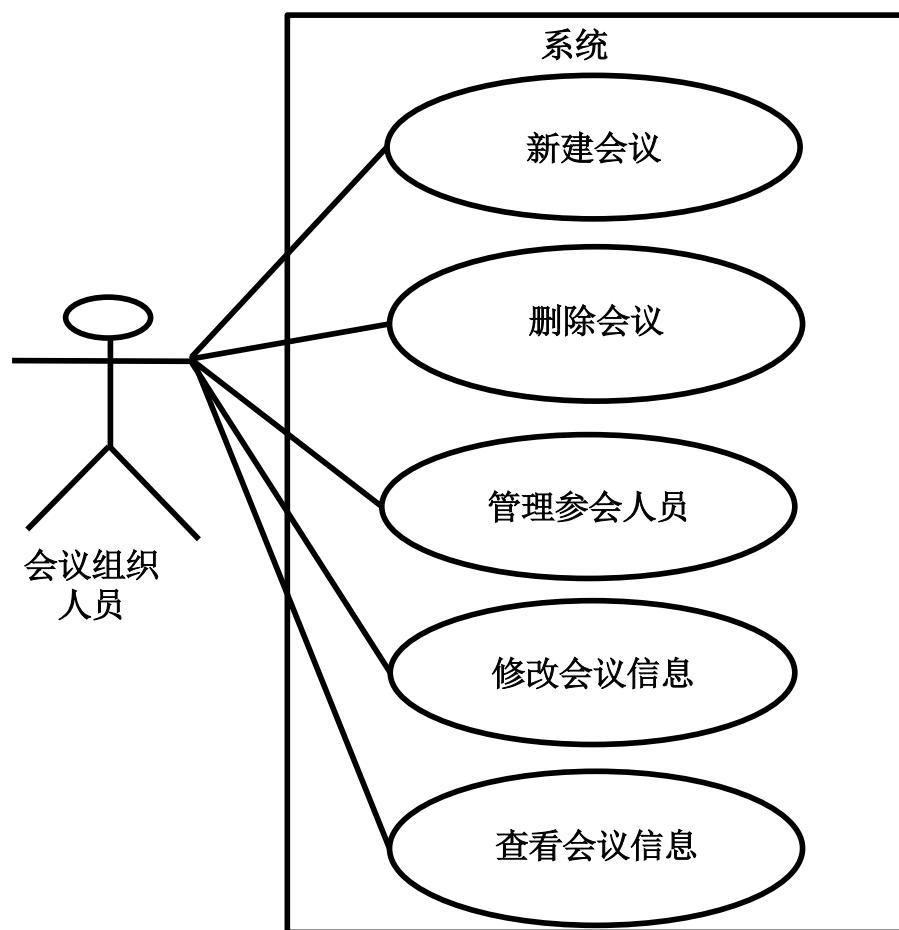


图5 顶层用况图



北京大学



注:有5个用况,
就应该有5个测试
功能点与其对应

图6 “会前管理”模块的用况图



北京大学



用例编号：2

用例名称：会前管理

用例描述：会议组织人员设置会议的初始信息，如时间、场所、主题等。同时，会议组织人员要把相关参会人员添加到该会议下。如果会议信息有改动，会议组织人员还可以对其进行更改。

前置条件：会议组织人员登录系统，并经过了身份验证。

后置条件：会议和人员信息被初始化。

活动步骤：会议组织人员登录系统。

会议组织人员新建会议。

会议组织人员录入会议的名称、场所、时间、主题、主办方等信息。

会议组织人员把相关的参会人员添加到该会议下。

扩展点：如果参会人员身份验证失败，则用例结束。

如果实际情况有变动，会议组织人员可对会议信息进行更改。

异常处理：无。



北京大学



表5 会议管理系统功能测试设计表

编号	功能名称	功能描述	输入/需要	输出/结果	发现问题	测试结果	测试人	测试时间
1*	登入登出	会议组织人员登录网站	用户名、密码、验证码	成功登入系统或显示出错信息				
2*	新建会议	会议组织人员新建会议	会议名称、场所、时间、主题、主办方	新建的会议信息存储在“会议表”中				
3	删除会议	会议组织人员删除某会议	要被删除的会议名称	“会议表”中删除有关该会议的记录				
4	管理参会人员	会议组织人员把相应的某些参会人员添加到特定会议下	参会人员ID、会议ID	“签到表”中增加了相应的记录				



