

1、需求评审

充分准备评审 好的软件需求说明书，是进行有效需求评审的前提。

首先，需求人员在与用户确认需求的过程中，一定不要放过任何一个细节，仔细体会用户的每一个要求。

对于用户的要求，需求人员需要对其加以梳理：哪些是合理的需求，哪些是不合理的需求，还有一些可能是必要的但是用户没想到的需求。

软件需求说明书不应该只是用户意愿的表达，而应该是从软件层面上对用户需求的总结。

软件需求说明书对需求用例的描述一般分为基本流和扩展流，

基本流是大家很容易想到的主要业务流程，而实际设计开发及测试过程中，最耗费时间的是实现扩展流的过程。

因此不能只注重基本流，好的软件需求说明书，扩展流一定远远多于基本流，扩展流写得越完善，说明需求人员考虑得越周全。

而实质上，如果扩展流写得不完善，后期的设计、开发及测试人员往往在相应的细节处理上无所适从。

用户的需求是可以分层次的，一般而言分成以下层次：

- ①目标性需求，定义整个系统需要达到的目标；
- ②功能性需求，定义了整个系统必须完成的任务；
- ③操作性需求，定义了完成每个任务的具体的人机交互；

2、兼容性测试

IE Tester

IE、chrome、firefox

iOS8、9、10

android：4.——5.——6.

屏幕尺寸的适配性测试

android的多尺寸，iOS的多尺寸、pc的多分辨率。

3、selenium

使用ide录制

运行（Selenium Remote Control）

Selenium的核心Selenium Core基于JsUnit，完全由JavaScript编写

Selenium IDE是基于FIREFOX浏览器的一个插件，提供GUI界面来运行Selenium测试。

1 通过ID,name选择元素：如 `selenium.type("id=q","百度");`
`selenium.type("name=search","百度")`

2 link= 根据链接文字来操作:如 `selenium.click("link=个人资料");`

3 根据XPath来选择元素：XPath Checker

css选择器

遇到的坑：

- 1、上传控件不是input类型，网上找了一个模块调用后，可以上传。
- 2、屏幕需要下拉滚动。
- 3、去数据库读取数据mysql、redis。
- 4、目前存在的问题是图片验证码识别的问题，无解。
- 5、模块化写法。
- 6、读写配置文件。

4、loadrunner

测试组件

1.Virtual User Generator（**虚拟用户生成器**）用于捕获最终用户业务流程和创建自动性能**测试脚本**（也称为虚拟用户脚本）。

2.**Controller（控制器）**用于组织、驱动、管理和监控**负载测试**。

3.**Analysis（分析器）**有助于您查看、分析和比较性能结果。

LoadRunner使用虚拟用户（Virtual users）来模拟实际用户对业务系统施加压力。虚拟用户在一个中央控制器（controller station）的监视下工作。

在做一个测试方案时，要做的第一件事就是创建虚拟用户执行脚本。

LoadRunner提供了Virtual User Generator来录制或编辑虚拟用户脚本。

E. 点击工具栏中的"执行"按钮来执行我们刚才录制的脚本，确保执行无误。

A. 从菜单中选择运行Controller，创建一个新的Scenario，选择刚才录制的脚本（simpapp）

"Think Time"用于设置Vusers的反应和思考时间，以尽量做到和正常人一样来施压。"Ignore think time"表示忽略思考时间，

这是理想状态，一般不使用。"As recorded"表示按照录制时的实际操作时

间。"Multiply recorded think time by"表示Vusers的思考时间是实际录制时间的若干倍。

在"Miscellaneous"中设置一些杂项，如使用进程还是使用线程等

1. 平均事务响应时间

Average Transaction Response Time 优秀：<2s

良好：2-5s

及格：6-10s

不及格：>10s

2. 每秒点击率

Hits per Second

(每秒客户端发出的请求数)

当增大系统的压力（或增加并发用户数）时，吞吐率和TPS的变化曲线呈大体一致，则系统基本稳定。若压力增大时，吞吐率的曲线增加到一定程度后出现变化缓慢，甚至平坦，很可能是网络出现带宽瓶颈，

