

testerhome

app快速回归测试

seveniruby
黄延胜

About Me

- ❖ 黄延胜 (思寒)

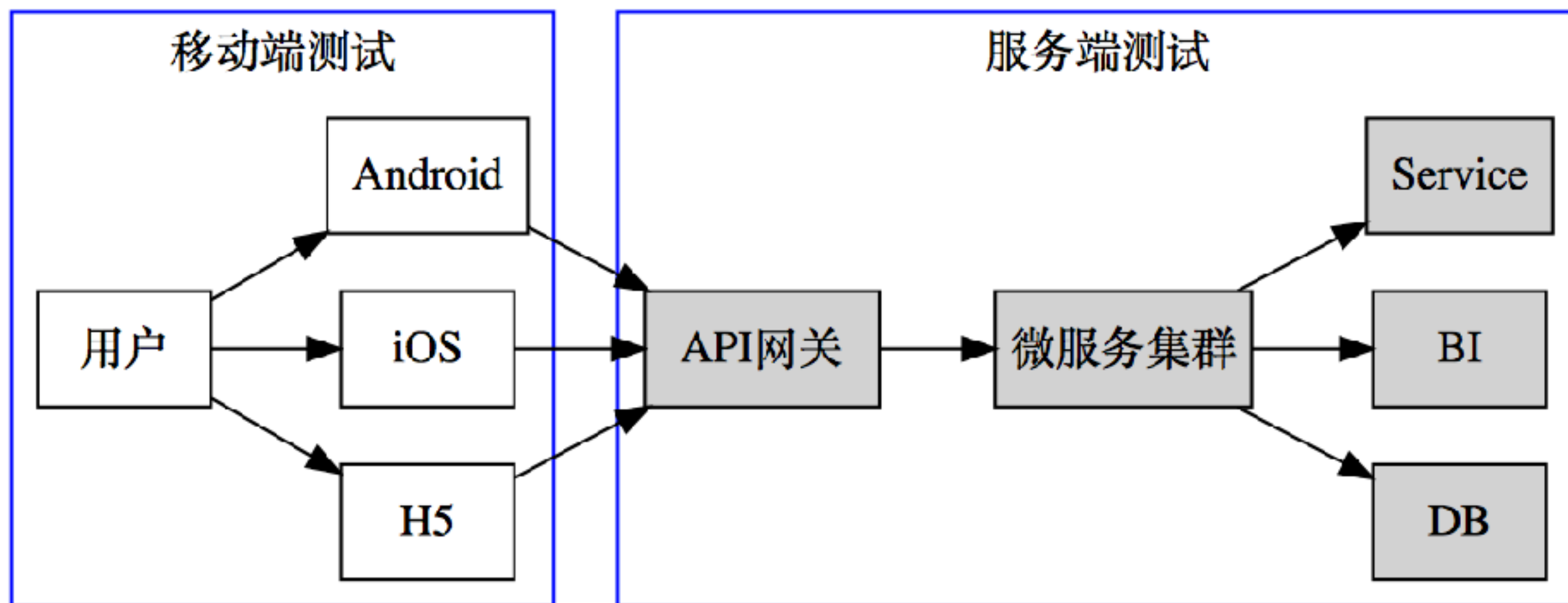
- ❖ TesterHome测试社区技术专家
- ❖ 现任多家公司的测试技术顾问

