

测试用例设计实践总结

by:授客 QQ: 1033553122

步骤 1: 用例划分

1. 按系统模块划分
2. 按性质分类划分
3. 按关联紧密程度划分

1. 按系统模块划分

一般设计比较好的系统软件，都会把功能进行分类，并以模块的方式布局在用户界面上，如图：

【目标管理】，【课程管理】，【学员管理】，大模块下再分小模块，比如【课程管理】模块又分【课程列表】，【项目资源管理】。



用例划分：针对每个模块、子模块或子模块的模块设计用例

优点：容易展开，简单明了

缺点：

1. 业务逻辑容易被忽视。模块与模块之间往往是关联的。
2. 容易忽略非 UI 功能的测试，比如安装测试

举例：数据库审计系统，【规则模块】，【对象模块】

【规则模块】：存放规则，比如操作表名 xx 的规则

【对象模块】：存放对象，比如表名对象，操作方式对象

关联：规则引用对象

业务流程：客户操作->产生浏览数据->系统捕获数据->检测操作对象与规则引用的对象->如果

对象匹配则触发规则并执行规则中指定的动作

单独测两个模块可能都没问题，但是结合起来，在【对象模块】中把【规则模块】中规则正在引用的对象删除，那结果会咋样？难说吧

2. 按性质分类划分

用例划分：兼容性测试，压力测试，安装测试，容量测试，可靠性...

好处：对按模块划分的有效补充。

3. 按关联紧密程度划分

用例划分：也是按模块划分，区别是把关联比较紧密的模块归到某个模块

好处：有利于任务分配，减少人力资源的重复投入

举例：手机在线教育 APP 应用，打开应用有 我的课程，我的笔记，我的问题等模块，其中，我的笔记，笔记记录来自课程模块，观看课件学习时进行提交的。

如果按模块来，测试我的笔记的人需要去观看课件并提交笔记，而测试课件观看的人又要测提交笔记，很明显的，“提交笔记”重复投入了劳力。如果把提交笔记归到我的笔记模块，这样按模块分配用例，分配给同一个人去测，这就减少了交叉，减少重复的劳动

步骤 2：用例设计

1、设计思想

2、用例编写

1、设计思想

a) 测试点来源与定位

➤ 来源

测试点来源：一、显式需求 二、隐式需求。一个需求点可以对应多个测试点

➤ 定位测试点

测试点其实也就是测试目的。用例定义了“怎么测”，而测试点则定义了“为何测”，所以，设计前必须明白测试点是什么，且一个用例仅对应一个测试点。

理由：便于统计，测试用例对整个测试过程的质量控制和评估有很重要的意义：

一、测试需求覆盖率分析。如果一个用例包含几个测试点，那么不利于需求覆盖分析

二、用例成功率分析。一个用例中有多个测试点，肯定会造成用例数量减少，用例失败率大大增加，那么你做的用例成功率还有什么意义？

三、缺陷分析。如果用例失败了，就生成一个缺陷。如果一个用例中写了多个测试点，回归的时候如果有指定回归用例，那用例中那些与缺陷不相关的测试点也可能也被回归，增加工作量。

以下 3 点想法帮助你更好的定位测试点

1. 站在用户使用角度来考虑，看你定位的“测试点”是否有实际意义

2. 考虑你定位的“测试点”的完成能否标志着用户实际业务流程的一个阶段性结束？

3. 考虑你定位的“测试点”的完成，是否可以为用户或其他业务提供输入数据，以供完成下面的工作？

综合 2-3 点：划清界线，点到即止

例子：QQ 邮箱注册

QQ 注册

简体中文 | 繁体中文

邮箱帐号 @qq.com ❗ 请输入邮箱

昵称 ❗ 昵称不可以为空

密码 长度为6-16个字符
不能包含空格
不能是9位以下纯数字

确认密码

性别 ☒ 男 ☐ 女

生日 公历 年 月 日

所在地 中国 福建 厦门

验证码 [点击换一张](#)

☒ 我已阅读并同意相关服务条款

[立即注册](#)

如上图，单独把任意一个选框拿出来并为其设计一个用例，站在用户角度来看，都无实际意义。用户关注的是我填写完资料并点击注册，能生成一个可用帐号，为登录功能提供输入数据。所以设计用例时，这里的测试目的应该定位为帐号注册，而不是某个选框的特性测试。那输入框的特性，比如上述咋办？这个就是方法问题了，类似这样的，可以考虑用场景法来设计。