同理若点击率/TPS曲线出现变化缓慢或者平坦，很可能是服务器响应时间增加，观察服务器资源使用情况，确定是否是服务器问题

4. 每秒系统处理事务数

Transaction per second

6. CPU利用率

Processor / %Processor Time 好：70%

坏：85%

很差：90%+

7. 数据库操作消耗的CPU时间

Processor / %User Time 如果该值较大，可以考虑是否能通过友好算法等方法降低这个值。

如果该服务器是数据库服务器，Processor\%User Time 值大的原因很可能是数据库的排序或是函数操作消耗了过多的CPU时间，

此时可以考虑对数据库系统进行优化。

常用函数：

lr_start_transaction 为性能分析标记事务的开始

lr_end_transaction 标记 LoadRunner 事务的结束

lr_rendezvous 在 Vuser 脚本中设置集合点

lr_think_time 暂停 Vuser 脚本中命令之间的执行

8、python

是一种面向对象的解释型计算机程序设计语言

Python具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起

Python开发人员尽量避开不成熟或者不重要的优化。一些针对非重要部位的加快运行速度的补丁通常不会被合并到Python内。所以很多人认为Python很慢。

不过，根据二八定律，大多数程序对速度要求不高。在某些对运行速度要求很高的情况，Python设计师倾向于使用JIT技术，或者用使用C/C++语言改写这部分程序。可用的JIT技术是PyPy。

控制语句

if语句，当条件成立时运行语句块。经常与else, elif(相当于else if) 配合使用。

for语句，遍历列表、字符串、字典、集合等迭代器，依次处理迭代器中的每个元素。

while语句，当条件为真时，循环运行语句块。

try语句。与except,finally配合使用处理在程序运行中出现的异常情况。

class语句。用于定义类型。

def语句。用于定义函数和类型的方法。

pass语句。表示此行为空，不运行任何操作。

assert语句。用于程序调试阶段时测试运行条件是否满足。

raise语句。制造一个错误。

import语句。导入一个模块或包。

from import语句。从包导入模块或从模块导入某个对象。

import as语句。将导入的对象赋值给一个变量。

in语句。判断一个对象是否在一个字符串/列表/元组里。

主要的算术运算符与C/C++类似。+, -, *, /, //, **, ~, %分别表示加法或者取正、减法或者取负、乘法、除法、整除、乘方、取补、取模。

>>, <<表示右移和左移。&, |, ^表示二进制的AND, OR, XOR运算。>, <, ==, !=, <=, >=用于比较两个表达式的值，分别表示大于、小于、等于、不等于、小于等于、大于等于。

在这些运算符里面，~, |, ^, &, <<, >>必须应用于整数。

Python使用and, or, not表示逻辑运算。

is, is not用于比较两个变量是否是同一个对象。in, not in用于判断一个对象是否属于另外一个对象。

Python区分列表(list)和元组(tuple)两种类型。list的写法是[1,2,3]，而tuple的写法是(1,2,3)。可以改变list中的元素，而不能改变tuple

Python使用'(单引号)和"(双引号)来表示字符串。与Perl、Unix Shell语言或者Ruby、Groovy等语言不一样，两种符号作用相同。

表达式前加r指示Python不解释字符串中出现的\

Python支持列表切割(list slices)，可以取得完整列表的一部分。支持切割操作的类型有str, bytes, list, tuple等。

它的语法是...[left:right]或者...[left:right:stride]。假定nums变量的值是[1, 3, 5, 7, 8, 13, 20]，那么下面几个语句为真：

nums[2:5] == [5, 7, 8] 从下标为2的元素切割到下标为5的元素，但不包含下标为5的元素。

函数

Python的函数支持递归、默认参数值、可变参数，但不支持函数重载。为了增强代码的可读性，

可以在函数后书写“文档字符串”(Documentation Strings，或者简称docstrings)，用于解释函数的作用、参数的类型与意义、返回值类型与取值范围等。

对象的方法

对象的方法是指绑定到对象的函数。调用对象方法的语法是 `instance.method(arguments)`。

它等价于调用 `Class.method(instance, arguments)`。当定义对象方法时，必须显式地定义第一个参数，一般该参数名都使用 `self`，用于访问对象的内部数据。

