Software Testing Techniques

-- 软件测试技术

刘超
北京航空航天大学软件工程研究所
2015年12月

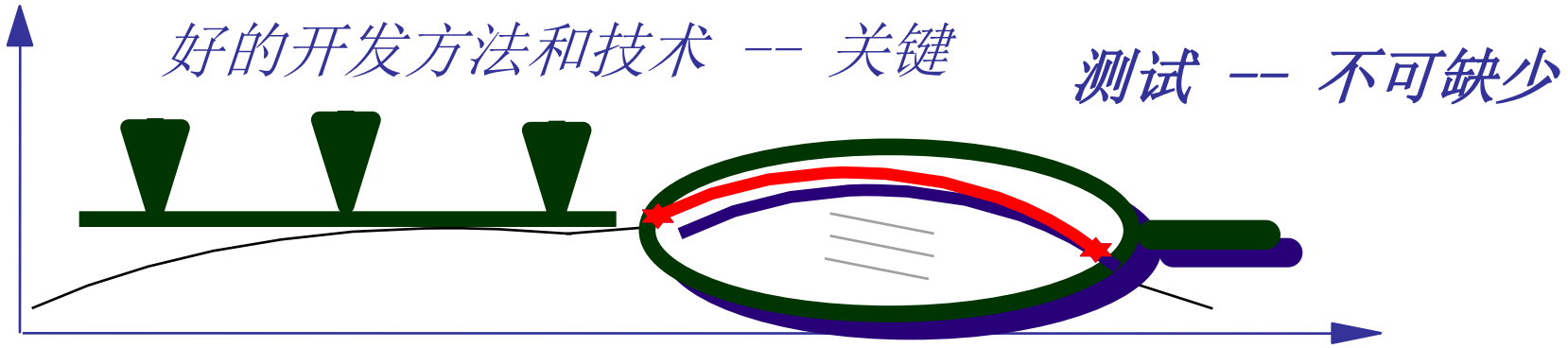
刘超：liuchao@buaa.edu.cn
电话：(10)82317496

软件测试的目的

- 证明程序的正确性 -- 除非仅处理有限种情况
- 检查系统是否满足需求 -- 期望目标
- 发现程序错误 -- 直接目标

软件测试的重要性

- 软件质量的重要性 -- 不言而喻
- 软件质量保证的难度 -- 众所周知
- 现实问题 工程问题 理论问题
- 保证和提高软件质量 -- 两种途径
- 尽量在开发期间减少错误
- 通过分析和测试发现和纠正错误



北航软件所

软件测试的重要性

- 软件测试是保证软件质量和可靠性的关键技术手段
- 在软件开发的瀑布模型中，软件测试被视为是紧随代码实现之后、揭示软件中的错误、保证和提高软件质量的重要阶段
- 在其它各种软件工程开发模型中，如快速原型、螺旋模型、以及面向对象的软件开发模型等，也都明确确认了软件测试的重要性和不可取代的地位。

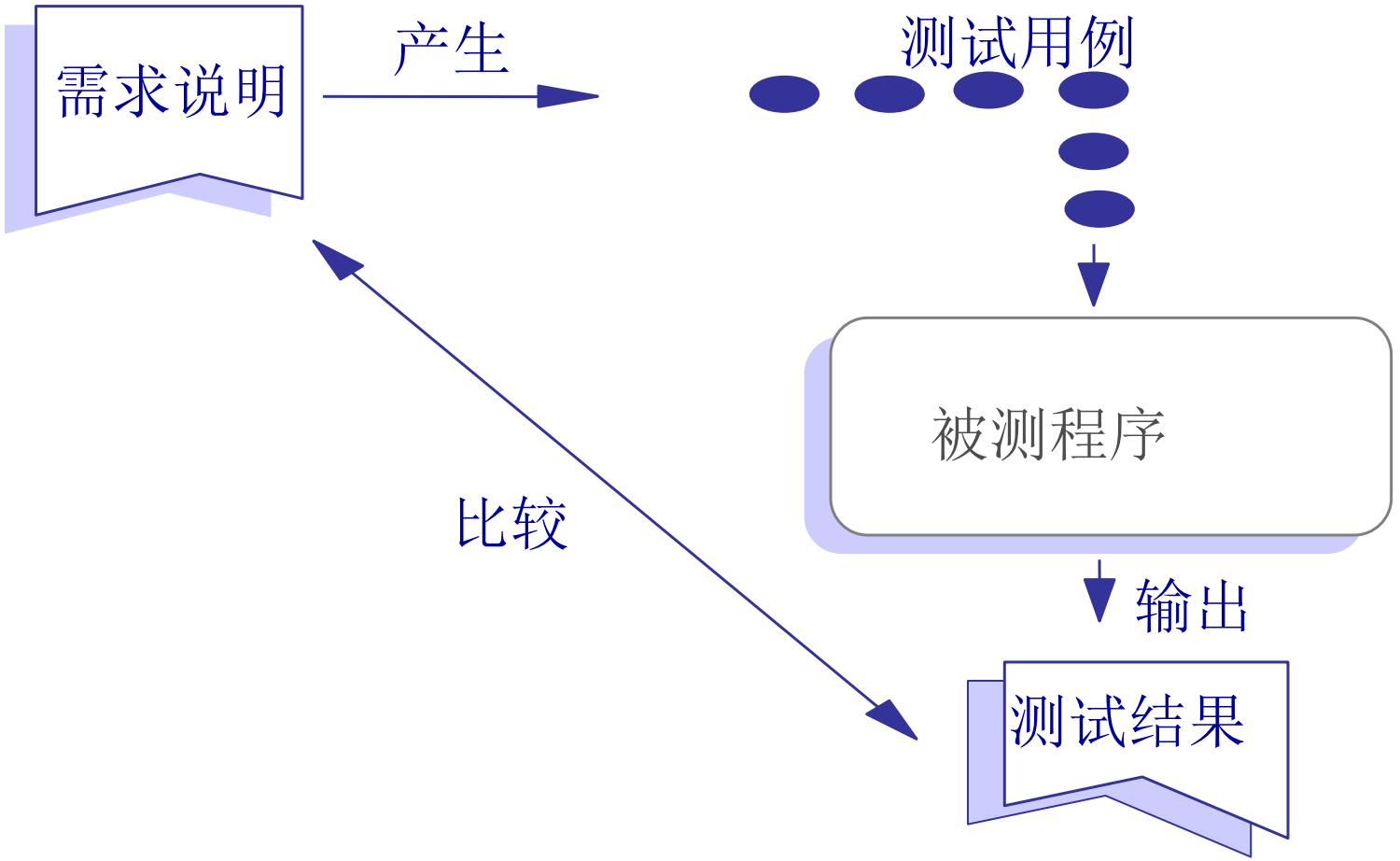
测试用例

- 发现错误可能性大的输入数据/操作
- 结果是可判定的
- 测试的执行过程和结果是可再现的
- 一个好的测试用例在于能够发现至今尚未发现的错误
- 一个成功的测试是发现了至今为发现的错误的测试

测试的手段

- 静态分析
  - 人工走查
  - 逆向分析
    - 如：引用分析、算法分析等
- 动态执行
  - 功能测试(黑盒子)：
    - 等价类、因果图、边界、强度等
  - 结构测试(白盒子)：
    - 语句测试、分支测试、条件测试、路径测试等

功能测试



黑盒子测试 -- 基本测试策略

- 正常情况
- 非正常情况
- 边界情况
- 非法情况
- 极端情况(强度测试)
- 性能测试
- 兼容性 用户友好性 ...
- 测试准则: 何时结束? 覆盖度?
- 等价类划分
- 因果图:
  - 例: 命令行 cmdname arg1 arg2 ...

软件可靠性

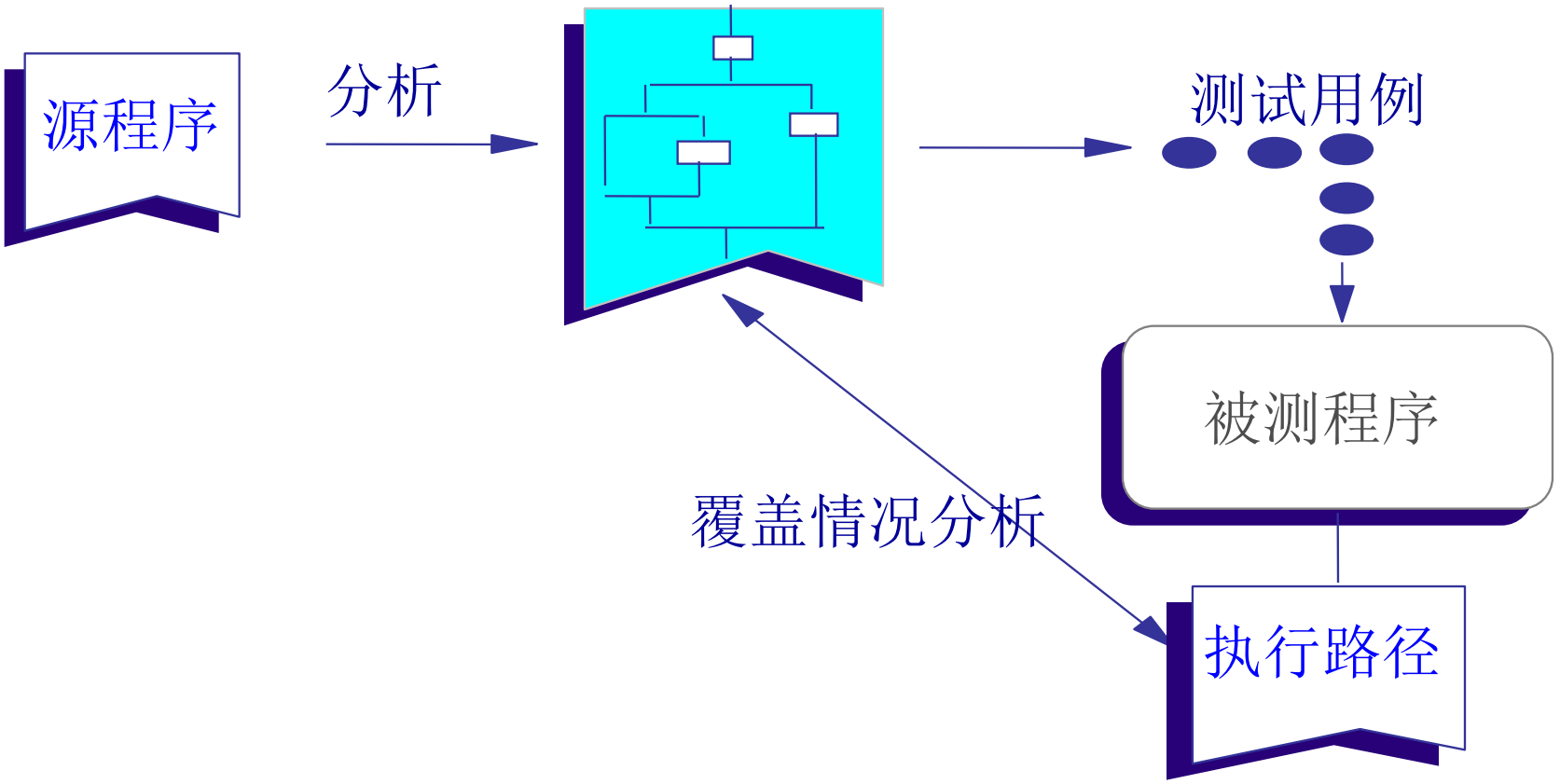
- 软件可靠性是软件在给定的时间间隔及给定的环境条件下，按设计要求，成功地运行程序的概率。
- 平均失效等待时间：

- 失效率为常数时，

  MTTF = 1/失效率
- 两次相继失效的平均时间
  - 当n很大时，第n次和第n+1次失效之间的平均时间，对于失效率为常数且修复时间很短的情况：

    MTBF = MTTF

结构测试-程序的内部逻辑



白盒子测试

- 静态分析
  - 控制流分析
    - 模块调用/被调用关系
    - 模块控制流程图
  - 数据流分析
    - 变量的定义 赋值与引用情况
  - 复杂度分析
    - 圈复杂度(Cyclematic Complexity,McCabe)
    - 科学复杂度(Science Complexity,Halstead)
  - 源代码走查(Walkthrough/Inspection)

白盒子方法：动态测试

- 运行程序，并覆盖程序中的基本控制结构
- 测试覆盖准则
  - SC1:语句覆盖测试
  - SC2:分支覆盖测试
- 其它:
  - 逻辑路径:循环--进入/不进入循环体
  - 条件覆盖

```
if( x != 0 )/* 错! 应为 x==0 */
  x = 1;
z = y/x;
```

```
while( i < n )
    s  +=  a[ i ];
```

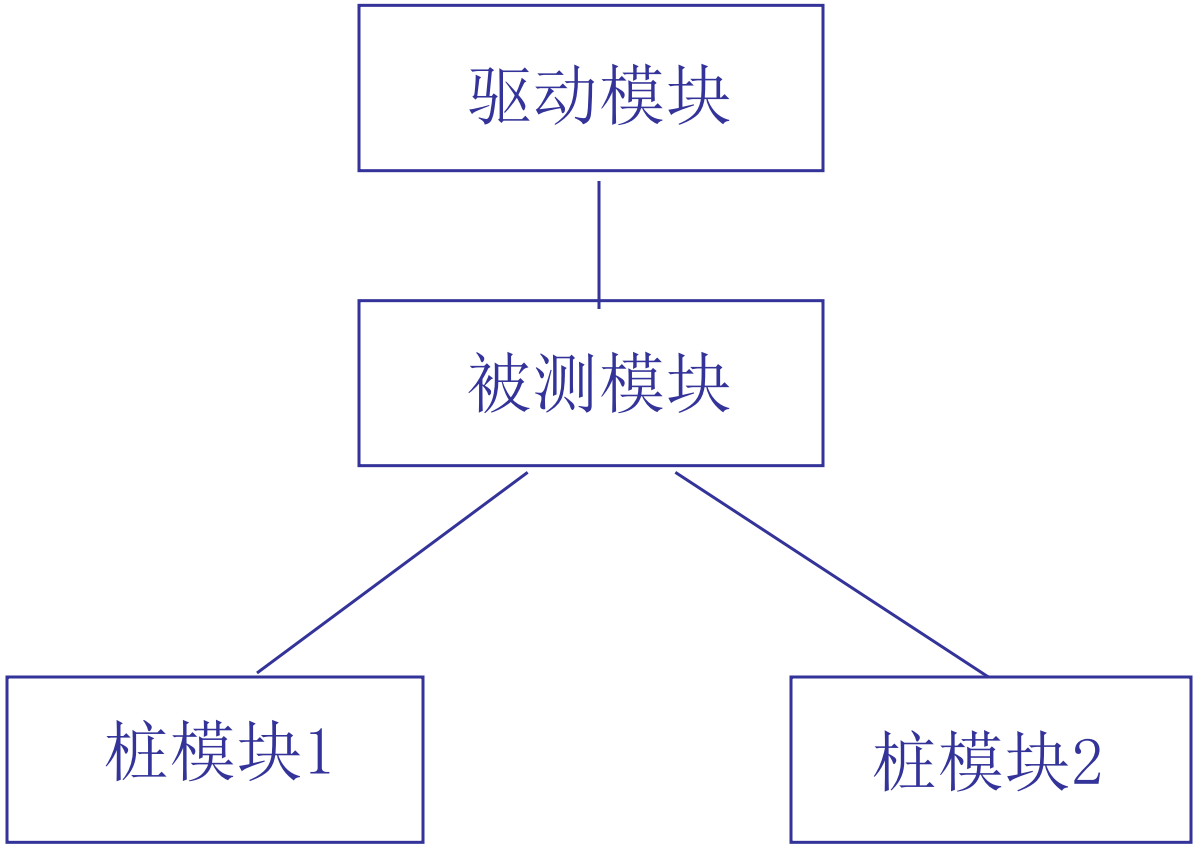
```
if( x>0 && y > 0 || x<0 && y<0 )
    z = x * y;
else
    z = - x * y;
```

