Geekbang 极客邦科技

InfoQ

QCon 全球软件开发大会【北京站】2016

你从来没有想过的新Monkey测试

腾讯 社交产品质量部 小V

Intemational Software Development Conference

自我介绍

- 小V， 两个娃的爸

- 移动终端专项测试，性能专项，稳定性专项

大纲

- monkey从前：传统仅仅是可靠性测试
- monkey现在：发现过万BUG, 高效率，高覆盖蜕变，我们遇到的难点和解决方法
- monkey未来：如何让性能与Monkey结合，不是像那些云测，我们提供的是开发可以直接解决的BUG

Monkey的从前

Monkey测试

- 随机测试: 生成一系列随机事件进行测试
- 引入电子产品的可靠性指标：MTTF

Monkey的现在

感觉一切都很好

- 在android的测试工具： Monkey
- 在ios官方不提供，但是有个叫ynm3000类monkey工具

我遇到的三个痛点

- BUG发现**太慢**
- BUG改**太难**
- 最后一点：嫌弃BUG**太多**

“还有一天要发布了，发现CRASH了么？”

分析

取版本	测试	提单	解单	回归测试

效率隐藏在每一处

分析

取版本 | 测试 | 提单 | 解单 | 回归测试

效率隐藏在每一处

解决方案：测试效率

- 加强测试的控制力
- 基于控件
- 加权算法

加权算法的来由

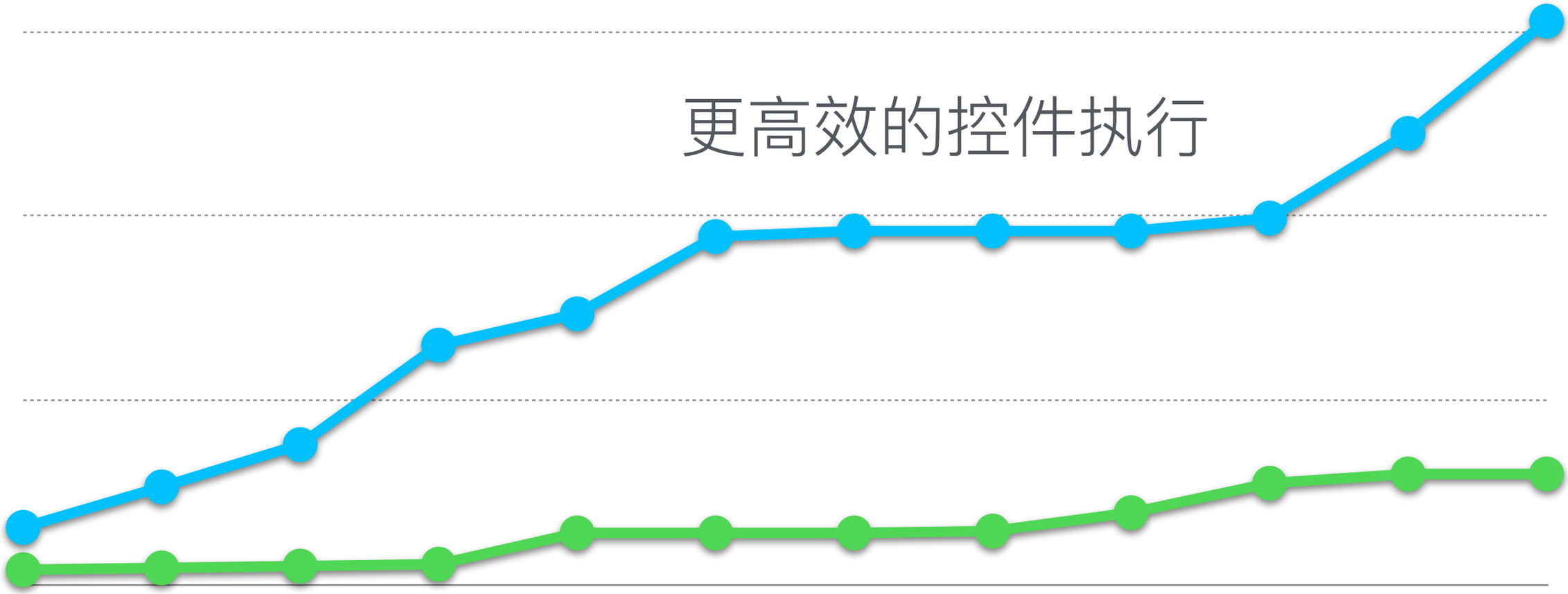
- 保持随机
- 尽量让其他控件有均等机会
- 尽量操作有意义的控件
- 真的只是尽量