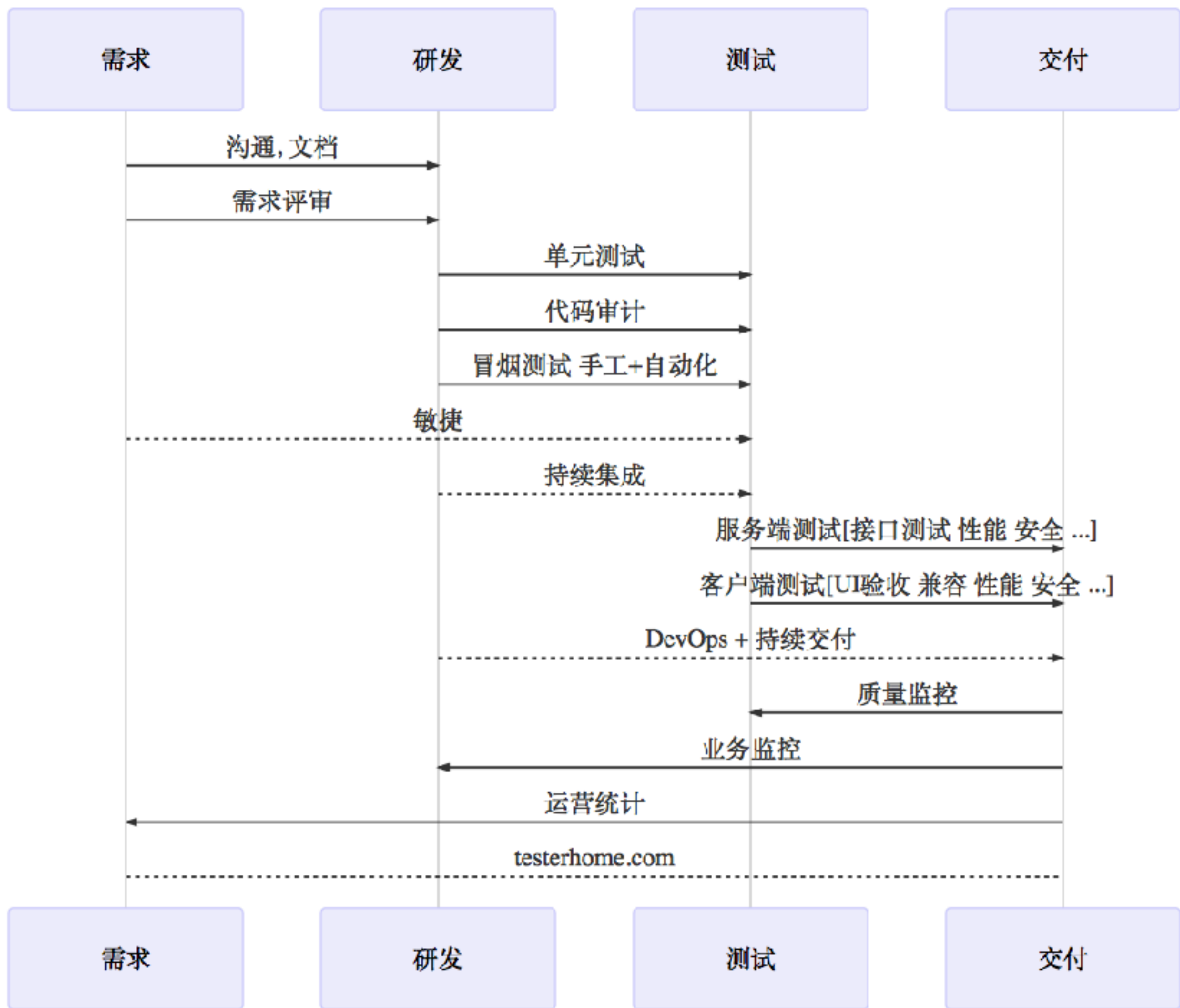
- ❖ 主要工作经历

- ❖ 阿里巴巴, 百度
- ❖ 中航信独立咨询师
- ❖ Testin产品总监
- ❖ 雪球测试技术专家

app传统测试流程

移动互联网公司的简化架构





研发阶段的质量保证实践

- ❖ code review
- ❖ 代码审计
- ❖ 单元测试
- ❖ 自动化冒烟测试
- ❖ 研发自测

测试阶段的质量保证实践

- ❖ 人工测试体系

- ❖ 测试组功能测试
 - ❖ 内测 + 众测 + 公测 等外部测试资源

- ❖ 自动化测试

- ❖ UI自动化测试
 - ❖ 接口自动化测试
 - ❖ 自动遍历测试

- ❖ 专项测试

- ❖ 客户端专项测试: 新老版本兼容 机型兼容 性能 卡顿 电量 流量 FPS 安全
 - ❖ 服务端的专项测试: api回归测试 流量对比 安全

竞争的本质--奥林匹克格言

- ❖ 更快

- ❖ 更快的产品迭代速度
- ❖ 更快的质量反馈

- ❖ 更高

- ❖ 更高的质量
- ❖ 更高的用户体验

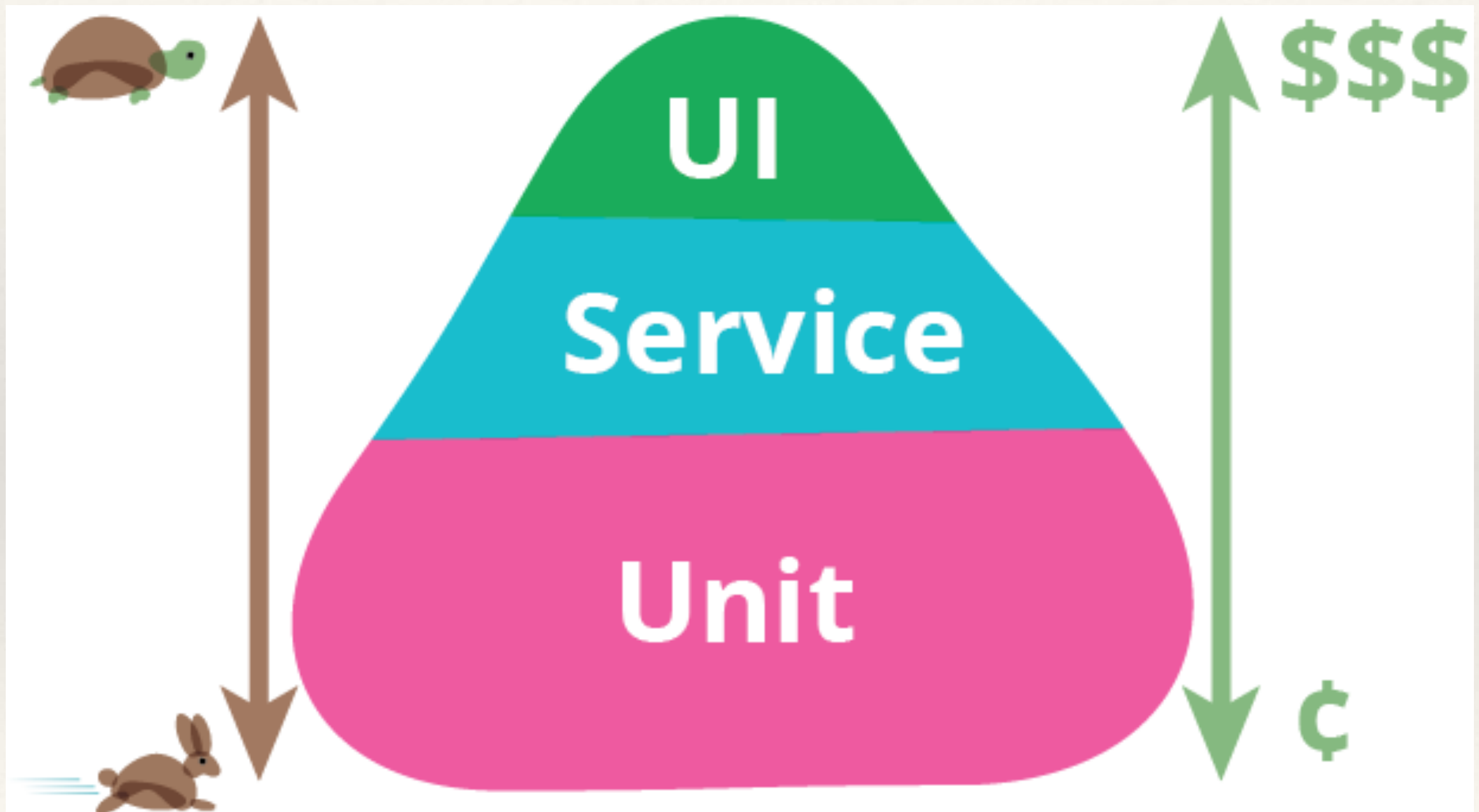
- ❖ 更强

- ❖ 更加专业 更加全能
- ❖ 产品 研发 测试 角色的融合 devops “全栈”