编号	功能名称	功能描述	输入/需要	输出/结果	发现问题	测试结果	测试人	测试时间
5*	修改会议	会议组织人员可以对某会议的信息进行修改	新的会议信息	“会议表”中相应记录的内容被更新				
6	查看会议信息	会议组织人员可以查看会议信息	查询条件	满足查询条件的会议的基本信息				
7	发放会前通知	会议组织人员可以在会前群发邮件通知相关人员参会	参会人员的E-mail地址, 参会通知信息	邮件发送成功				
8	查看参会信息	会议组织人员可以在会中查看人员的参会状况	系统自动统计, 不需输入	参会人员到会状况的统计结果				
9	签到登记	参会人员会前进行签到登记, 会议组织人员也可以帮助参会人员登记	参会人员ID	“签到表”中相应记录的“参会状态”项内容被更新				
10	管理反馈表	会议组织人员可以新建、删除、更新反馈表	反馈表的基本信息	“反馈表”中的内容被相应地更新				
11	管理反馈表问题	会议组织人员可以对特定反馈表的问题进行新建、删除、修改	问题的基本信息	“反馈表”中的内容被相应地更新				
12	通知填写反馈表	会议组织人员发送邮件通知参会人员填写反馈表	参会人员的E-mail地址、反馈表	邮件发送成功				
13	填写反馈表	参会人员填写反馈表	反馈表	反馈表被填写				
14	查看反馈结果	会议组织人员可以查看反馈的统计结果	所有反馈表	统计结果				



8.1.2 对性能设计测试用例—根据软件需求规格说明书

表6 性能测试设计表

编号	性能名称	性能描述	输入	系统响应	发现问题	测试结果	测试时间	测试人
1	正确性需求	系统要能够把会议组织人员所创建的会议的相关信息以及添加的人员信息准确地导入数据库中	会议信息、人员信息	数据库中的信息被更新				
2	正确性需求	与会者使用IC卡签到时，系统应能正确地读取相关信息并对到会情况进行统计	IC卡	系统识别IC卡，修改“签到表”中相应记录的参会状态，更新到会情况的统计信息				
3	正确性需求	系统要能够正确地将会议通知、反馈表填写通知等发送到参会人员邮箱地址	通知、参会人员邮箱地址	系统群发邮件，且发送到正确的地址				
4	时间特性需求	在参会人员使用IC卡签到时，系统在特定的时间内做出反应	IC卡	系统应能在0.2秒内识别出参会人员并进行签到数据录入				
5	时间特性需求	查看参会人员信息、到会情况信息和反馈表信息时，系统在特定的时间内返回正确结果	查询条件	系统能在1秒内返回正确结果				



北京大学



8.2 第二阶段--单元测试阶段

- ❖ 在编码同步进行的过程中，编码人员每完成一个单元的代码，就应该用上述的白盒测试方法对其进行测试，直到所有的单元模块都被测试通过。
- ❖ “单元”的含义根据实际情况有所不同。
 - 在C语言中，单元一般指一个函数；
 - 在Java和C++里，单元一般指一个类；
 - 在图形化的软件中，单元一般指一个窗口、一个菜单等。
- ❖ 可用的单元测试工具有**C++Test**、**JUnit**、**JTest**和**.Test**、**Mocha**和**Chai**、**python的unittest**等
 - C++Test是针对C/C++的单元测试工具，自动为函数、类生成测试用例，测试驱动函数和承接函数、
 - Junit是针对Java语言的单元测试框架，继承TestCase类，就可以用Junit进行自动测试
 - JTest也是针对Java语言的单元测试，可以自动生成测试代码和测试用例，还可以检查代码书写规范，代码覆盖率等
 - .Test是针对.net框架的单元测试工具
 - Visual Studio2019自带的单元测试功能:测试各个类中的函数。按要求输入,测试与预期的结果是否吻合,如果不吻合,则单元测试结果将显示失败或其他错误信息,若成功则单元测试结果将显示“通过”
 - Mocha是JavaScript的单元测试框架，使用describe和it块，就可以进行自动测试，测试期间用Chai实现断言功能
 - unittest是针对python语言的单元测试框架，继承TestCase类，就可以用unittest进行自动测试