加权算法的元素

- 均等机会
  - 控件操作次数
  - Activity进入层次（返回概率）
- 有意义点击
  - 控件isClickable
  - 控件面积

每5分钟控件操作个数（去重）

- 加权算法
- 非加权算法



分析

取版本 | 测试 | 提单 | 解单 | 回归测试

效率隐藏在每一处

不能忽略的流程低效

1. 打开QQ主题就crash
2. 提单给了不对此负责的开发
3. 同样的bug, 提给了另外一个开发
4. 提正确了，但开发在忙新需求
5. 解决了，一直没有用上新包测试

解决方案：流程效率

1. 根据堆栈和配置自动提单给正确的开发
2. 根据堆栈合并缺陷
3. 自动重点知会影响关键路径的bug
4. 利用持续集成，尽快用上新包

“慢”的解决方案

1. 测试效率
2. 流程效率

“这个BUG不知道怎么重现，改不了！”

难解的BUG

- OOM或者其他因内存问题的crash

HPROF

- ANR

RUNLOOP日志/堆栈
StrictMode

- 无法想象的NullPointException

最近5 activity记录
日志纪录
静态扫描

“时间有限，BUG那么多，要改到什么时候?”

解决方案：找重点

- 根据BUG出现的次数和Activity决定严重程度
- 根据发生CRASH函数找是否最近3天修改

成效

接入产品：
应用宝(Android)全民K歌(Android) QQ音乐(Android)
手机QQ空间(Android)　手机QQ(Android)
腾讯课堂(Android)，QQ教育等产品

风险预防：
2015年全年发现上万个BUG

小结

- 慢：更高效的测试与流程
- 难：更多信息让，bug更好解决
- 多：让问题更有重点

Monkey的未来

Monkey的未来

- 可以辅助发现更多类型的缺陷
- 可以测试得很快，甚至加入到CI检查中

可以辅助发现更多类型的缺陷

低成本

不用维护脚本

不依赖脚本的断言

Crash　LowMemoryKill

ANR/WatchDog

内存泄漏 CPU持续过高

卡顿，黑屏，白屏 重复下载 没有压缩

切后台摄像头，GPS没释放

翻版歌曲 私有API 错别字

已经做到的

- 监控Destroy的activity - LeakCanary
- hook文件open,close,read,write
- hook sqlite数据库
- pcap抓包

activity leaks

重复读写
buffer过小
主线程读写

全表扫描
没有使用事务

重复下载
没有压缩

[Magnifier分析云][db_exception]db:2872971611.db table:(产生大量的磁盘I/O(2 times)　ID：52771119　状态：新

详细信息 | 需求 (0) | 代码评审 (0) | SVN提交 (0) | 变更历史 (9) | 更多

Call Stack:

android.widget.BaseAdapter.notifyDataSetChanged(BaseAdapter.java:50)
|-android.widget.CursorAdapter.swapCursor(CursorAdapter.java:345)
|-android.widget.CursorAdapter.changeCursor(CursorAdapter.java:313)

[redacted]

|-android.os.Handler.dispatchMessage(Handler.java:98)
|-android.os.Looper.loop(Looper.java:136)
|-android.app.ActivityThread.main(ActivityThread.java:5001)
|-java.lang.reflect.Method.invokeNative(Native Method)
|-java.lang.reflect.Method.invoke(Method.java:515)
|-com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:785)
|-com.android.internal.os.ZygoteInit.main(ZygoteInit.java:601)
|-dalvik.system.NativeStart.main(Native Method)