北航软件所

结构测试覆盖准则之间的关系

- 语句覆盖
- 判定覆盖（分支覆盖）
- 条件覆盖
- 判定/条件覆盖
- 多重条件（条件组合）测试
- 路径覆盖（逻辑路径）

单元测试



McCabe圈复杂度

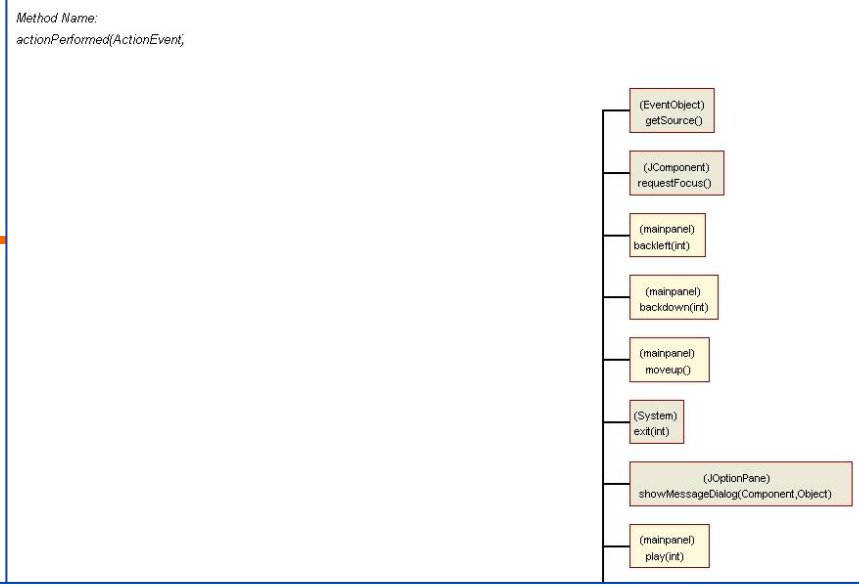
- 对于一个单入口、单出口的连通图：

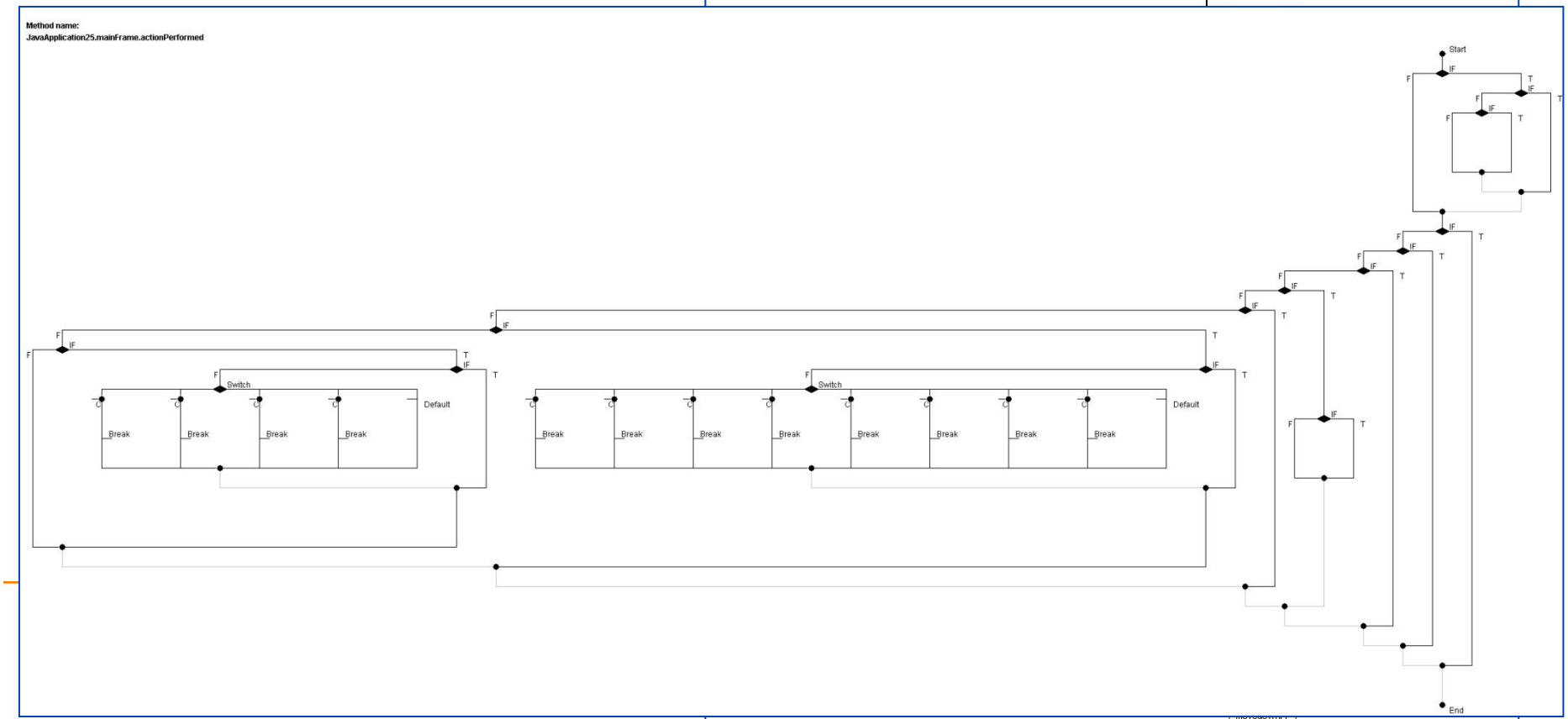
  圈复杂度 = 弧数 - 结点数 + 2

  圈复杂度 = 分支点数 + 1

例1 PushingBox

- 函数过于复杂！
- 都少个分支？





例2 学生注册系统SRS（Student Registration System）

我们承担了开发一个学生注册系统的项目（SRS）。该系统允许学生在大学的校园网络上进行在线注册每一个学期的课程，也可以用于跟踪学生的学习进展，直到其获得学位。

当学生被大学录取后，学生便需在SRS中建立学习计划，即确定为满足特定学位程序所需要的课程，并选择一位导师。SRS要检验学生所提出的学习计划是否满足他/她所修学位的要求。

一旦建立了学习计划，则在以后每个学期的注册期间，学生都可以在线查看课程计划，选择要选修的课程，如果课程有多名教授讲授，则还可以指定期望的课程班和授课时间（每周星期几，每天什么时间听课）。

SRS要检查对学生选择的课程进行必要条件的检查：

（1）参考学生已完成课程的成绩单（学生随时可以查看自己的成绩单），检查学生是否已经通过所选课程的预修课程，并取得必要的成绩；

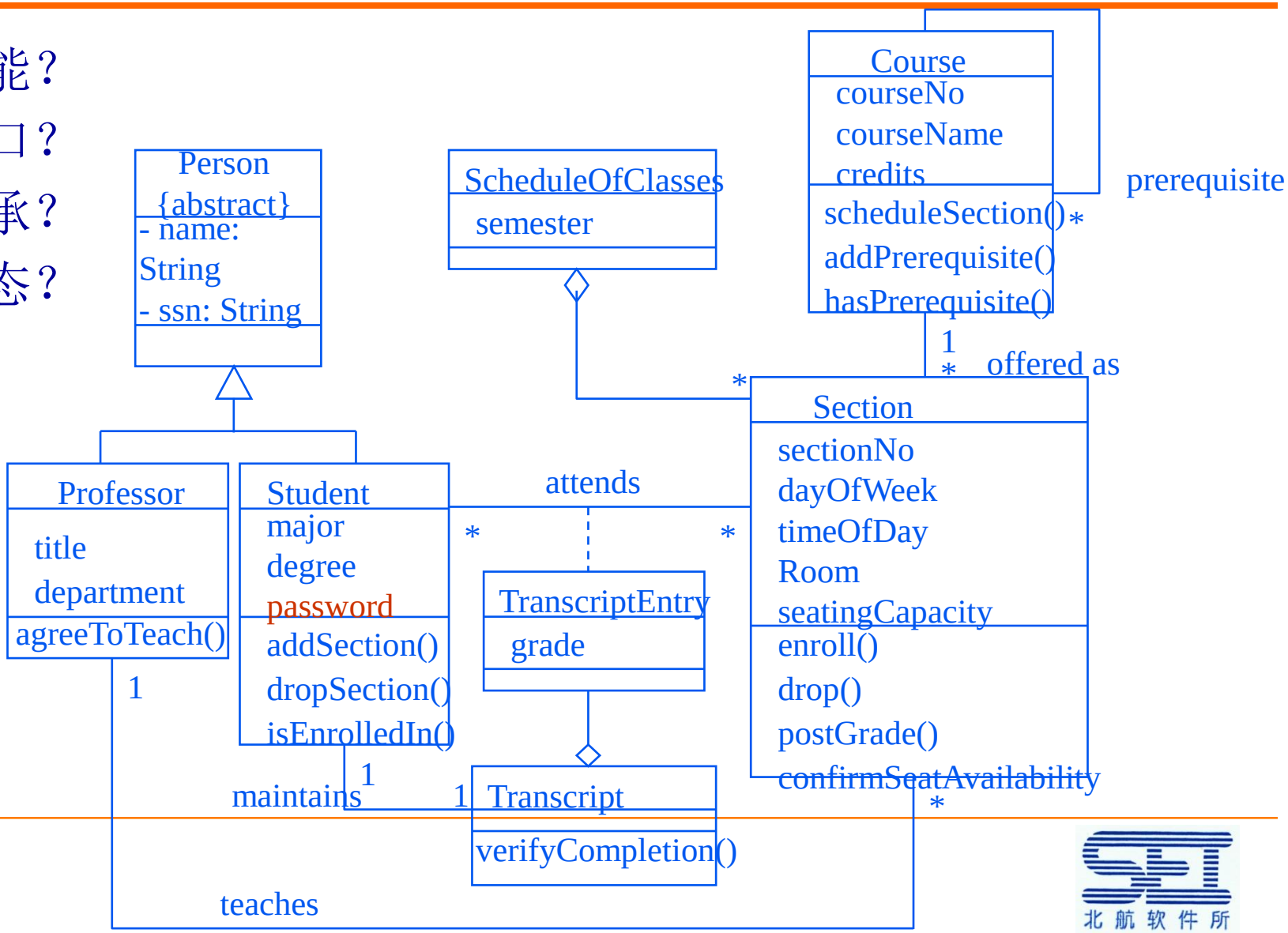
（2）该课程满足该学生学习计划要求之一；

（3）该课程班中仍有空位。

只有当上述三个条件都满足时，学生的选课请求才被接受。

问题1： SRS的“单元测试”（类及其方法）

- 功能？
- 接口？
- 继承？
- 多态？



问题2：白盒

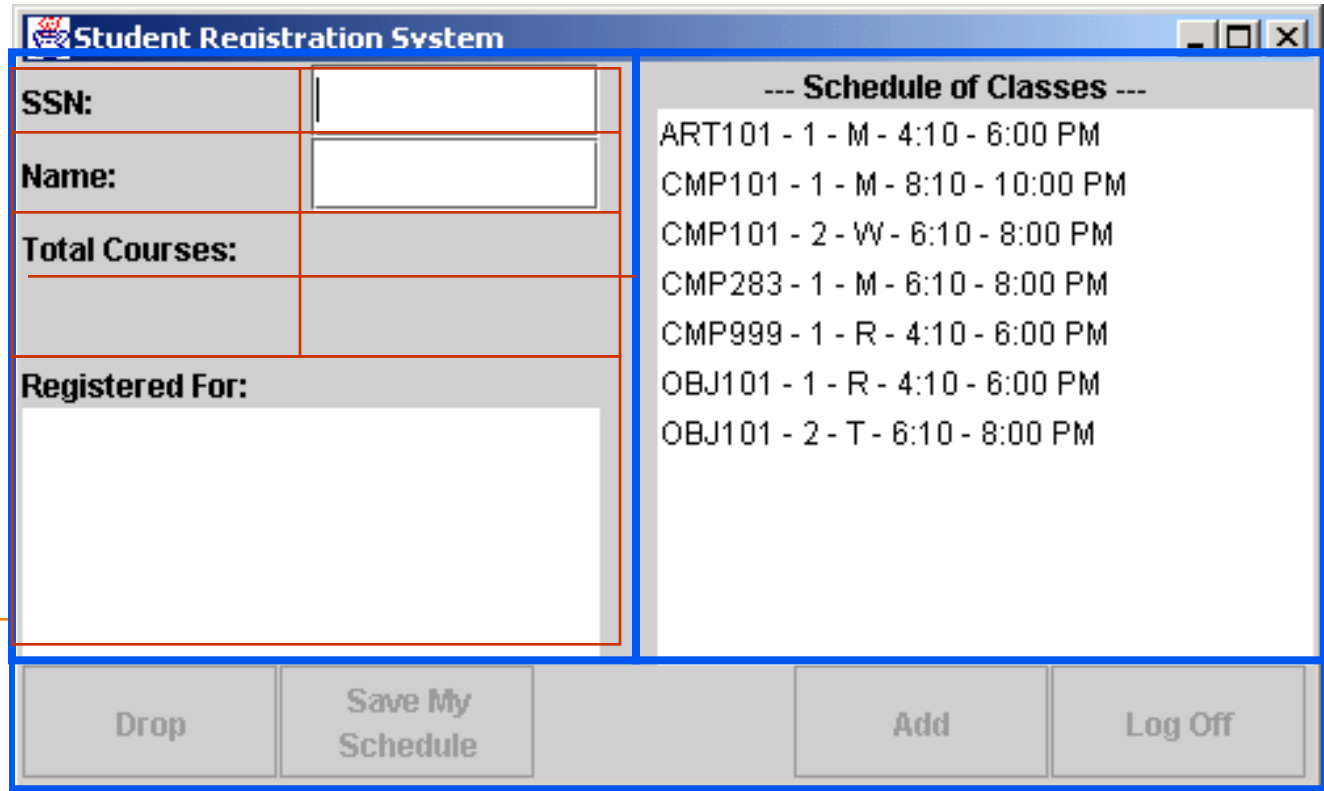
- 语句覆盖
- 分支覆盖

- 异常处理？
- 多态影响？

```
public abstract class CollectionWrapper {
    // attributes omitted
   public boolean initializeObjects(String pathToFile, boolean primary) {
      // variables omitted
      try {
         // Open the file.
         bIn = new BufferedReader(new FileReader(pathToFile));
         line = bIn.readLine();
         while (line != null) {
            if (primary)
               parseData(line);// for basic format of the line
            else
               parseData2(line);// for the second format of the line
            line = bIn.readLine();
         }
         bIn.close();
      }
      catch (FileNotFoundException f) {
         outcome = false;
      }
      catch (IOException i) {
         outcome = false;
      }
      return outcome;
   }
   public abstract void parseData(String line);
   public abstract void parseData2(String line);
}
```

