开发思想，逻辑能力

目录

测试用例

测试用例（Test Case）是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，以便测试某个程序路径或核实是否满足某个特定需求。指对一项特定的软件产品进行测试任务的描述，体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等，并形成文档。测试用例构成了设计和制定测试过程的基础。

编制测试用例的具体做法：

1、测试用例文档 2、测试用例的设置 3、设计测试用例

测试用例在软件测试中的作用：

1、指导测试的实施。测试用例主要适用于集成测试、系统测试和回归测试。

2、规划测试数据的准备

3、编写测试脚本的"设计规格说明书"

4、评估测试结果的度量基准。完成测试实施后需要对测试结果进行评估，并且编制测试报告。判断软件测试是否完成、衡量测试质量需要一些量化的结果。例：测试覆盖率是多少、测试合格率是多少、重要测试合格率是多少，等等。

5、分析缺陷的标准

测试用例的设计

（一）白盒技术

白盒测试是结构测试，所以被测对象基本上是源程序，以程序的内部逻辑为基础设计测试用例。

1、逻辑覆盖

程序内部的逻辑覆盖程度，当程序中有循环时，覆盖每条路径是不可能的，要设计使覆盖程度较高的或覆盖最有代表性的路径的测试用例。

(1)语句覆盖。

为了个提高发现错误的可能性，在测试时应该执行到程序中的每一个语句。语句覆盖是指设计足够的测试用例，使被测试程序中每个语句至少执行一次。

(2)判定覆盖。

判定覆盖指设计足够的测试用例，使得被测程序中每个判定表达式至少获得一次“真”值和“假”值，从而使程序的每一个分支至少都通过一次，因此判定覆盖也称分支覆盖。

(3)条件覆盖。

条件覆盖是指设计足够的测试用例，使得判定表达式中每个条件的各种可能的值至少出现一次。

(4)判定/条件测试。

该覆盖标准指设计足够的测试用例，使得判定表达式的每个条件的所有可能取值至少出现一次，并使每个判定表达式所有可能的结果也至少出现一次。

(5)条件组合覆盖。

条件组合覆盖是比较强的覆盖标准，它是指设计足够的测试用例，使得每个判定表达式中条件的各种可能的值的组合都至少出现一次。

(6)路径覆盖。

路径覆盖是指设计足够的测试用例，覆盖被测程序中所有可能的路径。

在实际的逻辑覆盖测试中，一般以条件组合覆盖为主设计测试用例，然后再补充部分用例，以达到路径覆盖测试标准。

2.循环覆盖

3.基本路径测试

（二）黑盒技术

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

1.等价类划分

(1)划分等价类。

①如果某个输入条件规定了取值范围或值的个数。则可确定一个合理的等价类(输入值或数在此范围内)和两个不合理等价类(输入值或个数小于这个范围的最小值或大于这个范围的最大值)。

②如果规定了输入数据的一组值，而且程序对不同的输入值做不同的处理，则每个允许输入值是一个合理等价类，此处还有一个不合理等价类(任何一个不允许的输入值)。

③如果规定了输入数据必须遵循的规则，可确定一个合理等价类(符合规则)和若干个不合理等价类(从各种不同角度违反规则)。

④如果已划分的等价类中各元素在程序中的处理方式不同，则应将此等价类进一步划分为更小的等价类。

(2)确定测试用例。

①为每一个等价类编号。

②设计一个测试用例，使其尽可能多地覆盖尚未被覆盖过的合理等价类。重复这步，直到所有合理等价类被测试用例覆盖。

③设计一个测试用例，使其只覆盖一个不合理等价类。

2.边界值分析

使用边界值分析方法设计测试用例时一般与等价类划分结合起来。但它不是从一个等价类中任选一个例子作为代表，而是将

测试边界情况作为重点目标，选取正好等于、刚刚大于或刚刚小于边界值的测试数据。

(1)如果输入条件规定了值的范围,可以选择正好等于边界值的数据作为合理的测试用例，同时还要选择刚好越过边界值的数据作为不合理的测试用例。如输入值的范围是[1，100]，可取 0，1，100，101 等值作为测试数据。

