

# 如何高效的保证移动测试联网应用质量

{ By Monkey陈晔晔

# 常见场景

- ↳ 变的比天还快的需求
- ↳ 项目生命周期越来越短
- ↳ 少的可怜的测试人员
- ↳ 很难高效的自动化
- ↳ 碎片化太大
- ↳ 屎一样的用户体验
- ↳ 发酵的测试用例
- ↳ 只有测试在测试
- ↳ 沟通困难
- ↳ 很少人知道测试是什么

高效是既快又好，而不是两者选一

- ↳ 测试篇章
- ↳ 流程篇章
- ↳ 自动化篇章

- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习

slide show	同moment show	<ol style="list-style-type: none"> <li>1. 自己的moment, 可修改语音, 描述, poi, tag等, 播放slide show查看效果</li> <li>2. 多次修改1, 保存, , 播放slide show查看效果</li> <li>3. 删除照片, 删除语音, 描述, poi, tag等, , 播放slide show查看效果</li> <li>4. 删除最后一张照片, , 播放slide show查看效果</li> <li>5. 左右滑动, , 播放slide show查看效果</li> <li>6. 点击可查看图片, 播放slide show查看效果</li> <li>7. 多点触控放大, 播放slide show查看效果</li> <li>8. tag和poi有显示限制, 播放slide show查看效果</li> <li>9. @好友的显示, 播放slide show查看效果</li> <li>10. 照片的数字和总数, 播放slide show查看效果</li> <li>11. 头像, 故事名称, 故事描述等, 播放slide show查看效果</li> <li>12. like故事, 评论故事, 数字的显示, 播放slide show查看效果</li> <li>13. like和评论的整合, 播放slide show查看效果</li> <li>14. 各种分享, 播放slide show查看效果</li> <li>15. 隐私用户的moment show显示, 播放slide show查看效果</li> <li>16. 其他人的故事的分享, like等, 播放slide show查看效果</li> <li>17. weixin播放查看</li> <li>18. 各种ie播放查看</li> <li>19. 在moment show和slide show之间切换</li> <li>20. slide show的同时切换进程等</li> </ol>
故事详情界面	点击故事详情	<ol style="list-style-type: none"> <li>1. 故事名称, 描述显示</li> <li>2. 可修改1</li> <li>3. 地图点击, 缩放</li> <li>4. 图片load more</li> <li>5. 添加新照片之后, 查看moment show</li> <li>6. 照片排序</li> <li>7. 时间轴排序</li> <li>8. poi点击</li> <li>9. 点击图片跳转moment show</li> <li>10. 排序界面多点触控</li> </ol>

- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习

☞ 高优先级:

新需求

主要核心功能

数据相关

主要功能路径

...

低优先级:

深入口的界面

用户查看少的界面

...

必须:

兼容性

...



- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习



问自己一个问题：你用过多少竞争对手或体验好的应用？

当你一直忙碌于项目、需求、代码  
当你一直忙碌于同事，上下级的关系  
当你一直忙碌于工作、加班、吐槽

我们需要不停的审视和要求自己

那么你已经离测试越来越远  
那么你已经离用户越来越远  
那么你已经离迷茫越来越近

- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习

了解产品开发的一切并不是为了学习开发、或是放弃测试，转向开发

更多的是为了从各个角度去设计用例、高效的完成测试

- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习

很久之前我看到过这样一段话：

测试的最终境界是什么？有的人回答是全部自动化

测试的最终境界是什么？有的人回答是自动生成有效的用例

测试的最终境界是什么？有的人回答是根本不需要测试

而我却认为，测试最终的境界就是告诉更多的人什么是质量，从点点滴滴中提升所有人对于质量的意识。



- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习



如果你还在办公室进行工作

那么你永远都不会知道外面的世界是什么样的

如果你还在办公室进行工作

那么对你而言仅仅是份工作，永远做不到最好

- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习

就一句话，你有没有让自己的产品融入自己的生活？

YES

NO

- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习

“招聘之前自己先体验一下这个岗位，这不仅有助于找到合适的人，还会帮助你日后更好的管理”

- ✎ 编写测试点，而不是测试用例
- ✎ 给功能、测试点定优先级
- ✎ 使用100个应用
- ✎ 了解产品开发原理
- ✎ 提升所有人对于质量的意识
- ✎ 必须移动测试
- ✎ 必须真正的使用产品
- ✎ 了解其他岗位在做什么
- ✎ 不停的学习