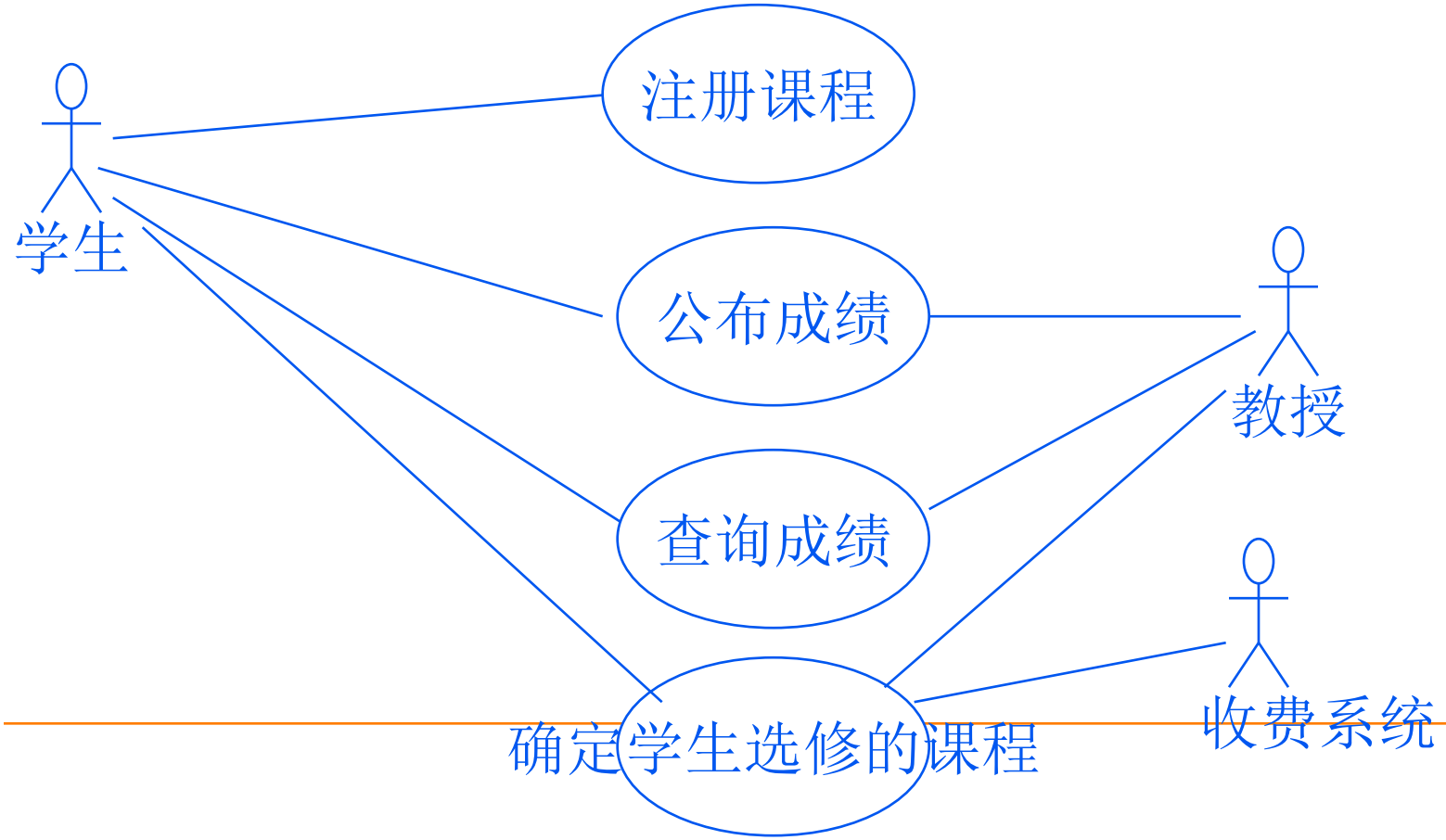
问题3：软件集成测试

- 功能
  - 用户操作
- 安装
  - 环境
- 性能
- 安全



问题4：基于需求的测试？

- 用例：
  - 使用者？典型使用场景？可能的、异常的场景？



测试的代价

- 开发费用的40-60%, 甚至80% !!??
- 为什么?
  - 有多少测试项?
  - 有多少运行平台?
  - 硬件/软件环境
  - 环境设置
  - 环境变化
  - 测试周期: 测试准备 测试实施 回归测试
  - 测试资源: 人力 设备 场地 消耗 ...

测试的“心理”障碍

- 涉及的人
  - 测试人员(QE&QA) 开发人员 设计人员 管理人员
  - 用户
- 不同人的心理
- 不同人之间的交流和沟通中的主要障碍
  - 角色差异
  - 心理差异
  - 专业背景差异
  - "文化"背景差异

软件测试过程与方法

- 软件质量问题
- 软件测试的目的
- 软件测试的手段
- 软件测试的代价
- 软件测试的"心理学"问题
- 软件测试阶段性与过程性
- 过程的控制与管理
- 软件测试的自动化技术和工具

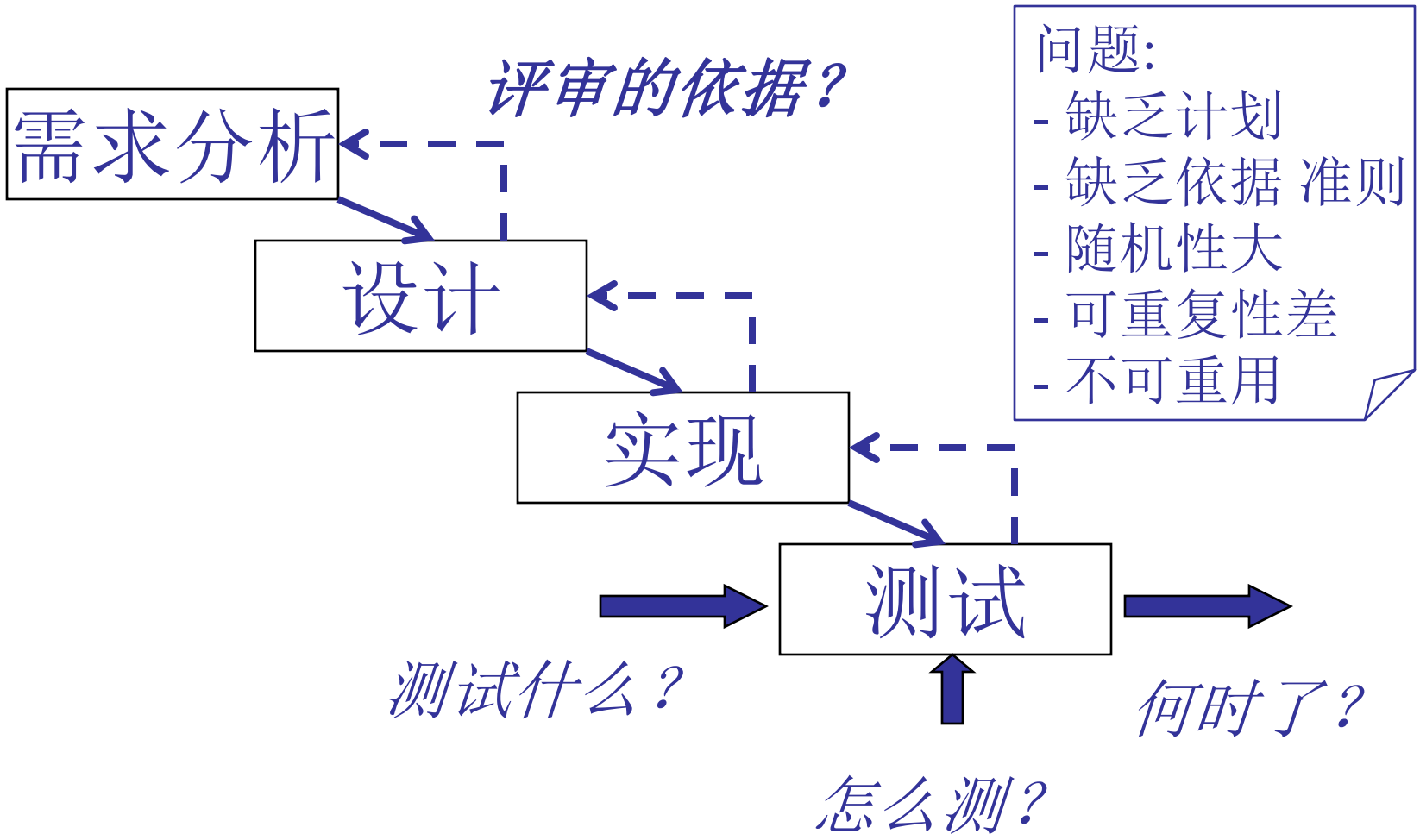
软件测试的基本原则

- 执行结果的可预见性(和可描述的)
- 第三者测试原则: Users, Developpers, Testes(Quality Engineers)
  - programmers <--> testers
  - development units <--> QE(QA) units
- 彻底检查每一个测试结果
- 非法的和非预期的情况的测试
- 做了该做的，没做不该做的
- 测试用例的保存、积累和重用
- 总假定程序是有错的
- 一段程序中存在的错误数与其中已发现的错误数成正比

软件测试的基本任务与时机

- 测试的任务
  - 运行环境? 程序执行? 测试用例? 测试步骤? 结果采集和记录? 结果分析? 错误报告? 错误定位?
- 测试的时机
  - 编程之后
  - 编程期间？
  - 编程之前？