SQL:

select * from (select _id# extraflag# frienduin# isread# issend# istroop# NULL as msg# msgData# msgId# msgseq# msgtype# selfuin# senderuin# shmsgseq# time# versionCode# longMsgIndex# longMsgId# longMsgCount# isValid# msgUid# vipBubbleID# uniseq# sendFailC

Log Download:

[Magnifier分析云][activity_leak][no_used]com.tencent.mobileqq.app.QQAppInterface leaked(67 times)　ID: 51470965　状态: 已关闭

详细信息 | 需求 (0) | 代码评审 (0) | SVN提交 (0) | 变更历史 (94) | 更多

GC Path:

com.tencent.mobileqq.app.[redacted]@43191908

|-java.lang.Thread(RedPacketPrecreate)@4335b318

Hprof and Log Download:

http://r[redacted]/v3/mobile/downloadDumpFile?model=dumpfile&uploadTime=1452080632113&fileId=7632

Revision:

184293

Device and OS Version:

GT-I9100@android+4.1.2

Analysis Report URL:

[redacted]

Bug Owner:

{"com.tencent.mobileqq.app.QQ[redacted]": "[redacted]"}

FAQ -- Any problem with this bug, please click:

[redacted]

可以测试得很快

覆盖更广
代码覆盖率
控件覆盖率
界面覆盖率

覆盖更快
算法优化

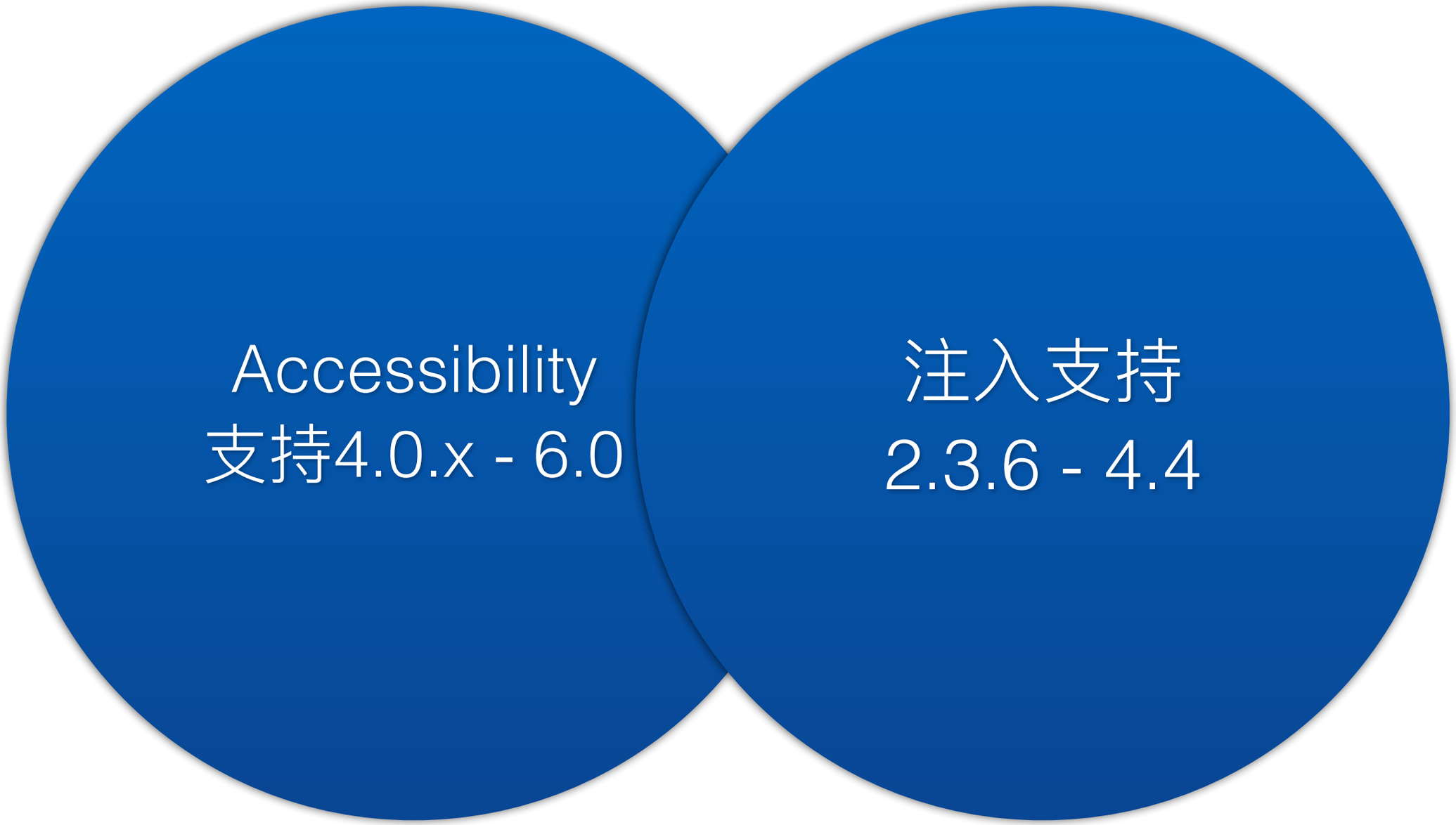
覆盖更更快
分布式monkey
与精准测试结合

与合流结合
与持续集成结合

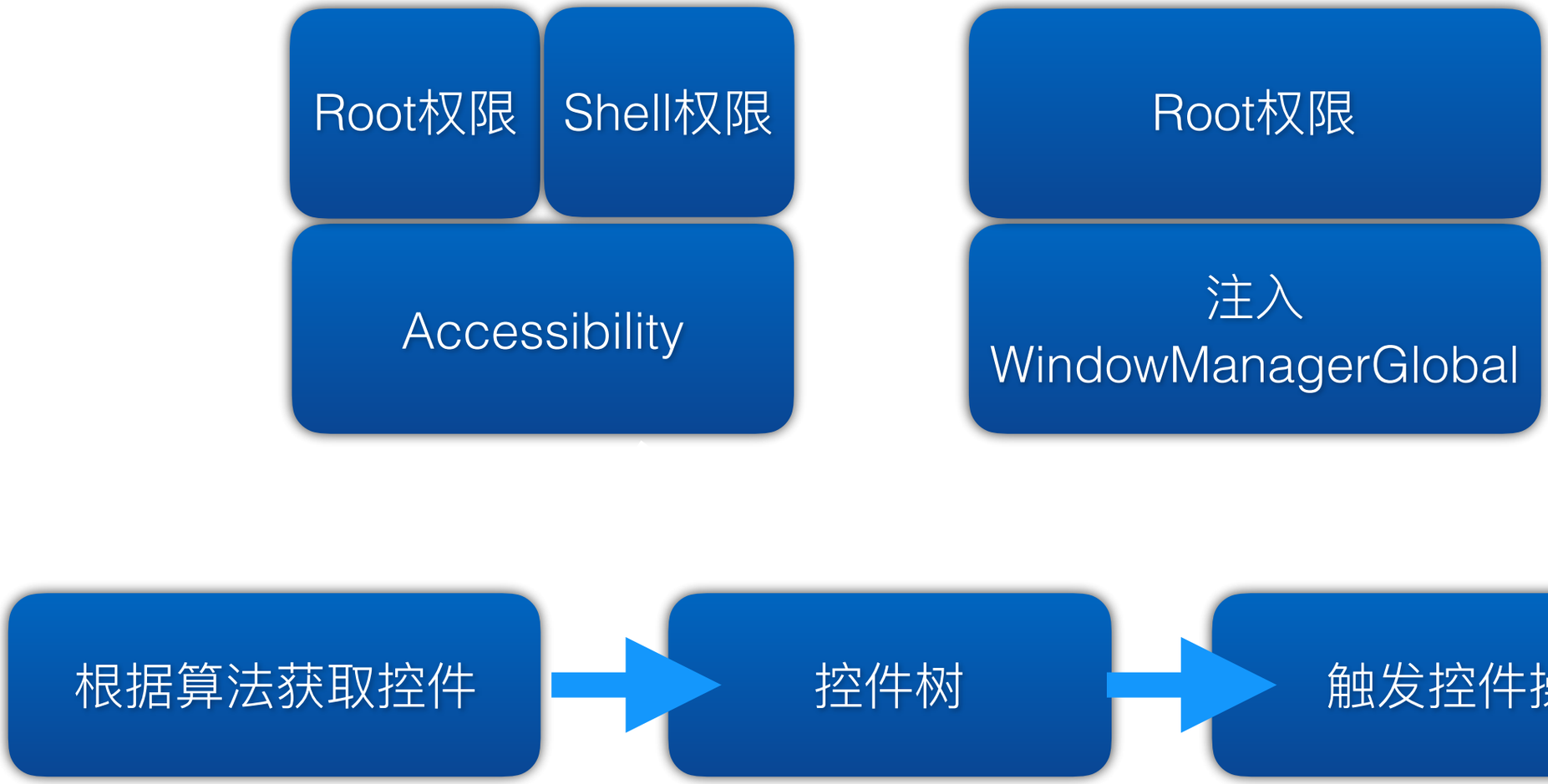
已经做到的：更广

- 更广需要面对的困难
  - 机型兼容性
  - 双引擎
  - 数据上报
  - 兼容性库
  - 复杂前提
  - 触发式脚本
  - 外部资源
  - 断网 / 弱网模拟
  - 账号资源申请

工具兼容性：双引擎



工具兼容性：双引擎


Root权限
Shell权限
Accessibility
Root权限
注入
WindowManagerGlobal
根据算法获取控件
控件树
触发控件操作

AccessibilityAPI

反射获取UiAutomationShellWrapper实例

↓

反射调用connect

↓

反射调用getUiAutomation获取
UiAutomation实例

↓

反射调用UiAutomation的
getRootInActiveWindow()

注入WindowManagerGlobal

注入SystemServer，通过
WindowManagerService.getFocusedWindow反射获取WindowInfo

↓

注入到被测应用

反射获取实例
api17>= WindowManagerGlobal

api17< WindowManagerImpl

↓

反射获取成员变量mRoots

↓

通过WindowInfo.token找当前界面的根节点

复杂前提：触发式脚本

- 登录脚本
- 进入应用场景：例如QZONE进入

```
//获取 文本内容为“登录”的控件
UiView btn_login=new UiView(new UiSelector().text("登 录"));
//获取控件Id为“id/0x20e” 的控件
UiView text1=new UiView(new UiSelector().viewIdName("id/0x20e"));
//如果text1控件在5秒内出现，则设置该text控件内容为UIN字符串
if(text1.waitForExist(5000))text1.setText(UIN);
//获取控件Id为“id/password” 的控件
UiView text2=new UiView(new UiSelector().viewIdName("id/password"));
//则设置该text2控件内容为PWD字符串
text2.setText(PWD);
//点击 文本内容为“登录”的控件
btn_login.click();
//等待MainActivity 10秒
UiHelper.CheckCurrentActivity("MainActivity",10000);
```