这里的 `self` 相当于 C++、Java 里面的 `this` 变量，但是我们还可以使用任何其它合法的参数名，比如 `this` 和 `mine` 等，`self` 与 C++、Java 里面的 `this` 不完全一样，它可以被看作是一个习惯性的用法，我们传入任何其它的合法名称都行

类型

Python 采用动态类型系统。在编译的时候，Python 不会检查对象是否拥有被调用的方法或者属性，而是直至运行时，才做出检查。所以操作对象时可能会抛出异常。不过，虽然 Python 采用动态类型系统，它同时也是强类型的。Python 禁止没有明确定义的操作，比如数字加字符串。

`str`
`list`
`tuple`
`set`
`dict`
`int`
`float`
`complex`
`bool`

CGI(Common Gateway Interface)，通用网关接口，它是一段程序，运行在服务器上如：HTTP 服务器，提供同客户端 HTML 页面的接口。CGI 程序可以是 Python 脚本、Perl 脚本、Shell 脚本、C 或者 C++ 程序等。

丰富的库：Python 标准库确实很庞大。它可以帮助处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV 文件、密码系

统、

GUI（**图形用户界面**）、Tk和其他与系统有关的操作。这被称作Python的“功能齐全”理念

。除了标准库以外，还有许多其他高质量的库，如wxPython、Twisted和Python图像库等等。

用Python写简单爬虫

首先，要通过urllib2这个Module获得对应的**HTML**源码。

```
1
2
3
4  import urllib2 #调用urllib2
url='http://www.baidu.com/s?wd=cloga' #把等号右边的网址赋值给url
html=urllib2.urlopen(url).read() #html随意取名 等号后面的动作是打开源代码页面，并阅读
print html #打印
```

通过上面这三句就可以将URL的源码存在content变量中，其类型为字符型。

接下来是要从这堆HTML源码中提取我们需要的内容。用Chrome查看一下对应的内容的代码（也可以用Firefox的Firebug）。

可以看到url的信息存储在span标签中，要获取其中的信息可以用正则式。

使用requests方式

遇到的坑：

- 1、写爬虫，遇到图片验证码的，解决不了。
- 2、遇到过验证cookies的、刷新时间限制的。
- 3、写备份脚本、写自动化脚本前面写了。
- 4、备份脚本，遇到的坑是：各种在Linux一条命令就可以获取到的结果在python里面不好实现。
- 5、发送邮件的坑，从a服务器跳转到b服务器执行脚本，通过ssh方式。
- 6、格式化邮件输出、读写文件权限不足，写目录经常写错。
- 7、正则匹配在爬虫和备份脚本里面都用到了。关键在于常用。
- 8、发送邮件服务器配置问题，经常发不出去。

9、使用github做脚本的管理。

9、github

git init

初始化一个仓库，但是一般常用的都是克隆现有的仓库，这里还是用刚才创建的helloworld仓库

git clone <https://github.com/hk029/hello-world.git>

我们可以先输入命令看看当前仓库的状态

git status

+1 表示目前有1个新文件

~0 表示0个修改的文件

-0 表示0个删除的文件

! 表示未保存

我们把它暂存到暂存区

git add NewFile

这个命令会把这个文件放到暂存区（还是在本地）到时候提交的时候就会把暂存区的东西提交到网上。

这里的commit只是保存到了本地。如果你只需要一个本地仓库，那么现在也就够了。

git commit -m "my first commit"

因为你是直接从远程仓库拷贝的，所以你输入

git remote

会发现已经有个original了 这个时候你直接git push就行了

git push

但是，建议还是使用ssh的模式，安全点。下面说下ssh的模式怎么用，首先你要创建一个私钥，就是在自己电脑里的钥匙。

ssh-keygen -t rsa -C "your email addr"

这时候，就算在github上注册了你的私钥，然后在输入

ssh -T git@github.com

看看是不是能ssh连上github

git push -u origin master

master是你的分支，origin是你的远程仓库

第一次git push -u origin master后，就可以用直接用git push指令了

10、wireshark

1、网卡一定要选中混杂模式“use promiscuous mode on all interfaces”
进行到这里已经看到报文以绿色，蓝色，黑色显示出来。Wireshark通过颜色让各种流量的报文一目了然。

