

如何提高 case 的广度与深度

首先，给一款产品设计测试用例，从总体上把握，测试用例应该主要覆盖以下三个方面：UI, function, non-function, 其实小到某一个 Feature, 无非也是从这三方面来设计，只要保证了这三方面的覆盖面，那么设计出来的测试用例应该是有广度也有深度。那么如何保证 UI, Function, Non-function 三个维度的深度与广度呢？其实这也就涉及到了一个很常见的问题，如何设计出高质量的测试用例，我们在《软件测试的艺术》一书中读到：一个好的测试用例是发现了软件中尚未发现的错误，不难看出，我们要拿这个测试用例是否发现了软件中的错误来衡量其质量。那么来详细看下如何从 UI, Function, Non-function 三个维度设计高质量的测试用例。

先说 UI 方面，UI 是手机呈现给用户的直观感觉，设计好了 UI 方面的测试用例，就保证了带给用户的第一感觉，UI 的测试既简单又重要。设计一款产品或者一个 Feature 的 UI 用例时，首先是考虑的是界面的完整性，严格按照 Spec, 把整个界面用一条 case 详细直观的呈现在 Tester 面前，这条 case 的描述要包含整体的布局，界面的颜色，包含的内容，按钮的位置，按钮上面显示的文字等等，比如下面的描述：

Status bar with time, signal, battery and other info is shown on top of the screen.
The screen title is with a question as "WOULD YOU LIKE HELP SETTING UP BLUETOOTH?".
A series options to answer the question:

-Yes

-Remind me later

-Don't show again

Only RSK is labeled as "BACK", LSK is empty

当然，在此要求设计者对手机的界面布局比较熟悉，能够知道手机的基本布局，以及每个布局的专业名称，一般来说，手机的 UI 界面分为三个部分，Status pain, Main pain, Control pain, Status pain 包含一些像 signal, battery, indicator 之类的信息，Main pain 就包含了界面的主要内容，Control pain 就包含了 softkeys。此外，像手机上呈现出的一些 note, note 的叫法依据其类型也不一，比如说 confirmation note, 此类 note 一般包含 Yes, No 的按钮，例如删除一个 contact 后会像提示 Deleted. 还有一类叫做 query note, 此类 note 一般会有选项指导用户选择，比如用户在删除一个 contact 前会提示 Delete? Xx Yes, No 因此设计在描述这些细节的时候切不可粗心大意，也不可误用词语，描述的准确性越大，Tester 测试的也就越准确，测试的质量也就越高。因此，专业的描述 UI, 设计者对手机应该有一定的熟悉程度。

当把整体界面介绍之后，就要描述一些细节甚至是 feature interaction 的 case。一般来说，界面的 control pain 都是 softkeys 的位置，那么点击这些 softkey 一般都会产生些子菜单，或者子菜单里还有下一级子菜单，因此这这也是一个检查点。另外，比如手机在连接上一些外设，像数据线，耳机，充电器，蓝牙耳机之类的，它们的 indicator 能否正确显示，而且表现其状态。此外如果手机在做了一些设置的更改，比如说设置了 alarm, 开启了蓝牙功能等等，这些能否在 UI 上正确表现。当然了有些细节的 UI 并不一定单纯的生成一条 case, 比如说手机在搜索蓝牙设备的时候，蓝牙标志会一直处于闪烁状态，像这样的细节也可以融合在功能 case 里，以防止漏掉功能点。此外能如果手机支持 portrait, landscape 两种模式，那么两种模式的 UI 同等重要都需要检查，也可以设计一些旋转屏幕的 case, 依据经验，手机在 portrait 和 landscape 切换时最容易产生一些 mess screen.

接下来就来看看 Function 这块，Function 这块的 case 可以说是整个产品或者整个 features 的重中之重，它所占的比例应该在整体的 80%左右，也是最能发现 error 的一块，