表7 单元测试

序号	测试函数名	函数功能	是否成功	是否更正
1	insertConventioner	将conventioner类的一个实例的内容插入到数据库的conventioner表中	是	/
2	GetUserRole	从用户输入的用户名和密码判断用户的类型	是	/
3	deleteConferenceByConfId	通过confId（会议号）删除指定的会议	是	/
4	insertConference	将conference 实例中的内容插入到数据库的Conference 表中	是	/
5	updateConference	将conference 实例中的内容更新到数据库的Conference 表中confId与实例的confId 相同的行	是	/
6	Vote	通过将相应的选项和对应的confId更新到Conference相应的表中，记录投票数	否	是





8.3 第三阶段--集成测试阶段

运用黑盒测试的方法对整个系统进行测试。

❖ 集成测试关注：

- 把各部分连接在一起，穿越模块接口的数据是否会丢失
- 一个模块的功能是否会对另一个模块的功能产生不利的影响
- 各子功能组合起来是否能达到预期的功能
- 各个模块的误差累积起来是否会放大，从而达到不可接受的程度

❖ 对于类，在集成测试阶段

- 主要检查其他模块对类的数据有没有意外的篡改，尤其是对静态的属性和方法的调用

❖ 对于前端页面，集成测试阶段

- 主要看集成进系统后，这个页面进入的前置条件是否得到其他页面的支持，同时从这个页面能否顺利转到其他页面
- 各页面对Session、Cookie等的更改会不会产生模块间的干扰，从而使一些页面出现反常行为
 - 尽可能找出Bug出现的原因



北京大学



8.8.4 第四阶段--系统测试

❖ 系统测试注意:

➤ 测试环境的搭建

- 各种程序设计语言都有强大的IDE, 集成了强大的测试环境, 在系统测试初期可以利用这些环境进行功能的测试
- 系统测试的第二阶段尽量配置真实的应用环境, 然后进行系统的性能、用户友好性方面的测试

➤ 测试过程的记录

- 认真记录问题发生的环境、性质和简要描述

❖ 系统测试包括两个方面

➤ 对功能的测试

- 对各功能点的测试
- 对某些严格要求数据正确性的模块, 可以采用等价类划分或边界值分析方法编写测试用例, 进行测试

➤ 对性能的测试



北京大学



测试报告案例:

Kernel会议管理系统测试报告

From:

吕云翔等. 软件工程课程设计. 机械工业出版社, 2009.

一、引言

1. 编写目的

本文档是在项目具体代码完成过程中, 项目团队中主要负责测试的人员记录在测试过程中发现的问题以及对于问题改进的建议。

本文档将供本系统的开发人员、维护人员查阅和使用。

2. 背景

a. 被测试软件系统的名称: “Kernel会议管理系统”。

b. 该软件的开发者: 刘浩、朱鑫、周俊、王昕鹏、周建、马识路。

c. 用户: 各个会议主办方、会议组织部门。

d. 测试环境与实际运行环境之间可能存在的差异以及这些差异对测试结果的影响。

实际运行环境: 联网的Web平台(信息管理)、局域网、桌面程序(IC卡管理)

测试环境: 单机的Windows平台、Microsoft Visual Studio 2005、SQL SERVER 2005 Express

测试环境与实际运行环境之间存在的差异、单机测试与联网的实际运行环境存在的差异以及并发操作等情况难于处理, 很难做到完全基于实际情况的测试。

3. 参考资料

- Kernel团队, 《Kernel会议管理系统需求规格说明书》
- 赵池龙、杨林、孙伟编著, 《实用软件工程(第2版)》, 电子工业出版社, 2006
- 朱三元等编著, 《软件工程技术概论》, 科学出版社, 2005
- 陈宏刚等编著, 《软件开发的科学与艺术》, 电子工业出版社, 2002
- 陈伟柱、陶文译, 《单元测试之道C#版》, 电子工业出版社, 2005

二. 测试概要

1. 测试采用的标准及技术(如表1、表2和表3所示)

表1 开始/ 中断/ 完成测试

开始/中断/完成测试	标准说明
开始测试标准	硬件环境可用且软件正确安装完成
中断测试标准	安装无法正确完成或程序的文档有相当多的失误或系统服务异常
完成测试标准	完成测试计划中的测试规划并达到程序和测试质量的目标, 并由周俊、马识路确认