比如默认绿色是TCP报文，深蓝色是DNS，浅蓝是UDP，黑色标识出有问题的TCP报文——比如乱序报文。

过滤报文：

如果正在尝试分析问题，比如打电话的时候某一程序发送的报文，可以关闭所有其他使用网络的应用来减少流量。但还是可能有大批报文需要筛选，这时要用到Wireshark过滤器。

最基本的方式就是在窗口顶端过滤栏输入并点击Apply（或按下回车）。例如，输入“dns”就会只看到DNS报文。输入的时候，Wireshark会帮助自动完成过滤条件。

11、Linux的常用命令

ls	显示文件或目录
mkdir	创建目录
-p	创建目录，若无父目录，则创建p(parent)
touch	创建空文件
cat	查看文件内容
cp	拷贝
mv	移动或重命名
rm	删除文件
-r	递归删除，可删除子目录及文件
-f	强制删除
find	在文件系统中搜索某文件
grep	在文本文件中查找某个字符串
rmdir	删除空目录
tree	树形结构显示目录，需要安装tree包
pwd	显示当前目录

ln 创建链接文件
more、less 分页显示文本文件内容
head、tail 显示文件头、尾内容

系统管理命令

uname 显示系统信息
top 动态显示当前耗费资源最多进程信息
ps 显示瞬间进程状态 ps -aux
du 查看目录大小 du -h /home带有单位显示目录信息
df 查看磁盘大小 df -h 带有单位显示磁盘信息
ifconfig 查看网络情况
ping 测试网络连通
netstat 显示网络状态信息
man 命令不会用了，找男人 如：man ls
clear 清屏
alias 对命令重命名 如：alias showmeit="ps -aux"，另外解除使用unalias showmeit
kill 杀死进程，可以先用ps 或 top命令查看进程的id，然后再用kill命令杀死进程。

打包压缩相关命令

tar: 打包压缩
-c 归档文件
-x 压缩文件
-z gzip压缩文件
-j bzip2压缩文件
-v 显示压缩或解压缩过程 v(view)
-f 使用档名

例：

tar -cvf /home/abc.tar /home/abc 只打包，不压缩
tar -zcvf /home/abc.tar.gz /home/abc 打包，并用gzip压缩
tar -jcvf /home/abc.tar.bz2 /home/abc 打包，并用bzip2压缩
当然，如果想解压缩，就直接替换上面的命令 tar -cvf / tar -zcvf / tar -

jcvt 中的“c”换成“x”就可以了。

关机/重启机器

shutdown

-r	关机重启
-h	关机不重启
now	立刻关机

halt 关机

reboot 重启

Linux管道

将一个命令的标准输出作为另一个命令的标准输入。也就是把几个命令组合起来使用，后一个命令除以前一个命令的结果。

例：grep -r "close" /home/* | more 在home目录下所有文件中查找，包括close的文件，并分页输出。

vim使用

vim三种模式：命令模式、插入模式、编辑模式。使用ESC或i或：来切换模式。

命令模式下：

:q	退出
:q!	强制退出
:wq	保存并退出

用户及用户组管理

/etc/passwd 存储用户账号

/etc/shadow 存储用户账号的密码

useradd 用户名

adduser 用户名

groupadd 组名

passwd root 给root设置密码

source /etc/profile 系统环境变量

bash_profile	用户环境变量
.bashrc	用户环境变量
su user	切换用户，加载配置文件.bashrc
su - user	切换用户，加载配置文件/etc/profile，加载bash_profile