已经做到的：更快

- 加权算法->极端加权算法（探索算法）

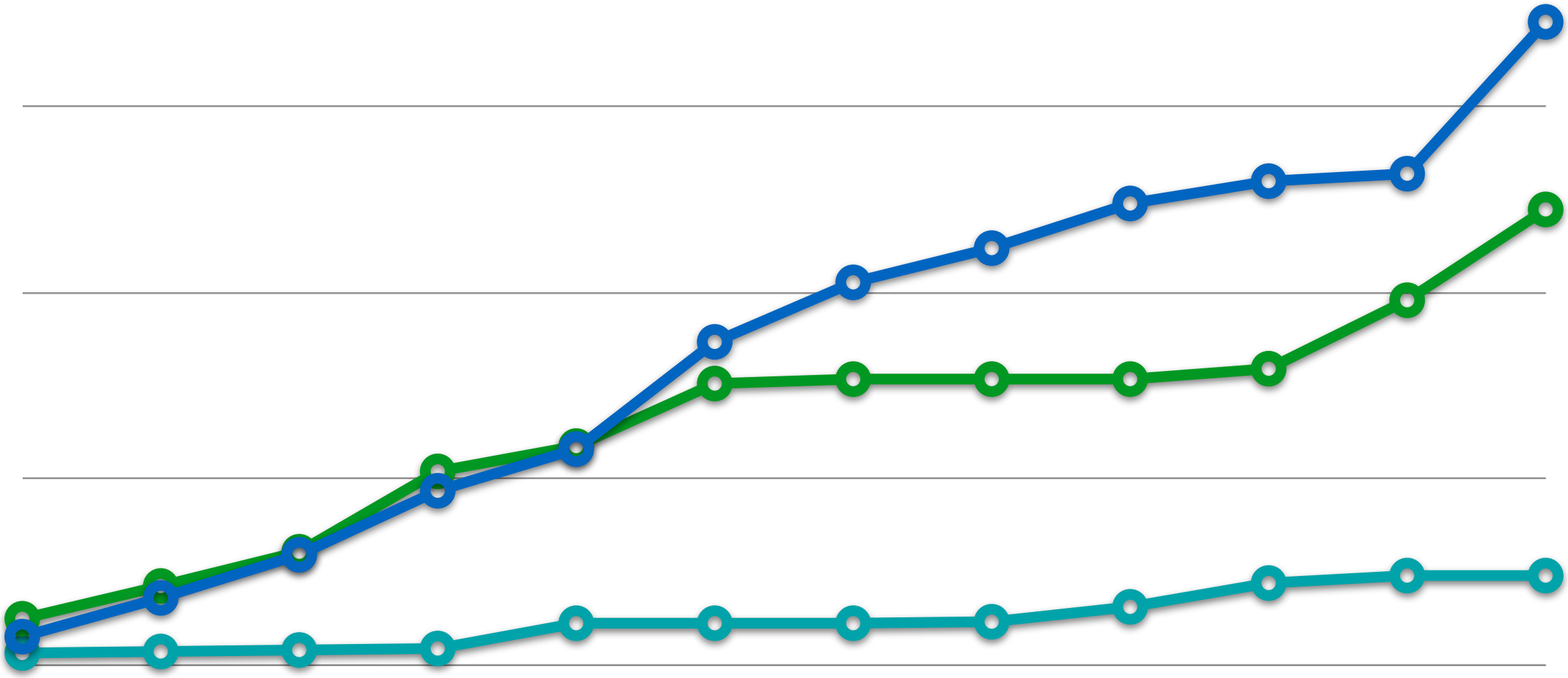
优先级	是被操作过	是否触发跳转	控件树是否变化
P0	0	N/A	N/A
P1（分享）	0	0	1
P2（分享）	0	1	0
P3	1	0	1
P4	1	1	0


遍历 原生 加权
NewMonkey
选择配置：
Magnifier
运行设置:
密码：
默认到发生crash为止
其他：
启用黑名单完增加黑名单
Monkey一下
20:02
3月11日 星期五
Nanshan District
20°C
Google
ChatON
照相机
三星应用商店
Play 商店
手机
联系人
应用程序
完成遍历

