

# 中小型游戏公司 如何发挥自身优势 并利用好开源工具

DeNA 黄洪鵬

## MTSC2018

第四届中国移动互联网测试开发大会

TesterHome | IT大咖说

TesterHome

## AGENDA

讲师简介

游戏测试现状

测试匠心VS效率

游戏审核标准

常用工具

## 讲师简介

简介

DeNA



Health Care



Automotive

大会

Mobile Game

## 游戏测试现状

游戏测试工程师将来都要去做运营和策划吗

## 游戏测试现状

## 游戏测试的困境

- 需求变化快，文档混乱
- 测试投入成本低
- 测试复杂度高
- 测试上升发展受限

MTSC  
第四届中国



## 游戏测试现状

## 丰富的语言

- 客户端

Lua JavaScript C# C++

- 服务器

Lua JavaScript C++ PHP  
JAVA

- 测试

python JS C++

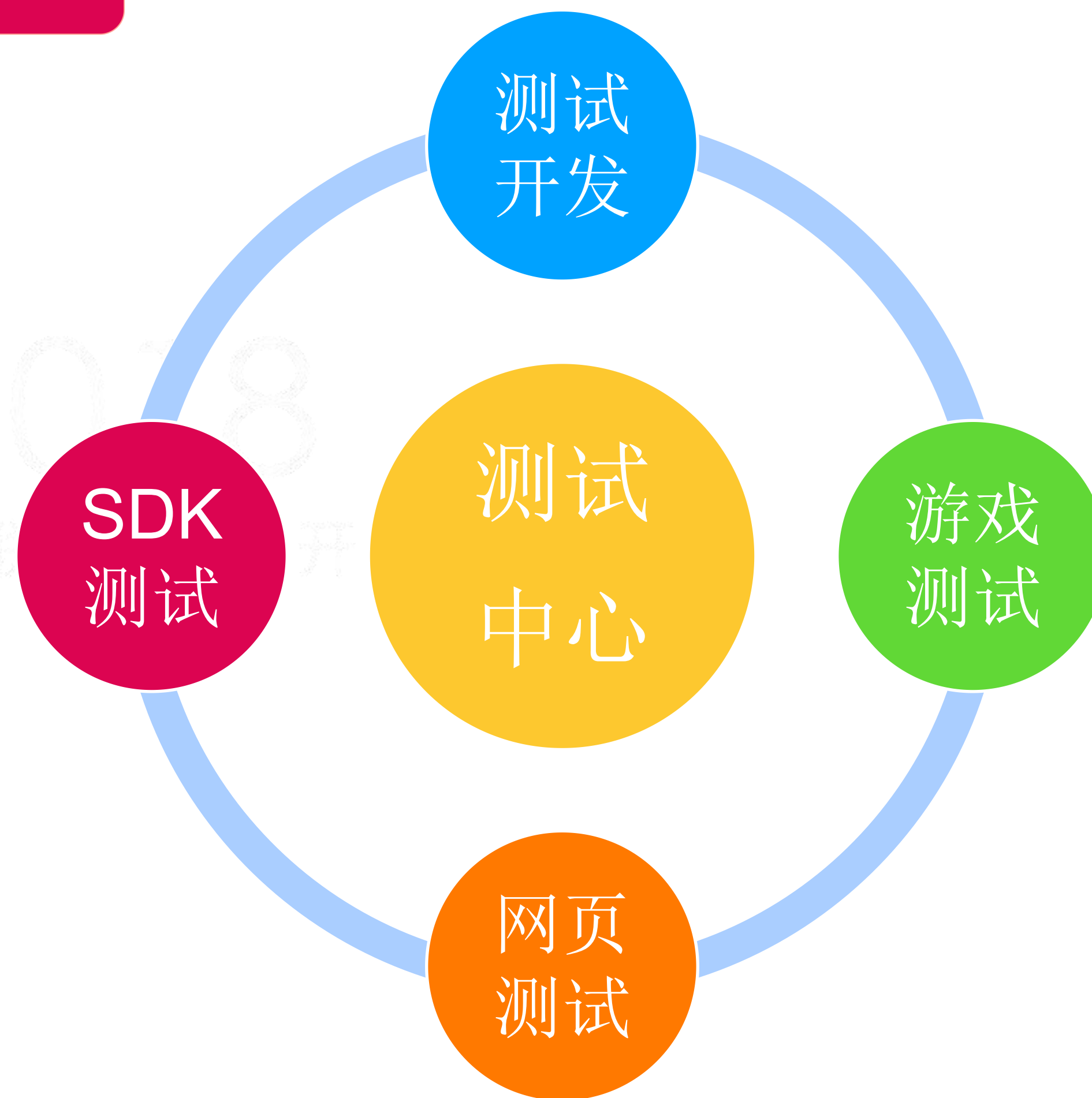


游戏测试现状

游戏公司测试中心建设

## 游戏公司建设测试中心的优势

- 测试人员的上升通道不再是策划和运营岗位
- 游戏测试可以和测试开发形成互补
- 紧急的集中测试任务可以跨团队支援
