软件测试期末

1. 软件缺陷

1.1 定义：

- 软件缺陷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，没有满足用户的需求。

1.2 种类：

1. 未达到产品说明书中已经标明的功能；
2. 出现了产品说明书中指明不会出现的错误；
3. 未达到产品说明书中虽未指出但应当达到的目标；
4. 功能超出了产品说明书中指出的范围；
5. 测试人员认为软件难以理解、不易使用，或者最终用户认为该软件使 用效果不良；

1.3 缺陷等级：

等级	描述
致命的	造成系统或者应用程序崩溃、死机、系统悬挂或者造成数据丢失、主要功能完全丧失
严重的	功能或者特性没有实现、主要功能丧失，导致严重的问题或知名的错误声明
一般的	虽不影响使用，但没有很好的实现功能，没有达到预期的效果。信息提示不太准确、用户界面差
微小的	对功能几乎没有影响，产品及属性仍然可用，如有个别错误字、操作时间长

1.4 软件缺陷的产生：

1. 需求解释有错误；
2. 用户需求定义错误；
3. 需求记录错误；
4. 设计说明有误；
5. 编码说明有误；
6. 程序代码有误；
7. 数据输入有误，问题修改不正确。

2. 什么是软件测试

- 软件测试就是在软件投入运行前，对软件**需求分析、设计规格说明**和**编码实现**的最终审查，它是软件质量保证的关键步骤。软件测试是为了尽快尽早地发现在软件产品中所存在的各种软件缺陷而展开的贯穿整个软件开发生命周期、对软件产品进行验证和确认的活动过程。

3. 软件测试的基本问题

3.1 软件质量反映的三个方面：

1. 软件需求是度量质量的基础。
2. 在各种标准中定义开发准则，用来知道软件人员用于工程化的方法来开发软件。
3. 往往会有一些隐含的需求没有明确地提出。

3.2 软件测试的对象：

- 不仅仅是对程序的测试，而是贯穿于软件定义和开发的整个过程。因此，软件开发过程中产生的需求分析、概要设计、详细设计以及编码等各个阶段所得到的文档都是软件测试的对象。

3.3 软件测试设计的四个关键问题：

1. 测试由谁来执行
2. 测试什么
3. 什么时候进行测试
4. 怎样进行测试

4. 软件测试的目的

- 执行软件来获得软件在可用性方面的信心并且证明软件能够满意的工作。证明、检测和预防已经成为一个良好测试的重要目标。

5. 测试在各个阶段的任务

阶段	任务
项目规划阶段	负责从单元测试到系统测试的整个测试阶段的监控。
需求分析阶段	确定测试需求分析、系统测试计划的制定，评审后成为管理项目。测试需求分析是对产品生命周期中测试所需的资源、配置、每阶段评判通过的规约；系统测试计划则是依据软件的需求规格说明书，制定测试计划和设计相应的测试用例。
详细设计和概要设计阶段	确保集成测试计划和单元测试计划完成。
编码阶段	由开发人员进行自己负责部分的代码测试。在项目较大时，由专人进行编码阶段的测试任务。
测试阶段	依据测试代码进行测试，并提交相应的测试报告和测试结束报告。

6. 测试信息流

- 测试过程需要两类输入信息：
    1. 软件配置：指测试对象。通常包括软件需求规格说明、软件设计规格说明、源代码等。
    2. 测试配置：通常包括测试计划、测试步骤、测试用例以及实施测试的测试程序、测试工具等。
- 对测试结果与预期的结果进行比较以后，即可判断是否存在错误，决定是否进入排错阶段，进行调试任务。由于修改可能会带来新的问题，我们需要对修改以后的程序重新测试，即进行回归测试。

7. 软件测试策略

7.1 定义

- 测试将按照什么样的思路和方式进行。

7.2 针对代码的软件测试策略

策略	描述
单元测试	测试对象是软件设计的最小单元（函数或子过程、类、类的成员函数、显示界面等）
集成测试	按照设计要求将通过单元测试的模块进行组装，有非增量式和增量式两种方法
确认测试	对照软件需求规格说明，对软件产品进行评估，以确认其是否满足软件需求
系统测试	软件和硬件进行了一系列的系统集成和测试，目标是证明系统的性能
验收测试	将最终产品与最终用户的当前需求进行比较的全过程，是交付之前的最后一次质量检验活动