举例：音乐台音乐短片 扑克牌花式技巧演示"五个窍门"视频播放

MV 艺人 悦单 悦友 日志 帖子 以下显示是第1-1

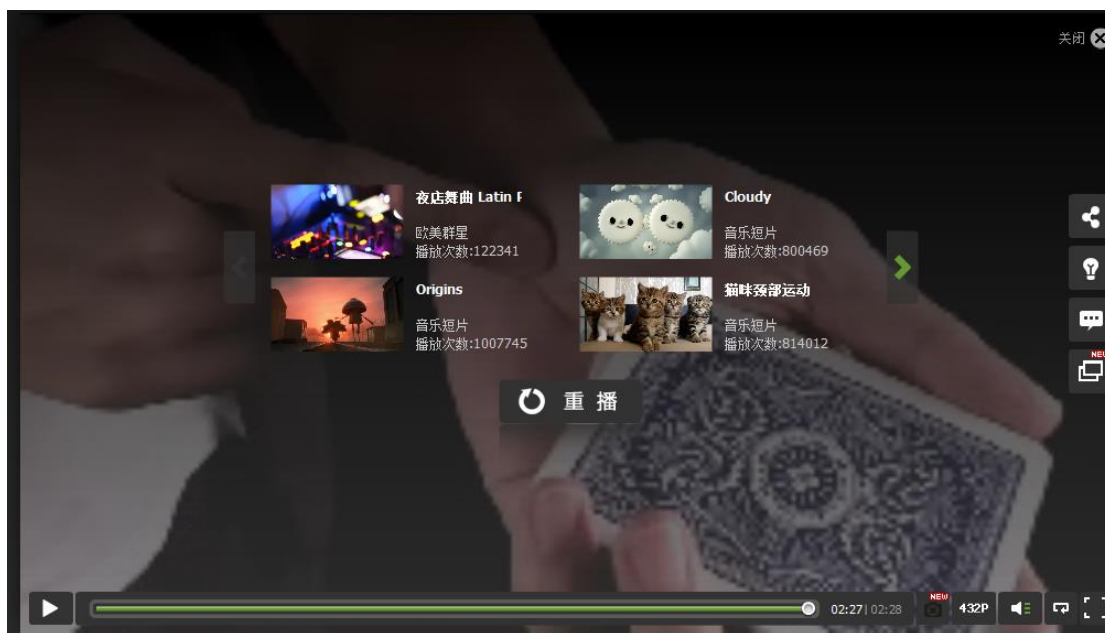
排序方式: 默认排序 关联最高 最多观看 最多收藏 最多推荐 最新上传

07:36 Storm Origins 音乐短片 播放101,5532次 上传时间: 2011-09-12

03:50 Sk8er Boi Avril Lavigne 播放17,8873次 上传时间: 2009-07-01

03:29 Jump Madonna 播放3,8616次 上传时间: 2009-08-18

02:28 扑克牌花式技巧演示 音乐短片 播放2,7112次 上传时间: 2012-12-01



操作流程：点击 扑克牌花式技巧演示“五个窍门”视频连接，自动打开视频播放界面，边缓冲，以边播放，播放完成，出现上述界面，给出“重播”按钮提示。

不考虑测试点粒度和分割(1 条用例)：

1. 点击视频连接--打开视频播放界面
2. 查看打开的播放器界面--一边进行视频缓冲，一边自动播放缓冲好的视频部分
3. 等待播放结束，查看播放器界面--出现重播按钮和推荐短片
4. 点击重播按钮--重新播放已缓冲完成的视频

考虑测试点粒度和分割(2 条用例)：

用例 1：在线视频播放功能

1. 点击视频连接--打开视频播放界面
2. 查看打开的播放器界面--视频以边缓冲，一边自动播放
3. 等待播放结束，查看播放器界面--出现重播按钮和推荐短片

用例 2：视频重播功能

1. 打开视频进行播放直到播放结束，查看播放器界面--出现重播按钮和推荐短片
2. 点击重播按钮--重新播放打开的视频

这里用例操作过程似乎有点冗余，但是从测试目的考虑是允许的：对用例 2 来说，步骤 1 可看成是为测试重播而必须经过的一条路线，不是测试重点，而对用例 1 而言，用例 2 中的步骤 1 则是测试重点。进一步，结束播放时的画面还有推荐视频。对这个设计用例，模仿用例 2 也显得很容易，思路清晰。