软件测试的阶段性

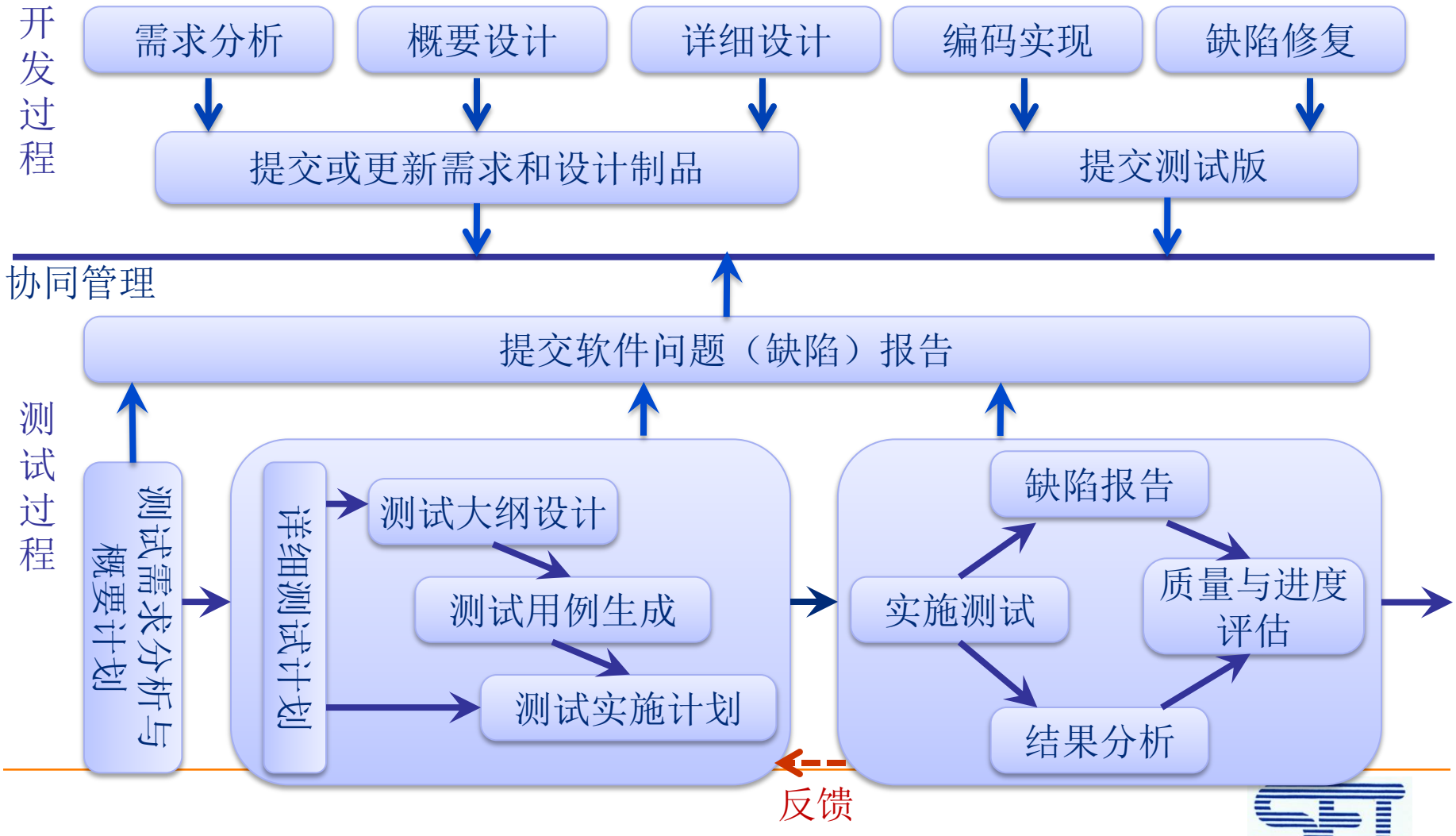


北航软件所

软件测试过程

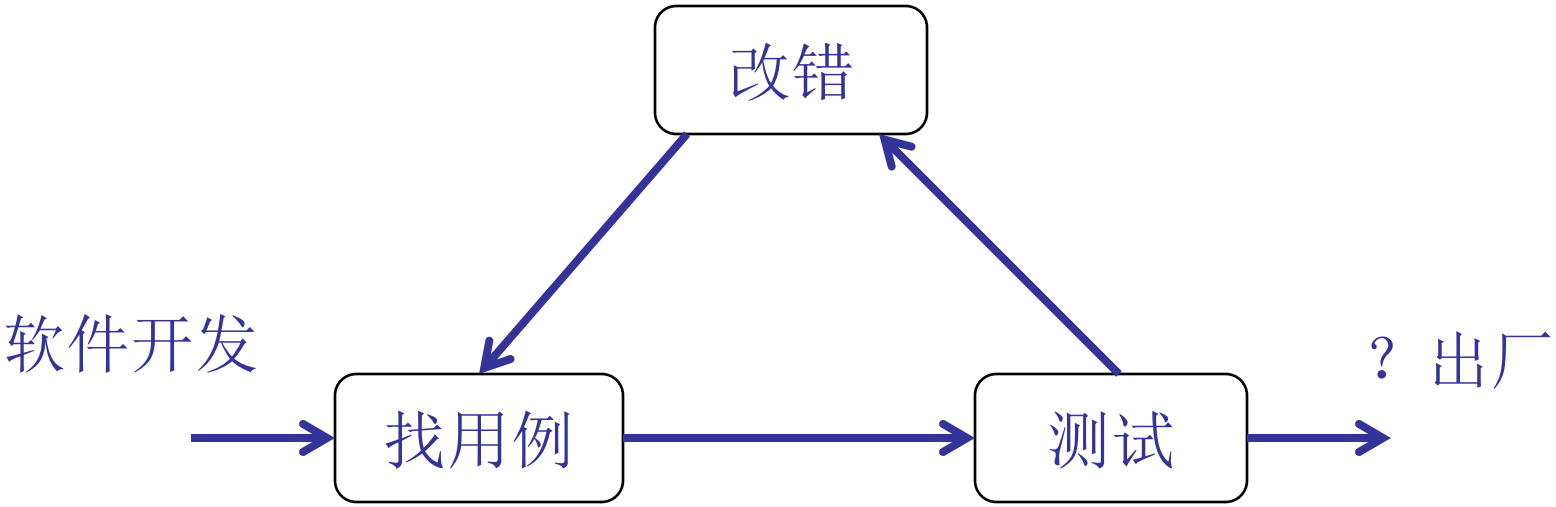
- 一种简单实用的软件测试过程模型POCERM
- 测试过程中必需的基本测试活动及其产生的结果
  - 拟定软件测试计划(Plans)
  - 编制软件测试大纲(Outlines)
  - 设计和生成测试用例(test Case generation)
  - 实施测试(Execution)
  - 生成软件测试报告(software testing Reports)
  - 软件问题报告SPR(Software Problem Report)
  - 测试结果报告(test result Reports)
  - 对整个测试过程进行有效的管理(Management)

软件开发与软件测试间的协同



北航软件所

常见的测试过程



基本特性

- 计划性：任务 人员 设备 时间 相关...
- 平行性：开发 编码 || 测试 再测试
- 完整性：计划+大纲+用例+SPRs+...
- 重用性：测试 再测试 回归测试 升级 多平台..
- 可重复性：SPRs 用例 大纲 -> 再现BUGs
- 周期性：test cycles, regression, update
- 可管理性：

well structured and organized QE group + well planned and prepared task

软件质量工程

- 软件开发是一项集体的 工程性的工作
  - 人 设备 场地 资金 进度等
- 沟通(Communication)与协同工作
  - 人与人 部门与部门之间
  - 及时交流情况
  - 确认理解一致
  - 充分研讨 "问题"
  - 问题记录与追踪
- 软件测试是软件开发的一部分

软件测试过程控制与管理

--软件测试的一些基本原则(补充)

- 测试必须是有计划的: 任务、时间、人员、设备、经费、方法与工具、问题等
- 测试必须是有组织的
- 测试必须是有准备的
- 测试必须是有记录的

测试是一项非常复杂的、创造性的和需要高度智慧的挑战性任务。

软件测试阶段性

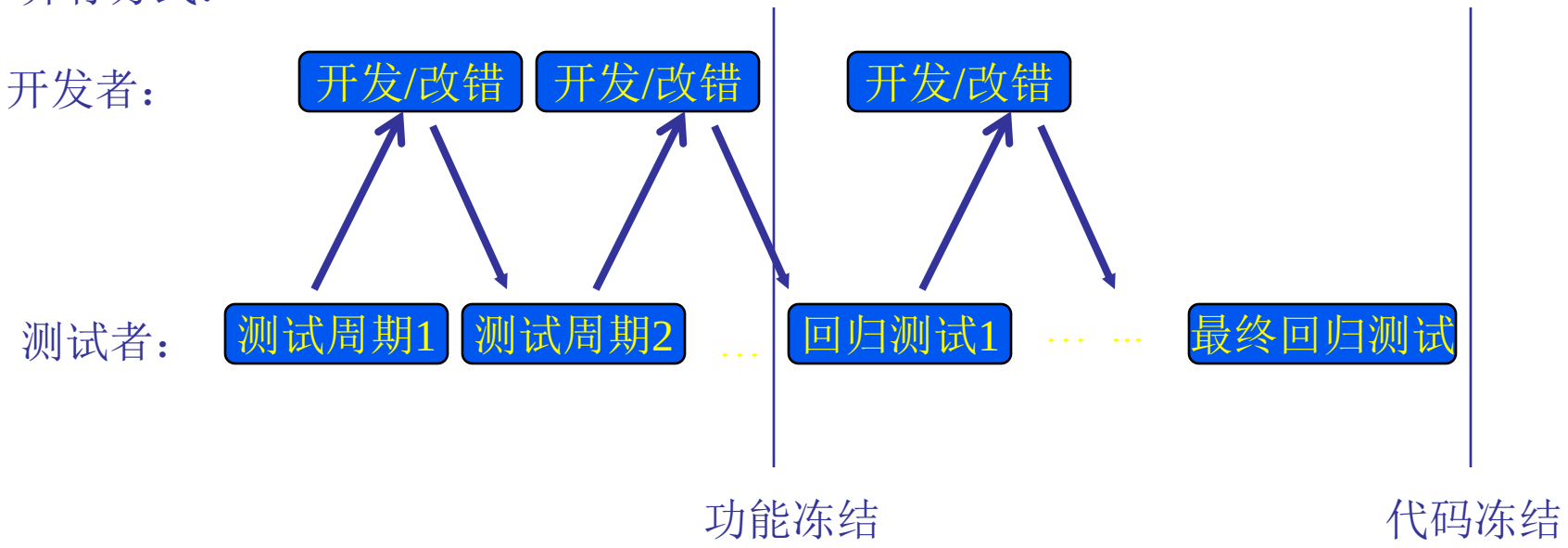
- 单元测试:
  - 静态分析: 结构分析 复杂度分析 代码走查
  - 白盒子测试:语句覆盖 分支覆盖 ...
  - 黑盒子测试: 合法 非法 边界 极端 ...
- 组合测试: 功能 性能 界面/接口（黑盒子）
- 集成测试: 功能 性能 界面 兼容 （黑盒子）
- 系统测试: 嵌入式系统 ...（黑盒子）
- BETA测试: 用户测试及试用（黑盒子）
- 验收测试: 对软件质量保证无直接贡献
- 测试的验收: 过程 方法 内容 结果 ...

软件测试的周期性

串行方式：

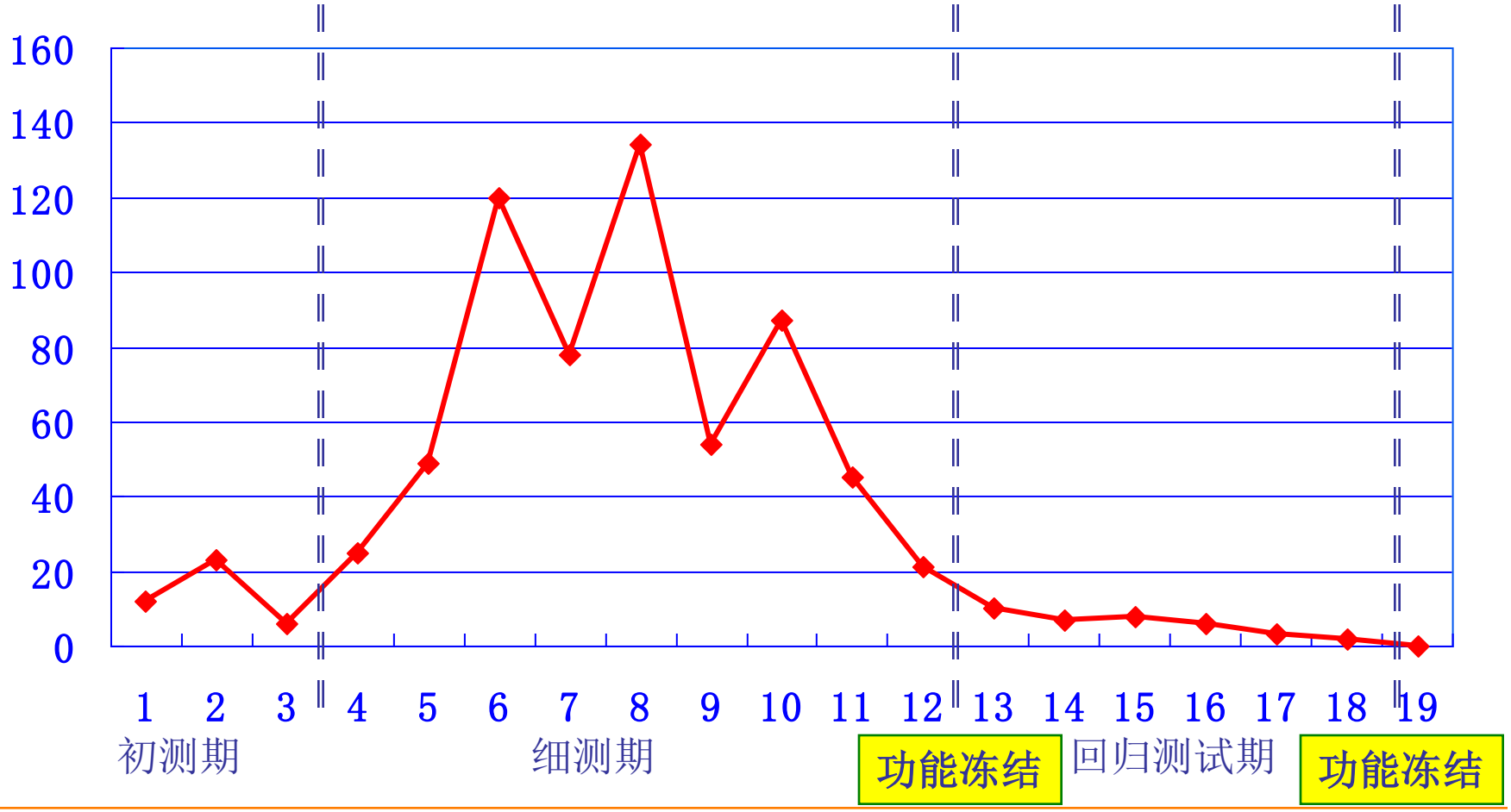
开发者：

并行方式：



北航软件所

三个测试期和两个里程碑



北航软件所

软件质量与测试

$$\text{软件质量} \sim \frac{\text{测试覆盖率}}{\text{软件问题数}}$$

软件集成测试的三个阶段

- 初测期：
  - 主要功能和关键的执行路径，排除主要障碍
- 细测期:
  - 依据测试计划和测试大纲
  - 逐一测试大大小小的功能、方方面面的特性、性能、用户界面、兼容性、可用性等等
  - 预期可发现大量不同性质、不同严重程度的错误和问题
- 回归测试期:
  - 系统已稳定，一轮测试中发现的错误已十分有限
  - 复查已知错误的纠正情况，确认未引发任何新的错误
- 终结回归测试(Final Regression)

软件集成测试的两个重要的里程碑

- 功能冻结(Function/Feature Freeze)：标志着整个系统的集成工作已经完成，系统的功能、性能、界面、兼容性等均已经过测试，符合设计要求，确认系统功能和其他特性均不再做任何改变；
- 代码冻结(Code Freeze)：
  - 理论上，无错误时冻结程序代码
  - 实际上，代码冻结只是标志着系统的当前版本的质量已经达到预期的要求，冻结程序的源代码，不再对其做任何修改。显然，这个里程碑是设置在软件通过了终结回归测试之后。

软件测试计划

- 概要测试计划
- 详细测试计划
- 测试实施计划

概要测试计划

- 在软件开发初期，即需求分析阶段制定
- 定义被测试对象和测试目标
- 确定测试阶段和测试周期的划分
- 制定测试人员、软硬件资源和测试进度等方面的计划, 任务与分配与责任划分
- 规定软件测试方法、测试标准，比如，
  - 语句覆盖率需达到95%
  - 三级以上的错误改正率需95%
  - 所有决定不改正的"轻微"错误都必须经过专门的质量评审委员会同意
- 支持环境和测试工具等
- *待解决的问题等*

北航软件所

详细测试计划

- 针对子系统在特定的测试阶段所要进行的测试工作制定详细计划
- 详细规定了测试小组的各项测试任务、测试策略、任务分配和进度安排等

测试人员的测试实施计划

- 测试者或测试小组的具体的测试实施计划
- 规定了测试者负责测试的内容、测试强度和工作进度
- 整个软件测试计划的组成部分，是检查测试实际执行情况的重要依据
- 主要内容:
  - 计划进度和实际进度对照表
  - 测试要点 测试策略 ...
  - 尚未解决的问题和障碍