- 测试标准统一化，测试结果客观





测试匠心



测试效率

第四届中国移动互联网测试开发大会

我们做的事，别人也在做。我们在想怎样才能把事情做得更好。

测试匠心

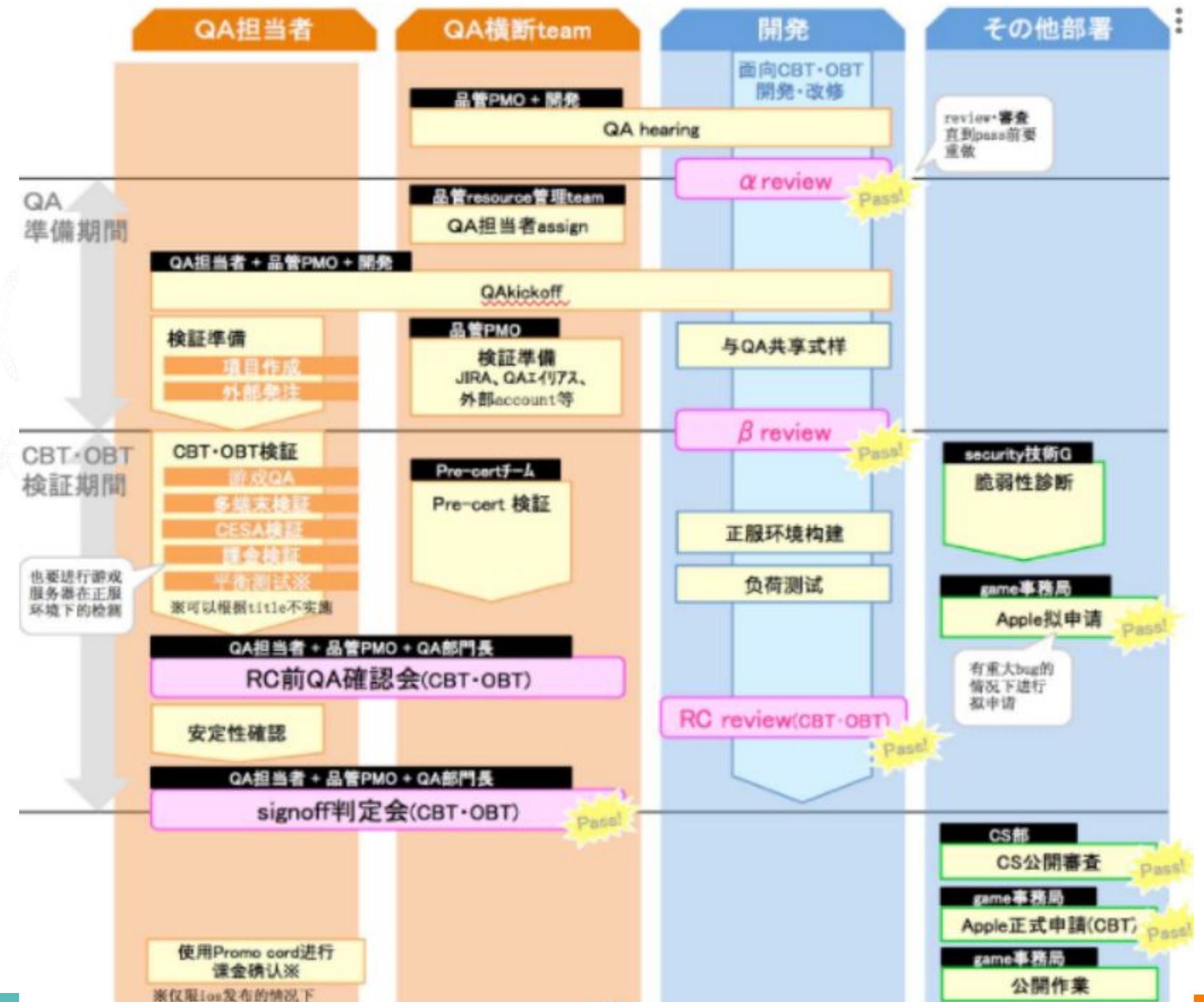
测试效率

日本游戏测试

## QA目标的世界观

与开发者保持适当的距离，独立，进行具有第三者性、客观性的质量管理

- 测试初期准备
- 测试实施
- 测试完成管理



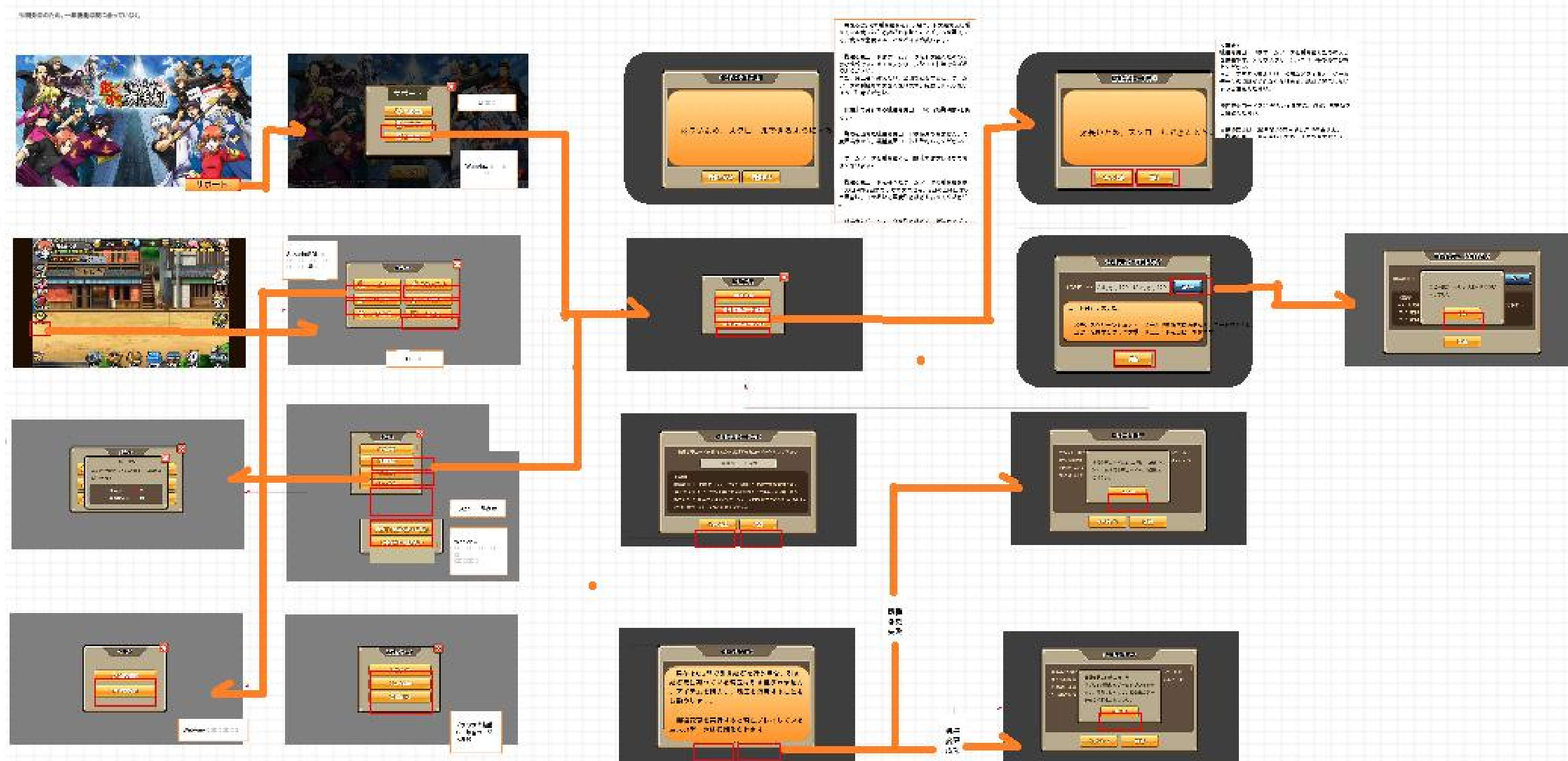
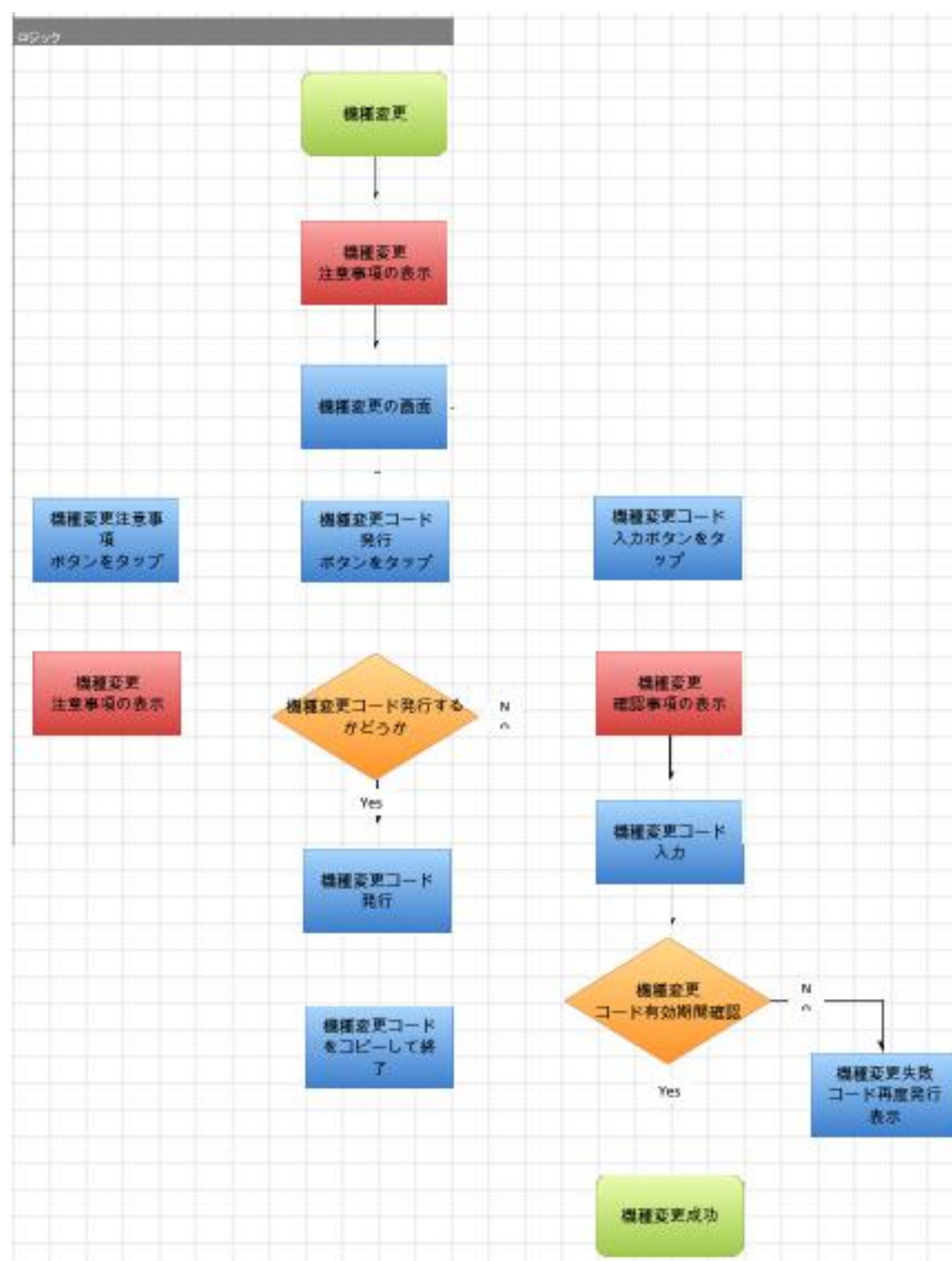
测试匠心