+ 新建笔记本

查找笔记本

名称

数

更新时间

monkeytest15's notebook

33

Git

1

HTML5

8

python

17

ios

21

ios学习

19

Android

29

互联网杂记

33

Trello

1

性能测试

4

Good Url

7

敏捷

5

网络

9

Google C + + 编写规范

10

Wphone

1

automation test

6

java

6

废纸篓

9

- ↳ 测试篇章
- ↳ 流程篇章
- ↳ 自动化篇章



- ↳ 流程没有银弹
- ↳ 该规范的就应该规范
- ↳ 先做再说
- ↳ 不要流于形式
- ↳ 会议只说重点
- ↳ 要坚持

- ❧ 测试篇章
- ❧ 流程篇章
- ❧ 自动化篇章

- ❧ 尽量避免界面自动化
- ❧ 自动化要从小工具做起
- ❧ 不要为了自动化去自动化
- ❧ 选择适合的技术
- ❧ 一定要持续集成
- ❧ 尝试所有你知道的技术

你的项目产品代码架构没有规划设计？

你的项目迭代很快吗？

你的项目产品界面变化很频繁吗？

你根本就不会写代码？

你的项目产品根本就没有界面？

那么请尽量不要做界面自动化！！！！

- ⌘ 尽量避免界面自动化
- ⌘ 自动化要从小工具做起
- ⌘ 不要为了自动化去自动化
- ⌘ 选择适合的技术
- ⌘ 一定要持续集成
- ⌘ 尝试所有你知道的技术

# 勿以自动化小而不为

```
1 import os
2
3 list = [["MD5-01cc9deb2ead839dce025f8f3b72ae97","MD5-ffac552ea5b820c7e82b90ebe162eb6c"],["MD5-5aabb07e116de0558879c607
4
5
6
7
8
9 #Str = str(list[0])
10 #print Str,Str.strip('[]')
11 a = []
12
13 os.system('./qboxrsctl login monkey test')
14
15 for i in range(len(list)):
16     if (len(list[i]) ==1):
17         a.append(str(list[i]).strip('[]'))
18     else:
19         a.append((str(list[i]))[2:38])
20         # a.append((str(list[i]))[42:78])
21         print a
22
23 for i in range(len(a)):
24     print i,a[i]
25     os.system('./qboxrsctl get iprint1 %s /Users/apple/Downloads/_package_darwin_amd64/%s.jpg'% (a[i],a[i]))
26
27
```

- ❧ 尽量避免界面自动化
- ❧ 自动化要从小工具做起
- ❧ 不要为了自动化去自动化
- ❧ 选择适合的技术
- ❧ 一定要持续集成
- ❧ 尝试所有你知道的技术

你问过如下问题吗？

请问robotium好用吗？

请问Android，ios用什么工具做自动化？

请问有什么工具做性能测试的？

Athrun和Coffee听说都不错，我项目是不是也要用？

你问这些问题之前是否考虑过你要测试的点是什么？  
是否考虑过自己产品的特性？



- ⌘ 尽量避免界面自动化
- ⌘ 自动化要从小工具做起
- ⌘ 不要为了自动化去自动化
- ⌘ 选择适合的技术
- ⌘ 一定要持续集成
- ⌘ 尝试所有你知道的技术

## Android

多渠道包——Python+Monkeyrunner

控件测试——instrumentation

功能测试——Robolectric

性能测试——Ngxinfo+CPU/memory Service+MAT+systrace

## Ios

压力测试——instruments+js

webapp——appium

逻辑测试——cocoa UT

```

- @Test
public void shouldHaveActivity() throws Exception {
    activity = Robolectric.buildActivity(MainActivity.class).create().get();
    assertThat((Class<MainActivity>) activity.getClass(), equalTo(MainActivity.class));
}

- @Test
public void shouldHaveHappySmiles() throws Exception {
    String hello = new MainActivity().getResources().getString(R.string.hello_world);
    assertThat(hello, equalTo("Hello world!"));
}

- @Test
public void shouldHaveAButtonThatSaysPressMe() throws Exception {
    activity = Robolectric.buildActivity(MainActivity.class).create().get();
    testButton = (Button) activity.findViewById(R.id.button1);
    assertThat((String) testButton.getText(), equalTo("monkeytest"));
}

- @Test
public void shouldHaveClickButtonSuccess() throws Exception {
    activity = Robolectric.buildActivity(MainActivity.class).create().get();
    testButton = (Button) activity.findViewById(R.id.button1);
    testText = (TextView) activity.findViewById(R.id.textView1);
    testButton.performClick();
    assertThat((String) testText.getText(), equalTo("the text is changed"));
}

```

```
@Test
public void shouldHaveActivity() throws Exception {
    activity = Robolectric.buildActivity(MainActivity.class).create().get();
    assertThat((Class<MainActivity>) activity.getClass(), equalTo(MainActivity.class));
}
```

```
@Test
public void shouldHaveAButtonThatSaysPressMe() throws Exception {
    activity = Robolectric.buildActivity(MainActivity.class).create().get();
    testButton = (Button) activity.findViewById(R.id.button1);
    assertThat((String) testButton.getText(), equalTo("monkeytest"));
}
```

```
@Test
public void shouldHaveClickButtonSuccess() throws Exception {
    activity = Robolectric.buildActivity(MainActivity.class).create().get();
    testButton = (Button) activity.findViewById(R.id.button1);
    testText = (TextView) activity.findViewById(R.id.textView1);
    testButton.performClick();
    assertThat((String) testText.getText(),equalTo("the text is changed"));
}
```

