

软件质量保证和管理结课作业

学院：软件学院

专业：软件工程

姓名：陈晗

学号：14126072

手机：15201346449

邮箱：14126072@bjtu.edu.cn

如何保证软件测试的质量

软件测试，就是描述一种用来促进鉴定软件的正确性、完整性、安全性和质量的过程。

软件产品的质量取决于软件开发过程，软件测试作为软件生存期中的一个重要阶段，受重视程度越来越高。软件测试是保证软件质量和可靠性的关键步骤，也是用来验证软件是否能够完成所期望功能的唯一有效的方法。测试已不仅仅局限于软件开发中的一个阶段，它已开始贯穿整个软件开发过程，进行测试的时间越早，整个软件开发成本下降就越多。大量统计表明，软件测试的工作量往往占到软件开发总量的 40% 以上，在极端的情况下，甚至可能高达软件工程其它步骤成本总和的三至五倍，其目的是尽可能的提高软件产品的质量和可靠性。

要保证软件测试的质量，需要对软件测试的基本定义及相关概念有一定了解，接下来及对软件测试的定义、本质、具体步骤进行介绍。

软件测试的定义：

软件测试的经典定义：

在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。

狭义的软件测试：

测试是为了发现错误而执行程序的过程。是根据软件开发的各个阶段的说明和程序的内部结构而精心设计的一批测试用例（有输入数据及预期的结果），并利用这些测试用例执行程序及发现错误的过程。

广义的软件测试：

是对软件计划、软件系统分析、软件设计、软件编码进行的查错活动，包括代码执行和人工审查活动，测试的目的是找出软件生命周期的各个阶段的错误，有利于以后进行修改和纠正。但测试本身不修正错误，调试才会修正错误。即找错的活动是测试；分析错误的性质与位置，进行纠错的活动是调试，保证算法的正确实现。软件测试与程序测试都是查找错误的活动，差别在于查找错误的范围不同。

正确理解软件测试的本质：

由于测试的目标是暴露程序的错误，从心理学角度看，由设计者自己进行测试是不恰当的，设计小组和测试小组应该分别设立，有利于进行客观和公正的软件测试。测试是有限的，由于通常的测试过程不可能穷尽一切情况，即使经过了严格的测试之后，仍然可能存在没有被发现的错误隐藏在程序中，不能证明程序中没有错误。

因此，测试仅仅有可能找出程序的错误，测试不能证明程序是正确的。软件工程中所有其它阶段都是“建设性”的，软件工程师力图从抽象概念出发，逐步设计出具体的软件系统，而测试人员的工作表面上看却是“破坏性”的，竭力证明软件中含有错误，不能按预定要求正确工作。凡是进行对比的方式均可理解为测试验证。

软件测试的具体步骤：

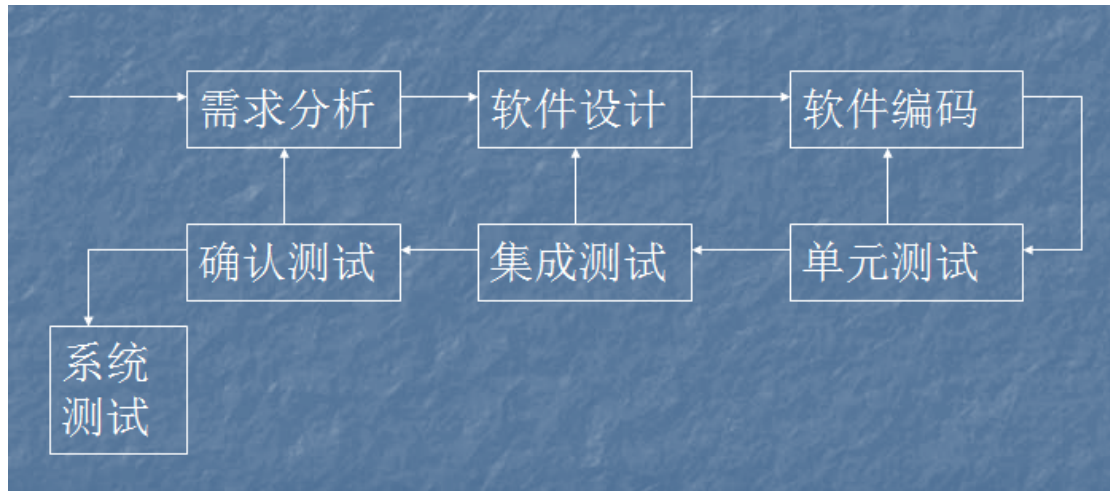
软件测试一般分为四个步骤：

- （1） 单元测试（也称模块测试）：针对软件设计的基本单元——程序模块，进行正确性检验的测试工作。目的在于发现各个模块内部可能存在的各种差错。单元测试需要从程序内部结构出发设计测试用例，多个模块可以平行、独立地进行测试；
- （2） 集成测试（也称组装测试，联合测试）：在单元测试的基础上，将所有模块按设计

