一、基础知识点：

1.软件测试基础知识

2.测试流程

3.测试方法

4.测试用例设计

5.bug管理

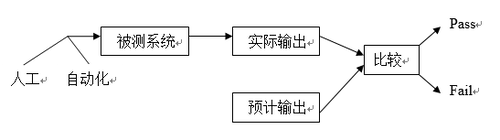
二、课程内容

1.1 软件测试基础知识

1． 软件测试的定义

1983年，IEEE提出的软件工程标准术语，软件测试定义如下：

“使用人工和自动手段来运行或测试某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别”。



缺点：只强调动态测试，忽略了静态测试。

G..J.Myers认为：

1）程序测试是为了发现错误而执行程序的过程；

2）好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案；

3）成功的测试是发现了至今为止尚未发现的错误的测试。

缺点：只强调了了发现错误，而忽视了缺陷。

以上两种定义都忽略了性能和效率测试。

2． 软件测试的含义（重在理解）：

l 软件测试是一个过程，包含若干活动，运行软件进行测试只是活动之一，它也包含一些不运行软件的活动

l 进行软件测试可以运用人工方式也可以借助于工具

l 进行软件测试可以运行软件也可以不运行软件

l 软件测试的目的是发现软件错误和不足（缺陷），观察角度要全面

3． 软件测试的目的：

证明（表明软件能够工作）-> 检测（发现错误）-> 预防（管理质量）

测试目的之证明：

l 获取系统在可接受风险范围内可用的信息；

l 尝试在非正常情况和条件下的功能和特性；

l 保证一个工作产品是完整的并且可用或者可被集成。

测试目的之检测：

l 发现缺陷、错误和系统不足；

l 定义系统的能力和局限性；

l 提供组件、工作产品和系统的质量信息。

测试目的之预防：

l 澄清系统的规格和性能；

l 提供预防或减少可能制造错误的信息；

l 在过程中尽早检测错误；

l 确认问题和风险，并且确认解决这些问题和风险的途径。

4． 软件测试的主要工作：

l 检视代码、评审开发文档

l 进行测试设计、写作测试文档（测试计划、测试方案、测试用例等）

l 搭建测试环境、执行测试，发现软件缺陷，提交缺陷报告，并确认缺陷最终得到了修正

l 通过测试度量软件的质量

1.1.1 软件测试分类

按测试技术分类

按测试技术，软件测试可分为：黑盒测试、白盒测试、灰盒测试

黑盒测试：即功能测试，测试人员不必了解代码，只需要检查软件是否按照需求文档完成。

白盒测试：测试人员需要了解代码程序结构和处理过程，按照代码逻辑进行测试，比如接口测试。

灰盒测试：介于黑盒和白盒之间的测试，既要像黑盒测试关注功能的正确性，也要对代码结构有一定的了解。

按测试方式分类

按测试方式，软件测试分为：静态测试、动态测试

静态测试：即需求文档下发到手以后，完成测试用例以前对需求文档进行检查的过程。

动态测试：具体的测试过程，需要对软件进行运行。

按测试阶段分类

按测试阶段，软件测试分为：单元测试、集成测试、系统测试、验收测试

单元测试：对代码的最小单元——模块进行正确性的验证工作。验证代码在语法、逻辑上的正确性。

集成测试：即将各个单元组装到一起以后进行的测试，以此来验证接口之间的调用是否正确。

系统测试：将已经开发好的软件放到测试环境进行测试，验证功能、UI、逻辑等正确性。

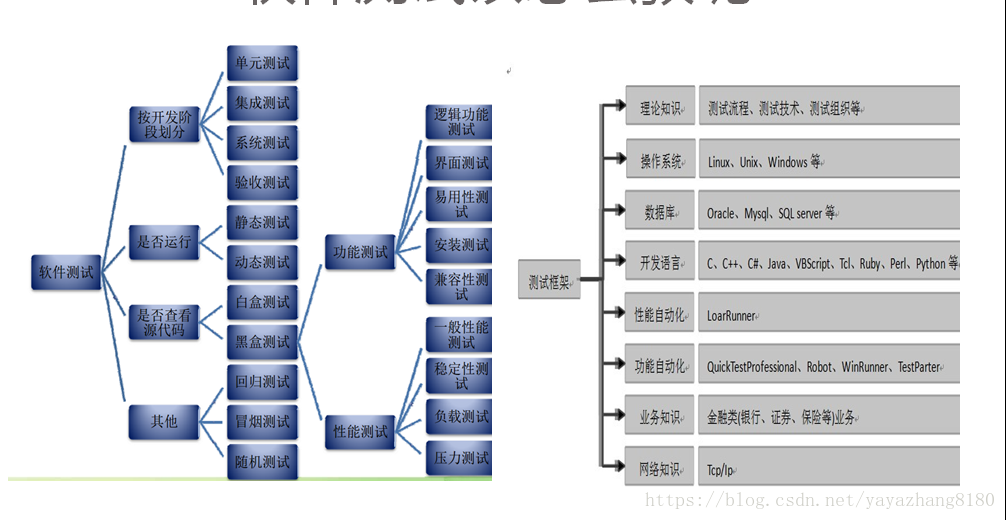
验收测试：测试的最后一轮，类似回归测试，即将软件的整体功能执行一遍，它不像系统测试那样细致，主要进行的是正向的测试流程。比如产品上线以后会在线上进行测试，而在线上测试的时候主要进行的是正向流程的测试。