每5分钟控件操作个数（去重）

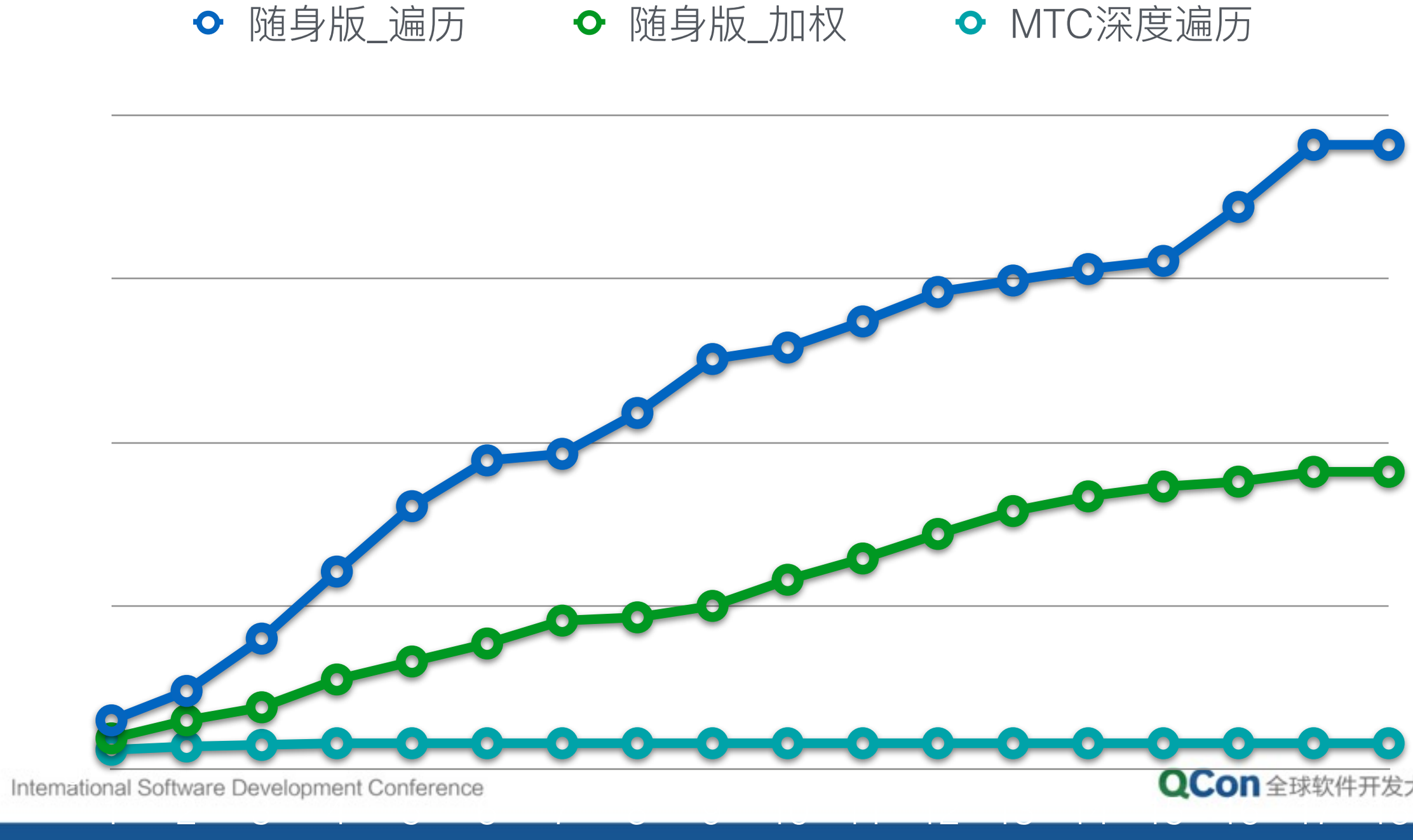
测试产品：手Q

遍历算法　　加权算法　　原生



每1分钟控件操作个数（去重）

测试产品：Google Sample Topeka



2016 1月－3月

提单总量：2493 有效单量：1994

问题类型	总量	有效
Crash问题	2274	1802（79.2%）
延伸专项问题 (io，内存，网络)	219	192（87.7%）

总结

- 稳定性测试：太慢，太难，太多
- 跨越稳定性测试，更广的前提下，更快
- 凝聚成一个随身的app

GIFT

我们team出品

纯干货纯案例《Android终端专项宝典》

Android Monkey 随身版

我的微信

Geekbang 极客邦科技　InfoQ

THANKS!