表2 测试技术

测试技术	说明
编写测试用例	在产品编码阶段编写测试用例
单元测试	对具体模块及函数进行测试
集成测试	检测模块集成后的系统是否达到需求
系统测试	对业务流程及数据流的处理是否符合标准，系统对业务流的处理是否存在逻辑不严谨和错误以及是否有在不合理的标准及要求

表3 测试类型

测试类型	说明
功能测试	根据《Kernel会议管理系统需求规格说明书》和《Kernel会议管理系统设计说明书》，检查产品是否正确实现了功能
边界值测试	选择边界数据进行测试，确保系统功能正常，程序无异常
界面测试	检查界面是否美观合理
文档测试	检查文档是否足够、描述是否合理

2. 单元测试（各类函数）

利用Visual Studio 2005中自带的单元测试功能进行单元测试，测试各个类中的函数。按要求输入，测试与预期结果是否吻合，如果不吻合，则单元测试结果将显示失败或其他错误信息，若成功则单元测试结果将显示“通过”，如表4所示。

表4 单元测试

序号	测试函数名	函数功能	是否成功	是否更正
1	insertConventioner	将conventioner类的一个实例的内容插入到数据库的conventioner 表中	是	/
2	GetUserRole	从用户输入的用户名和密码判断用户的类型	是	/
3	deleteConferenceByConfId	通过confId（会议号）删除指定的会议	是	/
4	insertConference	将conference 实例中的内容插入到数据库的Conference 表中	是	/
5	updateConference	将conference 实例中的内容更新到数据库的Conference 表中confId 与实例的confId 相同的行	是	/
6	Vote	通过将相应的选项和对应的confId更新到Conference相应的表中，记录投票数	否	是



3. 集成测试

本系统的集成测试重点是页面间的跳转是否正确以及页面间的参数传递是否正确。

对于页面间跳转的测试，主要测试点击本系统中的各个页面的导航链接提示是否跳转到相应的页面。对于参数传递的测试，主要测试页面间传递参数所产生的结果是否正确。

4. 系统测试

针对Kernel会议管理系统，这里将系统测试分成了两个部分：功能测试和性能测试。

对于功能测试，按照《Kernel会议管理系统需求规格说明书》中的功能模块设计，为系统的各项功能设计了测试方案，其中对数据要求不高的功能只是提出了大致的测试要求，但对于几项对数据正确性要求比较高的功能我们按照等价类划分的方法设计了测试用例。对于每个测试用例，均以清晰的表格形式记录了测试结果（通过或不通过）、发现问题、测试人及测试时间。

对于性能测试，按照《Kernel会议管理系统需求规格说明书》对性能的要求，设计了性能测试的表格。按照这个表格对系统中比较重要的几项性能需求进行了测试。每项测试同功能测试一样均记录了发现问题、测试结果、测试人及测试时间。

三、单元测试

1. 测试conventionerManager.cs类下的函数insertConventioner(conventioner conv)

conventionerManager.cs类在App_code目录下的Manager文件夹中。

函数声明如下：

```
public static bool insertConventioner(conventioner conv)
```

函数预期实现的功能：将convetioner类的一个实例的内容插入到数据库的conventioner表中。

输入：在conventionerManager.cs的insertConventioner函数的首行添入如下代码（因为在测试conventionerManagerTest.cs中无法初始化该类）：

```
conv = new conventioneer(0, "zhoujun", "abc@126.com", "help");
```

期望输出：

```
bool expected = true;
```

测试结果：通过

2. 测试AccountManager.cs中的GetUserRole(string id, string password)

AccountManager.cs在App_code目录下的Manager文件夹中。



北京大学

函数声明如下:

```
BizLogic.UserRole GetUserRole(string id, string password)
```

函数预期实现的功能: 依据用户输入的用户名和密码判断用户的类型。

输入:

```
string id = "zhoujun"
```

```
String password = "zhoujun"
```

期望输出:

```
confUnitTest1.BizLogic_UserRoleAccessor expected = BizLogic_UserRoleAccessor.Organizer;
```