而且也是最能体现 case 深度的一块。

Function 这块的 case 基本上可以分为 basic function 和 Feature interaction 两大块。Basic function 不言而喻就是其基本功能,基本上就是其菜单上能看的见得,以 Music player 为例,比如说 play a song, update library, create playlist, change some settings 等等,这些在菜单里或者其显而易见的功能称之为基本功能,当然在写关于这些基本功能的 case 时,为了提高 case 的执行力,不一定是一个功能一个点,有时候可以几个测试点合并到一条 case 里,如果说你改变了一个设置,接着可以叫一个步骤用一个操作来验证写此设这是否生效。除此之外就是一些 feature interaction 的 case 了,如果说 Function 的 case 是整体的重中之重的话,那么 Feature interaction 的 case 就是 Function 中的重中之重了。在下手下 Feature interaction 方面的 case 之前,要做好充分的 outline,想好所设计的 features 能跟其它的哪些 features 会发生交互,充分模拟用户的使用。再以 Music player 为例,Music player 甚至可以和手机上的任何一个 features 产生交互,那么此时就要考虑分类,比如可以分成常规操作和非常规操作,常规操作我们设计如下测试点:

- 1) Create or receive a call during music playing
- 2) Play music during active call
- 3) Send or receive sms, mms during music playing
- 4) Make some action about camera when playing music, such as take a picture or record a video
- 5) Slide show pictures with background music when music playing
- 6) Browse when music playing
- 7) Play a game with background audio when music playing
- 8) Receive BT transfer file during music playing

等等这些操作都是用户非常习惯的操作,往往也是最容易出问题的地方,在描述这些 case 的 expected result 的时候,要正确强调测试点在哪,比如说 3) Send or receive sms, mms during music playing, 这一条的重点就在于 playing music 是否被 interrupt 当 message 到达的时候,当 message alert tone 结束后, music 是否继续。

而对于非常规操作,往往也是用户一般不会去做,但是这种操作的确存在的:

- 1) Remove MMC when playing music file that is saved in MMC
- 2) Connect USB cable to make file transfer when playing music
- 3) Play FM radio when music playing and headset connected
- 4) Play a video during music playing
- 5) Record a sound clip when music playing
- 6) Switch off device when music playing then switch on it
- 7) Play the DRM music file whose license has been expired

等等的这些操作,这些非常规的 Feature interaction case,往往最容易导致手机产生死机,重启以及 crash 操作,因此在强调这些 case 的 expected result 时,可以额外强调通过这些操作手机可以正常作业,没有想死机,重启,crash 之类的现象发生。恰当的描述测试重点,期待结果,不要使描述偏离主题,从而提高了 case 的深度。

别的 feature 以此类推,先设计出详细的 basic function 以及 feature interaction outline,尽可能多的想到可能与之交互的 features,以实现不漏掉任何一个功能点。

再来说说 Non Function 方面,关于 Non Function 方面的 case 也主要分为两类:Performance 和 Reliability, 一个性能测试,一个压力测试,要想写好 Non Function 方面的 case,那就先了解下性能测试和压力测试。Performance 主要测 Memory , Response time, Data

throughput 和 Power management 等, 从名字就能猜出来, 分别测低内存时的表现, 响应时间, 数据传输速度和程序运行时的耗电量。Reliability 主要测 Endurance, Recovery 和 SRIT。分别测试程序多次运行后是否有内存泄露, 程序在极端情况下是否能恢复正常(比如 network lost, power lost 等)。知道了 Performance 测试和 Reliability 都包含哪些方面的测试, 就不难就 create case 了。再以 Music Player 为例来说如何写 Performance 方面的 case:

- 1) Play music file when device memory full
- 2) Update music library when there are 500 songs in device
- 3) Transfer music files to device when device memory full
- 4) Check power management when music playing with low memory

在写这些 performance 方面的 case 的 expected result 时, 一定要把握好重心, 比如说第一条, 在 memory full 的情况下播放歌曲, 理论上手机应该是可以完成的而且没有异常情况出现, 比如说不能播放, 有不友好的提示, 或者出现死机, crash 等现象, 这些都可以强调出来。

再看 Reliability 方面, 主要考虑手机的一些极端情况, 比如说低电量, 没有网络连接等:

- 1) Play music when device battery low until device switch off automatically
- 2) Play a song for more than 2 hours

Reliability 和 Performance 同样, 要理解这两类测试类型, 都包含哪些方面的测试, 才能写出全面的, 有深度的 case。

总之, 设计测试用例, 是一个能全面体现设计者测试功底的工作, 如何设计出既有广度又有深度的测试用例, 还是要不断地丰富自我的测试基本知识, 比如说一些测试专用术语, 手机 UI 界面常识, 手机基本常识, 测试类型等等, 这些都会自然而然的帮助设计者提高测试用例的质量。