软件测试大纲

- 软件测试的依据
- 测试项目
- 测试步骤
- 测试完成的标准
- 无论是自动测试还是手动测试，都必须满足测试大纲的要求。

软件测试大纲的本质

- 从测试的角度对被测对象的功能和各种特性的细化和展开。比如，
  - 针对系统功能的测试大纲是基于软件质量保证人员对系统需求规格说明书中有关系统功能定义的理解，将其逐一细化展开后编制而成的。
  - 测试大纲不仅是软件开发后期测试的依据，而且在系统的需求分析阶段也是质量保证的重要文档和依据。

软件测试用例

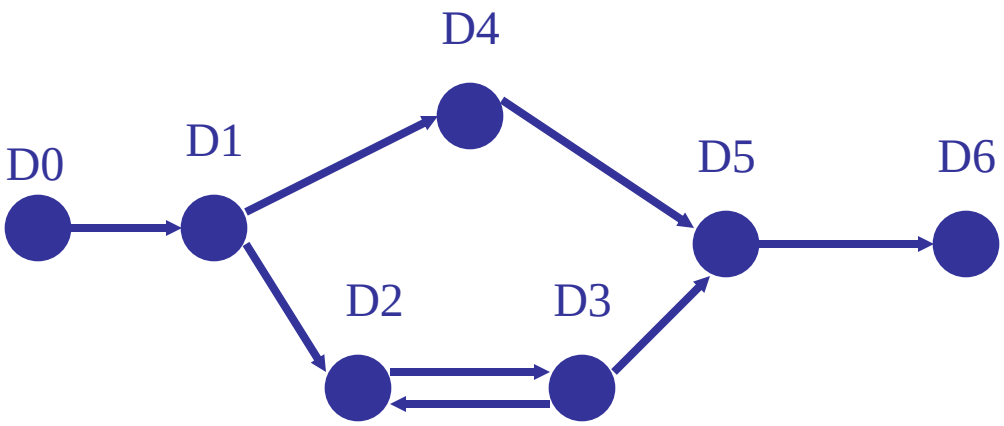
- 实施一次测试而向被测系统提供的输入数据、操作或各种环境设置
- 对交互式系统，软件交互执行过程的控制也是一种测试用例
- 测试用例的设计与生成是依据测试大纲对其中每个测试项目的进一步实例化。比如，
  - 对于一个输入项的测试，应当设计一组测试数据，包括合法的、边界的和非法的数据等。

测试用例生成的基本原则

- 测试用例的代表性：能够代表并覆盖各种合理的和不合理、合法的和非法的、边界的和越界的、以及极限的输入数据、操作和环境设置等;
- 测试结果的可判定性：即测试执行结果的正确性是可判定的;
- 测试结果的可再现性：即对同样的测试用例，系统的执行结果应当是相同的。

交互式软件的测试用例的生成

测试用例空间 T=D0xD1x(*(D2xD3)+D4)xD5xD6



t:d0 d1 d2 d3 d2 d3 d5 d6

控制点或输入点

控制转移

交互式软件的测试用例的生成

- 测试用例
  - 测试步骤: 控制点的序列 (测试大纲)
  - 代表值: 每个控制点上的每一项输入或操作(测试数据)
- 覆盖准则
  - 准则1: 控制点覆盖
  - 准则2: 转移边覆盖
  - 准则3: 典型路径覆盖
  - 准则4: 代表值覆盖

测试的实施

- 手工测试与自动测试
- 测试优先级
- 软件问题报告SPR
  - SPR的生存周期:
    - 状态:new,open,pending,close,...
    - 子状态：fix,defer,NPTF,Not a Bog, Transfer,...
- 测试状态报告
  - 进度 问题 调整 ...
- QE与Developers之间的通讯

北航软件工程研究所

- 软件测试过程管理平台QESuite
  - 异地群组协同工作平台
  - 软件测试计划、测试大纲和测试用例的管理
  - 软件问题报告的记录、追踪与控制
  - 软件测试过程的控制与管理