测试效率

日本游戏测试

## 设计式样书补充流程图

## 界面流程图



测试匠心

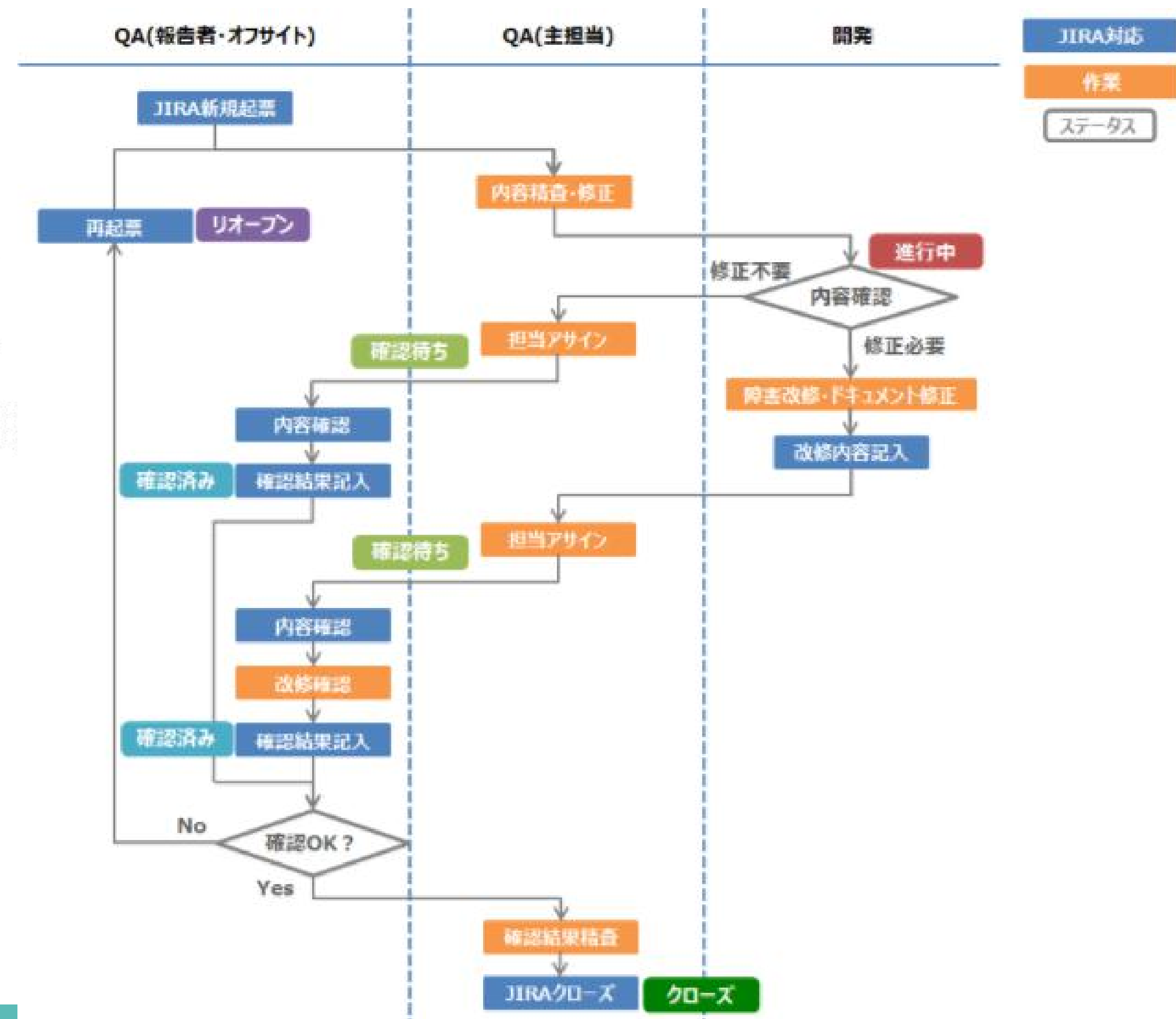
测试效率

日本游戏测试

## BUG 起票规范

セクション名	詳細
詳細	バグの詳細情報を記載するセクション。 Summaryだけの情報のみで伝わるのならば「標題の通り」だけでも十分。 不足している場合は、分かりやすく簡潔になるべく3行でまとめられるよう記載する。 補足がある場合は後述の補足のセクションに記載する。 敬語は使わない
補足	詳細に関連する補足を記載するセクション。特になければ削除する
再現手順	前提条件、手順、期待する結果、実際の結果を記載するセクション。
前提条件	箇条書きで1条件ずつ記載する
手順	番号付きリストで1手順ずつ記載する
期待する結果	最後の手順を実施した際のバグが発生しなかった場合の問題ない挙動を記載する ・基本的に項目書の期待結果にならう ・期待する結果が不明の場合は無頭に記載せず削除する
実際の結果	最後の手順を実施した際の問題となる挙動を記載する
③手順について	「○○を確認する」「○○に遷移する」といった状況説明の様な内容は厳密には手順ではないので、「○○ボタンをタップする」「○○を入力する」といった実際に操作する内容の記述を心がける。
検証環境	バグ発見時の検証環境を必要に応じて記載するセクション。 以下に例を上げるが、あくまで一例なので案件に応じて記載すべき情報は変わるので念頭に置いてください。
現象確認ページ	
パラメータ	内容 e.g.
URL	現象を確認したページを記載する。 ページ遷移が起因であれば、前後のURLも記載する。 http://sp.test.mbga.jp/
端末情報	
パラメータ	内容 e.g.
Device	端末のモデル名 SO-02F
OS	OS名とバージョンを記載する Android 4.4.2
Browser(Ver.)	ブラウザ名とバージョン Chrome(42.0.2311.109)
Environment	接続先環境名 jp_dev_service
Date	バグ発生時の日付と時間 2015-04-28 11:00
Results	結果、NGとOK {color:red}NG{color}
④端末情報を記載する場合は問題の切り分けがし易いように端末カバレッジを取りましょう。 たとえば、Android 4.2の標準ブラウザでバグが発生した場合、以下の様な事が考えられますので、他端末や他バージョンなどで再現性を取り、再現条件を絞り込んでいきましょう。	
・同じバージョンの他端末では問題ないのか	
・Android 4.2以外のバージョンでは問題ないのか	
・標準ブラウザ以外のブラウザでは問題ないのか	
・Android以外のOS (iOSやWindowsなど) では問題ないのか	
・異なる接続先環境では問題ないのか	
ユーザー情報	ユーザーに起因するバグの場合はユーザー情報を記載するとバグの原因を特定しやすくなります。
パラメータ	内容
mobage_user_id	検証で使ったモバゲーのユーザーID
Nickname	上記ユーザーのニックネーム

## BUG 流程



测试匠心

测试效率

测试监控

- 手工测试成本越来越昂贵
- 如何利用好每一次手工测试

手机测试时要求与电脑充电连接，  
电脑上的应用自动收集手机上的操作录像、截图  
和本地日志

电脑定时上传收集到的讯息到后台FTP服务器  
后台分析系统定时从FTP服务器拉取数据并作分析

在工作电脑上访问后台服务分析测试数据



## 测试匠心

## 测试效率

## 测试监控

表示：对比所有回归测试

Showing 1-15 of 15 items.