更改文件的用户及用户组

sudo chown [-R] owner[:group] {File|Directory}

例如：还以jdk-7u21-linux-i586.tar.gz为例。属于用户hadoop，组hadoop
要想切换此文件所属的用户及组。可以使用命令。

sudo chown root:root jdk-7u21-linux-i586.tar.gz

-rw-rw-r--一共十个字符，分成四段。

第一个字符“-”表示普通文件；这个位置还可能会出现“l”链接；“d”表示目录

第二三四个字符“rw-”表示当前所属用户的权限。所以用数值表示为

4+2=6

第五六七个字符“rw-”表示当前所属组的权限。所以用数值表示为

4+2=6

第八九十个字符“r--”表示其他用户权限。所以用数值表示为2

所以操作此文件的权限用数值表示为662

更改权限

sudo chmod [u所属用户 g所属组 o其他用户 a所有用户] [+增加权限 -
减少权限] [r w x] 目录名

例如：有一个文件filename，权限为“-rw-r----x”，将权限值改为“-rwxrw-r-
x”，用数值表示为765

sudo chmod u+x g+w o+r filename

上面的例子可以用数值表示

sudo chmod 765 filename

12、mysql

explain对sql进行优化

|较频繁的作为查询条件字段应该创建索引|

select * from emp where empno = 1

|唯一性太差的字段不适合单独创建索引，即使频繁作为查询条件

```
select * from emp where *** = '男'
```

更新非常频繁的字段不适合创建索引

```
select * from emp where logincount = 1
```

不会出现在WHERE子句中字段不该创建索引

mysql的基本命令：

显示数据库

命令：show databases （注意：最后有个s）

```
mysql> show databases;
```

例如：删除名为 xhkdb的数据库

```
mysql> drop database xhkdb;
```

例如：如果xhkdb数据库存在，尝试存取它：

```
mysql> use xhkdb;
```

屏幕提示：Database changed

use 语句可以通告MySQL把db_name数据库作为默认（当前）数据库使用，用于后续语句。该数据库保持为默认数据库，直到语段的结尾，或者直到发布一个不同的USE语句：

```
mysql> USE db1;
```

```
mysql> SELECT COUNT(*) FROM mytable; # selects from db1.mytable
```

```
mysql> USE db2;
```

```
mysql> SELECT COUNT(*) FROM mytable; # selects from db2.mytable
```

mysql表名大小写敏感

```
mysql> insert into MyClass values(1,'Tom',96.45),(2,'Joan',82.99), (2,'Wang',96.59);
```

2)、查询前几行数据

例如：查看表 MyClass 中前2行数据

```
mysql> select * from MyClass order by id limit 0,2;
```

语法：update 表名 set 字段=新值,... where 条件

```
mysql> update MyClass set name='Mary' where id=1;
```

命令：**alter table** 表名 **add**字段 类型 其他;

例如：在表MyClass中添加了一个字段passtest，类型为int(4)，默认值为0

```
mysql> alter table MyClass add passtest int(4) default '0'
```

1.导出整个数据库

导出文件默认是存在mysql\bin目录下

```
mysqldump -u 用户名 -p 数据库名 > 导出的文件名
```

```
mysqldump -u user_name -p123456 database_name > outfile_name.sql
```

13、oracle

1.监听

启动监听

```
lsnrctl start
```

停止监听

```
lsnrctl stop
```

查看监听状态

```
lsnrctl status
```

运行sqlplus命令，进入sqlplus环境，nolog参数表示不登录；

```
sqlplus /nolog
```

以管理员模式进入

```
conn /as sysdba
```

启动数据库

```
startup;
```

停止数据库

```
SHUTDOWN IMMEDIATE
```

远程连接数据库

```
sqlplus /nolog
```

```
conn sys/sys@IP:1521/orainstance as sysdba
```

```
SHUTDOWN <NORMAL|ABORT | IMMEDIATE | TRANSACTIONAL[LOCAL]>
```

删除用户

```
drop user "username" cascade;
```

注：cascade 参数是级联删除该用户所有对象，经常遇到如用户有对象而未加此参数则用户删不了的问题，所以习惯性的加此参数

授权

```
grant connect,resource,dba to "username" ;
```

修改用户口令

```
alter user "username" identified by "password";
```

显示当前用户

```
show user;
```

创建表空间

```
CREATE TABLESPACE data01 DATAFILE '/oracle/oradata/db/DATA01.dbf' SIZE 500M;
```

修改表空间大小

```
alter database datafile '/path/NADDate05.dbf' resize 100M;
```

增加表空间

```
ALTER TABLESPACE NEWCCS ADD DATAFILE '/u03/oradata/newccs/newccs04.dbf' SIZE 4896M;
```

查询数据库文件

```
select * from dba_data_files;
```

查询当前存在的表空间

```
select * from v$tablespace;
```

查看表结构

```
desc table;
```

修改连接数：要重启数据库

```
alter system set processes=1000 scope=spfile;
```

```
shutdown immediate;
```

```
startup;
```