欢迎进入QESuite Web Version1.0系统！

系统管理

功能分类

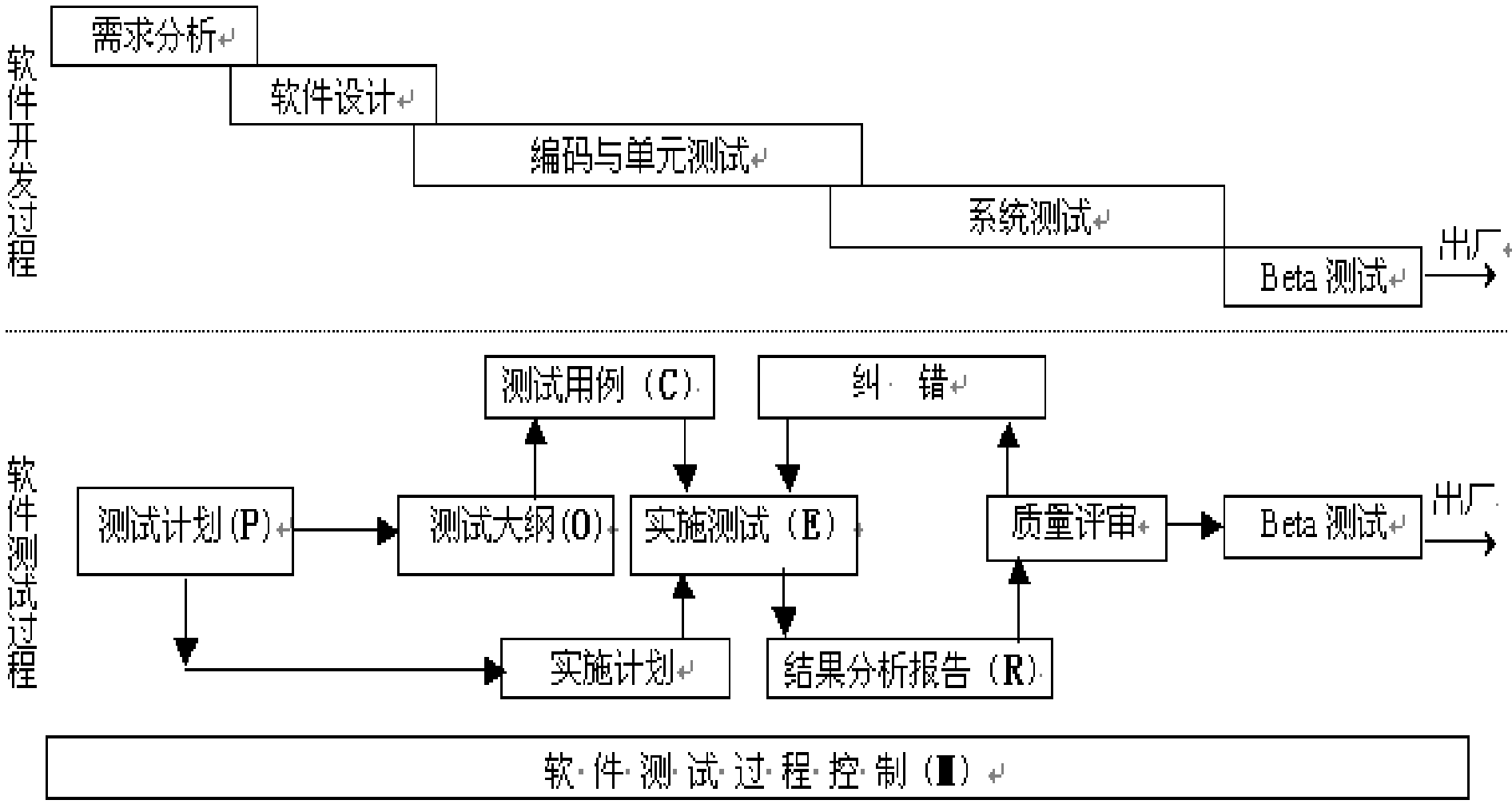
测试用例

问题报告

辅助文档

帮助

软件测试过程（POCERM-II）

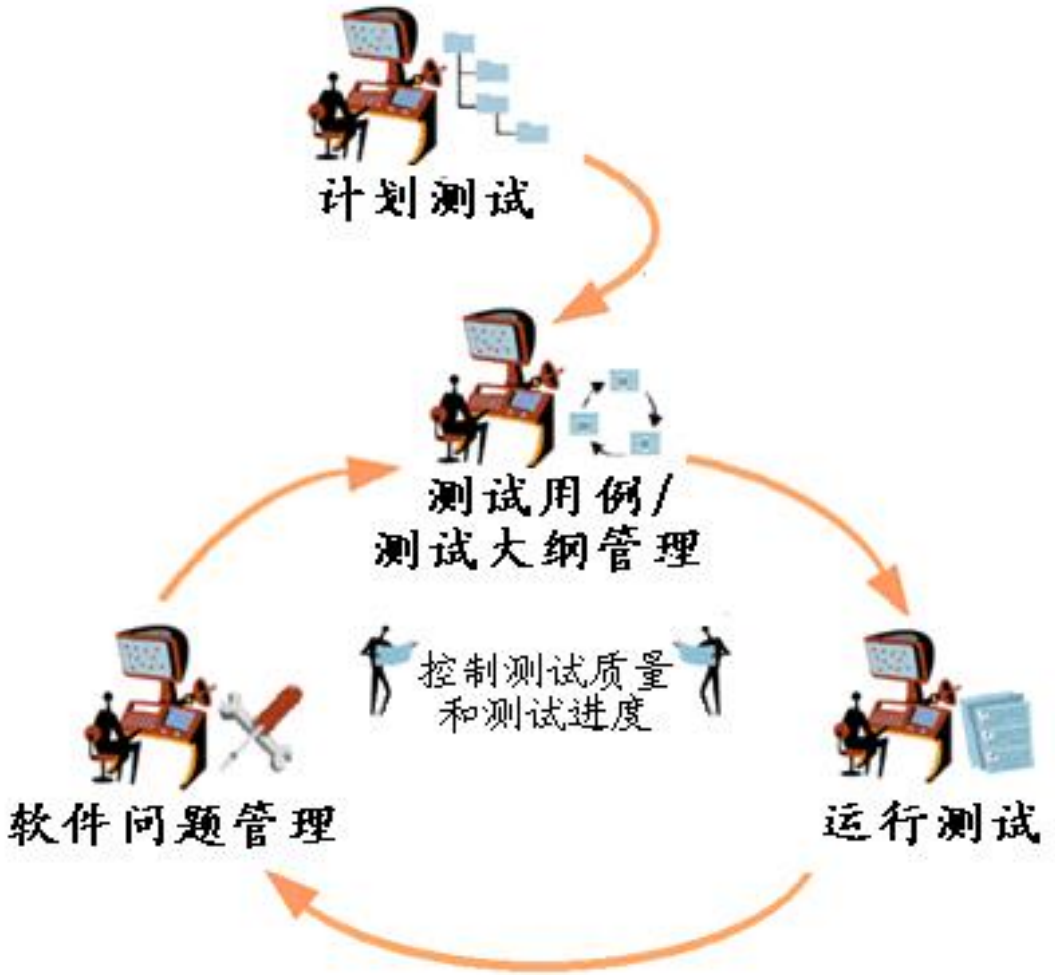


北航软件所

软件团队异地协同工作模式

异地协同

团队协作

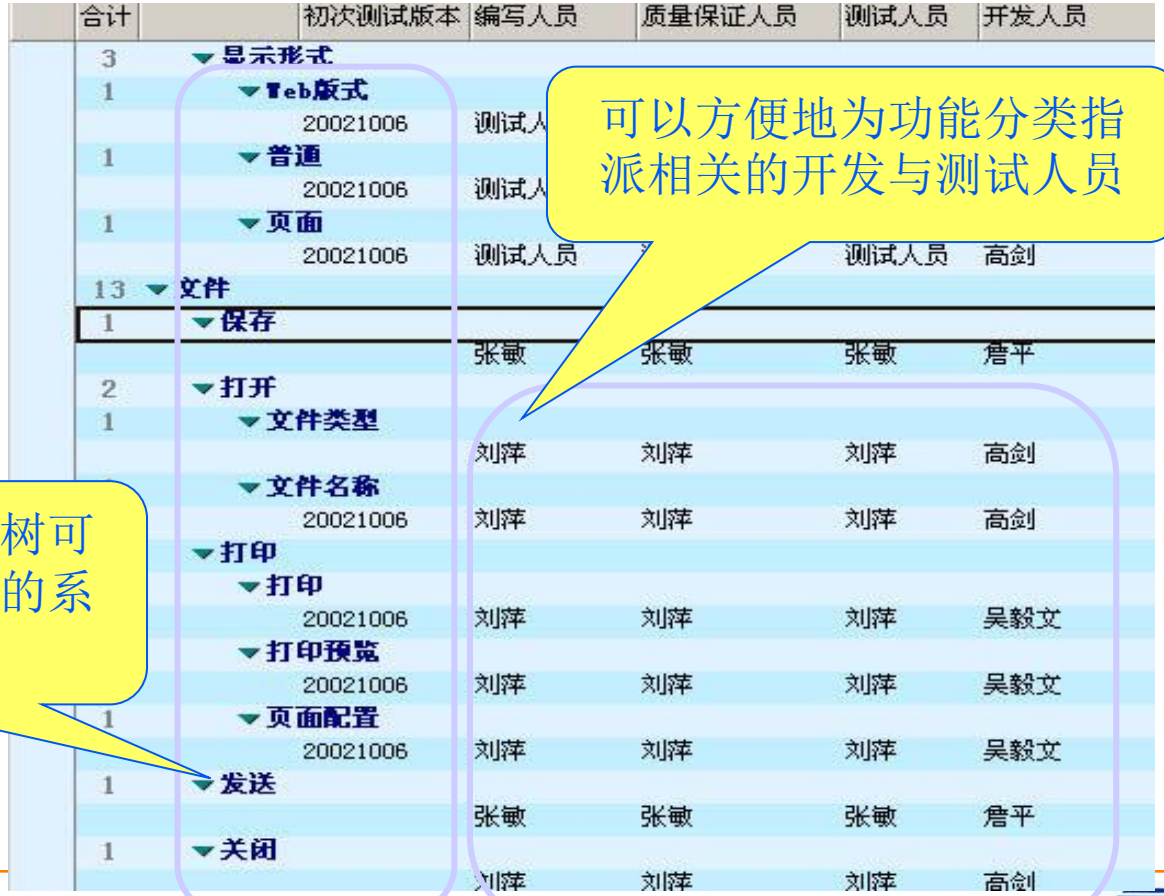


QESuite的功能架构

北航软件所

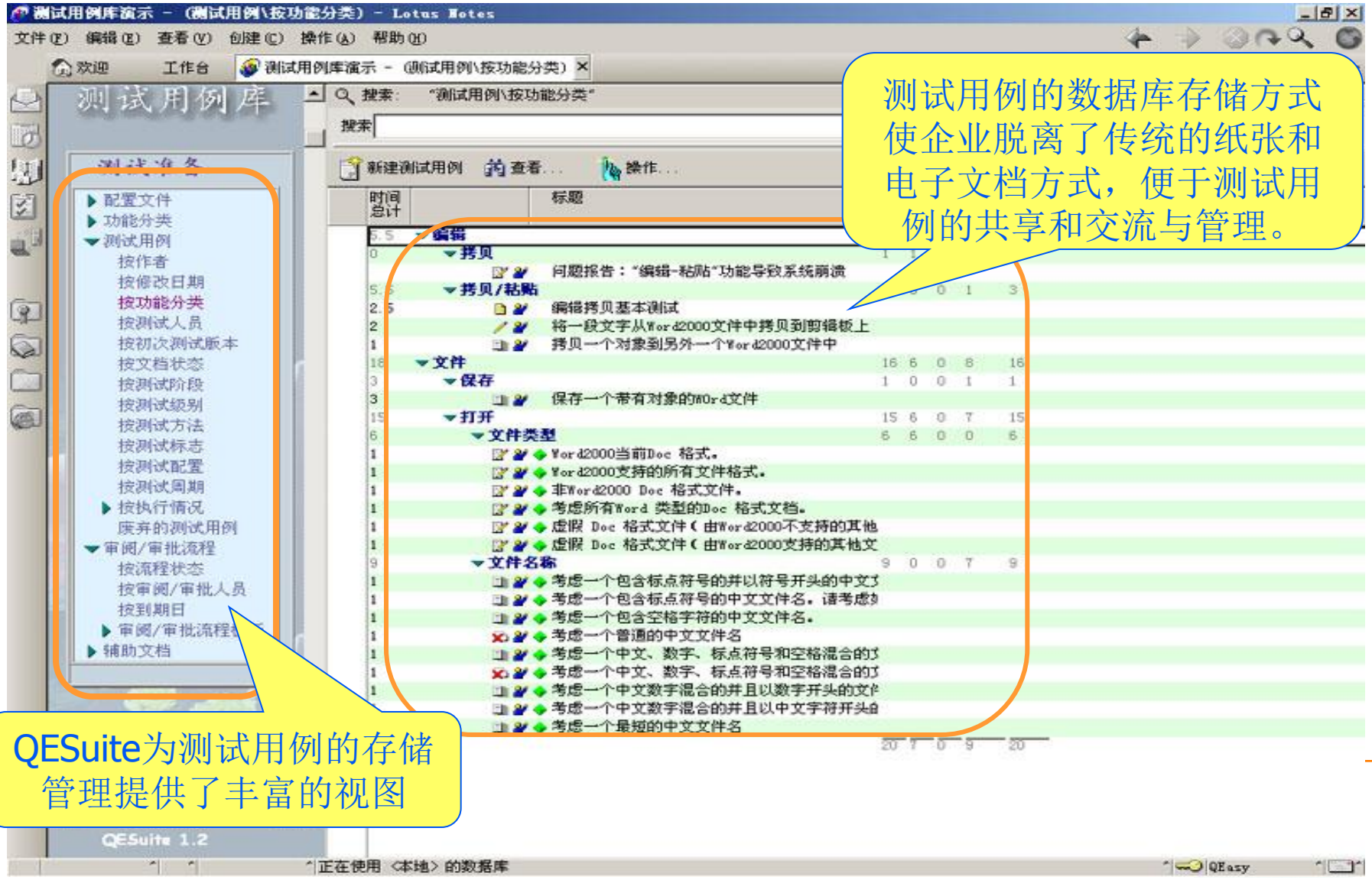
基于功能域分解的测试用例库管理

- 树状结构的功能分类管理可以满足大型软件的测试要求。

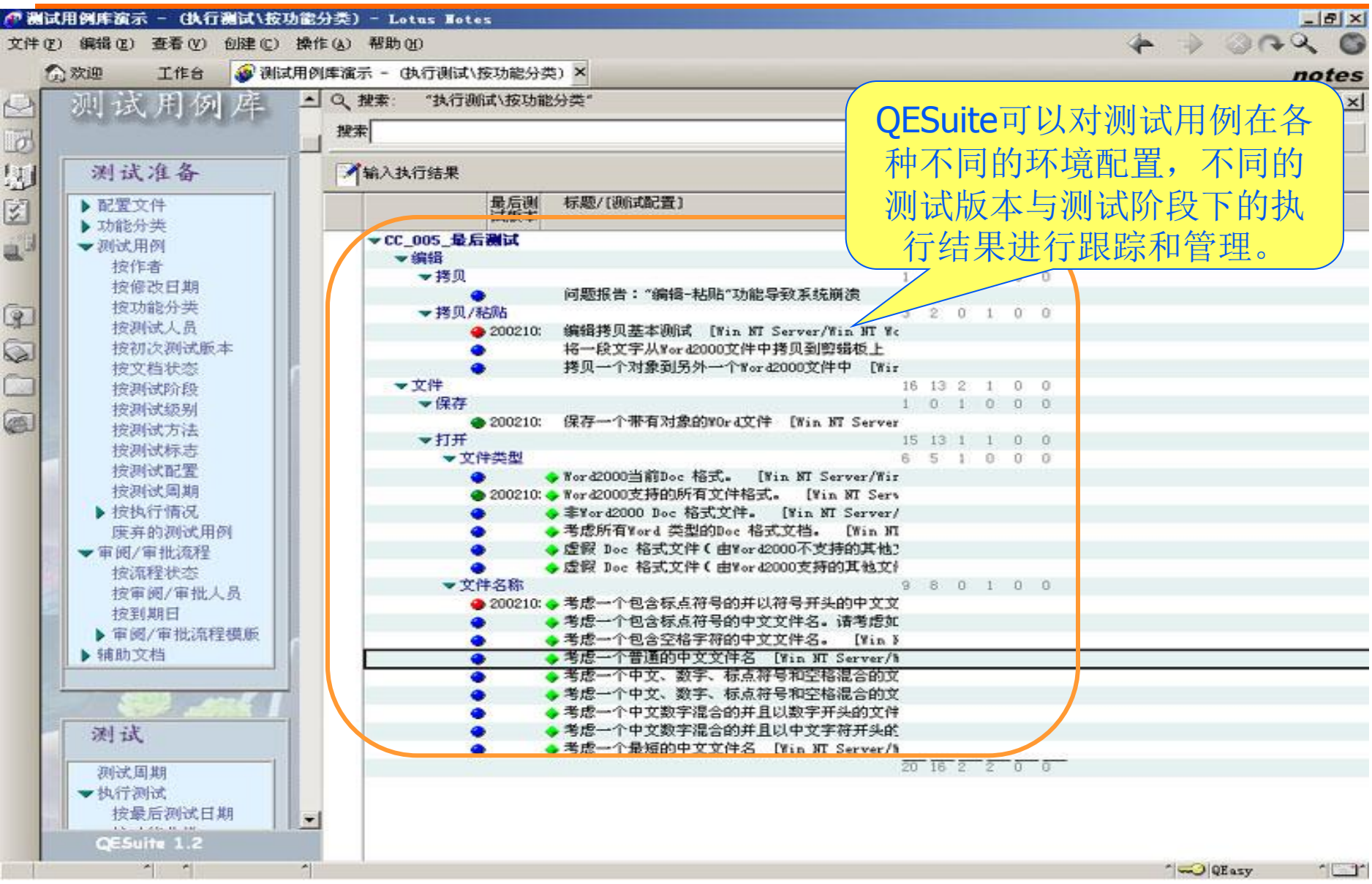


北航软件所

测试用例存储管理


测试用例的数据库存储方式使企业脱离了传统的纸张和电子文档方式，便于测试用例的共享和交流与管理。
QESuite为测试用例的存储管理提供了丰富的视图

测试用例执行状态监控


测试用例库演示 - (执行测试\按功能分类) - Lotus Notes
文件(F) 编辑(E) 查看(V) 创建(C) 操作(A) 帮助(H)
欢迎 工作台 测试用例库演示 - (执行测试\按功能分类)
测试用例库
测试准备
配置文件
功能分类
测试用例
按作者
按修改日期
按功能分类
按测试人员
按初次测试版本
按文档状态
按测试阶段
按测试级别
按测试方法
按测试标志
按测试配置
按测试周期
按执行情况
废弃的测试用例
审阅/审批流程
按流程状态
按审阅/审批人员
按到期日
审阅/审批流程模版
辅助文档
测试
测试周期
执行测试
按最后测试日期
QESuite 1.2
搜索: "执行测试\按功能分类"
搜索
输入执行结果
最后测试版本
标题/[测试配置]
CC_005_最后测试
编辑
拷贝
问题报告："编辑-粘贴"功能导致系统崩溃
拷贝/粘贴
200210: 编辑拷贝基本测试 [Win NT Server/Win NT
将一段文字从Word2000文件中拷贝到剪辑板上
拷贝一个对象到另外一个Word2000文件中 [Wir
文件
保存
200210: 保存一个带有对象的Word文件 [Win NT Server
打开
文件类型
Word2000当前Doc 格式。 [Win NT Server/Wir
200210: Word2000支持的所有文件格式。 [Win NT Ser
非Word2000 Doc 格式文件。 [Win NT Server/
考虑所有Word 类型的Doc 格式文档。 [Win NT
虚假 Doc 格式文件（由Word2000不支持的其他
虚假 Doc 格式文件（由Word2000支持的其他文
文件名称
200210: 考虑一个包含标点符号的并以符号开头的中文文
考虑一个包含标点符号的中文文件名。请考虑如
考虑一个包含空格字符的中文文件名。 [Win
考虑一个普通的中文文件名 [Win NT Server/
考虑一个中文、数字、标点符号和空格混合的文
考虑一个中文、数字、标点符号和空格混合的文
考虑一个中文数字混合的并且以数字开头的文件
考虑一个中文数字混合的并且以中文字符开头的
考虑一个最短的中文文件名 [Win NT Server/
QESuite可以对测试用例在各种不同的环境配置，不同的测试版本与测试阶段下的执行结果进行跟踪和管理。

软件问题报告生命周期管理

各种不同类型的人员可以从不同的视图中得到软件问题报告的不同分类信息，方便了软件问题的处理！

QESuite全力支持开发和测试部门的协同工作，可实现开发测试部门工作分配的流程化，方便快捷！

软件问题报告

VSHW62QALT ：可以使用中文字符创建虚拟域，但无法在此虚拟域中创建用户和部门 - Lotus Notes

文件(F) 编辑(E) 查看(V) 创建(C) 操作(A) 帮助(H)

工作台 人行UMC测试问题追踪库 - (按作者) VSHW62QALT ：可以使用中文字符创建虚拟域，但无法在此虚拟域中创建用户和部门 notes

编辑 退出 分类与分配 发信通知 关闭

中国人民银行UMC产品测试

软件问题报告

编号：VSHW62QALT　　状态：新建
作者：Vivian Shaw/QEasy　　创建日期：2004-07-09

简要描述*： 可以使用中文字符创建虚拟域，但无法在此虚拟域中创建用户和部门

基本信息 | 问题修复信息 | 测试环境 | 操作记录 | 历史信息

软件版本*：	4.3.9	测试版本*：	4.3.9
严重性*：	3. 数据丢失或功能失效	重现性：	
来源：		优先级：	
平台：	服务器操作系统：Sun Solaris 5.8;Linux Redhat 7.3 客户机操作系统：Win 2000 Client	浏览器：	
地点：		标志：	
状态改变后给作者发信？	○ 发信 ◉ 不发信	状态改变后给下列人员发信：	

保存缺省设置

再现步骤：

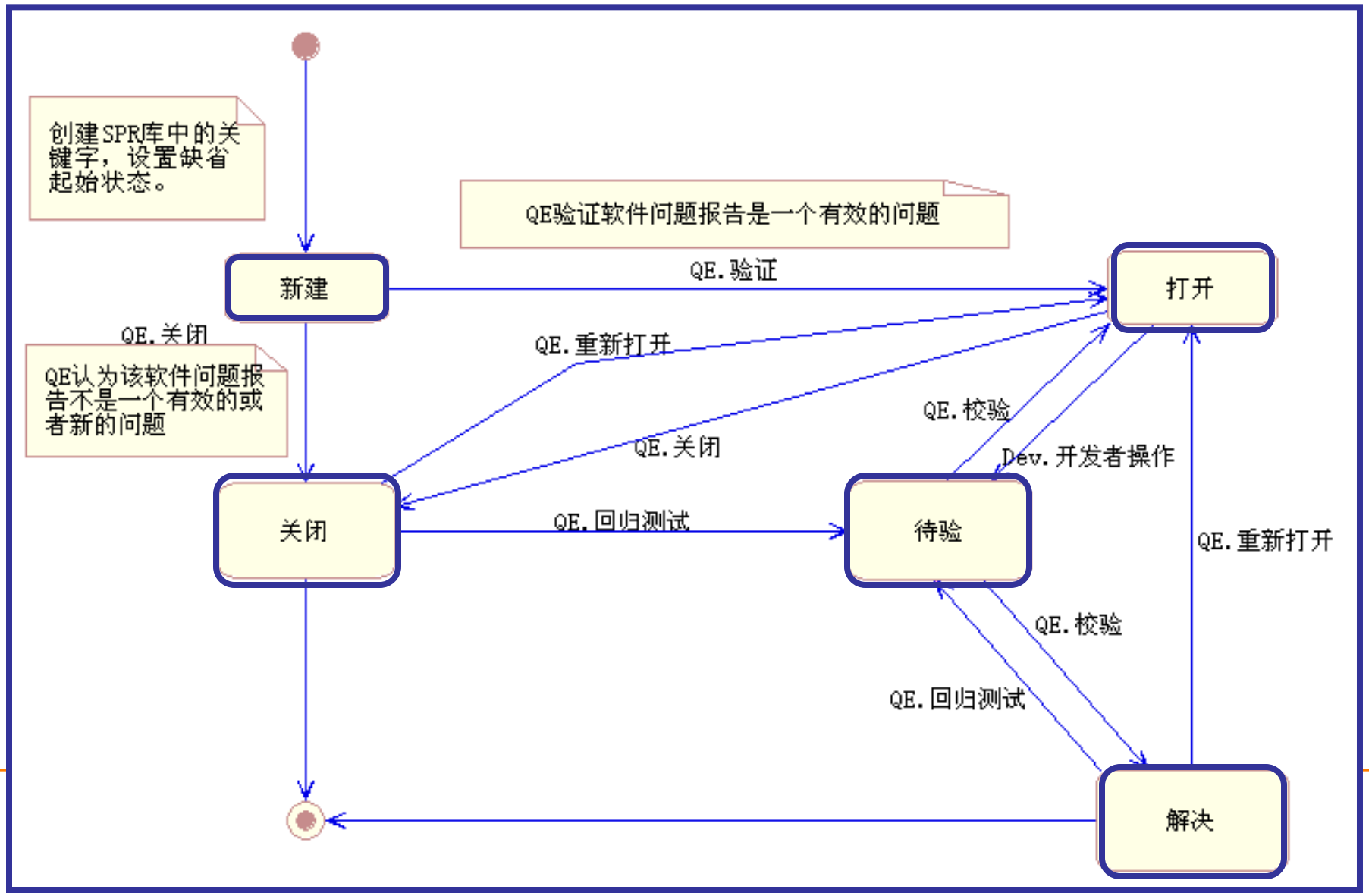
1. 创建虚拟域，在虚拟域名称和虚拟域前缀中使用中文字符，比如：虚拟域名称：北京，虚拟域前缀：中国。
2. 虚拟域创建成功。
3. 在此虚拟域下试图创建用户。