要求集成在一起进行测试，以检验总体设计中各模块间的接口设计问题、模块之间的相互影响、上层模块存在的各种差错及全局数据结构对系统的影响等方面。

- (3) 确认测试（也称验收测试，有效性测试）：主要检验软件的功能和性能是否与需求说明书中的规定一致。
- (4) 系统测试：将软件系统作为一个元素，放入整个实际的计算机系统中，与计算机硬件、其他软件、使用人员等系统元素结合在一起，在实际使用环境下进行综合全面的测试。

软件测试步骤与软件开发各阶段的关系如下图所示：



在对软件测试有了一定的了解之后，就可以对如何保证软件测试的方法进行讨论了。下面将会介绍软件的可测试性、测试原则以及提高软件测试质量的方法。

软件的可测试性：

可测试软件具有以下特征：

- (1) 可操作性。可操作性是指：被测软件的错误很少，可以避免重复测试的开销；没有阻碍测试连续执行的错误；在软件设计时应允许在开发阶段进行部分测试活动。可观察性。
- (2) 可观察性包括：每个输入有唯一的输出；系统状态和变量可见，或在运行中可查询；过去的系统状态和变量可见，或在运行中可查询；所有影响输出的因素都可见；容易识别错误输出；自动报告内部错误；可获取源代码；
- (3) 可控制性。可控制性是指：所有可能的输出都产生于某种输入组合；通过某种输入组合，所有代码都可能被执行；软件测试人员可直接控制软件和硬件的状态及变量；输入和输出格式保持一致且有规范的构成；能够便利地对测试进行说明，以及方便地执行和重构测试。
- (4) 可分解性。软件系统由众多独立模块构成。每个软件模块均可独立进行测试。
- (5) 简单性。简单性包括功能简单性、结构简单性、代码简单性。
- (6) 稳定性。软件的变化是不经常的，变化时可控制的，软件的变化不形象已有的测试，失效后能够得到良好的回复。
- (7) 易理解性。易理解性包括：设计能够被很好的理解；内部、外部和共享构件之间的依赖性能够很好地被理解；测试人员可方便的获取技术文档，并及时掌握设计更改清空；技术文档组织合理，明确详细。

软件测试原则：

- (1) 将软件测试贯穿于软件开发的各个阶段中，在开发过程中尽早地发现和预防错误，杜绝隐患，提高软件质量；
- (2) 测试用例必须包含输入数据和与之对应的预期输出结果，精心设计测试用例；
- (3) 测试时应避免设计者检查自己设计的程序；
- (4) 设计测试用例时，应包括合理的与不合理的输入条件；
- (5) 充分注意测试中出现的错误群集现象，若发现错误数目较多，则可能残存的错误数目也较多，这种错误出现的群集现象，已为许多程序测试实践所证实；
- (6) 严格执行测试计划，以软件需求说明书为基准设计测试用例，排除测试的随意性；
- (7) 对每一个测试结果做全面检查，不能遗漏错误出现的征兆，软件修改后要进行回归测试，即用修改前测试过的测试用例进行测试，再用新的测试用例测试；
- (8) 妥善保存测试计划、测试用例、出错统计数据和最终分析报告，为维护提供方便。

提升软件测试质量的方法：

1.采用测试性设计技术 软件测试是目前用来验证软件是否能够完成所期望的功能的唯一有效的方法。但是在测试的实施过程中，由于种种原因导致测试的难度相当大，甚至出现了无法测试的情形。为了提高软件的可测试性，我们在软件设计时应当遵循测试性设计原则，通过改变设计或代码、为软件增加专门测试结构等方法来提高软件的可测试性。

(1) 测试驱动设计。这种设计就是直接把软件需求变成测试代码。在确定软件测试性能要求的基础上优先编写测试代码。先写验收测试，再写单元测试，并在开发过程中不断修正。

(2) 每个操作对应一个方法，使方法小型化。使用小型化方法说明和重载带缺省方法参数的方法，使得测试中调用这些方法变的很容易。

(3) 显示与控制分离。把代码移到 GUI 视图的外面，各种 GUI 动作就能成了模型上的简单方法调用。这样，在修改程序功能不会影响视图，同时通过方法调用测试功能也比间接地测试功能更容易。