#	User ID	User Name	开始时间	结束时间	日志分析状态	测试类型	测试标签	
1	220	yue...	2017-11-01 16:57:56	(not set)	正在测试...	基准	base	👁
2	220	yue...	2017-11-01 18:17:46	(not set)	正在测试...	基准	base	👁
3	212	guorong...	2017-11-01 18:18:18	(not set)	正在测试...	基准	base	👁
4	220	yue...	2017-11-28 14:59:09	2017-11-28 14:59:14	等待解析中...	基准	base	👁
5	238	...	2017-11-28 15:44:44	2017-11-28 15:45:09	等待解析中...	基准	base	👁
6	238	...	2017-11-28 15:48:33	(not set)	正在测试...	基准	base	👁
7	238	...	2017-11-28 15:57:30	2017-11-28 15:57:36	等待解析中...	基准	base	👁
8	238	...	2017-11-28 15:59:07	2017-11-28 16:01:20	等待解析中...	基准	base	👁
9	238	...	2017-11-28 16:05:21	2017-11-28 16:08:26	等待解析中...	回归	base	👁

点击"基准测试次数"：可显示回归测试少测的动作

点击"回归测试次数"：可显示回归测试多测的动作

点击出现次数中的"数字"链接：可显示该动作的日志文件、所在行数等详细信息

Showing 1-6 of 6 items.

#	动作	基准测试次数	回归测试次数 ↓	是否覆盖
1	card:	21		否
2	skill	13	4	是
3	experience	17	6	是
4	format	169	23	是
5	index	174	52	是
6	...	...	...	...

测试匠心

测试效率

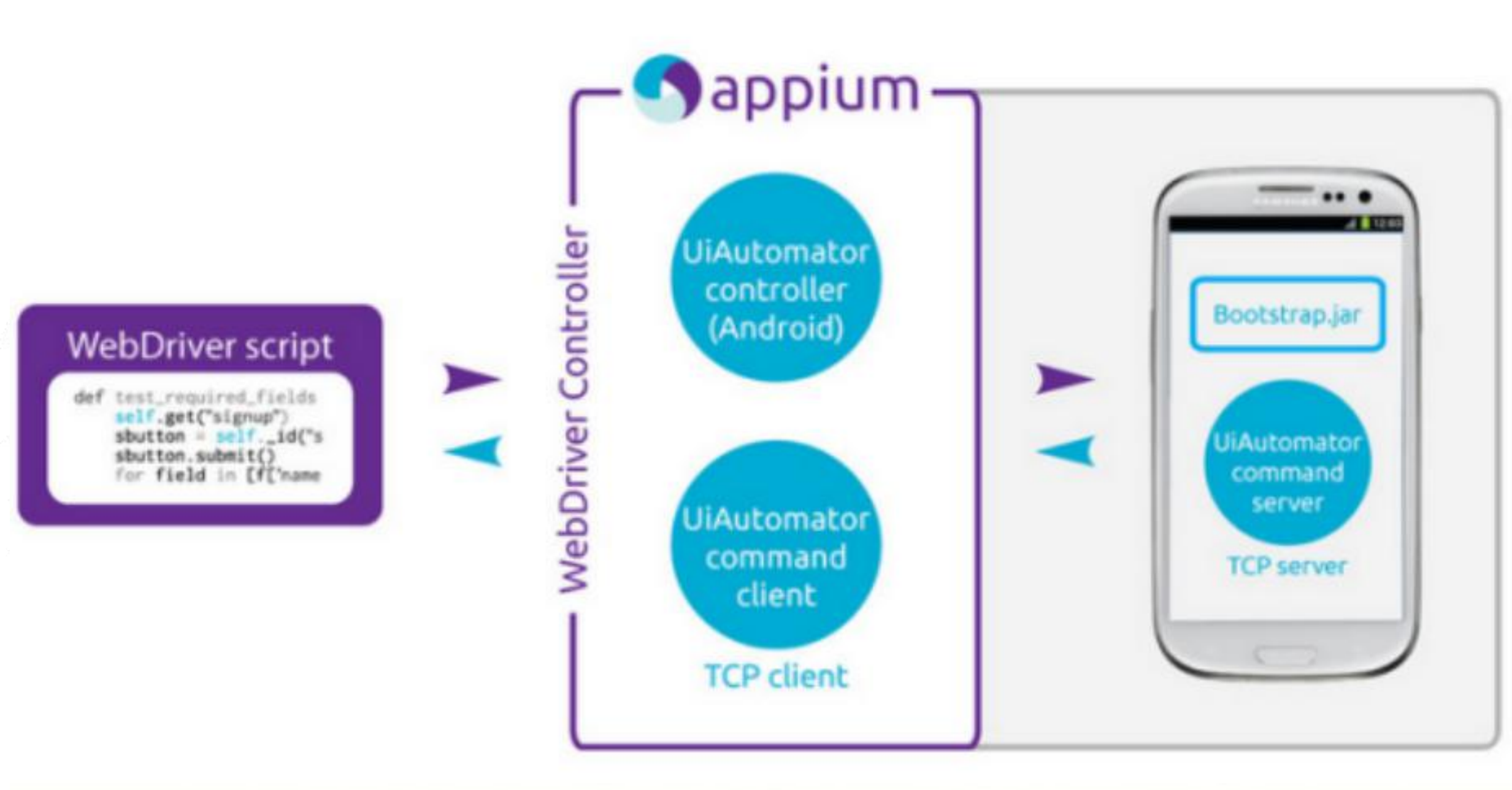
测试自动化

## • SDK测试选择使用自动化测试

结构是pageobject为主，ui2为主，  
采取常见多进程的方式运行，  
appium服务上做了从监听平台投递的文件夹到自  
动开始测试出报告。

扩展功能：

守护进程自动判断异常重启，  
监听执行进程内存，结论报告扩展，  
统一断言方式，自动读取手机配置信息，  
填写硬件和启动信息到报告，  
填写测试用例结果等