(2)如果输入条件指出了输入数据的个数，则按最大个数、最小个数、比最小个数少 1、比最大个数多 1 等情况分别设计测试用例。如，一个输入文件可包括 1--255 个记录，则分别设计有 1 个记录、255 个记录，以及 0 个记录的输入文件的测试用例。

(3)对每个输出条件分别按照以上原则(1)或(2)确定输出值的边界情况。如，一个学生成绩管理系统规定，只能查询 95--98 级大学生的各科成绩，可以设计测试用例，使得查询范围内的某一届或四届学生的学生成绩，还需设计查询 94 级、99 级学生成绩的测试用例(不合理输出等价类)。

由于输出值的边界不与输入值的边界相对应，所以要检查输出值的边界不一定可能，要产生超出输出值之外的结果也不一定能做到，但必要时还需试一试。

(4)如果程序的规格说明给出的输入或输出域是个有序集合(如顺序文件、线形表、链表等)，则应选取集合的第一个元素和最后一个元素作为测试用例。

3.错误推测

在测试程序时，人们可能根据经验或直觉推测程序中可能存在的各种错误，从而有针对性地编写检查这些错误的测试用例，这就是错误推测法。

4.因果图

等价类划分和边界值方法分析方法都只是孤立地考虑各个输入数据的测试功能，而没有考虑多个输入数据的组合引起的错误。

5.综合策略

每种方法都能设计出一组有用例子，用这组例子容易发现某种类型的错误，但可能不易发现另一类型的错误。因此在实际测试中，联合使用各种测试方法，形成综合策略，通常先用黑盒法设计基本的测试用例，再用白盒法补充一些必要的测试用例

测试用例模板：

模块描述 XX 项目/XX 模块

提交时间

测试人

测试时间

测试环境

测试工具

功能 1 描述 XX 项目/XX 模块/XX 功能

用例目的

前提条件

输入/动作

期望的输出/相应

实际情况

期待输出和实际比较

软件测试工具也分为自动化软件测试工具和测试管理工具。国内免费软件测试工具有：黑盒测试工具 AutoRunner 可以用来完成功能测试、回归测试、每日构建测试与自动回归测试等工作和 TestCenter 是一款功能强大测试管理工具，它可以帮助您：实现测试用例的过程管理，对测试需求过程、测试用例设计过程、业务组件设计实现过程等整个测试过程进行管理。

面试题

一、判断题
1．软件测试的目的是尽可能多的找出软件的缺陷。（Y）
2．Beta 测试是验收测试的一种。（Y）
3．验收测试是由最终用户来实施的。（N）
4．项目立项前测试人员不需要提交任何工件。（Y）
5．单元测试能发现约 80%的软件缺陷。（Y）
6．代码评审是检查源代码是否达到模块设计的要求。（N）
7．自底向上集成需要测试员编写驱动程序。（Y）
8．负载测试是验证要检验的系统的能力最高能达到什么程度。（N）
9．测试人员要坚持原则，缺陷未修复完坚决不予通过。（N）　　看情况有时候就是坚持原则。
10．代码评审员一般由测试员担任。（N）
11．我们可以人为的使得软件不存在配置问题。（N）
12．集成测试计划在需求分析阶段末提交。（N）
二、选折
1．软件验收测试的合格通过准则是：（ABCD）
A．　软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求。
B．　所有测试项没有残余一级、二级和三级错误。
C．　立项审批表、需求分析文档、设计文档和编码实现一致。
D．　验收测试工件齐全。
2．软件测试计划评审会需要哪些人员参加？（ABCD）
A．项目经理　　B．SQA 负责人　　C．配置负责人　　D．测试组
3．下列关于 alpha 测试的描述中正确的是：（AD）
A．alpha 测试需要用户代表参加　　D．alpha 测试是验收测试的一种
4．测试设计员的职责有：（BC）
B．设计测试用例　　C．设计测试过程、脚本
5．软件实施活动的进入准则是：（ABC）
A．需求工件已经被基线化　　B．详细设计工件已经被基线化
C．构架工件已经被基线化
三、添空
1.软件验收测试包括：正式验收测试，alpha 测试，beta 测试。
2.系统测试的策略有：功能测试，性能测试，可靠性测试，负载测试，易用性测试，强度测试，安全测试，配置测试，安装测试，卸载测试，文挡测试，故障恢复测试，界面测试，容量测试，兼容性测试，分布测试，可用性测试，（有的可以合在一起，分开写只要写出 15 就满分哦）
3.设计系统测试计划需要参考的项目文挡有：软件测试计划，软件需求工件和迭代计划。
4.对面向过程的系统采用的集成策略有：自顶向下，自底向上两种。
5.（这题出的有问题哦，详细的 5 步骤为~~）通过画因果图来写测试用例的步骤