如何平衡质量与效率

- ❖ 解决执行问题: 分层自动化测试
- ❖ 解决数据问题: 测试监控与测试建模
- ❖ 解决应用问题: 数据分析

分层自动化测试



测试监控与测试建模

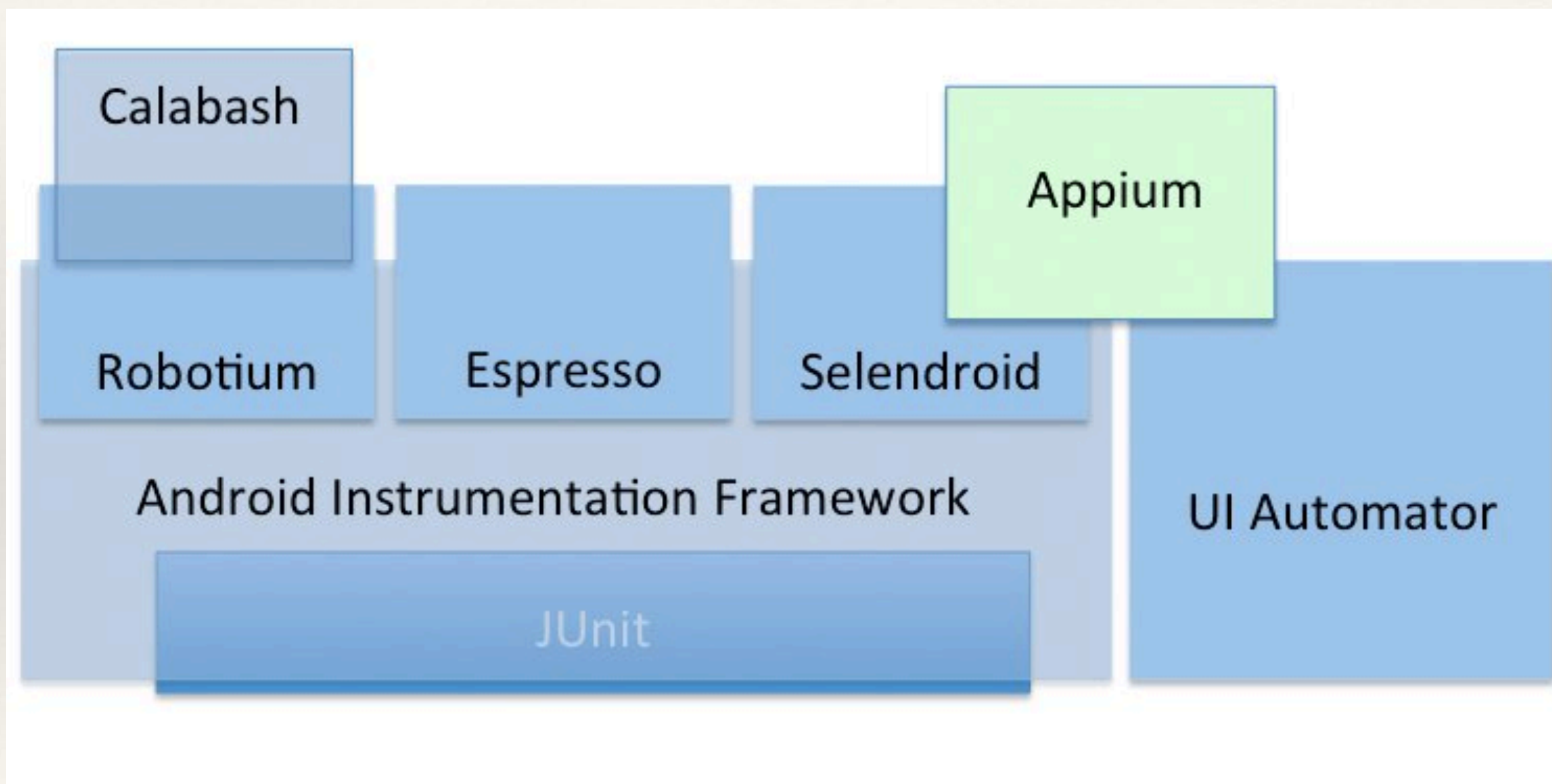
- ❖ app布局数据
 - ❖ 业务流程数据
 - ❖ 性能数据
 - ❖ 接口数据
 - ❖ 覆盖率数据
-
- ❖ 能监控就监控
 - ❖ 没有监控就建模

数据分析

- ❖ 业务流建模
- ❖ 像code diff那样比对新老版本的测试建模数据
- ❖ 可能的机器学习与深度学习

分层自动化测试之app测试

Android 自动化测试



iOS 自动化测试

Automation framework	Xcode version			iOS version		
	6.x	7.x	8.x	8.x	9.x	10.x
UIAutomation	✓	✓		✓	✓	
XCTest (KIF)		✓	✓	✓	✓	✓
XCUITest		✓	✓		✓	✓
Appium < 1.6	✓	✓		✓	✓	
Appium >= 1.6			✓	✓	✓	✓
			Xcode iOS support			
			iOS version			
				8.x	9.x	10.x
			Xcode 6.x	✓		
			Xcode 7.x	✓	✓	
			Xcode 8.x	✓	✓	✓

UI自动化测试的落地困难

- ❖ 投资回报率不理想
 - ❖ UI和业务流程变更是最主要变更
 - ❖ 容易被干扰 执行慢
- ❖ 技术门槛高
 - ❖ 框架技术门槛高
 - ❖ 技术人才难招
- ❖ 没有UI自动化的代价
 - ❖ 回归测试不能有效覆盖 人力疲于应付

UI自动化的常规建议

- ❖ 侧重于回归测试
- ❖ 控制规模
- ❖ 降低编写用例的成本 录制回放 录制生成用例
- ❖ 良好的维护模型 PageObject

UI自动化的最后一公里

- ❖ 自动遍历测试
- ❖ 自动化测试理念的改进 [Another Day]
- ❖ 用例生成 [Another Day]

自动遍历测试

- ❖ 通过不断遍历app中页面和路径尝试发现问题的方法
- ❖ 弥补现有自动化测试的不足
- ❖ 解决Monkey工具可控性差的问题

自动遍历的价值

- ❖ 验证UI的可用性
 - ❖ 基本功能
 - ❖ 兼容性
- ❖ 自动化的性能测试. 结合OneApm, NewRelic
- ❖ 发现内存泄漏 需要借助LeakCanary和MLeaksFinder
- ❖ 发现稳定性问题
- ❖ UI Diff 验证新老版本的功能差异
- ❖ 抓取接口请求分析

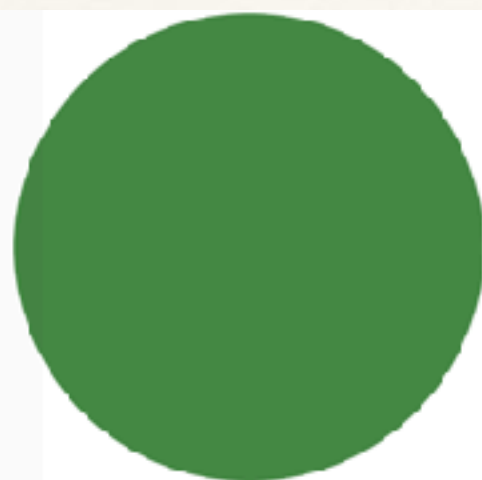
常见遍历工具

- ❖ google android原生monkey
- ❖ 百度smartmonkey
- ❖ 腾讯newmonkey
- ❖ vigossjj smart_monkey
- ❖ seveniruby appcrawler

appcrawler工具

- ❖ 支持遍历控制
- ❖ 数据建模
- ❖ 支持monkey事件
- ❖ 支持android和ios测试
- ❖ 支持native和hybrid开发
- ❖ 支持微信和微信webview自动化

自动遍历的报告



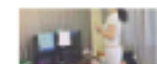
Succeeded	358	(100%)
Failed	0	(0%)
Canceled	0	(0%)
Ignored	0	(0%)
Pending	0	(0%)