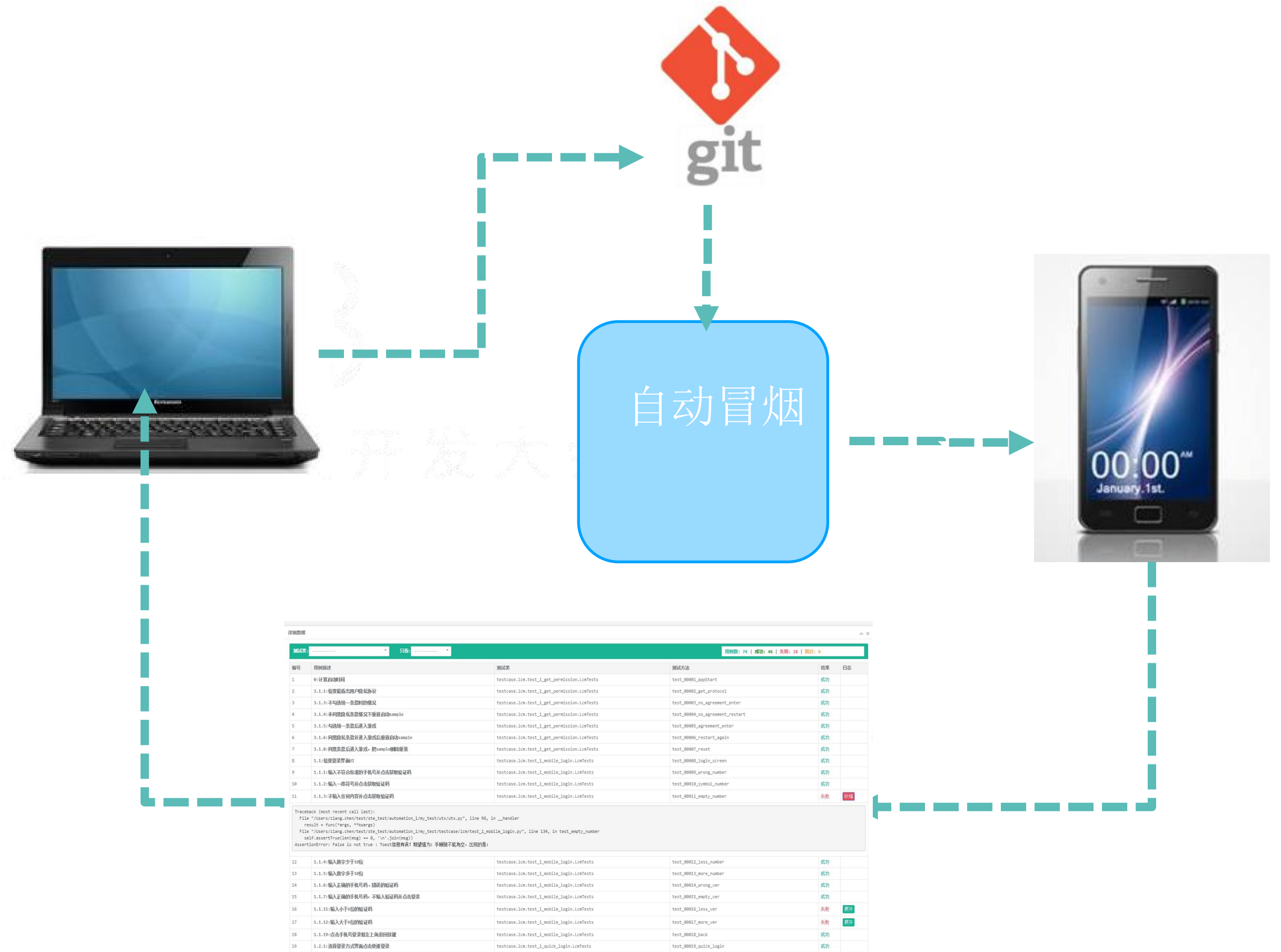
测试匠心

测试效率

测试自动化

- SDK测试重复性较高
- 功能较稳定
- BUG易于被程序识别

MTSC2018  
第四届中国移动互联网测试开发大会





测试匠心

测试效率

测试自动化

## 自动化测试目录结构

- Client
- Config
- Report
- Screenshot
- Src
- Testcase
- Util
- UTX

```

自动化appium工程
run.py #main文件

--apk #软件库
  new #通过自动打包服务的记录标识符（未来版本会废弃）
  package_list.csv #监听执行任务列表
  temp.json #临时文件，存放执行过程中的一些包名，报告名，测试结果
  weixin_1260.apk #业务用微信，微信登录，收费等

--client #工具类集合
  adbCommon.py #adb工具类
  base.py #前置业务和手势方法
  client.py #高级封装定位&截图等
  cmd.log #执行特定的call本地埋点
  data.json #appium进程性能文件
  filebase.py #监听进程和包分发处理
  log.py #日志工具类
  MailSession.py #邮件富文本工具类
  __init__.py

--config #数据驱动
  case_FT_Android_D.csv #整体xxx对应的case
  case_JT_Android_D.csv #简体xxx对应的case
  case_number.csv #读取用例编号
  config.ini #unittest跳过条件配置&相对路径
  config.py #同上
  data.csv #期望数据集
  excel_to_csv.py #辅助函数转换格式
  mobile.ini #手机配置信息，该手机的名称、手机号、账号、微信号、邮
  __init__.py #用来执行哪一个类型的用例（繁体，简体）等

--log
  .txt #存放unix时间日志-本地调式

--report #存放appium结果日志
  1.xlsx
  2.xlsx
  2018-01-08-20-16-12-ztest.html

--AppStore #用例执行完成后会在相应的文件夹下生成html格式的测试报告
  和xlsx格式的测试结果
  FT_Android_D
  FT_Android_Google_D
  FT_IOS_D
  GooglePlay
  JT
  JT_Android_D
  JT_IOS_D
  OneStore

--screenshot #截图存放路径
--src #存放测试包 测试包来源打包仓库监听
--testcase #测试用例
  readHelp.py
  __init__.py
    
```

```

--util #工具类
  require.py #支持更高级导入方式

  --util #工具类
    AndroidUtils.py
    common.py
    decorator.py
    geckodriver.log
    httpUtils.py
    SendMail.py
    __init__.py

--utx #报告系统
  BSTestRunner.py #魔法字典类
  case_tag.py #测试用例标签
  log.py #封装日志
  runner.py #封装unittest
  setting.py #数据启动
  template.html #前端
  utx.py #utx的工具类
  __init__.py

--test_change.py #监听新包，然后修改任务列表
--test_listen.py #监听任务列表，开始跑测试用例
--test_keep.py #守护进程，只用跑这个文件就行，会自动去调用test_change.py和te
t_listen.py
    
```