为：
（1）分析软件规格说明描述中，哪些是原因（即输入条件或输入条件的等价类），哪些是结果（即输出条件），并给每个原因和结果赋予一个标识符。
（2）分析软件规格说明描述中的语义，找出原因与结果之间，原因与原因之间对应的是什么关系？根据这些关系，画出因果图。
（3）由于语法或环境限制，有些原因与原因之间，原因与结果之间的组合情况不可能出现。为表明这些特殊情况，在因果图上用一些记号标明约束或限制条件。
（4）把因果图转换成判定表。
（5）把判定表的每一列拿出来作为依据，设计测试用例。
四、简答（资料是搜集整理的，感谢前辈的解题）无
1.区别阶段评审的与同行评审
同行评审目的:发现小规模工作产品的错误,只要是找错误;
阶段评审目的:评审模块 阶段作品的正确性 可行性 及完整性
同行评审人数:3-7 人 人员必须经过同行评审会议的培训,由 SQA 指导
阶段评审人数:5 人左右 评审人必须是专家 具有系统评审资格
同行评审内容:内容小 一般文档 < 40 页, 代码 < 500 行
阶段评审内容: 内容多,主要看重点
同行评审时间:一小部分工作产品完成
阶段评审时间: 通常是设置在关键路径的时间点上!

2.什么是软件测试
为了发现程序中的错误而执行程序的过程

3 简述集成测试的过程
系统集成测试主要包括以下过程：
1. 构建的确认过程。
2. 补丁的确认过程。
3. 系统集成测试测试组提交过程。
4. 测试用例设计过程。
5. 测试代码编写过程。
6. Bug 的报告过程。
7. 每周/每两周的构建过程。
8. 点对点的测试过程。
9. 组内培训过程。

4 怎么做好文档测试
仔细阅读，跟随每个步骤，检查每个图形，尝试每个示例。P142
检查文档的编写是否满足文档编写的目的
内容是否齐全，正确
内容是否完善
标记是否正确

5 白盒测试有几种方法
总体上分为静态方法和动态方法两大类。
静态：关键功能是检查软件的表示和描述是否一致,没有冲突或者没有歧义
动态：语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖。

6 系统测试计划是否需要同行审批，为什么
需要，系统测试计划属于项目阶段性关键文档，因此需要评审。

7Alpha 测试与 beta 的区别
Alpha 测试 在系统开发接近完成时对应用系统的测试；测试后仍然会有少量的设计变更。这种测试一般由最终用户或其它人员完成，不能由程序或测试员完成。黑盒、白盒、 压力、应力等等
Beta 测试 当开发和测试根本完成时所做的测试，最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其它人员完成，不能由程序员或测试员完成。
α、β、λ 常用来表示软件测试过程中的三个阶段，α 是第一阶段，一般只供内部测试使用；β 是第二个阶段，已经消除了软件中大部分的不完善之处，但仍有可能还存在缺陷和漏洞，一般只提供给特定的用户群来测试使用；λ 是第三个阶段，此时产品已经相当成熟，只需在个别地方再做进一步的优化处理即可上市发行。

8 比较负载测试，容量测试和强度测试的区别
负载测试：在一定的工作负荷下，系统的负荷及响应时间。
强度测试：在一定的负荷条件下，在较长时间跨度内的系统连续运行给系统性能所造成的影响。
容量测试：容量测试目的是通过测试预先分 析出反映软件系统应用特征的某项指标的极限值（如最大并发用户数、数据库记录数等），系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据的，并且它的目的是显示系统可以处理目标内确定的数据容量。