按软件质量特性分类

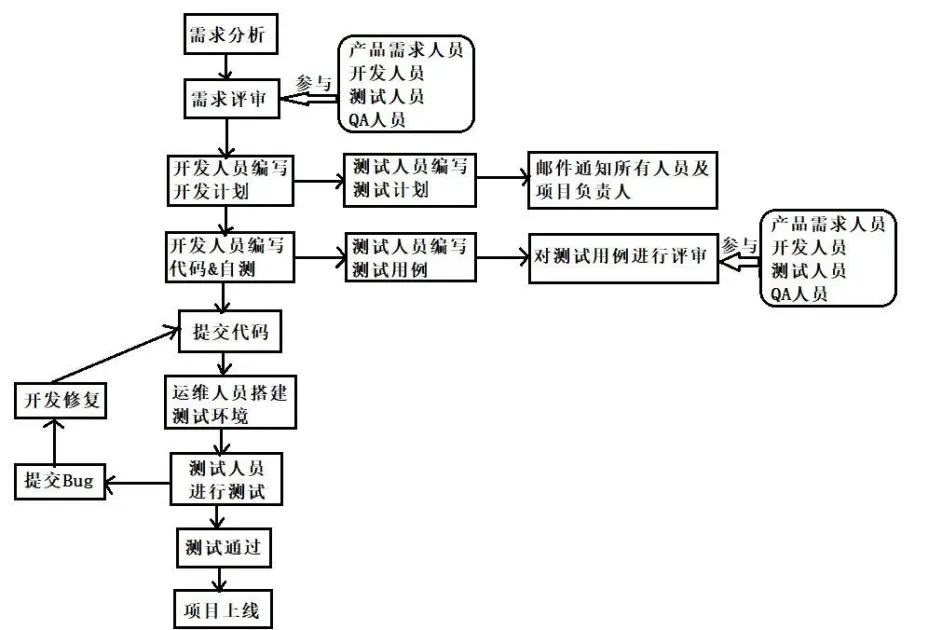
按软件质量特性，软件测试分为：功能测试、安全测试、性能测试、压力测试、兼容性测试

每个分类都可以单独作为一个岗位存在，比如安全测试、性能测试。

https://blog.csdn.net/yayazhang8180/article/details/80563378



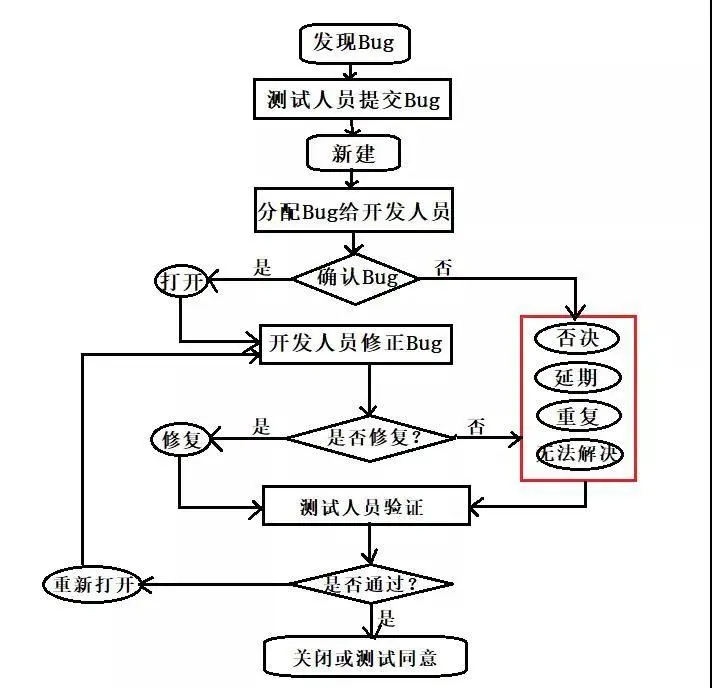
1.1.2 测试流程



缺陷管理流程

https://zhuanlan.zhihu.com/p/157567021

1.1.3 测试方法



常用的测试方法包括：等价类、边界值、正交排列、因果图、场景法。

等价类

1.适用场合: 有数据输入的地方,可以使用等价类划分,将大量的数据划分出若干范围,从每个范围中挑选代表数据进行测试,避免穷举,提高测试效率.

2.等价类方法划分：有效等价类，无效等价类 有效等价类:输入有意义,合理的数据集合； 无效等价类:输入无意义的,不合理的数据集合；

3.等价类划分法使用步骤:1.分析需求划分等价类(分为初步划分和细化);2.将等价类填写到<<等价类表>>中;3.从每个等价类中至少挑选一个代表数据,编写测试用例,执行测试

边界值

1.适用场合:常用于数据输入的地方,一般作为等价类划分的补充,和等价类划分一起使用

2.使用步骤:找到有效数据和无效数据之间的分界点,对分界点及其两边的点进行测试.