8. 软件测试过程

1. 测试计划，确定测试基本原则、生成测试概要设计。
2. 测试需求分析。
3. 测试设计，包括测试用例设计和测试规程规格说明。
4. 测试规程实现。
5. 测试执行。
6. 总结生成报告。

9. 动态测试

- 所谓动态测试是指通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率和健壮性等性能，动态测试包括功能确认与接口测试、覆盖率分析、性能分析、内存分析等。

10. 面向对象测试

10.1 定义

- 当考虑面向对象软件时，单元的概念发生了变化。封装驱动了类和对象的定义，这意味着每个类和类的实例(对象)包装了属性(数据)和操纵这些数据的操作。一个类可以包含一组不同的操作，而一个特定的操作也可能存在于一组不同的类中。因此，单元测试的意义发生了较大变化。我们不再孤立地测试单个操作，而是将操作作为类的一部分。此时最小的可测试单位是封装的类或对象，而不再是个体的模块。

10.2 分类

面向对象软件的单元测试	最小的可测试单元是封装的类或对象，着重考察类的对象对消息队列的响应和对象状态的正确性
面向对象软件的集成测试	面向对象软件没有层次的控制结构，因此传统的自下而上或自上而下的集成测

	试策略并不适用于面向对象方法构造的软件，需要研究适合面向对象特征的新的集成测试策略。其测试有两种不同策略：（1）基于类间协作关系的横向测试。（2）基于类间继承关系的纵向测试。
面向对象软件的系统测试	系统测试应该尽量搭建与用户实际使用环境相同的测试平台，应该保证被测系统的完整性，对临时没有的系统设备部件，也应有相应的模拟手段。系统测试时，应该参考 OOA 分析的结果，对应描述的对象、属性和各种服务，检测软件是否能够完全“再现”问题空间。系统测试不仅是检测软件的整体行为表现，从另一个侧面看，也是对软件开发设计的再确认。

11. 黑盒测试

11.1 定义

- 黑盒测试又称为功能测试或数据驱动测试，着眼于程序外部结构，将被测试程序视为一个不能打开的黑盒子，完全不考虑程序内部逻辑结构和内部特性，主要针对软件界面、软件功能、外部数据库访问以及软件初始化等方面进行测试。

11.2黑盒测试的基本方法

1. 边界值分析法
2. 等价类测试
3. 基于决策表的测试

12. 白盒测试

12.1 定义

- 白盒测试也称结构测试或逻辑驱动测试，是针对被测单元内部是如何进行工作的测试，它的突出特点是基于被测程序的源代码，而不是软件的规格说明。在软件测试中，白盒测试一般是由程序员完成的，当然也有专门做白盒测试的测试工程师。白盒测试人员必须对测试中的软件有深入的认识，包括其结构、各组成部分及之间的关联，以及其内部的运行原理、逻辑，等等。白盒测试人员实际上是程序员和测试员的结合体。

12.2 白盒测试的基本方法

1. 程序结构分析
2. 逻辑覆盖
3. 基本路径测试

13. 软件测试管理过程

- 一般由一位对整个系统设计熟悉的设计人员编写测试大纲，明确测试内容和通过准则
- 在实现组将所开发的程序经验证后，提交测试组，由测试负责人员组织测试，一般流程如下：
  - 测试人员仔细阅读相关资料，全面熟悉系统，编写测试计划，设计测试用例
  - 测试过程分为：代码审查、单元测试、集成测试、确认测试和系统测试
- 在整个过程中我们需要对测试过程中每个状态进行记录、跟踪和管理，并提供相关的分析和统计功能，生成和打印各种分析统计报表。形成较为完备的软件测试管理文档，保障软件在开发过程中，避免同样的错误在此发生，从而提高软件开发质量。

软件测试的人员组织

- 需求分析规格说明审查小组：一名组长+系统分析员，软件开发管理员，软件设计、开发、测试人员和用户
- 软件设计评审小组：一名组长+系统分析员，软件设计人员、测试负责人员
- 软件测试：组长+由具有一定的分析、设计和编程经验的专业人员组成（3-5人）