9 测试结束的标准是什么？
用例全部测试。
覆盖率达到标准。
缺陷率达到标准。
其他指标达到质量标准

10 描述软件测试活动的生命周期？
测试周期分为计划、设计、实现、执行、总结。其中：
计划：对整个测试周期中所有活动进行规划，估计工作量、风险，安排人力物力资源，安排进度等；
设计：完成测试方案，从技术层面上对测试进行规划；
实现：进行测试用例和测试规程设计；

执行：根据前期完成的计划、方案、用例、规程等文档，执行测试用例。
总结：记录测试结果，进行测试分析，完成测试报告。

11 软件的缺陷等级应如何划分？
A 类—严重错误，包括以下各种错误： 1． 由于程序所引起的死机,非法退出 2． 死循环 3． 数据库发生死锁 4． 因错误操作导致的程序中断 5． 功能错误 6． 与数据库连接错误 7． 数据通讯错误
B 类—较严重错误，包括以下各种错误：1．程序错误 2．程序接口错误 3．数据库的表、业务规则、缺省值未加完整性等约束条件
C 类—一般性错误，包括以下各种错误： 1． 操作界面错误（包括数据窗口内列名定义、含义是否一致） 2． 打印内容、格式错误 3． 简单的输入限制未放在前台进行控制 4． 删除操作未给出提示 5． 数据库表中有过多的空字段
D 类—较小错误，包括以下各种错误： 1． 界面不规范 2． 辅助说明描述不清楚 3． 输入输出不规范 4． 长操作未给用户提示 5． 提示窗口文字未采用行业术语 6． 可输入区域和只读区域没有明显的区分标志
E 类—测试建议

大体是这样，还会有一些变动，同时最后一道题出的是画流程图和控制图的题，等腰三角形那个，好了，仅供参考
一、等价类划分：三角形三条边 A、B、C 的数据类型不同
二、边界值分析：由于三角形的边长可以是正整数或正小数，所以就不对长度进行测试，那么边界值分析就不用了
三、因果图法：三角形的三条边数据输入组合
我们看一下三角形的流程图：

我们再分析一下三角形的等价类：
有效等价类：
输入 3 个正整数或正小数：
1、两数之和大于第三数，如 A<B+C；B<C+A；C<A+B
2、两数之和不大于第三数
3、两数相等，如 A=B 或 B=C 或 C=A
4、三数相等，如 A=B=C
5、三数不相等，如 A！=B，B！=C，C！=A
无效等价类：
1、空
2、负整数
3、非数字
4、少于三个数
三角形测试用例类别
输入条件 有效等价类 无效等价类
是否是三角形
(A>0) (1)
(B>0) (2)

(C>0) (3)
(A+B>C) (4)
(B+C>A) (5)
(C+A>B) (6) (A<=0) (7)
(B<=0) (8)
(C<=0) (9)
(A+B<=C) (10)
(B+C<=A) (11)
(C+A<=B) (12)
是否是等腰三角形
(A=B) (13)
(B=C) (14)
(C=A) (15) (A!=B)and(B!=C)and(C!=A) (16)
是否是等腰直角三角形 (A=B)and(A2+B2=C2) (17)
(B=C)and(B2+C2=A2) (18)
(C=A)and(C2+A2=B2) (19) (A!=B)and(B!=C)and(C!=A) (20)
是否是等边三角形 (A=B)and(B=C)and(C=A) (21) (A!=B) (22)
(B!=C) (23)
(C!=A) (24)
三角形测试用例：
序号 [A,B,C] 覆盖等价类 输出
1 [3,4,5] (1)(2)(3)(4)(5)(6) 是三角形
2 [0,1,2] (7) 非三角形
3 [1,0,2] (8) 非三角形
4 [1,2,0] (9) 非三角形
5 [1,2,3] (10) 非三角形
6 [1,3,2] (11) 非三角形
7 [3,1,2] (12) 非三角形
8 [3,3,4] (1)(2)(3)(4)(5)(6)(13) 等腰三角形
9 [3,4,4] (1)(2)(3)(4)(5)(6)(14) 等腰三角形
10 [3,4,3] (1)(2)(3)(4)(5)(6)(15) 等腰三角形
11 [2√2,2√2,4] (1)(2)(3)(4)(5)(6)(17) 等腰直角三角形
12 [4,2√2,2√2] (1)(2)(3)(4)(5)(6)(18) 等腰直角三角形
13 [2√2,4,2√2] (1)(2)(3)(4)(5)(6)(19) 等腰直角三角形
14 [3,4,5] (1)(2)(3)(4)(5)(6)(16)(20)(22)(23)(24) 是三角形
15 [3,3,3] (1)(2)(3)(4)(5)(6)(16)(21) 等边三角形
16 [,,,] 无效等价类 错误提示
17 [-3,4,5] 无效等价类 错误提示
18 [a,3,@] 无效等价类 错误提示
19 [3,4] 无效等价类 错误提示