查看用户当前连接数

2、显示当前连接用户

SQL> show user –不能用在sql窗口 只能用在command命令窗口。

3、查看系统拥有哪些用户

```
SQL> select * from all_users;
```

84. 如何快速清空一个大表？

```
SQL>truncate table table_name;
```

87. 如何测试SQL语句执行所用的时间？

```
SQL>set timing on ;
```

```
SQL>select * from tablename;
```

90. 怎么把select出来的结果导到一个文本文件中？

```
SQL>SPOOL C:\ABCD.TXT;
```

```
SQL>select * from table;
```

```
SQL >spool off;
```

92. 如何在sqlplus下改变字段大小？

```
alter table table_name modify (field_name varchar2(100));
```

3. 只复制表数据：

如果两个表结构一样：

```
insert into table_name_new select * from table_name_old
```

98. 从网上下载的ORACLE9I与市场上卖的标准版有什么区别？

从功能上说没有区别，只不过oracle公司有明文规定；从网站上下载的oracle产品不得用于商业用途，否则侵权。

101. 如何搜索出前N条记录？

```
SELECT * FROM empLOYEE WHERE ROWNUM < n ORDER BY empno;
```

```
Select * from a where rownum<N;
```

110. tablespace 是否不能大于4G？

没有限制。

有限制的是xp系统对单文件的限制。

106. 如何查询当前日期

```
select sysdate from dual;
```

14、测试基础理论

具体的黑盒**测试用例设计**方法包括等价类划分法、边界值分析法、错误推测法、**因果图法**、判定**表驱动法**、正交试验设计法、功能图法、**场景法**等。

等价类划分的办法是把**程序**的输入域划分成若干部分（子集），然后从每个部分中选取少数代表性数据作为**测试用例**。

每一类的代表性数据在测试中的作用等价于这一类中的其他值。该方法是一种重要的，常用的黑盒**测试用例设计**方法。

以登录为例解释：

划分等价类

1) 划分等价类： 等价类是指某个输入域的子集合。在该子集合中，各个输入数据对于揭露**程序**中的错误都是等效的，并合理地假定：测试某等价类的代表值就等于对这一类其它值的测试.因此，可以把全部输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件，就可以用少量代表性的测试数据.取得较好的测试结果.等价类划分可有两种不同的情况：有效等价类和无效等价类。

有效等价类：是指对于**程序**的规格说明来说是合理的，有意义的输入数据构成的集合.利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能。

无效等价类：与有效等价类的定义恰巧相反。

设计**测试用例**时，要同时考虑这两种等价类.因为，软件不仅要能接收合理的数据，也要能经受意外的考验.这样的测试才能确保软件具有更高的可靠性。

1、正常的用户名密码输入登录

2、错误的用户名密码组合登录失败

边界值分析法

边界值分析是通过选择等价类边界的**测试用例**。边界值分析法不仅重视输入条件边界，而且也必须考虑输出域边界。它是对等价类划分方法的补充。

（1）边界值分析方法的考虑：

长期的测试工作经验告诉我们，大量的错误是发生在输入或输出范围的边界上，而不是发生在输入输出范围的内部.因此针对各种边界情况设计**测试用例**，可以查出更多的错误。

使用边界值分析方法设计**测试用例**，首先应确定边界情况.通常输入和输出等价类的边界，就是应着重测试的边界情况.应当选取正好等于，刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值作为测试数据。

1、使用：中英文、数字、特殊符号、是否为空、字段长度为1或者超过限制等作为输入条件进行测试。

2、密码有相同的校验。

错误推测法

错误推测法是基于经验和直觉推测程序中所有可能存在的各种错误，从而有针对性的设计测试用例的方法。

错误推测方法的基本思想：列举出程序中所有可能有的错误和容易发生错误的特殊情况，根据他们选择测试用例。例如，在单元测试时曾列出的许多在模块中常见的错误。以前产品测试中曾经发现的错误等，这些就是经验的总结。还有，输入数据和输出数据为0的情况。输入表格为空格或输入表格只有一行。这些都是容易发生错误的情况。可选择这些情况下的例子作为测试用例。

