# 软件测试期末报告

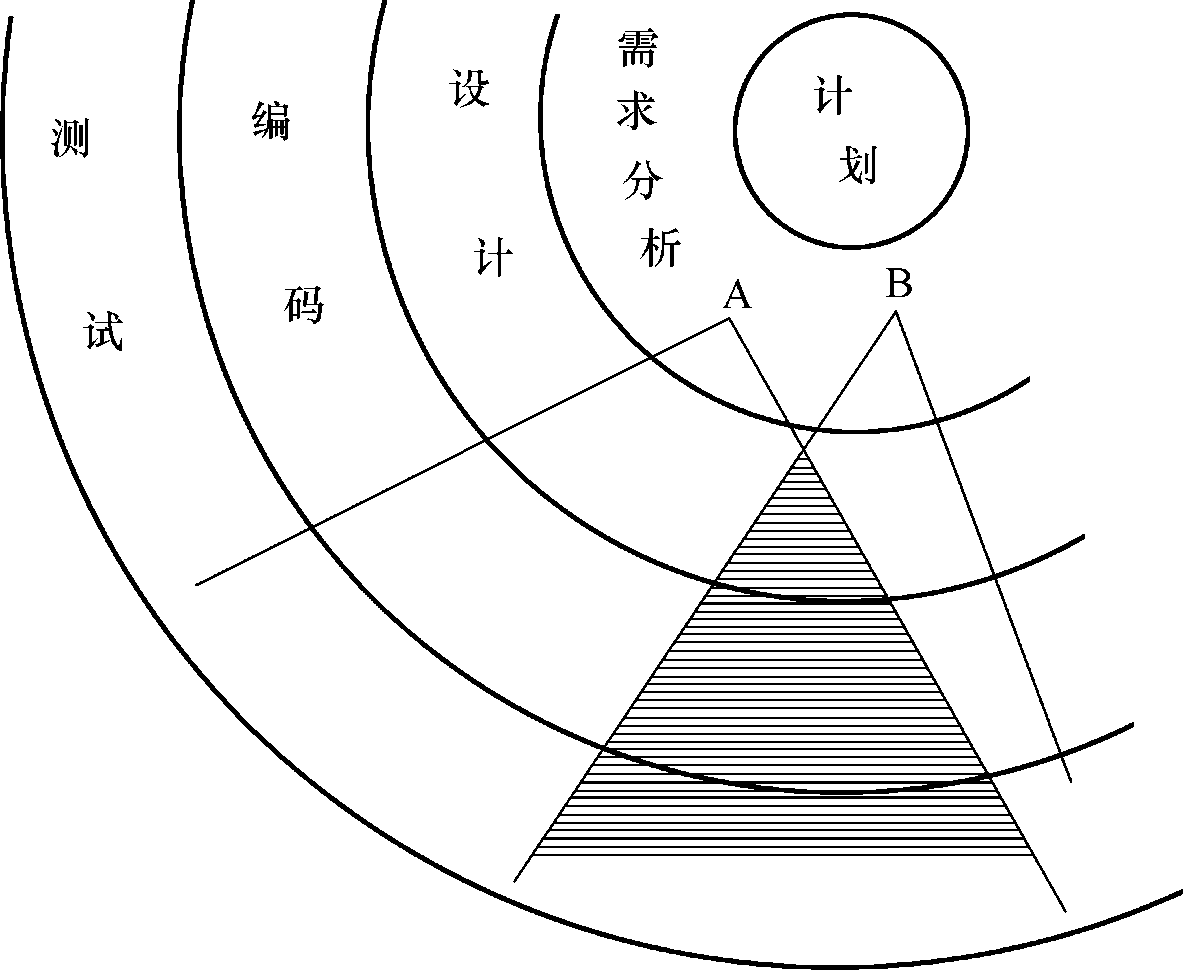
**姓名：曹辰鹏 学号：2019218117 班级：计算机19-4班**

软件测试是软件开发过程中必不可少的环节，在本次软件测试课程中，我循序渐进地学习了软件测试领域知识内容。下面我对本课程内容以自顶向下方式进行梳理总结。

第一章总结

第一章绪论与第二章软件测试策略是顶层总领概要内容。第一章绪论引入了缺乏软件测试造成的问题，介绍了软件危机和软件生存期（6个步骤，即计划、需求分析、设计、程序编写、测试、运行和维护），并给出了软件测试的定义（强调软件的正确性、除了正确性以外考虑健壮性与程序效率）。