针对缺陷采取怎么样的管理措施

只是对缺陷的生命周期进行管理和跟踪，Bugzilla 或者 TD 已经足够了，

1.要更好的管理缺陷，必须引入缺陷管理工具，商用的或者开源的都可。

2.根据缺陷的生命周期，考虑缺陷提交的管理、缺陷状态的管理和缺陷分析的管理。

3.所有发现的缺陷（不管是测试发现的还是走读代码发现的）都必须全部即时的、准确的提交到缺陷管理工具中，这是缺陷提交的管理。

4.缺陷提交后，需要即时的指派给相应的开发人员，提交缺陷的人需要密切注意缺陷的状态，帮助缺陷的尽快解决。缺陷解决后需要即时对缺陷的修复进行验证。这样的目的有两个：一个是让缺陷尽快解决；二是方便后面缺陷的分析（保证缺陷相关的信息准确，如龄期等），这是缺陷状态的管理。

5.为了更好的改进开发过程和测试过程，需要对缺陷进行分析，总结如缺陷的类别、缺陷的龄期分布等信息，这是缺陷分析的管理。

测试计划的目的是什么？

答：测试的目的是想以最少的人力、物力和时间找出软件中潜在的各种错误和缺陷，通过修正种错误和缺陷提高软件质量，回避软件发布后由于潜在的软件缺陷和错误造成的隐患带来的商业风险。

软件测试应该划分几个阶段?简述各个阶段应重点测试的点?各个阶段的含义？

答：大体上来说可分为单元测试, 集成测试, 系统测试, 验收测试, 每个阶段又分为以下五个步骤:

测试计划，测试设计，用例设计，执行结果，测试报告

初始测试集中在每个模块上，保证源代码的正确性，，该阶段成为单元测试，主要用白盒测试方法。

接下来是模块集成和集成以便组成完整的软件包。集成测试集中在证实和程序构成问题上。主要采用黑盒测试方法，辅之以白盒测试方法。

软件集成后，需要完成确认和系统测试。确认测试提供软件满足所有功能、性能需求的最后保证。确认测试仅仅应用黑盒测试方法。

单元测试

单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等等。它是软件动态测试的最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。

集成测试

集成测试是在软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。

系统测试

系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求，检查软件的行为和输出是否正确并非一项简单的任务，它被称为测试的“先知者问题”。

验收测试

验收测试旨在向软件的购买者展示该软件系统满足其用户的需求。它的测试数据通常是系统测试的测试数据的子集.