测试结果: 通过。

3. 测试conferenceManager.cs类中的deleteConferenceByConfId(int confId)

conferenceManager.cs在App_code目录下的Manager文件夹中。

函数声明如下:

```
public static bool deleteConferenceByConfId(int confid)
```

函数预期实现的功能: 通过confId(会议号)删除指定的会议。

输入:

```
int confId = 4;
```

期望输出:

```
bool expected = true;
```

测试结果: 通过。

4. 测试conferenceManager.cs类中的insertConference(conference conf)

conferenceManager.cs在App_code目录下的Manager文件夹中。

函数声明如下:

```
public static bool insertConference(conference conf)
```

函数预期实现的功能:

将conference实例中的内容插入到数据库中的Conference表中。

输入:

在 conferenceManager.cs 的 insertConference 函数的首行添入如下代码 (因为在测试 conferenceManagerTest.cs 中无法初始化该类):

```
DateTime dt = System.DateTime.Now;
```

```
conf = new conference(0, "helloBUAA", "software", dt, "software development", "BUAA");
```

期望输出:



北京大學


```
bool expected = true;
```

测试结果：通过。

5. 测试conferenceManager.cs类中的updateConference(conference conf)

conferenceManager.cs在App_code目录下的Manager文件夹中。

函数声明如下：

```
public static bool updateConference(conference conf)
```

函数预期实现的功能：将conference 实例中的内容更新到数据库的Conference表中confId 与实例的confId相同的行。

输入：

在conferenceManager.cs的updateConference函数的首行添入如下代码（因为在测试conferenceManagerTest.cs中无法初始化该类）：

```
DateTime dt = System.DateTime.Now;
```

```
conf = new conference(6, "helloChina", "software", dt, "China department", "BUAA");
```

期望输出：

```
bool expected = true;
```

测试结果：通过。

6. 测试feedBackManager.cs类中的Vote(int option, int confID)

feedBackManager.cs 类在App_code目录下的Manager文件夹中。

函数声明如下：

```
public static bool Vote(int option, int confID)
```

函数预期实现的功能：通过将相应的选项和对应的confId更新到Conference相应的表中，记录投票数。

输入：

```
int option = 'A' ;
```

```
int confID = 1;
```

期望输出：

```
bool expected = true;
```

测试结果：

Assert.AreEqual 失败。应为< True >，实际为< False >。feedBackManager.Vote未返回所需的值。

查找原因：

实际的SQL语句:

```
dbc.SqlText = "update feedback set question_total" + option + " = question_total" + option + " +  
1 " + "where question_conf_id = @ confID";
```

option 对应A、B、C、D四个字符,所以option应该为字符型char,修改为public static bool Vote(char option, int confID)。

重新构建单元测试:

输入:

```
char option = 'A';
```

```
int confID = 1;
```

期望输出:

```
bool expected = true;
```

测试结果: 通过。

四、集成测试

本系统的集成测试重点是页面间的跳转是否正确以及页面间的参数传递是否正确。对于页面间跳转的测试,主要测试点击本系统中的各个页面的导航链接提示是否跳转到相应的页面,对于参数传递的测试,主要测试页面间传递参数所产生的结果是否正确。

1. 页面跳转测试

下面根据Kernel会议管理系统列出各页面跳转的测试结果(如表5所示)。

表5 页面跳转测试结果

编号	跳转链接名称	所在页面	测试结果	测试人	测试时间
1	新建会议	MeetingList.aspx	通过	周俊	2007/12/14
2	编辑会议	MeetingHome.aspx	通过	周俊	2007/12/14
3	删除会议	MeetingHome.aspx	通过	周俊	2007/12/14
4	管理参会人员	MeetingHome.aspx	通过	周俊	2007/12/14
5	配置邮箱账户	首页	通过	周俊	2007/12/14
6	管理反馈表问题	MeetingHome.aspx	不通过	周俊	2007/12/14
7	添加编辑反馈表问题	ManageQuestions.aspx	通过	周俊	2007/12/14
8	预览反馈表	ManageQuestions.aspx	不通过	周俊	2007/12/14
9	填写反馈表	邮件中链接	通过	周俊	2007/12/14
10	查看反馈结果	MeetingHome.aspx	通过	周俊	2007/12/14