测试匠心

测试效率

测试自动化

详细数据

测试类: ----- 只看: ----- 用例数: 74 | 成功: 46 | 失败: 28 | 跳过: 0

编号	用例描述	测试类	测试方法	结果	日志
1	0:计算启动时间	testcase.lcm.test_1_get_permission.LcmTests	test_00001_appStart	成功	
2	3.1.1:检查能看出用户隐私协议	testcase.lcm.test_1_get_permission.LcmTests	test_00002_get_protocol	成功	
3	3.1.3:不勾选统一条款时的情况	testcase.lcm.test_1_get_permission.LcmTests	test_00003_no_agreement_enter	成功	
4	3.1.4:未同意隐私条款情况下重新启动sample	testcase.lcm.test_1_get_permission.LcmTests	test_00004_no_agreement_restart	成功	
5	3.1.5:勾选统一条款后进入游戏	testcase.lcm.test_1_get_permission.LcmTests	test_00005_agreement_enter	成功	
6	3.1.6:同意隐私条款并进入游戏后重新启动sample	testcase.lcm.test_1_get_permission.LcmTests	test_00006_restart_again	成功	
7	3.1.8:同意条款后进入游戏,把sample删除重装	testcase.lcm.test_1_get_permission.LcmTests	test_00007_reset	成功	
8	1.1:检查登录界面UI	testcase.lcm.test_1_mobile_login.LcmTests	test_00008_login_screen	成功	
9	1.1.1:输入不符合标准的手机号并点击获取验证码	testcase.lcm.test_1_mobile_login.LcmTests	test_00009_wrong_number	成功	
10	1.1.2:输入一串符号并点击获取验证码	testcase.lcm.test_1_mobile_login.LcmTests	test_00010_symbol_number	成功	
11	1.1.3:不输入任何内容并点击获取验证码	testcase.lcm.test_1_mobile_login.LcmTests	test_00011_empty_number	失败	收缩

```

Traceback (most recent call last):
  File "/Users/ziang.chen/test/ste_test/automation_1/my_test/utx/utx.py", line 96, in __handler
    result = func(*args, **kwargs)
  File "/Users/ziang.chen/test/ste_test/automation_1/my_test/testcase/lcm/test_1_mobile_login.py", line 134, in test_empty_number
    self.assertTrue(len(msg) == 0, '\n'.join(msg))
AssertionError: False is not true : Toast信息有误! 期望值为: 手机号不能为空, 出现的是:
    
```

12	1.1.4:输入数字少于10位	testcase.lcm.test_1_mobile_login.LcmTests	test_00012_less_number	成功	
13	1.1.5:输入数字多于10位	testcase.lcm.test_1_mobile_login.LcmTests	test_00013_more_number	成功	
14	1.1.6:输入正确的手机号码、错误的验证码	testcase.lcm.test_1_mobile_login.LcmTests	test_00014_wrong_ver	成功	
15	1.1.7:输入正确的手机号码,不输入验证码并点击登录	testcase.lcm.test_1_mobile_login.LcmTests	test_00015_empty_ver	成功	
16	1.1.11:输入小于6位的验证码	testcase.lcm.test_1_mobile_login.LcmTests	test_00016_less_ver	失败	展开
17	1.1.12:输入大于6位的验证码	testcase.lcm.test_1_mobile_login.LcmTests	test_00017_more_ver	失败	展开
18	1.1.19:点击手机号登录框左上角返回按键	testcase.lcm.test_1_mobile_login.LcmTests	test_00018_back	成功	
19	1.2.1:选择登录方式界面点击快速登录	testcase.lcm.test_1_quick_login.LcmTests	test_00019_quick_login	成功	