回归测试

回归测试是在软件维护阶段，对软件进行修改之后进行的测试。其目的是检验对软件进行的修改是否正确。

1.测试退出标准

测试退出标准为完成测试需求中列出的所有功能及测试过程中发现缺陷的回归测试。

1. 单元测试退出标准

1) 单元测试用例设计已经通过评审

2) 核心代码 100% 经过 Code Review

3) 单元测试功能覆盖率达到 100%

4) 单元测试代码行覆盖率不低于 80%

5) 所有发现缺陷至少 60%都纳入缺陷追踪系统且各级缺陷修复率达到标准

6) 不存在 A、B 类缺陷

7) C、D、E 类缺陷允许存在

8) 按照单元测试用例完成了所有规定单元的测试

9) 软件单元功能与设计一致

2. 集成测试退出标准

1) 集成测试用例设计已经通过评审

2) 所有源代码和可执行代码已经建立受控基线，纳入配置管理受控库，不经过审批不能随意更改

3) 按照集成构件计划及增量集成策略完成了整个系统的集成测试

4) 达到了测试计划中关于集成测试所规定的覆盖率的要求

5) 集成工作版本满足设计定义的各项功能、性能要求

6) 在集成测试中发现的错误已经得到修改，各级缺陷修复率达到标准

7) A、B 类 BUG 不能存在

8) C、D 类 BUG 允许存在,但不能超过单元测试总 BUG 的 50%

9) E 类 BUG 允许存在

3. 系统测试退出标准

1) 系统测试用例设计已经通过评审

2) 按照系统测试计划完成了系统测试

3) 系统测试的功能覆盖率达 100%

4) 系统的功能和性能满足产品需求规格说明书的要求

5) 在系统测试中发现的错误已经得到修改并且各级缺陷修复率达到标准

6) 系统测试后不存在 A、B、C 类缺陷

7) D 类缺陷允许存在，不超过总缺陷的 5%

8) E 类缺陷允许存在，不超过总缺陷的 10%

如果能够执行完美的黑盒测试，还需要进行白盒测试吗？

黑盒测试：从用户角度出发，根据规格说明设计测试用例，并不涉及程序的内部特性和内部结构，只依靠被测程序输入和输出之间的关系或程序的功能设计测试用例。黑盒测试有两个显著特点：

（1）黑盒测试与软件的具体实现过程无关，在软件实现的过程发生变化时，测试用例仍然可以用。

（2）黑盒测试用例的设计可以和软件实现同时进行，这样能够压缩总的开发时间。

黑盒测试主要是为了发现以下几类错误：

1、是否有不正确、遗漏或额外的功能实现？

2、在接口上，输入是否能正确的接受？能否输出正确的结果？

3、是否有数据结构错误或外部信息（例如数据文件）访问错误？

4、性能上是否能够满足要求？

5、是否有初始化或终止性错误？

白盒测试：已知程序的内部结构，检查内部操作是否按规定执行。主要对程序细节进行严密检验，针对特定条件和循环设计测试用例，对程序的逻辑路径进行测试。通过在程序的不同点检查程序状态，确定实际状态是否与预期的状态一致。

白盒测试主要是想对程序模块进行如下检查：

1、程序的所有语句至少执行一次。

2、对所有的逻辑条件都能至少执行一次。

3、在循环的边界和运行的界限内执行循环体。

4、测试内部数据结构的有效性，等等。

从以上可以看出就算执行了完美的黑盒测试也是无法测试程序内部特定部位，另外当规格说明本身有误，也不能发现问题。而白盒测试能对程序的内部特定部位进行覆盖测试，所以黑盒和白盒测试为互补关系，结合起来进行测试用例的设计更为合理。

经验表明，通常在进行单元测试时采用白盒测试方法，集成测试采用灰盒测试方法，系统测试采用黑盒测试方法。

软件的缺陷等级应如何划分？

1．致命错误，可能导致本模块以及其他相关模块异常，死机等问题；

2．严重错误，问题局限在本模块，导致模块功能失效或异常退出

3．一般错误，模块功能部分失效；

4．建议问题，由问题提出人对测试对象的改进意见；

静态测试和动态测试的区别有哪些？

静态方法是指不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。对需求规格说明书、软件设计说明书、源程序做结构分析、流程图分析、符号执行来找错。静态方法通过程序静态特性的分析，找出欠缺和可疑之处，例如不匹配的参数、不适当的循环嵌套和分支嵌套、不允许的递归、未使用过的变量、空指针的引用和可疑的计算等。静态测试结果可用于进一步的查错，并为测试用例选取提供指导。