问题：报错（如下图）

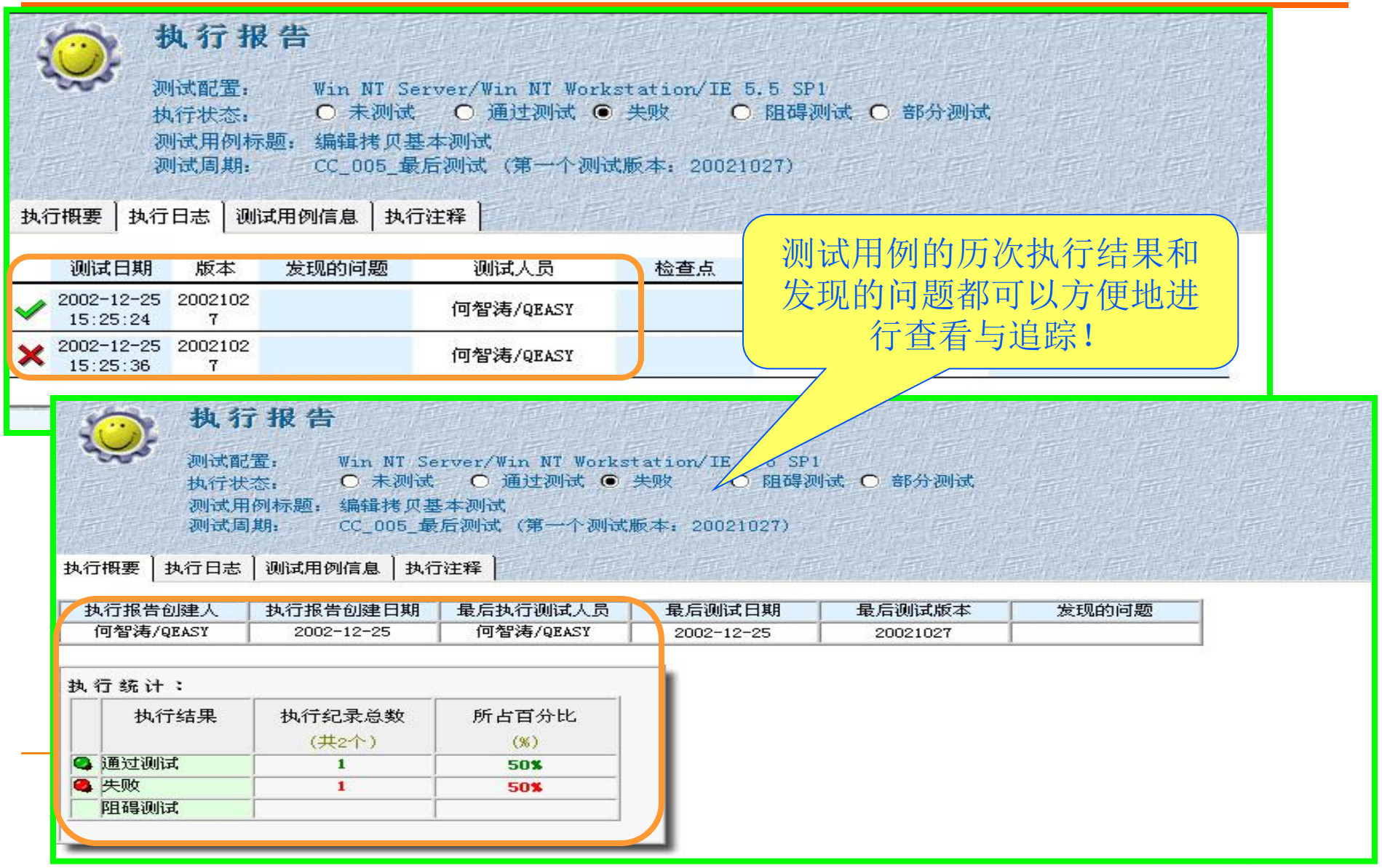
附件：（请使用菜单“文件 -> 附加”添加一个或多个附件）

开始 | QESuite1.3概要介绍 | 精品成人论坛 - p... | VSHW62QALT ：可以... | 办公室 | 99% | 18:40

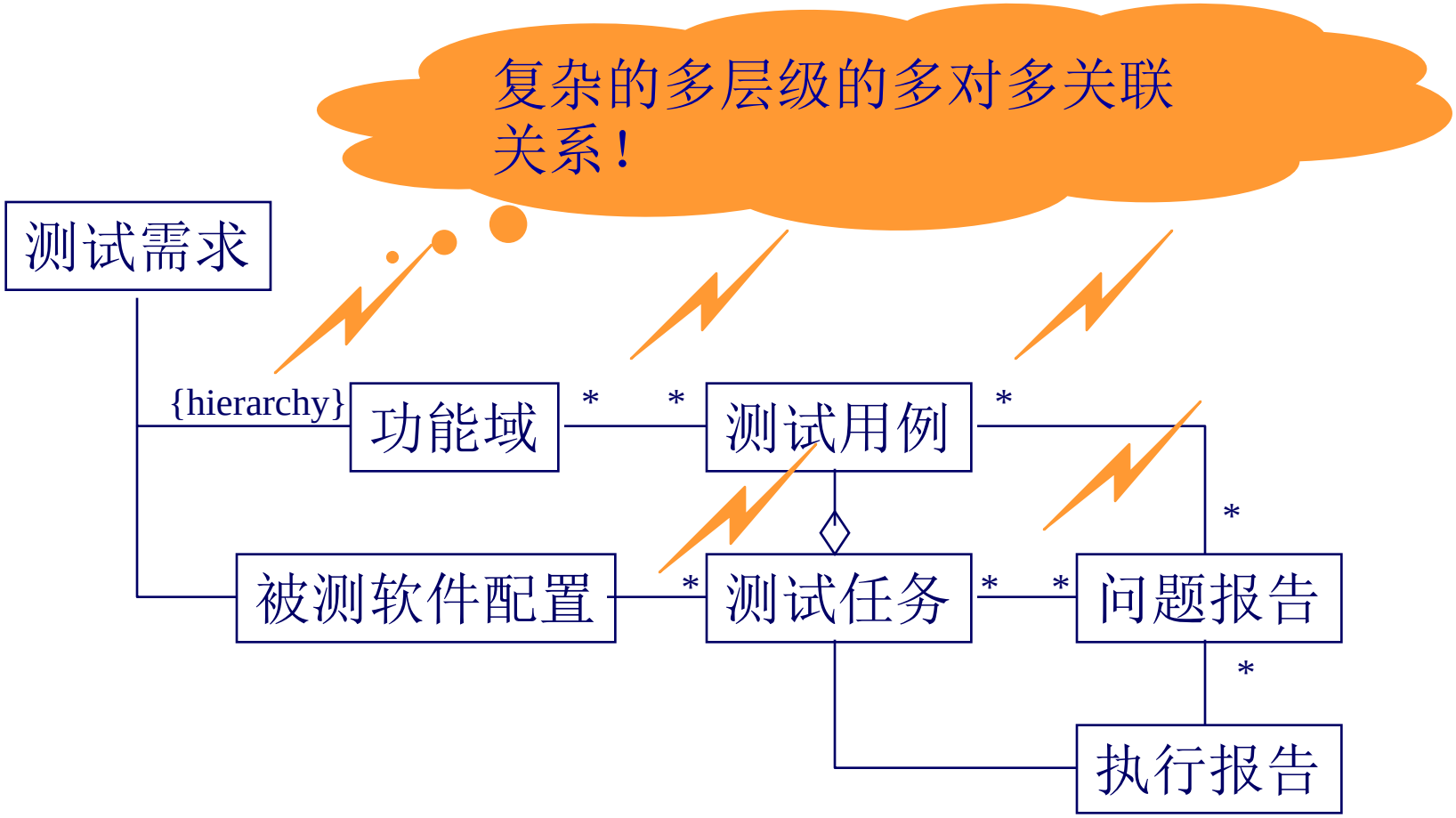
软件问题报告的生命周期



测试执行报告及测试结果的追踪



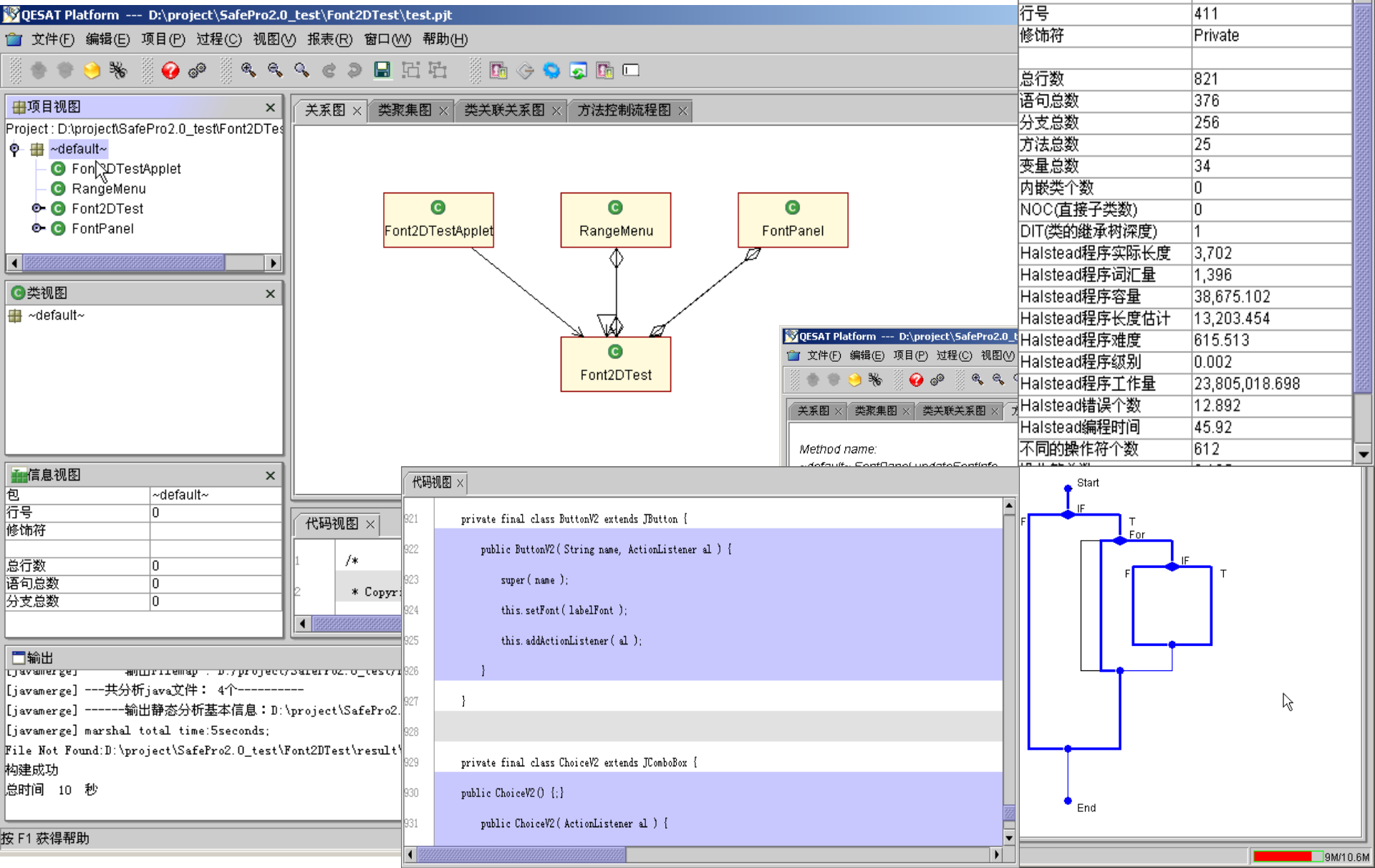
测试用例及其关联性


复杂的多层级的多对多关联关系！
测试需求
{hierarchy}
功能域
*
*
测试用例
*
*
被测软件配置
*
测试任务
*
*
问题报告
*
*
执行报告

