精简测试用例编写

by:授客 QQ：1033553122

大家都知道，测试用例的一个核心作用就覆盖测试需求，尽可能的减少漏测，同时提高测试效率。再细想想，这种核心作用的本质也就是一种“提醒”作用。

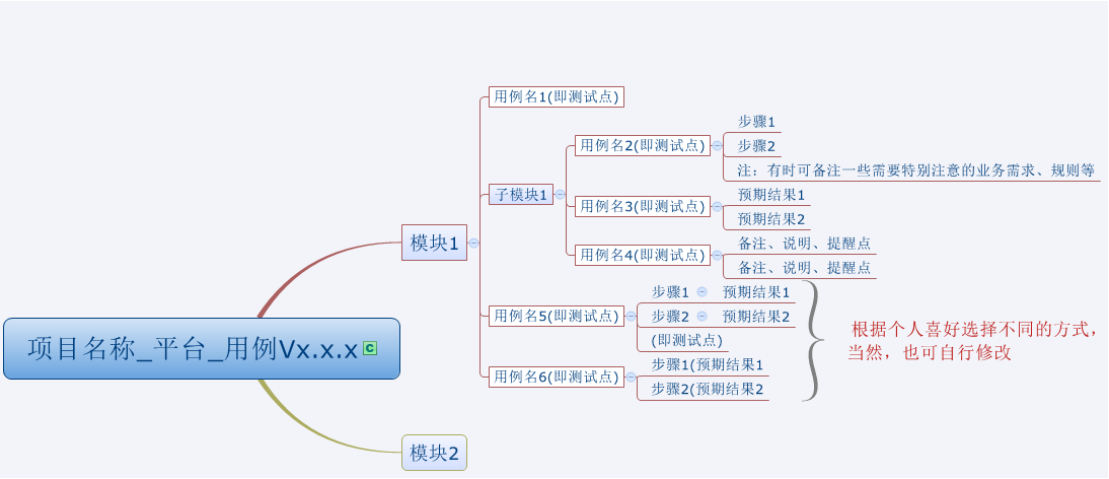
你可能会说“对呀，本来就是这样的呀，没啥问题呀”。我也觉得这个本身没错，那关键的问题是什么呢？ 问题在于时间和可执行性。

话说，写用例、设计用例是需要时间的，而在追求速度的时代，似乎连这点时间都给不起。这不是我吹的，特别是互联网，大部分公司都采用敏捷开发模式，讲究快速，所以，时间往往是有限的，再加上很多公司对敏捷仅停留在概念阶段，以为“敏捷=快速”，把时间花在用例设计上简直就是一种浪费….这样一来，用来设计用例的时间有时候，真是少之又少。如果按传统方式，把用例写得很详细，各种前提条件，步骤，说明，预期结果，编写人，日期啥的，这个时间成本是很高的。

再说可执行性。经常看到一些人写用例写得很认真，很详细、具体，把每一步的操作都写了，还有一些人一个用例下来，十几个步骤，包含了 n 个验证点。这种用例的可执行性咋样呢？按我的经历来看，这种类型用例，在测试的时候，用例编写人自己都懒得看用例，完全是按自己的当前时间的想法来测试，而非用例编写人呢？同样的，他也不会完全按照上面的步骤来执行，人嘛，都喜欢按自己的方式来做事，谁会看一下用例步骤，然后操作一下呢？所以，最后的结果是，用例文档就是摆设，放在那里，没人看。

怎么破？

我的观点是：测试用例必须写，而且还要“精炼”：能不写的尽量不写，能“简写”的尽量简写----“精简”的思维->简而言之，当前面的部分，已经起到了足够的提醒作用时，后面部分就可以不写。这样做的好处是，不仅可以节省时间，而且还给执行用例的人留下一定的思考空间，有利于 TA 的成长。