☒ Succeeded ☒ Failed ☒ Canceled ☒ Ignored ☒ Pending

Suite	Duration (ms.)	Succeeded	Failed	Canceled	Ignored	Pending	Total
com.android.launcher3-Launcher	10	2	0	0	0	0	2
com.android.launcher3-NewOptInActivity	2	1	0	0	0	0	1
com.google.android.googlequicksearchbox-NewOptInActivity	2	1	0	0	0	0	1
com.xueqiu.android-GuideTutorialActivity	19	5	0	0	0	0	5
com.xueqiu.android-H5Activity	141	21	0	0	0	0	21
com.xueqiu.android-H5Activity-实盘交易	15	3	0	0	0	0	3
com.xueqiu.android-HashTagSearchActivity	5	1	0	0	0	0	1
com.xueqiu.android-HotStocksActivity	12	1	0	0	0	0	1
com.xueqiu.android-LoginOptionActivity	157	7	0	0	0	0	7
com.xueqiu.android-LoginOptionActivity-明细	5	1	0	0	0	0	1
com.xueqiu.android-MainActivity	590	106	0	0	0	0	106
com.xueqiu.android-MainActivity-实盘交易	37	5	0	0	0	0	5
com.xueqiu.android-MultiImageSelectorActivity	2	1	0	0	0	0	1
com.xueqiu.android-PostStatusActivity	86	17	0	0	0	0	17
com.xueqiu.android-ReplierProfileActivity	12	4	0	0	0	0	4
com.xueqiu.android-SingleFragmentActivity	27	4	0	0	0	0	4
com.xueqiu.android-StatusDetailActivity	386	81	0	0	0	0	81
com.xueqiu.android-StockDetailActivity	28	3	0	0	0	0	3
com.xueqiu.android-StockDetailActivity-分时	208	33	0	0	0	0	33
com.xueqiu.android-StockFollowersActivity	46	8	0	0	0	0	8

霉 主采出现多条裂缝

半润上海一楼盘卷入“质量门”：墙面发霉 窗...
股市头条 - 沪深



家电杂谈：被你所忽略的大家电产品特性（二）

今日话题：苏宁V馆文 居民生活的提高，...
家电圈 - 沪深

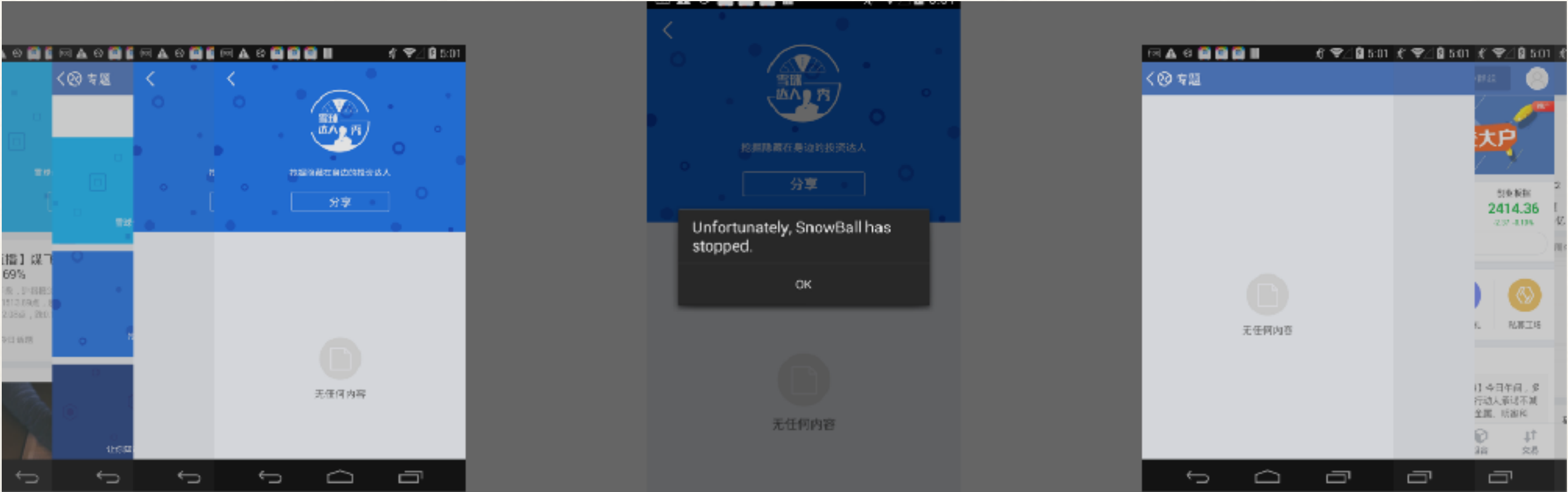


阿里、恒大概念继续扩张，傍上“大款”就真涨！



after clicked





74_MainActivity_-qj_专题-qi_-android.widget.TextView--无任何内容.jpg

名称	修改日期	大小	种类
73_MainActivity_-qj_专题-qi_-android.widget.TextView--挖掘隐藏在身边的投资达人.jpg	今天 下午1:01	121 KB	JPEG 图像
74_MainActivity_-qj_专题-qi_-android.widget.TextView--无任何内容.jpg	今天 下午1:01	135 KB	JPEG 图像
75_MainActivity_-qj_专题-android.view.View-action_bar_overlay_layout-.jpg	今天 下午1:01	57 KB	JPEG 图像
76_MainActivity_-qj_专题-android.view.View-action_bar-.jpg	今天 下午1:01	57 KB	JPEG 图像
77_MainActivity_-qj_专题-android.widget.ImageView-home-.jpg	今天 下午1:01	461 KB	JPEG 图像
78_MainActivity_-qj_专题--android.widget.TextView--上证指数.jpg	今天 下午1:01	200 KB	JPEG 图像
79_qj_专题--StockDetailActivity_-Back-Back-Back.jpg	今天 下午1:01	597 KB	JPEG 图像
80_MainActivity_-android.widget.TextView--组合风云榜.jpg	今天 下午1:01	250 KB	JPEG 图像
81_MainActivity_-jd_组合风云榜-android.widget.TextView--总排行榜.jpg	今天 下午1:01	317 KB	JPEG 图像
82_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--总排行榜.jpg	今天 下午1:02	322 KB	JPEG 图像
83_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--组合综合实力秀.jpg	今天 下午1:02	322 KB	JPEG 图像
84_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--最近6个月.jpg	今天 下午1:02	322 KB	JPEG 图像
85_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--最近3个月.jpg	今天 下午1:02	336 KB	JPEG 图像
86_MainActivity_-jd_组合风云榜-jc_-android.widget.TextView--涨停.jpg	今天 下午1:02	197 KB	JPEG 图像

app功能遍历思维导图



UI Diff



- ❖ 对比新老版本的UI
- ❖ 控件变化
- ❖ 数据变化
- ❖ 降噪

端到端api的diff分析

- ❖ 用proxy拦截网络流量
- ❖ 自动遍历
- ❖ 分析相同事件在新老版本之间的差异
 - ❖ 请求变化
 - ❖ 响应变化

内存问题检测

- ❖ 导致问题

 - ❖ crash

 - ❖ 卡顿

- ❖ 解决办法

 - ❖ 传统解决方案 Android Monitor + iOS Instruments

 - ❖ 自动化解决方案 LeakCanary MLeaksFinder BlockCanary

 - ❖ 自动遍历+sdk

弱网测试

- ❖ 弱网模拟方案

- ❖ FaceBook ATC + 树莓派
- ❖ Mac Charles + Windows Fiddler + 全平台BurpSuite
- ❖ 定制代理服务器