举例：在线教育系统，手机端离线视频学习

操作流程：网络连接下，下载课件视频，网络断开下，查看打开视频学习，提交离线笔记，新增待同步学识记录。网络连接，点击同步学识记录按钮进行服务端与手机端的同步。

从以上 3 点想法来考虑，可定位以下两个测试点：

1. 保存离线笔记
2. 同步离线笔记

可能有人会觉得，以上 2 个测试点也可以合并在一起。对的，但是你再结合一个用例对应一个测试点就好理解为何不合并了。

备注：用例是死的，人是活的，例中所举的例子都存在一定的冗余，执行的时候可以考虑适当的用例执行顺序来减少操作冗余。

举例：教师端学员信息修改

姓名: 身份证: 审核状态: 全部 付款: 全部

姓名	身份证	性别	目标	审核	单位	报名金额	手机	QQ	加入日期	操作
林超	350123198808290319	男	临时测试 -10.28	通过	福州	未支付	13960918480	454545	2014-03-20	<input type="button" value="修改"/> <input type="button" value="删除"/>

点击修改，弹出修改界面，继续点击单位，出现如图界面

帮助
用于学员信息修改

身份证: 350123198808290319

*姓名: 林超

邮箱: 124534182@qq.com

*手机号码: 13960918480

QQ: 454545

所在地: 福建省 福州

出生日期: 2013-10-10

*单位: 福州

毕业时间: 2013-08-15

*时间: 2014-04-16

查找单位 快速搜索

请选择:

天渔

测试 开发部

福建天渔公司	天	aa
成大放	很好	aaa
坑爹么	dfgfert	

找不到单位请联系客服

点击修改界面中的【单位】设置框，弹出的是一个单位搜索和选择对话框，如果不独立出来，对搜索框的准确性验证也加到修改功能的用例里，用例会显得庞大，而且测试点不单一，咋办？单独出来，目的就是对其搜索或展示数据(单位)准确性，找不到单位联系客服等功能验证，比如，是否错乱，是否少了等进行验证，是有意义的，因为这个测试点的输出数据为这个资料修改模块提供了输入数据，使其可往下执行。而修改中则仅关注单位选择作用。

b) 分离测试数据与测试逻辑（步骤）

方法：将用例中的一些输入、输出等作为参数，数据则单独列出，在执行时选择相应的数据执行。

理由：为什么要参数化？

a、没有将测试数据和测试逻辑分开的测试用例可能显得非常庞大，不利于测试员理解，导致难以控制和执行；

b、通过将用例参数化，可以简化用例，使测试用例逻辑清晰，数据不逻辑的关系明了，易于

理解;

c 、有利于提高测试用例的重用;

选择参数化内容

测试用例中需要通过使用不同数据来重复执行测试的部分;

包括:

a 、输入 (数据或操作等)

b 、输出 (结果数据或预期结果等)

举例

例一: 系统登陆

步骤(逻辑) :
1、输入<<<用户名>>>
2、输入<<<口令>>>
3、点击[OK]按钮
结果 :
<<<预期结果>>>

测试数据:

“用户名”	“口令”	“预期结果”	说明
“user10”	“pass10”	进入系统	正确的用户名和口令(6 位)
“user789”	“pass789”	进入系统	正确的用户名和口令(7-9 位)
“user000010”	“pass000010”	进入系统	正确的用户名和口令(10 位)
“”	“pass”	提示输入用户名 不能进入系统	用户名为空
“空格”	“pass”	提示无效用户名 不能进入系统	用户名为空格
“user”	“userpass”	提示用户名太短 不能进入系统	用户名小于 6 位
“user0000011”	“userpass”	提示用户名太长 不能进入系统	用户名大于 10 位

.....