思维导图编写用例为例：

1． 一看用例名，就知道步骤及预期结果的，仅写用例名

这里的用例名，也就是我们的测试点、需求验证点。

这里对编写测试用例的人有个要求：语言组织能力+思维能力，尽量做到划分合理，且见名知意。

2． 仅看用例名，不能预知操作步骤的，还须把操作步骤写出来

3． 仅看用例名，不能预知预期结果的，还须把预期结果写出来

4． 预期结果、操作步骤有时候都可简写：直接以备注、说明、提醒点替代

对比上面的，这样写可能会给人有点“乱”的感觉，但是换个思路想，

这里实际是把预期结果、操作步骤当作是子验证点(即子用例)，采用第一条规则，这里的子用例仅写了用例名,即提醒点，验证点。也就是说，这条用例是由一组子用例构成的。

注：用例粒度可粗可细，结合时间成本考虑，做到合理的划分即可。

5． 针对一些步骤比较复杂的用例，步骤和预期结果都要写出来。

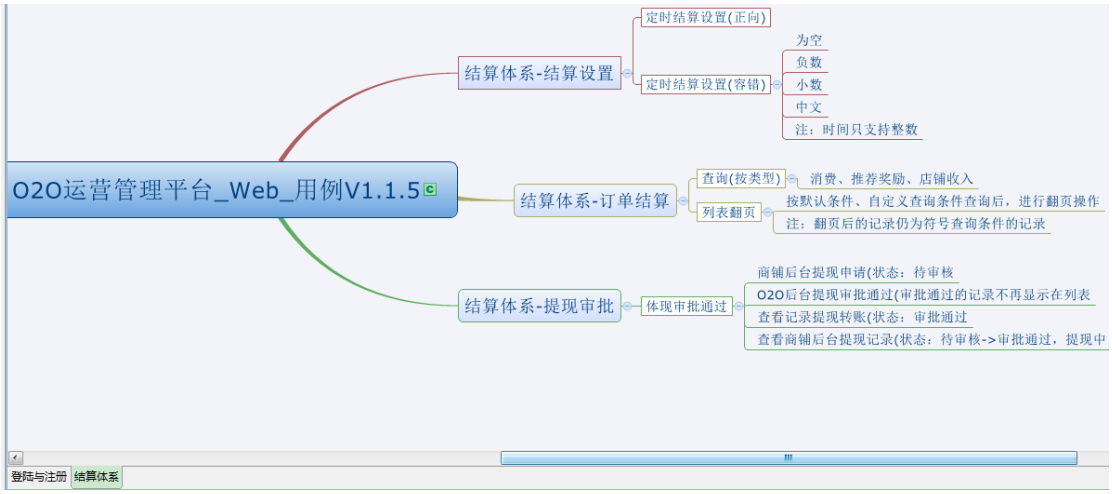
复杂情况下，一个用例包含多个操作步骤，这些操作步骤中，每个步骤可能都对应一个子测试点(预期结果)。如上，可以新增一个结点，填写预期结果，也可以和操作步骤写在同一个结点，以括号等不同方式进行区分，具体根据个人喜好或者大家达成共识的统一风格。

6． 技巧

根据具体情况，可以适当做些备注(可以是一些业务逻辑、规则、需求、预期结果等)，让人看得更明白

为了避免模块层级过多，可以不进行模块划分就不划分，当然也可以采用其它技巧，比如模块名称写成“大模块-子模块”的形式

7． 例子



有的人可能会想，我们可以写“通用用例”呀，对，这个似乎是可行的，但是我想说的是：

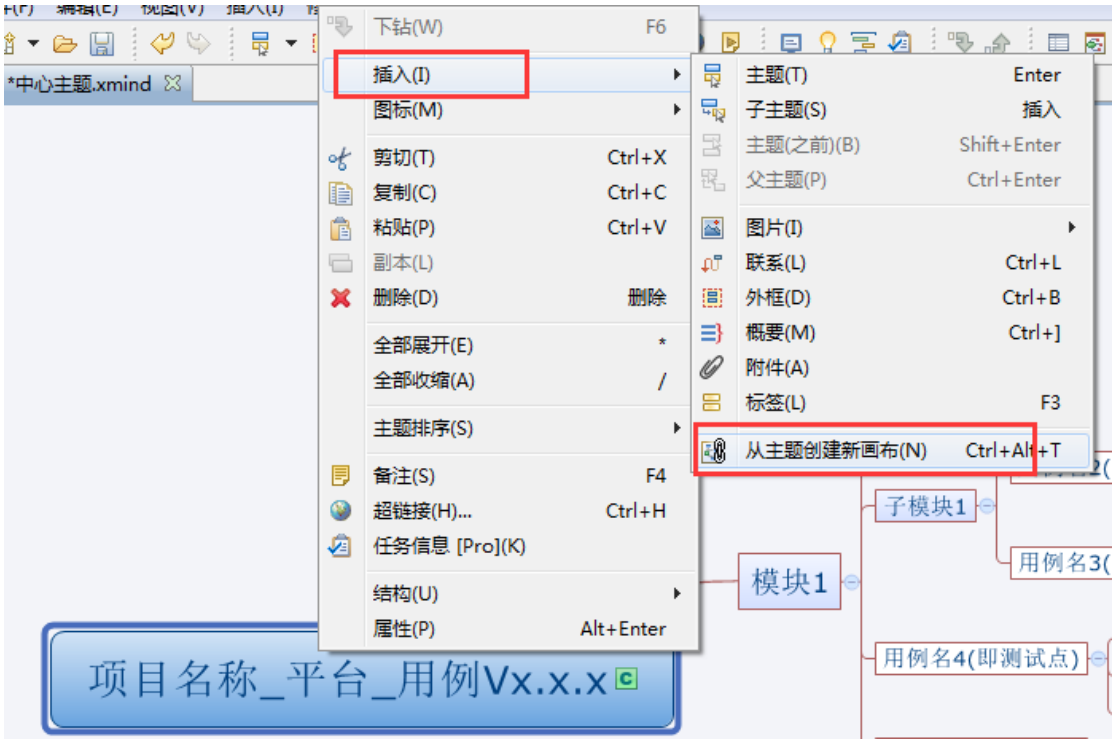
1． 用例设计是根据需求来写的，这里的“通用”部分其实并不多，仅小部分，而且单独把某个“点”拿出来考虑，有点类似“断章取义”，意义不大。

2. 所谓的通用用例，本质上看，也就是设计方法的体现，与其去“记忆”这些会变化的“通用”，还不如多想想你的用例设计里面，哪里体现了、到了这些设计方法。

写在最后

1、多去研究下产品，ui 交互等，对产品有感觉，自然的不用需求说明，你也知道点击哪个功能会跳出啥结果。

2、假如用 Xmind 编写用例，最好能合理的划分模块，然后每个画布放一个模块，具体操作如下，右键中心主题 -> 插入 -> 从主题创建新画布



如下，点击不同模块，展示不同模块的用例，这样做的一个好处是啥呢？假如一个画布存放所用用例，可能因为内存的原因，操作时会出现“卡顿”的现象，有时候还打不开文件，保存时也很耗时。

3、用例不仅编写者自己看，别人也要看的，所以不管咋写，一定要让人看得懂。