1、错误推测的话，比如登录用户名使用手机号注册，我用1000000000000这种的手机号，绝对不行。

因果图法

前面介绍的等价类划分方法和边界值分析方法，都是着重考虑输入条件，但未考虑输入条件之间的联系，相互组合等。考虑输入条件之间的相互组合，可能会产生一些新的情况。但要检查输入条件的组合不是一件容易的事情，即使把所有输入条件划分成等价类，他们之间的组合情况也相当多。因此必须考虑采用一种适合于描述对于多种条件的组合，相应产生多个动作的形式来考虑设计测试用例。这就需要利用因果图（逻辑模型）。

因果图方法最终生成的就是判定表。它适合于检查程序输入条件的各种组合情况。

1、因果图的话，把输入条件排练组合，检查输错。登录输入框+密码输入框+登录按钮的排列组合。

判定表组成法

条件桩（Condition Stub）：列出了问题的所有条件。通常认为列出的条件的次序无关紧要。

动作桩（Action Stub）：列出了问题规定可能采取的操作。这些操作的排列顺序没有约束。

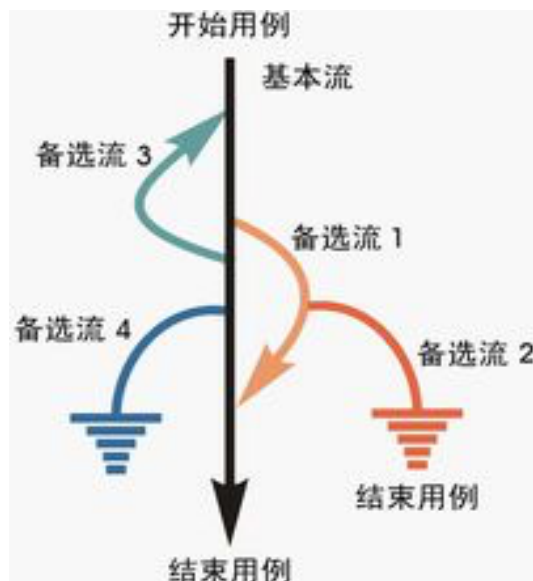
条件项（Condition Entry）：列出针对它左列条件的取值。在所有可能情况下的真假值。

动作项（Action Entry）：列出在条件项的各种取值情况下应该采取的动作。

规则：任何一个条件组合的特定取值及其相应要执行的操作。在判定表中贯穿条件项和动作项的一列就是一条规则。显然，判定表中列出多少组条件取值，也就有多少条规则，既条件项和动作项有多少列。

场景法

软件几乎都是用事件触发来控制流程的，事件触发的情景



基本流和备选流

便形成了场景，而同一事件不同的触发顺序和处理结果就形成事件流。这种在软件设计方面的思想也可以引入到软件测试中，可以比较生动地描绘出事件触发时的情景，有利于测试设计者设计测试用例，同时使测试用例更容易理解和执行。

正交试验设计

就是使用已经造好了的正交表格来安排试验并进行数据分析的一种方法，目的是用最少的测试用例达到最高的测试覆盖率。

测试执行

执行测试开发阶段建立的自动测试过程，并对所发现的缺陷进行跟踪管理。

测试执行一般由单元测试、组合测试、集成测试、系统联调及回归测试等步骤组成，测试人员应本着科学负责的态度，一步一个脚印地进行测试。

常用方法

功能测试就是对产品的各功能进行验证，根据功能测试用例，逐项测试，检查产品是否达到用户要求的功能。常用的测试方法如下

1. 页面链接检查：每一个链接是否都有对应的页面，并且页面之间切换正确。
2. 相关性检查：删除/增加一项会不会对其他项产生影响，如果产生影响，这些影响是否都正确。
3. 检查按钮的功能是否正确：如update, cancel, delete, save等功能是否正确。