- ❖ 重要场景

- ❖ 弱网场景 响应慢 超时
- ❖ Mock 验证健壮性, 比如修改字段类型和服务端返回的null值
- ❖ Fuzz 验证边界值 利用代理自动生成边界值
- ❖ 自动遍历+定制proxy 批量检测弱网

自动化的冒烟测试

- ❖ 基于debug包的自动化测专项试
 - ❖ monkey健壮性测试
 - ❖ 自动遍历+LeakCanary自动检测内存泄漏
 - ❖ 自动遍历+BlockCanary自动检测卡顿
 - ❖ 自动遍历+Bugly崩溃检测
 - ❖ 自动遍历+接口分析检测接口质量和变更跟踪
- ❖ 基于test包的自动化冒烟
 - ❖ 少量的自动化
 - ❖ 自动遍历

app测试的总结

- ❖ 充分利用自动化
 - ❖ 借助于自动遍历
 - ❖ 借助于专项sdk分析
 - ❖ 借助于api hook proxy等技术
- ❖ 自动遍历不是银弹

分层自动化测试之接口测试

推荐的接口测试优秀实践

- ❖ 选择优雅的接口测试框架
- ❖ 录制+生成测试用例
 - ❖ 基于har或者tcpdump的数据源
 - ❖ 直接生成接口测试框架代码模板
 - ❖ 直接生成配置驱动的用例 excel yaml
- ❖ 新老版本之间的接口对比分析
 - ❖ 结构对比
 - ❖ 重构时的数据对比

开源接口测试框架Rest-Assured

- ❖ 简约的接口测试DSL
- ❖ 支持xml json的结构化解析
- ❖ 支持xpath jsonpath gpath等多种解析方式
- ❖ 对spring的支持比较全面

Rest-Assured示例

```
given().  
    parameters("firstName", "John", "lastName", "Doe").  
when().  
    post("/greetXML").  
then().  
    body("greeting.firstName", equalTo("John")).  
    body("greeting.lastName", equalTo("Doe"));
```

```
given().  
    config(RestAssured.config().xmlConfig(xmlConfig().with().namespaceAware(true)).  
when().  
    get("/package-db-xml").  
then().  
    body(hasXPath("/db:package-database", namespaceContext));
```

```
when().  
    get("/shopping").  
then().  
    body("**.find { it.@type == 'groceries' }", hasItems("Chocolate", "Coffee"));
```

自动生成接口测试用例

- ❖ 用于手工测试场景生成接口测试用例
- ❖ 数据来源 Proxy Har
- ❖ 利用模板技术生成测试用例代码模板
- ❖ 包含请求和断言的自动生成
- ❖ 自动生成数据驱动的用例模板

通过抓包自动生成接口测试用例

The screenshot displays a web browser's developer tools interface, specifically the Network tab. A list of network requests is visible, with the following details:

No.	URL	Method	Path	Status	Size	Type	Content-Type
59	https://www.answern.com	GET	/productCode/033011.html	200	58222	HTML	html
62	https://www.answern.com	POST	/getAllProvinces.json	200	745	JSON	json
64	https://www.answern.com	POST	/getRegion/belongid/				
65	https://www.answern.com	POST	/getTrademark/region				
66	https://clients4.google.com	POST	/chrome-sync/comm				
69	https://www.answern.com	GET	/checkLogin.json?{}				

The context menu is open for the selected request (No. 65), showing various actions. The 'Copy as TestCase with Rest-Assured' option is highlighted. The 'Request' tab is selected, showing the raw request data:

```
{ "dto": { "creator": null, "regionName": "北京市", "trade
```

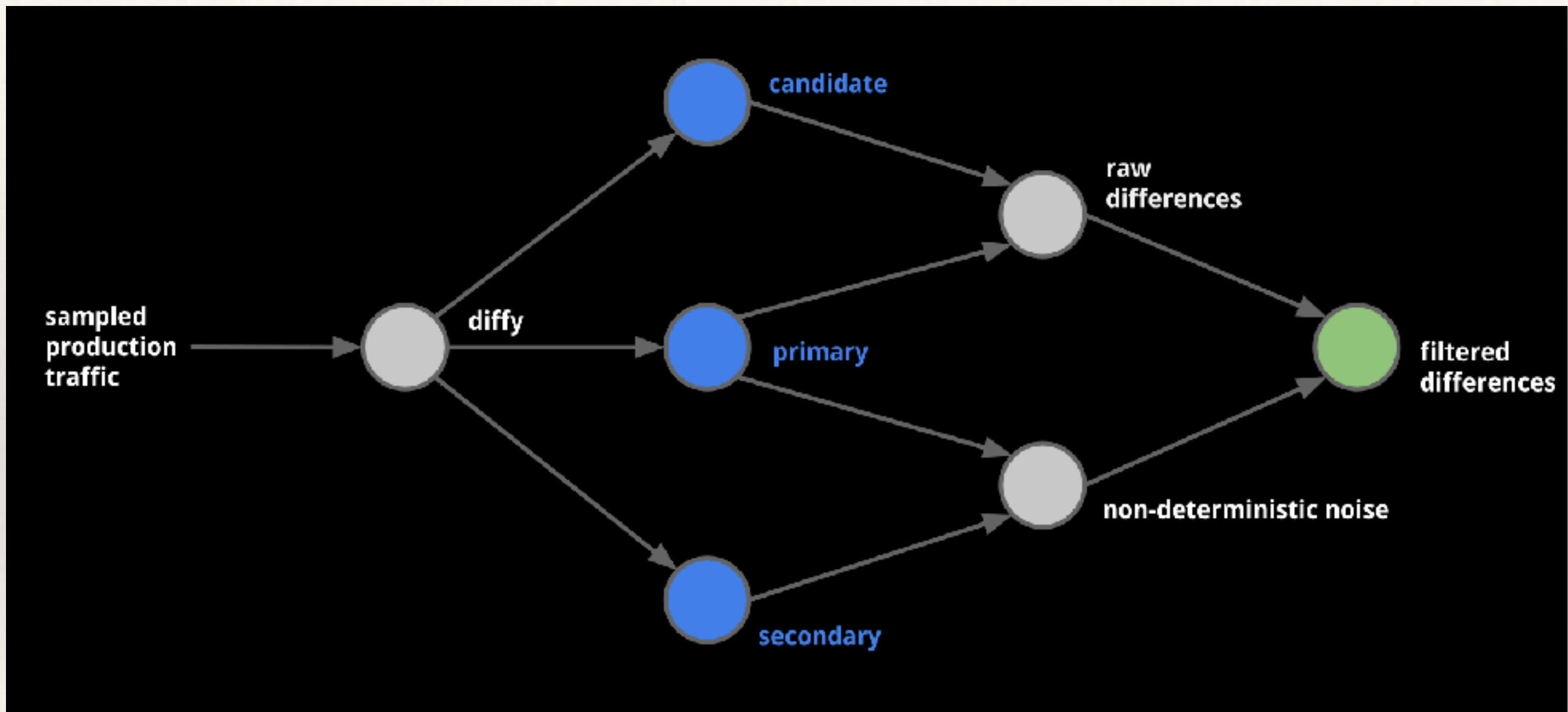
The 'Response' tab is also visible, showing the raw response data:

```
lt": "success"
```