2. 参数传递测试

下面根据Kernel会议管理系统列出参数传递的测试结果（如表6 所示）。参数传递主要测试的是页面间跳转的新页面依赖于所传递的参数的页面跳转。

表6 参数传递测试结果

编号	参数传递链接	所在页面	预期结果	发现问题	测试结果	测试人	测试时间
1	新建会议	MeetingList.aspx	会议名称、主题、时间、场所等信息均为空白		通过	周俊	2007/12/14
2	编辑会议	MeetingHome.aspx	会议名称、主题、时间、场所等信息与所点击的会议相同	会议信息均为空白	不通过	周俊	2007/12/14
3	删除会议	MeetingHome.aspx	删除相应的会议		通过	周俊	2007/12/14
4	管理反馈表问题	MeetingHome.aspx	显示会议对应的各反馈问题		通过	周俊	2007/12/14
5	添加编辑反馈表问题	ManageQuestions.aspx	编辑时，显示对应的问题信息，添加时，问题信息为空白		通过	周俊	2007/12/14
6	预览反馈表	ManageQuesstions.aspx	反馈表的内容与所点击的反馈表相同	显示的是其他反馈表的内容	不通过	周俊	2007/12/14
7	填写反馈表	邮件中链接	反馈表的内容与所点击的反馈表相同		通过	周俊	2007/12/14
8	查看反馈结果	MeetingHome.aspx	反馈表的内容与所点击的反馈表相同		通过	周俊	2007/12/14

五、系统测试

系统测试是按照贴近实际的场景进行部署后对系统进行的一系列严格有效的测试。它主要包括两个方面，一是功能测试，二是性能测试。

1. 对功能的测试

下面根据《Kernel会议管理系统设计说明书》所列出的各功能模块，对于有些与数据库中表内的数据项直接相关，且数据项对其类型有较高的要求的功能模块，在功能测试设计表（表7）之后进行了较详细的测试，以契合这些模块对数据正确性的高要求。

编号	功能名称	功能描述	输入/需要	输出/结果	发现问题	测试结果	测试人	测试时间
1*	登入登出	会议组织人员登录网站	用户名、密码、验证码	成功登入系统或显示出错信息	见后表			
2*	新建会议	会议组织人员新建会议	会议名称、场所、时间、主题、主办方	新建的会议信息存储在“会议表”中	见后表			
3	删除会议	会议组织人员删除某会议	单击一个会议对应的删除按钮，并在弹出的提示框中点击确定	“会议表”中删除有关该会议的记录	无	通过	朱鑫	2007/12/16
4	管理参会人员	会议组织人员把相应的某些参会人员添加到特定会议下	参会人员ID、会议ID	“签到表”中增加了相应的记录	无	通过	朱鑫	2007/12/16
5*	修改会议	会议组织人员可以对某会议的信息进行修改	新的会议信息	“会议表”中相应记录的内容被更新	见后表			
6	查看会议信息	会议组织人员可以查看会议信息	点击 MeetingList.aspx 中的每个会议对应的链接	显示对应会议的基本信息	无	通过	朱鑫	2007/12/16
7	发放会前通知	会议组织人员可以在会前群发邮件通知相关人员参会	参会人员的E-mail地址，参会通知信息	邮件发送成功	未测试			
8	查看参会信息	会议组织人员可以在会中查看人员的参会状况	系统自动统计，不需输入	参会人员到会状况的统计结果	未测试			
9	签到登记	参会人员会前进行签到登记，会议组织人员也可以帮助参会人员登记	参会人员ID	“签到表”中相应记录的“参会状态”项内容被更新	未测试			

编号	功能名称	功能描述	输入/需要	输出/结果	发现问题	测试结果	测试人	测试时间
10	管理反馈表	会议组织人员可以新建、删除、更新反馈表	反馈表的基本信息	“反馈表”中的内容被相应地更新				
11	管理反馈表问题	会议组织人员可以对特定反馈表的问题进行新建、删除、修改	问题的基本信息	“反馈表”中的内容被相应地更新				
12	通知填写反馈表	会议组织人员发送邮件通知参会人员填写反馈表	参会人员的E-mail地址、反馈表	邮件发送成功				
13	填写反馈表	参会人员填写反馈表	反馈表	反馈表被填写				
14	查看反馈结果	会议组织人员可以查看反馈的统计结果	所有反馈表	统计结果				