当然，这里的案例也存在不妥的地方，也就说包含了多个测试点，另外，要是再加个验证码，那就更不妥了。。。

c) 依据业务逻辑进行设计

（关于业务逻辑的详细说明可参考另一文档）

举例说明什么是业务逻辑：

网上购物

业务流程：用户登录-选择商品-结算-下订单-付款-确认收货，这是一个流程

业务规则：当用户下单付款后必须通知卖家，有顾客光临

业务实体：订单信息，包含购买物品，买家，金额等

业务实体完整性：如：订单信息中，买家不可少，物品 id 不能为空

根据上述，可以得出优先级：业务流程→业务规则→实体完整性

当然这顺序不是绝对的。

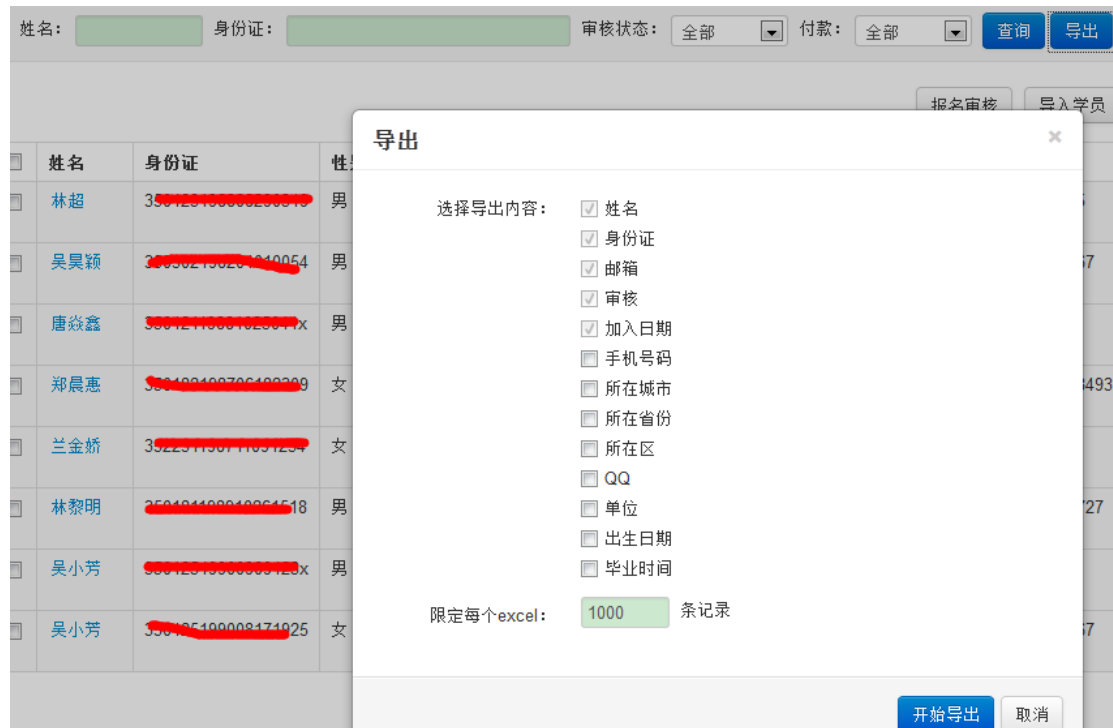
思想：

根据 80/20 原则，百分之 80 的用户只使用了产品 20% 的核心功能，测试要多站在用户角度进行模拟测试，有些测试站在测试的角度看是有意义的，站在用户的角度看却没多大意义，因为有些类似边界值的数据用户极少或根本不会用。（注意我这里的用词），所以要保证基本的核心功能可

用。这样写出来的用例优先级也比较好分，一目了然

方法：这里针对业务流程的测试推荐使用“场景法”。（当然，个人理解业务流程是从系统整体来把握的，局部角度来看，有些只算是“操作流程”，但是这个区别并不影响方法的使用）

举例：



分析：先考虑用户使用场景

场景 1：列表有数据，用户把数据按默认方式导出

点击导出->开始导出->查看导出文件

场景 2：用户突然不想导出

点击导出->点击取消

场景 3：列表有数据，用户把数据按自定义方式导出

点击导出->开始导出->查看导出文件

点击导出->设置导出列->开始导出->查看导出文件

点击导出->设置导出列->设置导出记录数->查看导出文件

场景 4：找不到导出的文件，重复导出

点击导出->导出列和记录数设置和上次一样->开始导出->查看导出文件

这些主要的考虑了，接下来考虑容错啥的

1. 列表没数据，进行导出

2. 导出列的边界值测试

好了，接下来就是细分和组合等了

细分：比如上面的导出列设置时，可以是全部选中，可以添加部分选中

组合：比如那个取消导出操作可以和其它场景的写在一起

更多详情，搜索 场景法

再举个例子说明按逻辑设计的好处:

教师端学员信息修改

姓名: 身份证: 审核状态: 全部 付款: 全部

姓名	身份证	性别	目标	审核	单位	报名金额	手机	QQ	加入日期	操作
林超	350123198808290319	男	临时测试 -10.28	通过	福州	未支付	13960918480	454545	2014-03-20	<input type="button" value="修改"/> <input type="button" value="删除"/>

点击修改, 弹出修改界面, 继续点击单位, 出现如图界面

帮助

用于学员信息修改

身份证: 350123198808290319

*姓名: 林超

邮箱: 124534182@qq.com

*手机号码: 13960918480

QQ: 454545

所在地: 福建省 福州

出生日期: 2013-10-10

*单位: 福州

毕业时间: 2013-08-15

*时间: 2014-04-16

查找单位

快速搜索

请选择:

天渔

测试 开发部

福建天渔公司	天	aa
成大放	很好	aaa
坑爹么	dfgfert	

找不到单位请联系客服

要点分析:

此处的修改是服务端管理员对学生端学员信息的修改, 如果按业务逻辑来, 这里的修改会同步学生端学员信息的修改, 这个点不容易遗漏的

说明: 按业务逻辑来设计用例, 容易让自己陷入矛盾的地方

背景: 某个在线教育产品, 功能模块包含了 我的笔记, 课程-视频课件播放, 其中, 我的笔记中, 笔记内容记录, 来源视频播放界面提交的笔记

举例: 按业务逻辑来, 可能会如下方式编写

- 1、打开视频播放界面, 输入笔记内容, 提交---(预期结果)
- 2、打开我的笔记--可见提交的笔记

这样看好像没问题, 但是细想下, 测试 我的笔记 模块时, 会漏掉步骤 2 的验证么? 不会吧, 所以这里的步骤 2 是多余的, 可去掉, 这里应该对步骤 1 进行重点测试, 不输入、输入字符过长, 输入字符含特殊字符, 输入字符含换行等

那步骤 2 怎么办? 在我的笔记模块新增用例, 把步骤 1 当做一条线, 如下

- 1、打开视频播放界面提交一条笔记 (预期结果可免了, 视频播放模块已验证过了)

2、打开我的笔记--预期结果（提交时间，内容显示，字符类型支持等）

这里也告诉我们，仅当某个点不会被单独作为一个用例检测点时，才需要进行一个“关联”，好比上面的学员信息修改，数据同步

这样看好像是没错的，但是很大的不足是啥呢？还是上面提到的，人力的重复投入：测试提交笔记时至少测输入字符串的长度，类型支持；测试笔记模块的查阅时也要测试笔记内容是否被截断，要测试特殊字符的显示是否正常等，也要进行提交笔记时执行的测试操作

解决方案：没错，还是按逻辑设计用例>>输入笔记->提交笔记->显示笔记，

1、打开视频播放界面，输入笔记内容，提交---(预期结果)

2、打开我的笔记--可见提交的笔记

这里可以根据本文中提到的，检测点的思想，进行细化，分成多条用例

比如用例 1. 记笔记(字符长度测试)；用例 2. 记笔记(字符类型验证)，当日对应的用例内容也跟着改，如下

1、打开视频播放界面，输入超长字符的笔记内容，提交---(预期结果)

2、打开我的笔记--笔记显示不截断，过长以…结尾

接着可以根据本文中提到的，归到同一个模块，比如笔记模块，分配给同一个人

d) 独立出公共用例

思想：把某些公用的模块或功能独立出来设计，减少冗余

举例：常见的智能手机，很多模块中选择文字，文字变底色，通常伴随弹出操作面板，类似全选，复制等，那可以考虑在某个模块中把这个功能单独出来设计用例，其它模块则不再重复写

e) 提高用例复用性

设计用例应该多考虑用例的复用性，可以从以下几个方面来考虑：

1)通用性。通用性是指可复用测试用例并不局限于具体的应用，不过分依赖于被测软件的需求、设计和环境，能够在某一类型、某一领域的相似软件的测试中广泛使用。(可以尝试去构建自己的用例库)