动态测试方法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，这种方法由三部分组成：构造测试实例、执行程序、分析程序的输出结果。

软件测试过程基本上是这样的
1。软件测试计划和控制
2。测试分析和设计
3。测试实施和执行
4。退出测试标准
5。测试报告
6。测试结束活动

为什么要在一个团队中开展软件测试工作？

因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比 ISO 质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

你的测试职业发展是什么？

测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级测试工程师奔去。不断的更新自己改正自己，做好测试任务。

简述一下缺陷的生命周期
软件缺陷的生命周期指的是一个软件缺陷被发现、报告到这个缺陷被修复、验证直至最后关闭的完整过程。

简单的软件缺陷生命周期：
1、发现——打开：测试人员找到软件缺陷并将软件缺陷提交给开发人员；
2、打开——修复：开发人员再现、修复缺陷，然后提交测试人员去验证；
3、修复——关闭：测试人员验证修复过的软件，关闭已不存在的缺陷。

写出你所知道的 3 种常用的排序方法，并用其中一种方法设计出程序为数组 a[100]排序。
常用的排序算法有很多：
冒泡，快速排序，直接插入，希尔排序，选择排序，堆排序，归并排序！
就举冒泡排序(c++)：

```
void bubblesort()
{
    for (i = 1; i < max; i ++)
    {
      for (j = max - 1; j >= i; j --)
      if (a[j + 1] < a[j]) //小则交换
      {
        a[0] = a[j + 1];
```

```
            a[j + 1] = a[j];
            a[j] = a[0];
        }
    }
}
```