接口大数据回归方案

- ❖ 大数据对比
 - ❖ Diffy
 - ❖ TcpCopy
 - ❖ Gor
- ❖ 不同环境diff
 - ❖ 测试环境 预发布环境 生产环境

twitter的diffy介绍



接口测试中的回归测试

- ❖ diffy需要数据一致的环境. 有一定限制
- ❖ 仍然需要结合接口测试框架做补充
- ❖ 结构diff
 - ❖ 字段类型变更检测
 - ❖ 字段缺失与新增检测
- ❖ 数据diff
 - ❖ 重构项目验证数据正确性
- ❖ 使测试框架智能判定新老版本的变更并进行预警

测试监控与测试建模

测试建模

- ❖ UI测试建模
 - ❖ app的界面数据本质是dom树
 - ❖ 控件的事件本质是一次请求事件
- ❖ 接口测试建模
 - ❖ 请求数据与响应数据
 - ❖ 接口请求
- ❖ 其他数据建模
 - ❖ 性能 覆盖率 ...
 - ❖ 代码调用流 数据流

回归测试的本质

- ❖ UI测试与接口测试的本质就是函数式 $\text{res} = f(\text{req})$
- ❖ 回归测试的本质是测试2个函数
 - ❖ $\text{res_new} = f_new(\text{req})$
 - ❖ $\text{res_old} = f_old(\text{req})$
 - ❖ diff