使用等价类+边界值测试的思路：A.先对有效数据进行测试- 1个测试用例尽可能的将多个控件的有效数据组合起来测(优化)B.再对无效数据进行测试- 无效数据需要单独测试(为了避免屏蔽现象)C.最后对多个无效数据组合测试(适当强化)

因果图

适用场合: 界面中考虑控件的组合和限制关系的情况(组合数量较少)

因果图中常用的9个图形符号:恒等,与,或,非;互斥(E),唯一(O),包含(I),要求(R),屏蔽(M);

使用步骤:

a.找出输入条件的所有组合和限制.

b.编写测试用例,每一列对应一条测试用例.

c.每组输入条件组合对应的输出结果,画因果图,填判定表(画因果图可以省略)

d.找出所有的输入条件(因),和所有的输出结果(果);

1 或 0 （默认表达方式， Default ）

  1代表真   0代表假

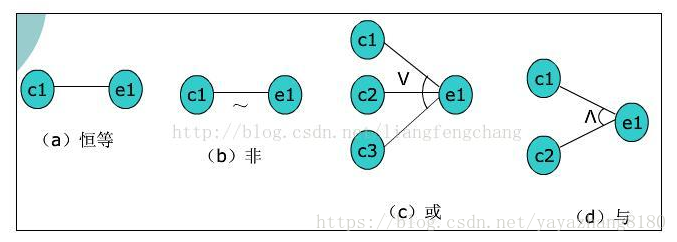
Y 或 N

  Y=Yes代表真  N=No代表假

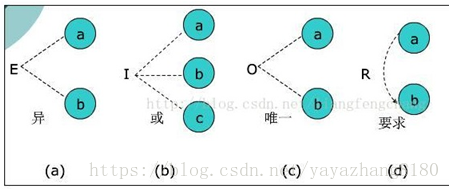
T 或 F

  T=True代表真  F=False代表假

4种原因与结果的关系



4种原因与原因的约束



E约束（排他性约束、Exclusive）：C1和C2中最多有一个可能为1，即C1和C2不能同时为1

I约束（包含性约束, Inclusive）：： C1、C2、C3中至少有一个必须是1，即： C1、C2、C3不能同时为0

O约束（唯一性约束, Only）：C1和C2必须有一个且仅有一个为1

R约束（必要性约束, Request）：： C1是1时，C2必须是1

M约束（强制约束，Masking）：：唯一的针对结果的约束；若结果E1是1，则结果E2强制为0

场景法

适用场合:当需要测试软件的业务流程(逻辑)时,适合用场景法,场景法是基于业务的方法,有测试人员模拟用户在使用软件的各种不同的情况;

场景法划分：基本流和备选流基本流：也叫有效流或正确流,模拟用户正确的操作的过程;备选流：也叫无效流或错误流,模拟用户错误的操作的情景;

场景法的使用步骤:

a.根据列出场景填写场景表;

b.为每个场景编写适当的测试用例(不一定是1:1的)

c.分析需求,整理业务流程(逻辑),列出场景;

正交排列

适用场合:对于参数配置类软件,以及兼容性测试时需要考虑各个控件之间的组合情况(组合较多),使用正交排列法选择较少的组合达到最佳的测试效果.

使用步骤:

a.选择合适的正交表(确定 m = 控件取值个数, k = 控件数)

b.编写测试用例

c.完成控件,控件取值对应因子和状态的映射.

d.分析需求,列出参与组合的控件以及每个控件的取值

判定表法Decision Table Method：

判定表是分析和表达多种输入条件下系统执行不同动作的工具，它可以把复杂的逻辑关系和多种条件组合的情况表达得既准确又明确。

   一般情况下，我们在画出因果图后写出判定表，两者绑定使用。但是无论是因果图法也好，判定表法也好，它们两者都是可以单独使用的。

   根据个人喜好，熟练了以后，可以考虑直接使用判定表法，省去画图步骤（Normally）。

因果图+判定表的经验结论

判定表法的优点：

1、充分考虑了输入条件间的组合，对组合情况覆盖充分；

2、最终每个用例覆盖多种输入情况，有利于提高测试效率；

3、设计过程中，对输入条件间的约束关系做了考虑，避免了无效用例，用例的有效性高；

4、能同时得出每个测试项的预期输出。

判定表法的缺点：

1、当被测试特性输入较多时，会造成判定表规格过于庞大；

2、输入之间的约束条件不能有效区分输入是否确实需要进行组合测试，会造成不需要组合测试的输入做了组合，从而产生用例冗余。

错误推测法

任何有意义的错误推测都值得单独写一条测试用例， 一般情况下，推测开发需求中没有明确指明的， 错误推测法很随意，就是个头脑风暴

1.1.4 测试用例设计

测试用例

所谓“测试用例”即测试执行流程的文档表述。测试用例包括：用例编号、功能模块、测试功能点、前置条件、操作场景及步骤、预期结果、测试结果、优先级。