1. 目的

指导测试组为完成软件工程制定合理的测试计划。以及指导整个测试过程。

2. 范围

此文件适用于软件的测试过程。

3. 职责

3.1 本程序的责任机构是 SEPG. 负责对本流程的意见收集、改进更新、培训和解释工作。

3.2 本过程的知晓部门：研发部、测试质保部、技术管理部、客户代表。

3.3 相关部门或岗位职责：

3.3.1 测试人员：

根据项目组确定的软件开发过程，编写测试计划，（包括：集成测试计划、系统测试计划），编写测试用例，并实施测试，出具测试分析报告。

3.3.2 技术经理：负责配合编写测试计划，协调测试活动。

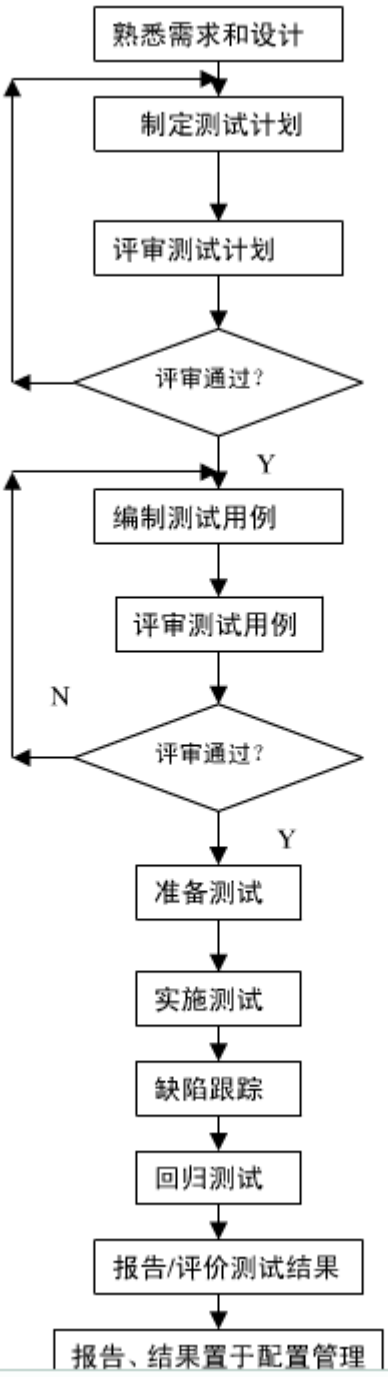
3.3.3 开发人员：配合测试。

3.3.4 评审人员：评审测试计划、测试用例、测试分析报告。

3.3.5 配置管理人员：对测试工作产品进行管理。

4. 术语

序号	术语	解释
1	测试用例	根据测试需求和设计所设计的测试输入、测试操作和预期结果。
2	软件测试	是根据软件开发各阶段的文档和程序的内部结构而精心设计一批测试用例（即输入数据及其预期的输出结果），并利用这些测试用例去运行程序，以发现程序缺陷的过程。
3	黑盒测试	把程序看成一只黑盒子，测试者完全不考虑程序的结构和处理过程。它根据需求说明书规定的功能来设计测试用例，检查程序的功能是否符合需求，又称“功能测试”。
4	集成测试	主要对系统内部的相互服务进行测试，是由底向上通过集成完成的功能模块，对程序内部具体单一功能模块的测试。
5	系统测试	将软件作为整个计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合在一起，在实际运行环境下，对计算机系统进行的一系列的测试。
6	验收测试	验收测试是以用户为主的测试。由用户设计测试用例，使用真实数据进行测试，分析测试的结果。
7	回归测试	指在第一次系统测试完，开发小组已经将所有的缺陷处理后，进行得第二次系统测试。


熟悉需求和设计
制定测试计划
评审测试计划
评审通过？
Y
编制测试用例
评审测试用例
N
评审通过？
Y
准备测试
实施测试
缺陷跟踪
回归测试
报告/评价测试结果
报告、结果置于配置管理

步骤描述	执行人
5.3.1 制定测试计划 技术经理在制定开发计划时（参见《软件项目策划过程》），应同时与测试质保部一起考虑项目的测试计划。测试质保部制定测试计划，协商安排集成测试和系统测试等工作。测试计划内容包括测试人员安排，测试方案、测试用例设计工作安排和测试执行工作安排等内容，按照《软件测试计划》模板制定。	技术经理、测试人员
5.3.2 评审测试计划 制定完成的测试计划要进行评审，评审参照《评审过程》进行。	技术经理
5.3.3 编制和评审测试用例 **5.3.3.1. 集成测试用例编制:** 测试人员根据《概要设计说明书》文档安排人员制定集成测试方案和测试用例，考虑需要集成的关键模块、集成的次序和需要测试的接口和测试的环境要求等工作。 **5.3.3.2. 系统测试用例编制:** 测试人员根据项目的《软件需求规格说明书》，参考项目设计文档，编写测试方案和测试用例，规定测试数据、测试预期结果、测试充分性评价等，重点在功能测试，兼顾性能测试，如确认需要测试的功能和不必测试功能；用户界面的确认；硬件、软件和通信接口的确认等等，完成后的测试用例交给技术经理审核，并组织有关人员参加评审，通过评审后入配置管理库进行管理。	技术经理、开发人员、测试人员
5.3.4 准备测试 **5.3.4.1 检查是否可以进行测试：** 集成测试的代码是否已完成单元测试；系统测试的代码是否已完成集成测试。 **5.3.4.2 代码及测试文档准备：** 集成测试和系统测试前开发或配置管理人员应向配置管理库归档最新测试代码及测试所必须的文档。 **5.3.4.3 测试环境搭建:** 对自己可以搭建测试环境的软件项目，测试时应从配置管理库中提取程序及测试所必需的文档，并根据测试方案搭建测试环境；对自己无法搭建测试环境，可以到现场进行测试，但必须采取措施保证不影响用户原有的系统（做测试方案时应给予考虑）。	开发人员、配置管理人员、测试人员
5.3.5 实施测试 **5.3.5.1.集成测试:** 测试人员根据设计说明书和测试方案进行集成测试. **5.3.5.2.系统测试:** 系统测试应尽量在与实际运行（使用）环境一致的环境下进行（如果不行，就要在与实际环境有可比性的模拟环境下进行），与计算机硬件、外设、支持软件、资料和人员等其它系统元素结合在一起，测试软件需求规格说明书列出的所有需求，确保所有的软件功能需求都能得到满足，所有的软件性能需求都能达到，所有的文档都是正确且便于使用；同时在测试过程中，应当按照顾客手	测试人员