像code diff那样去diff你的app与服务

- ❖ App Diff 代表作品AppCrawler

- ❖ UI Diff
- ❖ 接口Diff
- ❖ 性能Diff

- ❖ Service Diff 代表作品Diffy

- ❖ 结构Diff
- ❖ 数据Diff

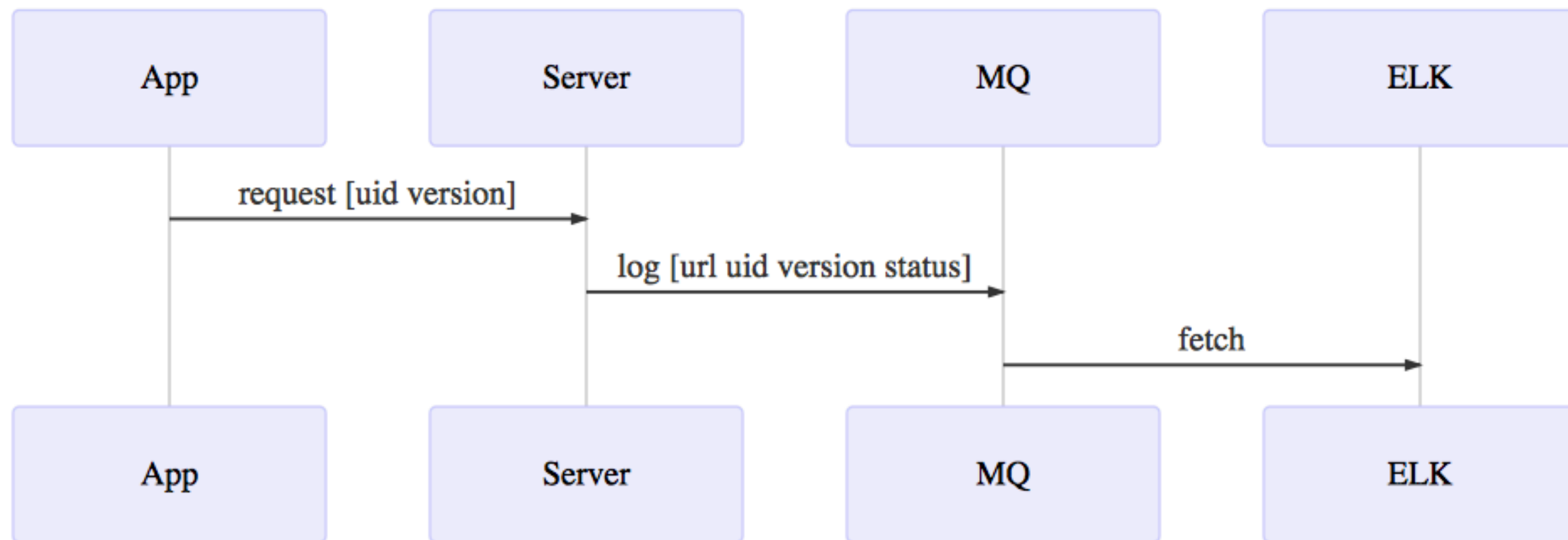
- ❖ 函数级别Diff 代表作品暂无

- ❖ 函数的大数据输入输出对比
- ❖ 数据流diff 调用链diff

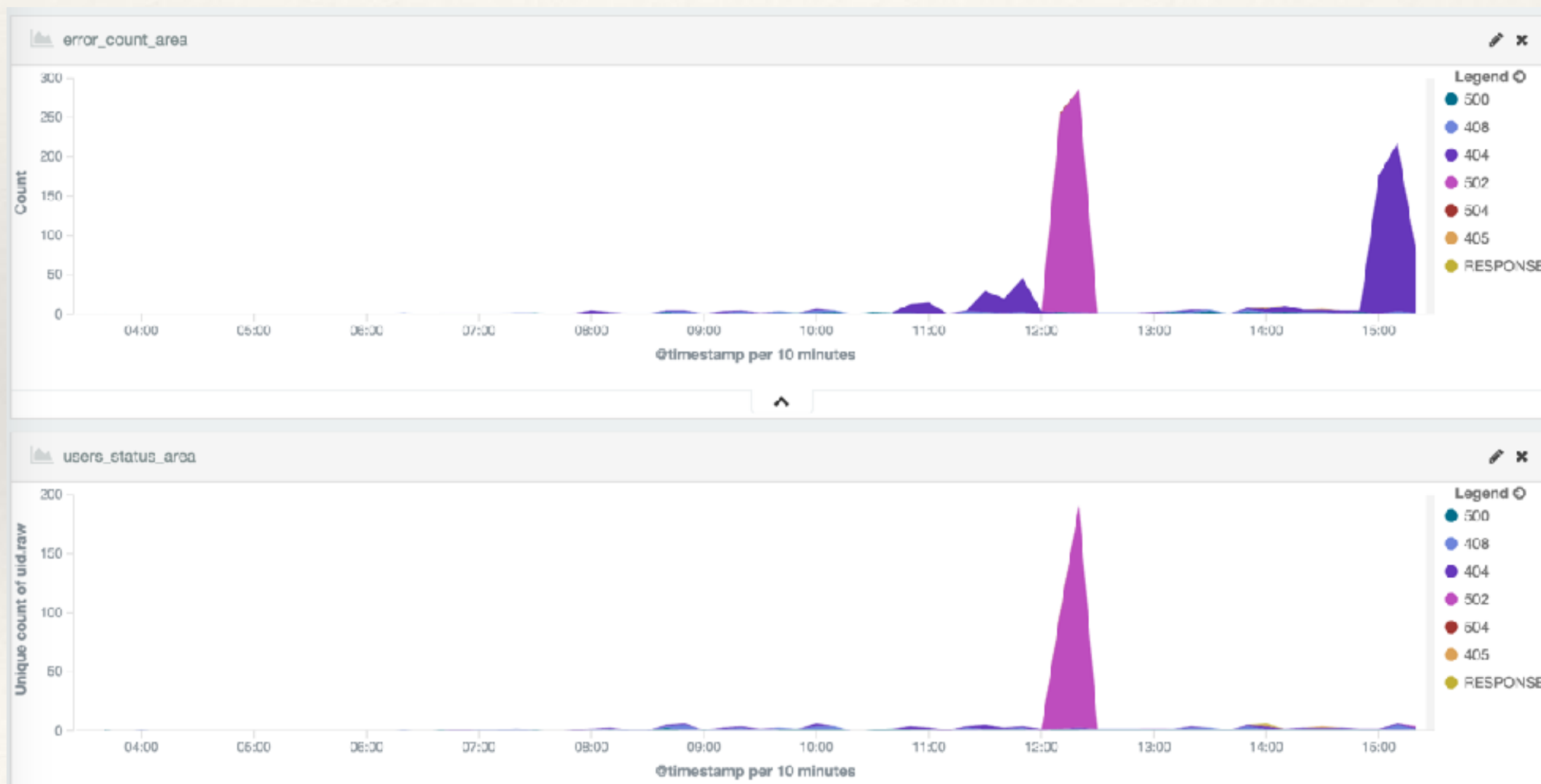
测试数据监控

- ❖ 研发自测
 - ❖ 监控研发内测程度
- ❖ 内部测试+公测+线上监控
 - ❖ 监控分析测试覆盖度
 - ❖ 定位问题
 - ❖ 监控使用人数
 - ❖ 监控使用深度