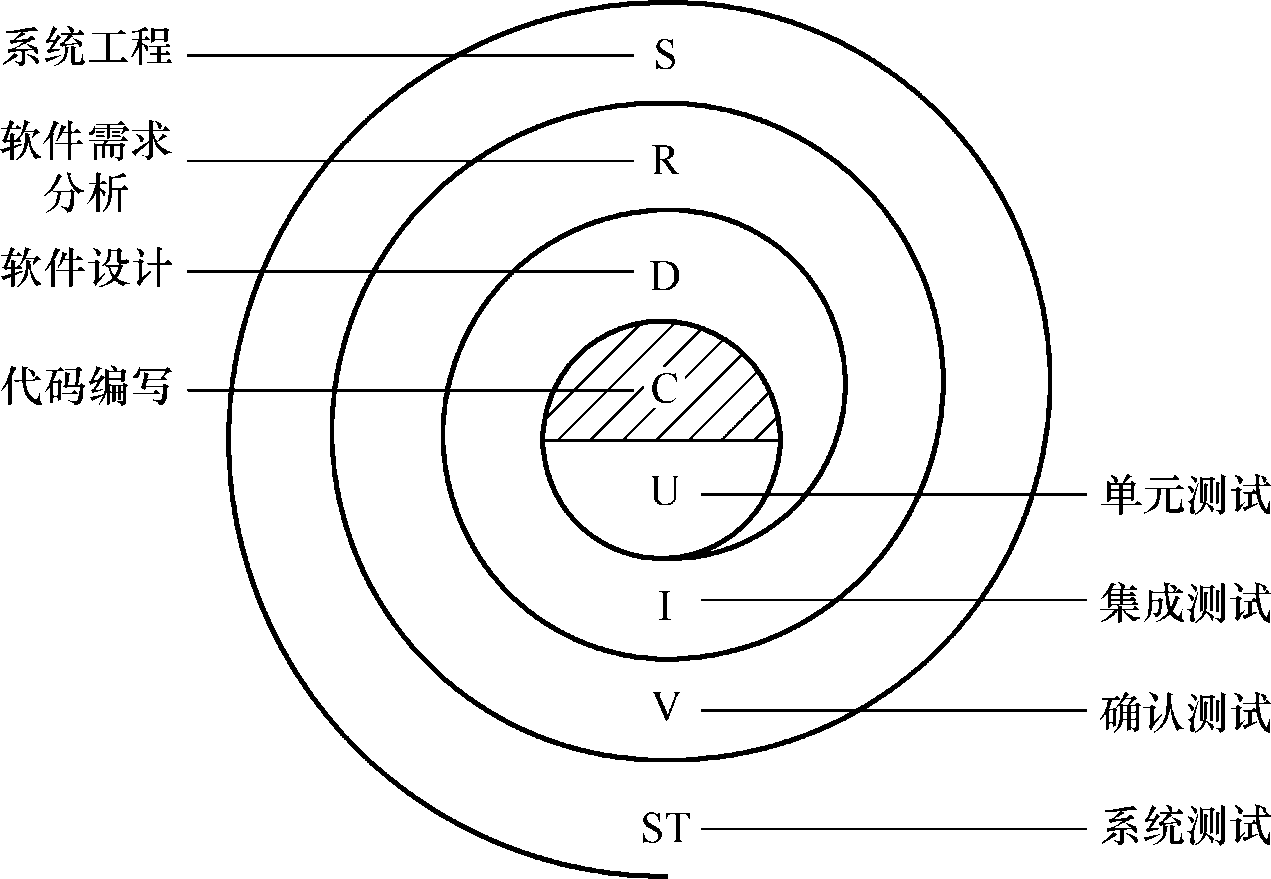
接下来介绍的软件测试的基本原则，软件测试与开发是并行的。确认、验证与测试在整个软件开发过程中作为质量保证的手段，应当最终保证软件产品的正确性、完整性和一致性。最后，软件错误是无法避免的，与软件测试相关的主要课题仍值得人们去探索研究。同时需要注意的是越是早发现错误，需要的成本就越小，影响就越小。



