Ch 09 软件测试

软件测试：使用人工或自动手段，运行或测定某个系统的过程，其目的是检验它是否满足规定的需求，或是清楚了解预期结果与实际结果之间的差异。

软件测试分为**静态分析**和**动态测试**

软件调试：发现所编写软件中的错误，确定错误的位置并加以排除，使之能由计算机或相关软件正确理解与执行的方法与过程。

在进行调试工作以前，首先要发现存在着某种错误的迹象。随后的调试过程通常分为两步：

1. 确定问题的性质并且找到该错误在软件中所处的位置
2. 修正这一错误

软件测试技术

黑盒测试

黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能，通过测试来**检测每个功能是否都能正常使用**

黑盒测试方法主要有**等价类划分**、**边界值分析**、**因果图**、**错误推测**等，主要用于**软件确认测试**

白盒测试

白盒测试也称结构测试或逻辑驱动测试，它是知道产品内部工作过程，可通过测试来**检测产品内部动作是否按照规格说明书的规定正常进行**，按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能

白盒测试的主要方法有逻辑覆盖、基本路径测试等，主要用于**软件验证**

白盒测试技术

依据程序的逻辑结构—白盒测试技术

控制流程图：一种表示程序控制结构的图形工具，其基本元素是节点、判定、过程块

1. **过程块**：既不能由判定、也不能由节点分开的一组程序语句
2. **判定**：是一个程序点，此处控制流可以分叉
3. **节点**：是一个程序点，此处控制流可以结合

各种测试方法

1. **路径测试**：执行所有可能的穿过程序的控制流程路径

   路径测试严格地限制为所有可能的入口/出口路径。如果遵守这一规定，则达到了 100% 路径覆盖率。在路径测试中，该策略是最强的，但一般是不可实现的

2. **语句测试**：至少执行程序中所有语句一次

   如果遵守这一规定，则达到了 100% 语句覆盖率

语句覆盖是最弱的逻辑覆盖准则

3. **分支测试**：至少执行程序中每一分支一次

   如果遵守这一规定，则达到了 100% 分支覆盖率

   分支覆盖是一种比语句覆盖稍强的逻辑覆盖。但不能保证一定能查出在判断的条件中存在的错误

4. **条件组合测试**：设计足够的测试用例，使每个判定中的所有可能条件取值组合至少执行一次

   如果遵守这一规定，我们就说实现了条件组合覆盖。只要满足了条件组合覆盖，就一定满足分支覆盖。

黑盒测试

黑盒测试（功能测试）——依据软件行为的描述的测试

事务流测试技术

1. **事务**：以用户的角度所见的一个工作单元

   一个事务由一系列操作组成，其中某些操作可含有系统执行成分，或含有设备执行成分，它们共同协作，完成用户的一项工作。

2. **事务处理流程**：

与程序控制流程图比较：

1. 事务流图是一种数据流图，从操作应用的历史，观察数据对象
2. 事务流图中的分支：“抽象”了一个复杂的过程
3. 事务流图存在“中断”，把一个过程等价地变换为具有繁多出口的链支

等价类划分技术

1. **等价类**：输入域的一个子集，在该子集中，各个输入数据对于揭示程序中的错误都是等效的。以等价类中的某代表值进行的测试，等价于对该类中其他取值的测试
2. **有效等价类**：指那些对于软件的规格说明书而言，是合理的、有意义的输入数据所构成的集合

   用于实现功能和性能的测试

3. **无效等价类**：指那些对于软件的规格说明书而言，是不合理的、无意义的输入数据所构成的集合

   用于测试那些所实现的功能和性能不符合规格说明书的要求

边界值分析

边界值分析是一种最常用的黑盒测试技术

因果图

因果图：是设计测试用例的一种工具，着重检查**各种输入条件的组合**。通过画因果图，把用自然语言描述的功能说明转换为判定表，最后利用判定表来检查程序输入条件的各种组合情况。因果图测试结束为判定表的每一列设计一个测试用例。

软件测试步骤

单元测试

主要检查软件测试设计的最小单元—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。**对应程序编码，主要采用**白盒测试**技术。

集成测试

对应软件的设计，主要检测各个单元集成过程中相互之间的接口错误以及形成新的组合后功能中的错误，多采用**黑盒测试技术**并辅以一些白盒测试技术

有效性测试（确认测试）

有效性测试的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。确认测试只使用**黑盒测试技术。**

系统测试

集中检验系统所有元素之间协作是否合适，整个系统的性能、功能是否达到。

1. **恢复测试**：采取人工干预方式强制使软件出错，检验系统的恢复能力
2. **安全性测试**：试图验证建立在系统内的预防机制，防止来自非正常的侵入
3. **强度测试**：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。强度测试需要在反常规数据量、频率和资源的方式下运行系统，以检查系统能力的最高实际限度
4. **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
5. **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，发现人为因素或使用上的问题。
6. **部署测试（配置测试）**：软件必须在多平台及操作系统上运行。在软件将要在其中运行的每一种环境中测试软件