(4) 对于可能要作为参数的类，做一个接口。用接口说明外部程序组件或在需要时改变接口形成一个空类作为参数传入。

2.选择合适的测试管理模型 模型是系统功能的形式化或半形式化的表示，支持输入状态组合的系统枚举。基于模型的测试主要考虑系统的功能，可以认为是功能测试的一种。测试模型体现了被测试系统的最本质的功能关系。而且要比系统本身更易于开发和分析。一个可测试的模型要能提供足够的信息用来产生测试用例。所以可测试的模型必须满足以下要求：

- (1) 必须是某种测试实现的完全准确的反映，模型必须表示要检查的所有特征；
- (2) 是对细节的抽象；
- (3) 可以表示所有事件和所有的动作；(4)可以表示系统的各种状态，以便由可知的方法来确定已达到或没有达到什么状态。

3.提高软件测试效率

为了提高软件测试效率，测试人员需要熟悉掌握软件涉及的领域知识，了解软件各项功能的重要程度和成熟程度，掌握测试理论和工具；用户是炎症需求正确性的主导力量，应允许发挥用户的积极作用。在组织软件测试时，可通过以下几个方面提高软件测试效率：

(1) 根据不同测试人员的特点进行测试分工，单元测试应以软件开发人员为主进行，以保证每个单元能够完成设计的功能。在很多情况下，集成测试也可以开发人员为主进行。当

软件体系结构完成后，独立测试人员机构计入；

（2）软件测试人员应注重与用户沟通，及早发现需求分析、理解不合理的问题，避免今后花费大量的资源和时间进行修改；

（3）对于软件开发人员，需加强测试方法的培训，提高自我测试的效率；

（4）在选择独立测试人员是，尽量选择比较熟悉了解被测试软件相关领域知识的人员；

（5）独立测试人员应该在软件开发的需求阶段就参与项目的研制，以便更好地制定测试计划、确定测试目标以及便携测试用例。通过找出项目中关键的模块和错误率高的模块，可使用测试首先集中在重要的部分，避免发生把过多的时间花费在非重要模块的测试而没有时间测试重要的模块的情况；

（6）被测试软件在测试中发现了问题，需要进行有组织的分析研究，然后权衡利弊进行规范化修改，避免反复修改，反复测试；

（7）规范软件配置管理，通过管理及技术手段，对软件和文档版本进行控制，保障软件测试的有效性。

小结：

总之，软件测试是一份需要完整流程的工作，我们必须对其有一个完备的计划。在接到测试任务后，基于种种其它因素的考虑，测试人员往往急于进度，立即投入到具体的测试工作去了，测试、记录、分析，忙的不亦乐乎，工作进行了一半才发现，或是硬件配置不符合要求，或是网络环境不理想，甚至软件版本不对，一时弄得骑虎难下，这都是没有做好测试准备惹的祸。那么我们应该如何做好性能测试的准备工作呢？

做软件项目有需求调查、需要分析，我们做测试也一样。在拿到测试任务后，我们首要的任务就是分析测试任务，在开始测试前，我们至少要弄清以下几个问题：

a)要测试什么或测试的对象是谁？

b)要测试什么问题或我们想要弄清楚或是论证的问题？

c)哪些因素会影响测试结果？

d)需要怎样的测试环境？

e)应该怎样测试？

只有在认真调查测试需求和仔细分析测试任务后，才有可能弄清以上一系列的问题，只有对测试任务非常清楚，测试目标极其明确的前提下，我们才可能制定出切实可行的测试计划。

软件生命周期每一阶段中都应包含测试，从静态测试到动态测试，要求检验每一个阶段的成果是否符合质量要求和达到定义的目标，尽可能早的发现错误并加以修正。如果不在早期阶段进行测试，错误的不断扩散、积累常常会导致最后成品测试的巨大困难、开发周期的延长、开发成本的剧增等等。

软件测试与软件质量的相同点在于二者都是贯穿整个软件开发生命周期的流程。软件质量保证的职能是向管理层提供正确的可视化的信息，从而促进与协助流程改进。软件质量保证还充当测试工作的指导者和监督者，帮助软件测试建立质量标准、测试过程评审方法和测试流程，同时通过跟踪、审计和评审，及时发现软件测试过程中的问题，从而帮助改进测试或整个开发的流程等，因此有了软件测试，测试工作就可以被客观的检查与评价，同时也可以协助测试流程的改进。而软件测试为提供数据和依据，帮助软件测试更好地了解质量计划的执行情况、过程质量、产品质量和过程改进进展，从而使软件测试更好地做好下一步工作。