4. 字符串长度检查：输入超出需求所说明的字符串长度的内容，看系统是否检查字符串长度，会不会出错。
5. 字符类型检查：在应该输入指定类型的内容的地方输入其他类型的内容（如在应该输入整型的地方输入其他字符类型），看系统是否检查字符类型，会否报错。
6. 标点符号检查：输入内容包括各种标点符号，特别是空格，各种引号，回车键。看系统处理是否正确。
7. 中文字符处理：在可以输入中文的系统输入中文，看会否出现乱码或出错。
8. 检查带出信息的完整性：在查看信息和update信息时，查看所填写的信息是不是全部带出，带出信息和添加的是否一致。
9. 信息重复：在一些需要命名，且名字应该唯一的信息输入重复的名字或ID，看系统有没有处理，会否报错，重名包括是否区分大小写，以及在输入内容的前后输入空格，系统是否作出正确处理。
10. 检查删除功能：在一些可以一次删除多个信息的地方，不选择任何信息，按“delete”，看系统如何处理，会否出错；然后选择一个和多个信息，进行删除，看是否正确处理。
11. 检查添加和修改是否一致：检查添加和修改信息的要求是否一致，例如添加要求必填的项，修改也应该必填；添加规定为整型的项，修改也必须为整型。
12. 检查修改重名：修改时把不能重名的项改为已存在的内容，看会否处理，报错。同时，也要注意，会不会报和自己重名的错。
13. 重复提交表单：一条已经成功提交的纪录，back后再提交，看看系统是否做了处理。
14. 检查多次使用back键的情况：在有back的地方，back，回到原来页面，再back，重复多次，看会否出错。
15. search检查：在有search功能的地方输入系统存在和不存在的内容，看search结果是否正确。如果可以输入多个search条件，可以同时添加合理和不合理的条件，看系统处理是否正确。
16. 输入信息位置：注意在光标停留的地方输入信息时，光标和所输入的信息会否跳到别的地方。
17. 上传下载文件检查：上传下载文件的功能是否实现，上传文件是否能打开。对上传文件的格式有何规定，系统是否有解释信息，并检查系统是否能够做到。
18. 必填项检查：应该填写的项没有填写时系统是否都做了处理，对必填项是否有提示信息，如在必填项前加*
19. 快捷键检查：是否支持常用快捷键，如Ctrl+C Ctrl+V Backspace等，对一些不允许输入信息的字段，如选人，选日期对快捷方式是否也做了限制。
20. 回车键检查：在输入结束后直接按回车键，看系统处理如何，会否报错。

一套完整的测试应该由五个阶段组成：

1.测试计划

首先，根据用户需求报告中关于功能要求和性能指标的规格说明书，定义相应的测试

需求报告，即制订黑盒测试的最高标准，以后所有的测试工作都将围绕着测试需求来进行，符合测试需求的应用程序即是合格的，反之即是不合格的；同时，还要适当选择测试内容，合理安排测试人员、测试时间及测试资源等。

（安排测试人员测试的模块，分析需要测试的范围，测试需要其他人员的配合、工具的提供、机器的准备）

2.测试设计

将测试计划阶段制订的测试需求分解、细化为若干个可执行的测试过程，并为每个测试过程选择适当的测试用例（测试用例选择的好坏将直接影响到测试结果的有效性）。

（把一次的需求，拆分后，估算测试周期后，开始准备些测试用例）

3.测试开发

建立可重复使用的自动测试过程。

（准备自动化脚本）

4.测试执行

执行测试开发阶段建立的自动测试过程，并对所发现的缺陷进行跟踪管理。测试执行一般由单元测试、组合测试、集成测试、系统联调及回归测试等步骤组成，测试人员应本着科学负责的态度，一步一个脚印地进行测试。

（单元测试给开发测，组合测试，一般是多个模块一块测，集成测试是一个维度可以测试，比如一个客户pc端，然后联调测试，客户pc+管家pc一块测试，最后发现bug，再复测。以及到

最后已经上线的模块，需要回归测试，验证相关的模块是正常好使的）

5.测试评估

结合量化的测试覆盖域及缺陷跟踪报告，对于应用软件的质量和开发团队的工作进度及工作效率进行综合评价。

显然，黑盒测试只有严格按照步骤进行，才可能对应用程序的质量进行把关。然而，如果没有一种优秀的测试工具的帮助，单纯凭借手工测试，不但将耗费大量的人力、物力和财力，而且有很多测试工作是难以实现甚至是无法实现的。