第二章总结

在第二章中，软件测试的生命周期包括计划阶段（标志测试条件和测试优先级）、设计阶段（设计测试用例）、开发阶段（测试开发、设计脚本）、执行阶段（执行测试用例）、评估阶段（测试结果与预期结果比较）。对于发现的错误数，可以多方面统计。

软件测试工作如下：



在螺旋线的核心点针对每个单元的源代码进行单元测试。在各单元测试完成以后，沿螺旋线向外前进，开始针对软件整体构造和设计的集成测试。然后是检验软件需求能否得到满足的确认测试，最后，来到螺旋线的最外层，把软件和系统的其他部分协调起来，当作一个整体，完成系统测试。这样，沿着螺旋线，从内向外，逐步扩展了测试的范围。

具体测试步骤包括单元测试检验程序最小单位错误、集成测试检验系统模块间连接正确性、确认测试保证确认的软件以及确认的配置、系统测试完成系统中各个组成部分的综合性检验。本章最后比较探讨了常用的测试技术，例如静态方法与动态方法，黑盒测试与白盒测试，回归测试、人工测试和自动测试。简单叙述了一下基本概念、他们的特点，以及进行比较。

第三章总结

基于第二章提到的黑盒测试与白盒测试的测试技术，第三章和第四章分别详细讲解了黑盒测试和白盒测试的核心思想以及多种方法。

黑盒测试又称功能测试，完全不考虑程序内部结构，是从用户观点出发的测试。具体的技术包括等价类划分、因果图、正交实验设计法、边值分析以及判定表驱动测试。

等价类划分：将数目极多的输入情况划分成若干等价类。分出来有效等价类和无效等价类。之后根据等价类确定测试用例。

因果图：列出原因和结果，根据具体情况得到原因和结果的对应关系，得到因果图，画出判定表，得到测试用例。

正交实验设计法：从大量实验点中挑出适量的、有代表性的点，按照“正交表”合理安排实验的设计方法。

边值分析法：使用五点法或者七点法设计测试用例。需要注意的是n变量函数的分析法。

判定表驱动：根据判定条件和对应的执行动作列出判定表，根据判定表设计测试用例。

最后就是功能测试，它完全不顾模块内部的实现逻辑，以检验输入/输出信息是否符合规格说明书中有关功能需求的规定为目标。包括了功能测试的系统化和模块功能的分解测试。

第四章总结

对于白盒测试技术来说，测试者可以看到被测的源程序，根据程序内部构造设计测试用例。单元测试大多从程序的内部结构出发设计测试用例，即多采用白盒测试方法。具体的技术包括程序结构分析（控制流图、数据流、信息流）、逻辑覆盖（语句覆盖、判定覆盖、路径覆盖、计算最少测试用例数、测试覆盖准则等）、域测试、符号测试、路径分析（路径表达式、路径数的计算、路径数、路径编码等）、程序插装、程序变异。

第五章总结

第五章详细介绍了集成测试。因为组件接口可能发生错误以及功能矛盾或者不相容，集成测试有其必要性。集成测试需要两种模块，驱动模块（模拟被测试上一级模块）和桩模块（模拟待测模块的下一级）。集成测试方法包括一次性集成（少数测试运行、分解的集成）、自顶向下集成（主控制模块开始，按程序控制层次向下进行集成）、自底向上集成（从层次的底部向上集成构件）、协作集成（支持一个特定的子功能）、基干集成（验证紧密耦合的子系统之间的互操作性）、层次集成（增量式检测）、客户/服务器集成（客户和服务器之间交互的稳定性）、分布服务集成（松散耦合的同等构件之间的交互稳定性）、高频集成（将新代码和已稳定的基线集成）和基于调用图集成（成对集成和相邻集成）。