# Android&Ios

```
139 self.assertEqual(json_print_value(return_json, 'result', 0), (0, 1))
140 self.assertEqual(json_print_value(return_json, 'id', 0), (1, 3))
141 self.assertEqual(json_print_value(return_json, 'unit_price', 0), (getPhotoBook6Price(24, 1), 3))
142 self.assertEqual(json_print_value(return_json, 'total_price', 0), (getPhotoBook6Price(24, 1) + Others_Express_price, 1))
143 self.assertEqual(json_print_value(return_json, 'express_price', 0), (Others_Express_price, 1))
144
145 return_json = Client.Get_products_query_price(['{"id":1,"product_category": "PhotoBook6", "pages": 30,"count": 1},{ "id":2
146 self.assertEqual(json_print_value(return_json, 'result', 0), (0, 1))
147 self.assertEqual(json_print_value(return_json, 'total_price', 0), (getPhotoBook6Price(30, 1) + getPhotoBook6Price(22, 2) + Others
148 self.assertEqual(json_print_value(return_json, 'express_price', 0), (Others_Express_price, 1))
149
```

```
150 def testQueryPhotoBook12Price(self):
```

```
151     return_json = Client.Get_products_query_price(['{"id":1,"product_category": "PhotoBook12", "pages": 20,"count": 1}'])
152     self.assertEqual(json_print_value(return_json, 'result', 0), (0, 1))
153     self.assertEqual(json_print_value(return_json, 'id', 0), (1, 3))
154     self.assertEqual(json_print_value(return_json, 'unit_price', 0), (getPhotoBook12Price(20, 1), 3))
155     self.assertEqual(json_print_value(return_json, 'total_price', 0), (getPhotoBook12Price(20, 1) + Others_Express_price, 1))
156     self.assertEqual(json_print_value(return_json, 'express_price', 0), (Others_Express_price, 1))
157
```

```
158 def testQueryPhotoCardPrice(self):
```

```
159     return_json = Client.Get_products_query_price(['{"id":1,"product_category": "PhotoCard", "pages": 23,"count": 1}'])
160     self.assertEqual(json_print_value(return_json, 'result', 0), (0, 1))
161     self.assertEqual(json_print_value(return_json, 'id', 0), (1, 3))
162     self.assertEqual(json_print_value(return_json, 'unit_price', 0), (PhotoCard_current_price, 3))
163     self.assertEqual(json_print_value(return_json, 'total_price', 0), (PhotoCard_current_price + Others_Express_price, 1))
164     self.assertEqual(json_print_value(return_json, 'express_price', 0), (Others_Express_price, 1))
165
```

```
166 def testQueryBigBagPrice(self):
```

```
167     return_json = Client.Get_products_query_price(['{"id":1,"product_category": "BigBag", "count": 1}'])
168     self.assertEqual(json_print_value(return_json, 'result', 0), (0, 1))
169     self.assertEqual(json_print_value(return_json, 'id', 0), (1, 3))
170     self.assertEqual(json_print_value(return_json, 'unit_price', 0), (BigBag_current_price, 3))
171     self.assertEqual(json_print_value(return_json, 'total_price', 0), (BigBag_current_price + Others_Express_price, 1))
172     self.assertEqual(json_print_value(return_json, 'express_price', 0), (Others_Express_price, 1))
173
```

```
174 def testMutiProductPrice(self):
```



- ⌘ 尽量避免界面自动化
- ⌘ 自动化要从小工具做起
- ⌘ 不要为了自动化去自动化
- ⌘ 选择适合的技术
- ⌘ 一定要持续集成
- ⌘ 尝试所有你知道的技术