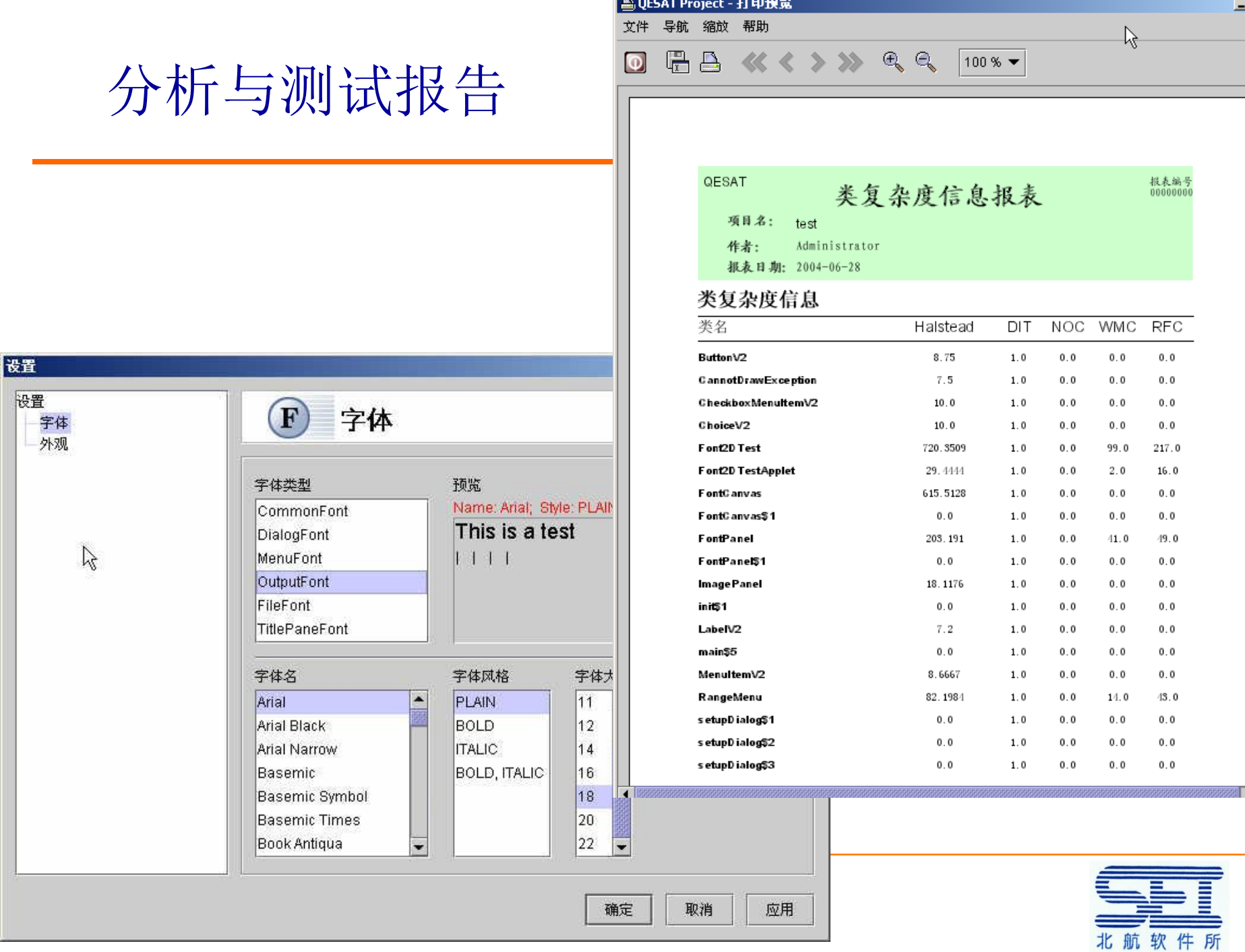
北航软件工程研究所

- C++/Java软件分析与测试工具系列
  - 类图、程序结构图、程序流程图
  - 语句覆盖测试和分支覆盖测试
  - ANSI C++、MS VC++/Windows、GNU C++/Linux
  - Java

C++/Java软件分析与测试工具

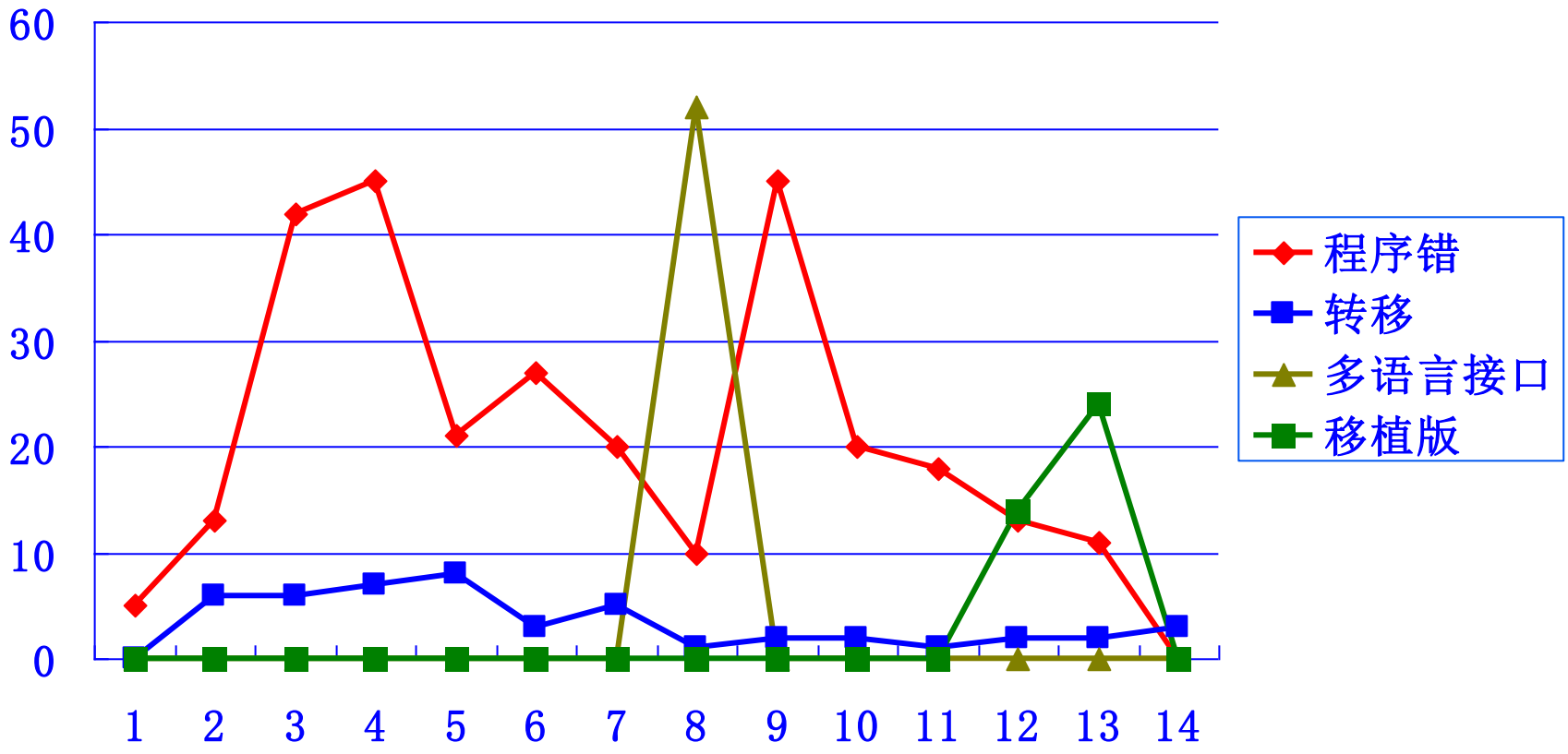


分析与测试报告



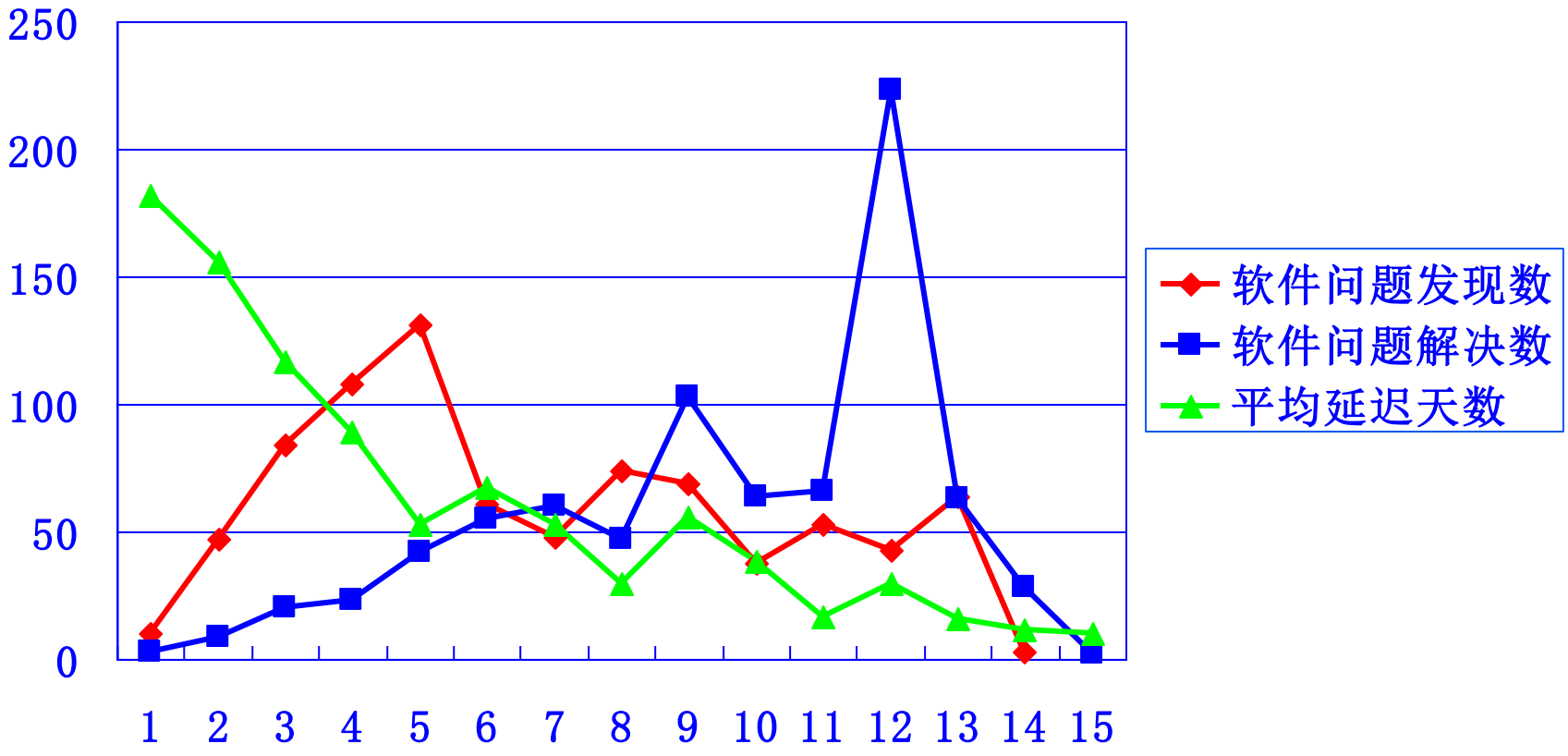
北航软件所

软件问题时间分布图



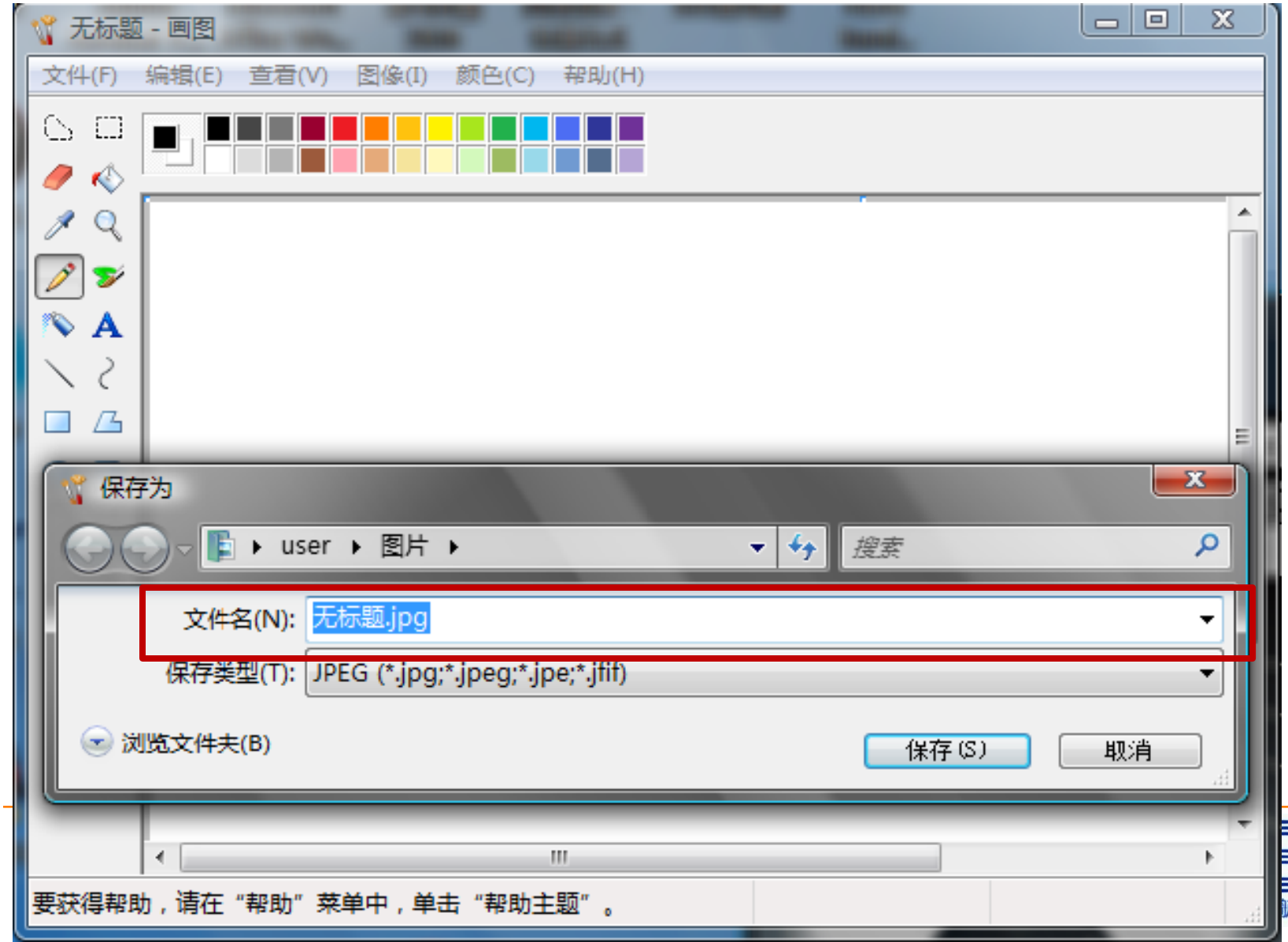
北航软件所

软件问题发现与解决延迟图(2009/10/29)



北航软件所

练习：测试—文件名


无标题 - 画图
文件(F) 编辑(E) 查看(V) 图像(I) 颜色(C) 帮助(H)
保存为
user ▸ 图片 ▸
搜索
文件名(N): 无标题.jpg
保存类型(T): JPEG (*.jpg;*.jpeg;*.jpe;*.jfif)
浏览文件夹(B)
保存(S)
取消
要获得帮助，请在“帮助”菜单中，单击“帮助主题”。

总结

- 软件测试实际上是一个相当复杂的软件质量保证工程
- 软件测试过程POCERM突出描述了软件测试的过程性，以及计划、大纲和软件问题报告等一系列文档的重要性。
- 测试的基本准则:覆盖率
- 测试自动化 -- 工具支持