第六章总结

在软件测试工作螺旋线中，第六章所讲述的系统测试就是螺旋线的最外层。系统测试包括非功能测试（包括安装测试、兼容性测试、安全性测试部、恢复测试等）、性能测试（包括负载测试、压力测试、容量测试等）和其他测试（阿尔法测试、贝塔测试等）。需要注意的是性能测试中的压力测试（对系统不断施加压力，确定系统的瓶颈或者不能接收的性能点，检查程序对异常情况的抵抗能力），以及容量测试（系统达到满负荷的情形下的承受能力，检查被测系统处理大数据量的能力）。

第七章总结

从第七章至第十章是软件测试的管理层面的知识。

前述第一章中我们提到了测试与开发是并行的观点，因此第七章中说明了项目开发过程中，各阶段需要做什么。依次需要测试准备，、测试实施、测试总结。准备过程中需要有需求分析和计划、完成测试环境的搭建、编制测试用例。测试实施时，执行测试用例，记录测试的数据（注意5W原则）、最后和相关人员沟通。最后将测试的结果进行总结，整理出数据，修订测试用例，进行用例库的维护。其中同时缺陷管理需要达到两个目标：一是确保每个被发现的缺陷都能够被解决，二是收集缺陷数据并根据缺陷趋势曲线识别和预防缺陷的频繁发生。需要对产生的缺陷进行一个统计分析，使用28原则，将产生缺陷的模块排列，找出占80%缺陷的模块，修复模块。在这一章中还介绍了成熟度模型CMM，将开发的软件按照各阶段缺陷占比分级，用于指导软件开发组织完善和改进开发过程。最后的缺陷度量，则有助于发现缺陷和剔除缺陷。

第九章总结

第九章阐述了软件评审方法，软件测试和软件评审都是产品验证和确认的活动。其主要目的是找出软件中的缺陷，及时排除软件开发过程中引入的缺陷，提高软件生产效率、降低消除缺陷的成本，为项目监控提供信息，找出测试无法发现的缺陷。软件评审有助于保证软件质量，提高团队生产效率。软件评审包括需求评审、设计评审、代码走查、代码审查。测试是软件开发过程中的一种质量控制活动，而质量控制又是质量管理的一个方面。

第十章总结

第十章讨论了软件质量与软件质量管理，当前软件质量问题的挑战，系统事故分析。可采用一些模型对程序中隐藏错误数量进行估计，如撒播模型、回归模型。软件产品表现为三种质量：内部质量、外部质量、使用质量。前两种质量有相同的六个质量特性，使用质量有四个质量特性。还介绍了软件质量管理的若干个过程，包括但不限于软件质量保证过程、软件验证过程。软件质量因素的分类方法从运行特性、软件经受修改的能力以及软件适应新环境的能力出发。最后就是知道程序排错和测试的区别，以及常用的排错方法如内存信息转储、跟踪等，还有常用的排错策略把，如试错法、回溯法等。

感悟

最后，通过本次课程学习，我掌握了软件测试的基本思想与一些技术方法。当下互联网软件应用正值蓬勃发展期，像谷歌、微软这样庞大体量的公司会因为软件漏洞导致货币和人员的损失，而保证优秀的软件测试可以避免如此庞大的损失。未来期望自己不论是作为软件的开发者还是管理者，都应当认识到软件测试是项目中不可缺少的一部分，是软件质量的重要保证。同时自己将在项目开发中保持对软件测试方法与技术的学习，尽可能发现并改正被项目软件中的错误，提高软件的可靠性，节省团队工作成本。