持续集成不仅仅只是自动化，更是一场长期的战斗。在你真正投入使用之前你是无法想象它的好处。

<a href="#">testreport_2013-10-11_17-10.html</a>	11-Oct-2013 09:10	1573714
<a href="#">testreport_2013-10-11_17-17.html</a>	11-Oct-2013 09:17	1583065
<a href="#">testreport_2013-10-11_17-50.html</a>	11-Oct-2013 09:50	1632123
<a href="#">testreport_2013-10-11_18-55.html</a>	11-Oct-2013 10:55	1642673
<a href="#">testreport_2013-10-11_19-15.html</a>	11-Oct-2013 11:16	1652393
<a href="#">testreport_2013-10-11_19-30.html</a>	11-Oct-2013 11:31	1662113
<a href="#">testreport_2013-10-12_11-10.html</a>	12-Oct-2013 03:10	1679634
<a href="#">testreport_2013-10-12_14-25.html</a>	12-Oct-2013 06:25	1689354
<a href="#">testreport_2013-10-12_17-40.html</a>	12-Oct-2013 09:40	289888
<a href="#">testreport_2013-10-12_18-40.html</a>	12-Oct-2013 10:41	1699074
<a href="#">testreport_2013-10-14_16-10.html</a>	14-Oct-2013 08:10	1709121
<a href="#">testreport_2013-10-14_19-40.html</a>	14-Oct-2013 11:40	1718841
<a href="#">testreport_2013-10-15_13-20.html</a>	15-Oct-2013 05:20	1728561
<a href="#">testreport_2013-10-15_16-00.html</a>	15-Oct-2013 08:00	1738281
<a href="#">testreport_2013-10-15_16-31.html</a>	15-Oct-2013 08:32	1760021
<a href="#">testreport_2013-10-15_16-40.html</a>	15-Oct-2013 08:40	1769741
<a href="#">testreport_2013-10-15_17-15.html</a>	15-Oct-2013 09:15	1812698
<a href="#">testreport_2013-10-15_17-25.html</a>	15-Oct-2013 09:25	1822506
<a href="#">testreport_2013-10-15_18-35.html</a>	15-Oct-2013 10:35	1851666
<a href="#">testreport_2013-10-15_18-38.html</a>	15-Oct-2013 10:38	1877847
<a href="#">testreport_2013-10-15_18-48.html</a>	15-Oct-2013 10:48	1832013
<a href="#">testreport_2013-10-15_19-00.html</a>	15-Oct-2013 11:00	1788389
<a href="#">testreport_2013-10-15_19-25.html</a>	15-Oct-2013 11:25	1858803
<a href="#">testreport_2013-10-16_12-00.html</a>	16-Oct-2013 04:00	1867443
<a href="#">testreport_2013-10-16_14-25.html</a>	16-Oct-2013 06:25	1876083
<a href="#">testreport_2013-10-16_14-30.html</a>	16-Oct-2013 06:31	1883853
<a href="#">testreport_2013-10-16_15-55.html</a>	16-Oct-2013 07:55	1893363
<a href="#">testreport_2013-10-16_17-20.html</a>	16-Oct-2013 09:20	1901133
<a href="#">testreport_2013-10-16_18-30.html</a>	16-Oct-2013 10:30	1910643
<a href="#">testreport_2013-10-16_19-05.html</a>	16-Oct-2013 11:05	1919823
<a href="#">testreport_2013-10-16_19-10.html</a>	16-Oct-2013 11:10	1927593
<a href="#">testreport_2013-10-16_19-15.html</a>	16-Oct-2013 11:15	1937103
<a href="#">testreport_2013-10-17_14-40.html</a>	17-Oct-2013 06:40	1945797
<a href="#">testreport_2013-10-17_15-30.html</a>	17-Oct-2013 07:30	1954437
<a href="#">testreport_2013-10-17_17-15.html</a>	17-Oct-2013 09:15	1963077

- ❧ 尽量避免界面自动化
- ❧ 自动化要从小工具做起
- ❧ 不要为了自动化去自动化
- ❧ 选择适合的技术
- ❧ 一定要持续集成
- ❧ 尝试所有你知道的技术



不要因为自己所在项目使用不到而不使用

只有尝试过尽可能多的东西才能够知道哪些合适自己的产品

能够让自己知道更多，了解更多，何乐而不为呢？

# 谢谢大家

By Monkey陳曄曄

QQ: 383750787

Gtalk: [snowangelsimon@gmail.com](mailto:snowangelsimon@gmail.com)

Weixin: monkey15chen

# 百度技术沙龙

畅想

交流

争鸣

聚会

关注我们：[t.baidu-tech.com](http://t.baidu-tech.com)

资料下载和详细介绍：[infoq.com/cn/zones/baidu-salon](http://infoq.com/cn/zones/baidu-salon)

“畅想·交流·争鸣·聚会”是百度技术沙龙的宗旨。百度技术沙龙是由百度与InfoQ中文站定期组织的线下技术交流活动。目的是让中高端技术人员有一个相对自由的思想交流和交友沟通的平台。主要分讲师分享和OpenSpace两个关键环节，每期只关注一个焦点话题。

讲师分享和现场Q&A让大家了解百度和其他知名网站技术支持的先进实践经验，OpenSpace环节是百度技术沙龙主题的升华和展开，提供一个自由交流的平台。针对当期主题，参与者人人都可以发起话题，展开讨论。

## InfoQ 策划·组织·实施

关注我们：[weibo.com/infoqchina](http://weibo.com/infoqchina)