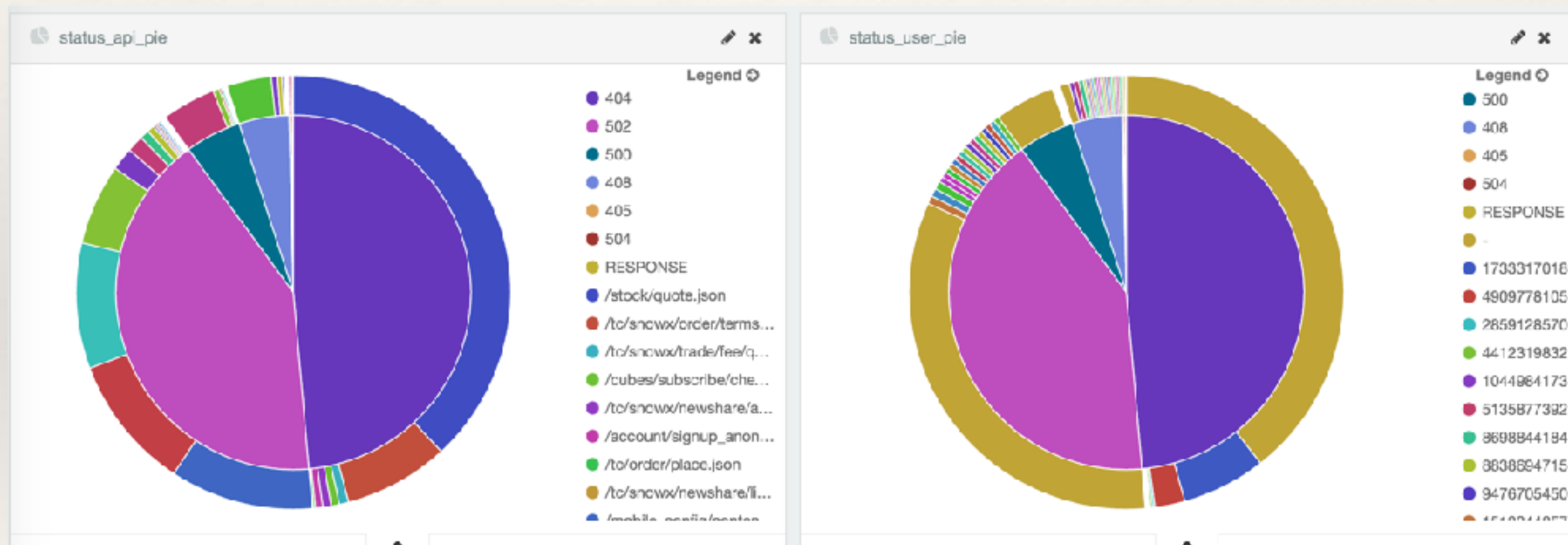
数据来源和流程



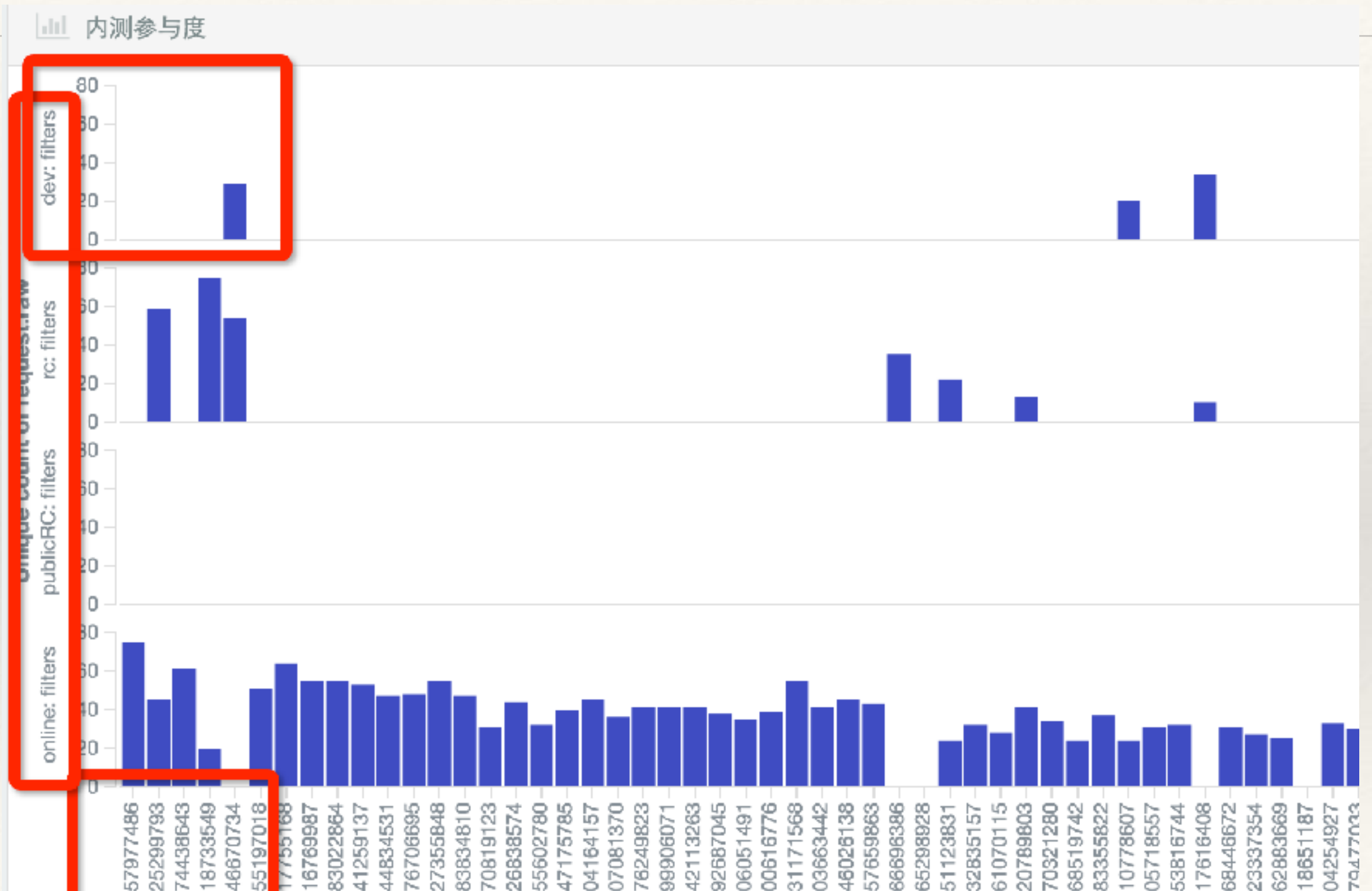
有没有问题



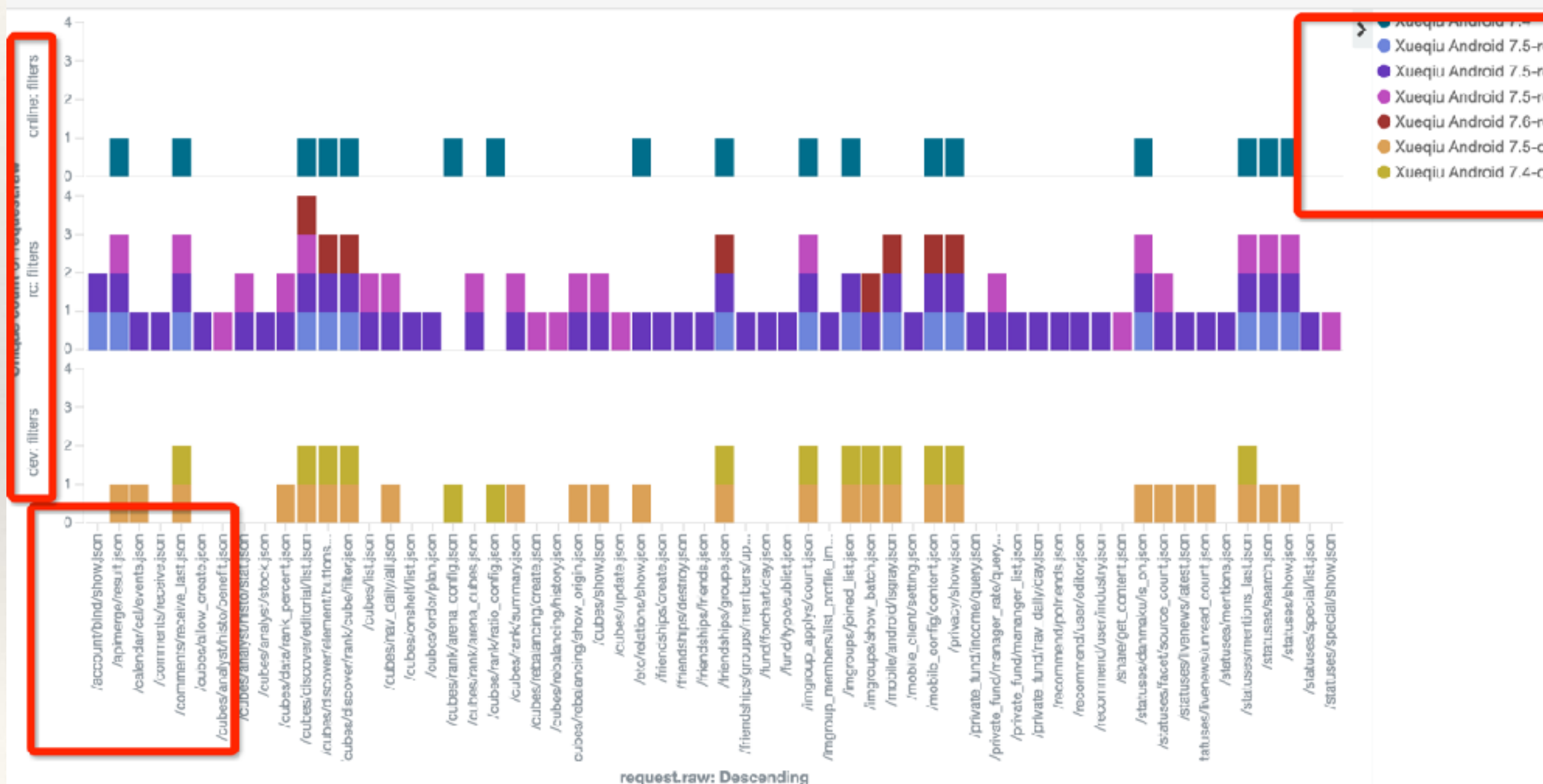
什么样的问题



测试进展分析



漏测分析



总结

- ❖ 分层自动化测试
 - ❖ 自动化冒烟测试
 - ❖ 自动遍历测试
 - ❖ 接口测试
- ❖ 测试数据监控与测试建模
 - ❖ 接口监控数据
 - ❖ app使用数据
 - ❖ 接口测试数据
- ❖ 数据分析
 - ❖ diff 最简单的回归测试策略
 - ❖ 机器学习or深度学习

测试工作中的数据一直没有得到有效的利用和积累
我们需要重视我们的质量数据
利用大数据相关的技术完善新时代的测试体系

seveniruby@testerhome.com