注：每个功能与《Kernel会议管理系统需求规格说明书》中的用例相对应。测试结果为“通过”或“不通过”。编号中带的功能测试项在下文中有详细的功能测试设计及结果。

下面是对某些严格要求数据正确性的模块的测试设计及测试结果，对于这些模块采用了等价类划分的测试方法。

- “登入登出”功能的测试设计

表8 等价类划分

输入条件	有效等价类	编号	无效等价类	编号
用户名	用户名与数据库中的用户名一致	1	用户名与数据库中的用户名不一致	4
密码	密码与数据库中的密码一致	2	密码与数据库中的密码不一致	5
验证码	输入验证码正确	3	输入验证码不正确	6



表9 测试用例设计

用例编号	输入			预期输出	覆盖范围
	用户名	密码	验证码		
1.1	Kernel	xyz	与系统提示一致	成功登录到系统	1, 2, 3
1.2	abcd	xyz	与系统提示一致	登录失败	4
1.3	kernel	12de	与系统提示一致	登录失败	5
1.4	kernel	xyz	与系统提示不一致	登录失败	6

注：假设数据库中包含用户名和密码分别是kernel和xyz的会议组织人员信息。

表10 测试结果记录

用例编号	预期结果	实际结果	测试结果（通过/不通过）	测试日期	测试人
1.1	登录成功	登录成功	通过	2007/12/16	朱鑫
1.2	登录失败	登录失败	通过	2007/12/16	朱鑫
1.3	登录失败	登录失败	通过	2007/12/16	朱鑫
1.4	登录失败	登录成功	不通过	2007/12/16	朱鑫

• “新建会议”功能的测试设计

表11 等价类划分

输入条件	有效等价类	编号	无效等价类	编号
会议名称	长度在1~32之间	1	长度为0，即空	6
			长度大于32	7
场所	长度在1~32之间	2	长度为0，即空	8
			长度大于32	9
时间	日期类型	3	非日期类型	10
主题	长度在1~60之间	4	长度为0，即空	11
			长度大于60	12
主办方	长度在1~32之间	5	长度为0，即空	13
			长度大于32	14

表12 测试用例设计

用例编号	输入					预期输出	覆盖范围
	会议名称	场所	时间	主题	主办方		
2.1	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	新建会议成功	1, 2, 3, 4, 5
2.2	空	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	新建会议失败	6
2.3	北航*****会议（长度超过32）	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	新建会议失败	7
2.4	北航党委年终会议	空	2007/12/16	对本年度的党建工作做出总结	北航党委	新建会议失败	8

(续)

用例编号	输入					预期输出	覆盖范围
	会议名称	场所	时间	主题	主办方		
2.5	北航党委年终会议	*****会议中心 (长度超过32)	2007/12/16	对本年度的党建工作做出总结	北航党委	新建会议失败	9
2.6	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	新建会议失败	10
2.7	北航党委年终会议	如心会议中心	2007/12/16		北航党委	新建会议失败	11
2.8	北航党委年终会议	如心会议中心	2007/12/16	***** (长度超过60)	北航党委	新建会议失败	12
2.9	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结		新建会议失败	13
2.10	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结	***** (长度超过32)	新建会议失败	14

表13 测试结果记录

用例编号	预期结果	实际结果	测试结果 (通过/不通过)	测试日期	测试人
2.1	新建会议成功	新建会议成功	通过	2007/12/16	朱鑫
2.2	新建会议失败	新建会议成功, 但由于会议名称为空, 在MeetingList.aspx中新建的会议未显示	不通过	2007/12/16	朱鑫
2.3	新建会议失败	新建会议失败	通过	2007/12/16	朱鑫
2.4	新建会议失败	新建会议成功	不通过	2007/12/16	朱鑫
2.5	新建会议失败	新建会议失败	通过	2007/12/16	朱鑫
2.6	新建会议失败	新建会议失败, 但没有提示信息	不通过	2007/12/16	朱鑫
2.7	新建会议失败	新建会议成功	不通过	2007/12/16	朱鑫
2.8	新建会议失败	新建会议失败, 但没有显示提示信息	不通过	2007/12/16	朱鑫
2.9	新建会议失败	新建会议成功	不通过	2007/12/16	朱鑫
2.10	新建会议失败	新建会议失败, 但没有显示提示信息	不通过	2007/12/16	朱鑫



- “修改会议信息”功能的测试设计
- 该功能的测试设计与“新建会议”功能的测试设计相似，因为新输入的会议信息仍然必须满足会议名称、时间、主题、主办方、场所的类型限制。