2)有效性。测试用例的目标是尽早发现软件问题

3)独立性。

1.用例之间不存在相互依赖关系

对于测试需求 R1 和 R2，测试用例集分别为 c1 和 c2，c1 和 c2 的交集为空，并且每个可复用测试用例能够独立运行。测试用例是否具有独立性，决定了测试用例可复用能力的强弱。

如果测试用例之间存在着相互关联，或测试用例的运行环境取决于其他测试用例的执行状态，那么，其中的测试用例不能复用时，与之相关的测试用例的可复用性也不复存在。

2.测试逻辑和测试数据分离

详情见下文

4)标准化

见“用例组成”

1、用例编写

1.1 用例组成

用例应遵循统一或规范的格式、结构，规范的命名规则，使用术语，用简明、易懂、无歧义的语言来描述，并且具有详细的文档。主要元素如下：

标识符 ID: 每个测试用例应该有一个唯一的标识符，它将成为所有和测试用例相关的文档、表格引用和参考的基本元素

测试项(用例名): 测试用例的标题，所给名称最好能清晰且简洁地表达测试用例的功能，见名知意，即测试目的、验证点。

建议格式：**【模块-子模块】**用例名

比如：**【登录】**密码大小写敏感测试

测试需求: 对要验证的测试需求的描述和测试要求，如登录验证需求：

a、用户名长度为 6 至 10 位（含 6 位和 10 位），

b、用户名由字符（a-z、A-Z）和数字（0-9）组成，。

测试环境: where-在哪里测？测试用例运行时所处的环境，包括系统的配置和设定等要求，也包括操作操作系统，浏览器，通讯协议等环境。即软硬件环境。一般来说，在整个的测试模块里面应该包含整个的测试环境的特殊要求，而单个测试用例的测试环境需要表征该测试用例所单独需要的特殊环境需求。

测试前提: 测试用例执行前必须满足的条件，如已登录、某个选项已经被勾选

输入数据: which-输入哪些数据？用来执行测试用例的数据。可能包括数据、文件，必要的时候，相关的数据库、文件也必须被罗列。

操作步骤: how-怎么做？操作步骤，如 1 打开软件，2 点击 xx 按钮

预期输出: 标识按照指定的环境和输入标准得到的期望输出结果(包含中间结果和最后结果)。

用例关联: 用来标识该测试用例与其它用例之间的依赖关系，例如，用例 A 需要基于 B 的测试结果正确的基础上才能进行，此时需要在 A 的测试用例中表明对 B 的依赖性，从而保证测试用例的严谨性。并非所有的测试用例之间都需要关联

优先级: 优先级越高的用例，应越早被执行

优先级通常是这样分的：

首先: 核心功能>次要功能

其次: 正向用例>逆向用例

也就是说，先确保核心功能正常，再次要功能测试；先确保常规路径运转正常-然后再逆向测试；

注意：

1. 优先级之分主要是从“关注对用户最有价值的东西”这个点出发来考虑的。

2. 上述的优先级的顺序，银行为例，银行账户登录，登录模块也包含逆向测试，但是涉及资金安全，登录模块的逆向测试的优先级也大于常规的正向用例的优先级，所以本质上优先级应该是：**核心功能（正向用例>逆向用例）> 次要功能（正向用例>逆向用例）**，而针对核心功能

所在模块: 按模块书写，通常情况下，建议 **【模块-子模块】**用例名称

版本号: 用于测试用例的版本管理，每个测试用例应按照定义的规则设定一个版本号。

测试阶段: 被测软件所处的测试阶段，包括单元测试、集成测试、确认测试、系统测试、验收测试等

测试类型: 有功能测试、性能测试、安全测试、用户界面测试、接口测试、安装测试等，可选择多项。

附件: 对测试用例附加的一些描述信息，例如文本、图像、模型、与测试用例有关的一些文档，方便测试人员进一步理解测试用例。

1.2 用例编写

1. 层次性
2. 明确性
3. 可测性
4. 可读性

1. 层次性

黑盒理论: 输入->处理->输出

设计应用: 测试步骤与预期结果对应

举例:

测试步骤 1--预期结果 1

测试步骤 2--预期结果 2

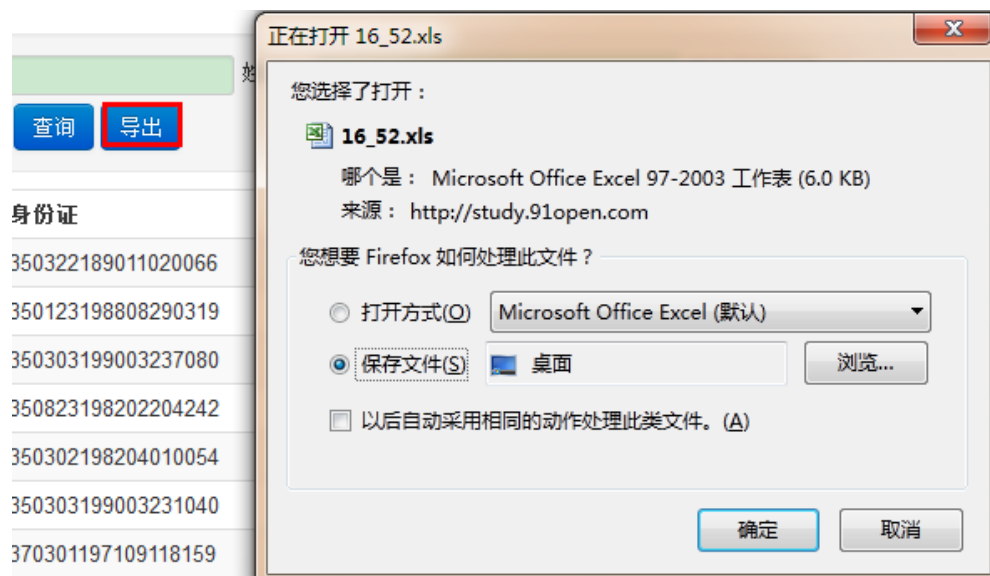
2. 完整性

黑盒理论: 输入->处理->输出

设计应用: 对输出物进行完整的检验

举例:

数据导出, 如图



点击导出--弹出导出确认框

确认导出--可在导出目录下看到导出文件

-----以下对步骤往往被忽略-----

打开导出文件--导出记录数正确, 内容完整, 准确

3. 可测性

黑盒理论: 预期结果 vs 实际结果 → 验证是否缺陷

设计应用: 预期结果必须可测

举例:

数据查询

身份证:	<input type="text"/>	姓名:	<input type="text"/>	报名目标状态:	<input type="text" value="全部"/>	注册时间:	<input type="text" value="2014-04-09"/>	至	<input type="text"/>
<input type="text" value="2014-04-20"/>		<input type="button" value="查询"/>		<input type="button" value="导出"/>					

姓名	身份证	手机	单位	注册时间	目标个数	操作
正式五零	350322189011020050	13635283752		2014-04-10	2	修改 查看 删除
正式五一	350322189011020051	13635283752		2014-04-10	3	修改 查看 删除

当前第 3 页/共 3 页，共有 22 条记录

[首页](#) [上一页](#) [1](#) [2](#) [3](#) [下一页](#) 到第 页 [跳转](#)

选择目标状态全部，输入注册时间，点击查询--列出注册时间范围内的的所有学员记录，数据正确，完整

分析：

情形一：列表的数据不是你自己造的，且测试不接触后台数据库，即数据源不知

这种情况下，预期结果的“列出所有的”，“数据正确”，“完整”，从何验证，这样的预期结果没实际意义

情形二：列表的数据是自己造的或者可通过后台查询，即数据源可知

这种情况下，预期结果的“列出所有的”，“数据正确”，“完整”是可验证的，有实际意义。

所以这里要根据实际情况来写预期结果，以情形一为例：

选择目标状态全部，输入注册时间，点击查询--列出学员记录的注册时间在给定注册时间查询范围内

4. 可读性

1. 数据和逻辑独立性，详见上面

2. 语言描述：尽量精炼，用词恰当等

3. 规范（我个人不是很赞同）

对用例中用到的元素，输入数据和非输入数据如按钮，控件等，添加标识规范，如输入数据用{ }，类似按钮控件，链接等非输入数据用【】

例子：

在密码框中输入 { 密码 }，点击【登录】按钮

关于这点我不是很赞成的，有待讨论，因为需求什么都在变，可能这个版本写“登录”，下个版本写“确认”，但是同一个意思，登录系统，所以我个人比较建议用自然语言描述，比如输入密码和用户名，登录系统，这样大大提高用例可复用性。再说了，稍微有点电脑基础的人，一看界面也应该大致知道类似删除，登录，修改之类的元素吧。。。