表14 等价类划分

输入条件	有效等价类	编号	无效等价类	编号
会议名称	长度在1~32之间	1	长度为0，即空	6
			长度大于32	7
场所	长度在1~32之间	2	长度为0，即空	8
			长度大于32	9
时间	日期类型	3	非日期类型	10
主题	长度在1~60之间	4	长度为0，即空	11
			长度大于60	12
主办方	长度在1~32之间	5	长度为0，即空	13
			长度大于32	14

表15 测试用例设计

用例编号	输入					预期输出	覆盖范围
	会议名称	场所	时间	主题	主办方		
5.1	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	修改会议信息成功	1, 2, 3, 4, 5
5.2	空	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	修改会议信息失败	6
5.3	北航*****会议 (长度超过32)	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	修改会议信息失败	7
5.4	北航党委年终会议	空	2007/12/16	对本年度的党建工作做出总结	北航党委	修改会议信息失败	8
5.5	北航党委年终会议	*****会议中心 (长度超过32)	2007/12/16	对本年度的党建工作做出总结	北航党委	修改会议信息失败	9
5.6	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结	北航党委	修改会议信息失败	10
5.7	北航党委年终会议	如心会议中心	2007/12/16		北航党委	修改会议信息失败	11

(续)

用例编号	输入					预期输出	覆盖范围
	会议名称	场所	时间	主题	主办方		
5.8	北航党委年终会议	如心会议中心	2007/12/16	*****（长度超过60）	北航党委	修改会议信息失败	12
5.9	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结		修改会议信息失败	13
5.10	北航党委年终会议	如心会议中心	2007/12/16	对本年度的党建工作做出总结	*****（长度超过32）	修改会议信息失败	14

表16 测试结果记录

用例编号	预期结果	实际结果	测试结果（通过/不通过）	测试日期	测试人
2.1	新建会议成功	新建会议成功	通过	2007/12/16	朱鑫
2.2	新建会议失败	新建会议成功，但由于会议名称为空，在MeetingList.aspx中新建的会议未显示	不通过	2007/12/16	朱鑫
2.3	新建会议失败	新建会议失败	通过	2007/12/16	朱鑫
2.4	新建会议失败	新建会议成功	不通过	2007/12/16	朱鑫
2.5	新建会议失败	新建会议失败	通过	2007/12/16	朱鑫
2.6	新建会议失败	新建会议失败，但没有提示信息	不通过	2007/12/16	朱鑫
2.7	新建会议失败	新建会议成功	不通过	2007/12/16	朱鑫
2.8	新建会议失败	新建会议失败，但没有显示提示信息	不通过	2007/12/16	朱鑫
2.9	新建会议失败	新建会议成功	不通过	2007/12/16	朱鑫
2.10	新建会议失败	新建会议失败，但没有显示提示信息	不通过	2007/12/16	朱鑫

2. 对性能的测试

表17 性能测试

编号	性能名称	性能描述	输入	系统响应	发现问题	测试结果	测试时间	测试人
1	正确性需求	系统要能够把会议组织人员所创建的会议的相关信息以及添加的人员信息准确地导入数据库中	会议信息、人员信息	数据库中的信息被更新	无	通过	2007/12/16	朱鑫



(续)

编号	性能名称	性能描述	输入	系统响应	发现问题	测试结果	测试时间	测试人
2	正确性需求	与会者使用IC卡签到时，系统应能正确地读取相关信息并对到会情况进行统计	IC卡	系统识别IC卡，修改“签到表”中相应记录的参会状态，更新到会情况的统计信息	无	通过	2007/12/16	朱鑫
3	正确性需求	系统要能够正确地将会议通知、反馈表填写通知等发送到参会人员邮箱地址	通知、参会人员邮箱地址	系统群发邮件，且发送到正确的地址	无	通过	2007/12/16	朱鑫
4	时间特性需求	在参会人员使用IC卡签到时，系统在特定的时间内做出反应	IC卡	系统应能在0.2秒内识别出参会人员并进行签到数据录入	无	通过	2007/12/16	朱鑫
5	时间特性需求	查看参会人员信息、到会情况信息和反馈表信息时，系统在特定的时间内返回正确结果	查询条件	系统能在1秒内返回正确结果	无	通过	2007/12/16	朱鑫



北京大学