## 测试报告样例

发大会

报告汇总

设备型号: {系统: '7.0', 品牌: 'samsung'}

启动时间: 0:00:00.230973

用例名称: 自动化测试报告

用例总数: 74

用例通过: 46

用例失败: 28

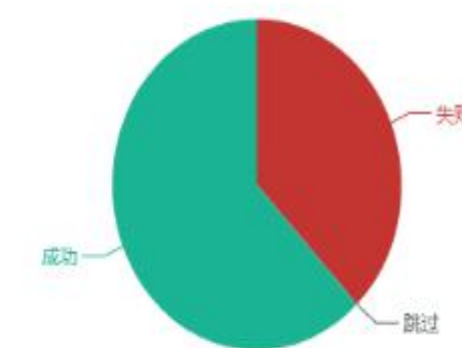
用例跳过: 0

开始时间: 2018-04-24 18:07:11

运行时间: 0:14:16.649476

失败  
跳过  
成功

测试用例运行结果



测试匠心

测试效率

平衡性测试

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1						team1	team2	team3	team4	team5	team6	team7	team8	team9
2	大熊	罗	雏菊	白星	萨波	6:0:4	10:0:0	0:0:10	3:0:7	0:0:10	0:0:10	0:0:9	0:0:10	0:0:10
3	大熊	黑胡子	雏菊	白星	萨波	1:0:9	2:0:8	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10
4	戈普	豪格巴克	突破后的奈美	佳妮法	白星	9:0:1	10:0:0	5:0:5	3:0:7	3:0:7	7:1:2	7:0:3	8:0:2	7:0:3
5	戈普	豪格巴克	突破后的奈美	佳妮法	毒Q	10:0:0	10:0:0	7:0:3	4:0:6	8:0:2	9:0:1	10:0:0	9:1:0	10:0:0
6	戈普	突破后的路	闪电	佳妮法	毒Q	10:0:0	10:0:0	1:1:8	3:0:7	7:0:2	8:0:2	0:0:10	0:0:10	0:0:10
7	戈普	豪格巴克	闪电	佳妮法	毒Q	10:0:0	10:0:0	4:0:6	2:0:8	10:0:0	9:0:1	0:0:10	0:0:10	10:0:0
8	戈普	闪电	突破后的撒	佳妮法	毒Q	10:0:0	10:0:0	1:0:9	0:0:10	9:0:1	0:0:10	1:0:9	0:0:10	0:0:10
9	戈普	闪电	突破后的撒	突破后的乔巴	佳妮法	10:0:0	10:0:0	3:0:7	1:1:8	1:0:9	0:0:10	0:0:10	9:0:1	10:0:0
10	戈普	豪格巴克	闪电	突破后的撒	佳妮法	10:0:0	10:0:0	1:0:9	1:0:9	0:0:10	10:0:0	10:0:0	0:0:10	0:0:10
11	战国	闪电	黑胡子	雏菊	白星	9:0:1	10:0:0	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10
12	战国	雷利	白胡子	闪电	突破后的乔巴	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10
13	撒谎布	阿龙	战国	闪电	突破后的奈美	4:1:5	10:0:0	0:1:9	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10
14	杰克斯	闪电	戈普	大熊	雏菊	3:0:7	10:0:0	9:1:0	10:0:0	0:0:10	1:0:9	6:0:4	6:0:4	7:0:3
15	杰克斯	青雉	闪电	雏菊	毒Q	8:1:1	7:1:2	5:1:4	2:0:8	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10
16	闪电	雷利	突破后的路	突破后的乔巴	佳妮法	10:0:0	10:0:0	1:2:7	1:0:9	0:1:9	0:0:10	0:1:9	0:0:10	0:0:10
17	杰克斯	闪电	突破后的撒	佳妮法	毒Q	10:0:0	10:0:0	5:0:5	4:0:6	9:1:0	0:1:9	10:0:0	10:0:0	10:0:0
18	闪电	鳄鱼	突破后的路	雏菊	毒Q	10:0:0	10:0:0	4:0:6	1:0:9	0:1:9	1:1:8	3:0:7	1:2:7	4:1:5
19	杰克斯	闪电	马尔高	突破后的路	突破后的乔巴	2:8:0	10:0:0	2:1:7	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10
20	杰克斯	鳄鱼	闪电	突破后的乔巴	佳妮法	10:0:0	10:0:0	5:0:5	1:2:7	8:0:2	2:0:8	9:0:1	10:0:0	10:0:0
21	杰克斯	闪电	雏菊	突破后的乔巴	佳妮法	10:0:0	10:0:0	2:3:5	4:2:4	4:2:4	1:1:8	8:0:2	10:0:0	9:0:1
22	杰克斯	戈普	闪电	佳妮法	毒Q	10:0:0	10:0:0	4:0:6	4:0:6	1:0:9	10:0:0	6:0:4	0:1:9	1:0:9
23	杰克斯	青雉	闪电	雏菊	佳妮法	7:0:3	9:1:0	7:0:3	5:0:5	1:0:9	2:0:8	3:2:5	4:1:5	3:0:7
24	闪电	雷利	萨波	旺·奥加	新世界·佐罗	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10	0:0:10
25	闪电	戈普	大熊	突破后的路	突破后的艾斯	6:0:4	10:0:0	3:0:7	3:0:7	0:0:10	0:0:10	2:0:8	0:0:10	2:0:8
26	杰克斯	青雉	闪电	突破后的乔巴	佳妮法	10:0:0	10:0:0	10:0:0	9:0:1	8:0:2	4:0:6	4:0:6	5:1:4	5:0:5

测试报告样例

## 游戏审核标准

坚守底线，QA SIGN OFF制度

## 游戏审核标准

## 必须通过项目

游戏业务功能测试	FAIL	检查redmine/teambition上游戏bug的状态 高优先级的BUG 0个，普通优先级的BUG低于(CBT50个,OBT 20个)
服务器接口脆弱性测试	FAIL	<a href="#">见说明页【服务器接口脆弱性测试说明】</a>
客户端闪退率分析	FAIL	是否接入了崩溃收集SDK(推荐腾讯bugly或者网易云捕) 用户crash率必须低于5%，次数crash率必须低于3%。
服务器验证和客户端加密混淆	FAIL	<a href="#">见说明页【服务器验证和客户端加密混淆说明】</a>
客户端帧率和界面切换时间	FAIL	<a href="#">1,2,3档机标准见【客户端基线】页</a>
		1档机型以上机型 在默认画质配置下，核心游戏场景默认中要求90%数值都不低于25FPS；常用界面间切换时间不高于1秒
		2档机型以上机型 在默认画质配置下，核心游戏场景默认中要求90%数值都不低于25FPS；常用界面间切换时间不高于1秒
3档机型 在默认画质配置下，核心游戏场景默认中要求90%数值都不低于18FPS。常用界面间切换时间不高于2秒		
游戏自更新能力测试	FAIL	<a href="#">见说明页【游戏自更新能力测试说明】</a>
服务器压力测试	FAIL	<a href="#">见说明页【服务器压力测试】</a>
手机机型适配测试 (testin/testbird/wetest)	FAIL	TOP300机型、基线以上机型适配通过率>=95%；1档机型通过率>=99%
IP检查	FAIL	涉及IP的资源全部通过监修
合规测试	FAIL	通过发行地区的法律法规检查

游戏审核标准

技术打分项目

## 热更新特别注意CDN

- CDN的问题可能带来的问题
- 热更新下载策略
- 热更新打包方式

MTSC2018  
第四届中国移动互联网测试开发大会

## 游戏审核标准

## 技术打分项目

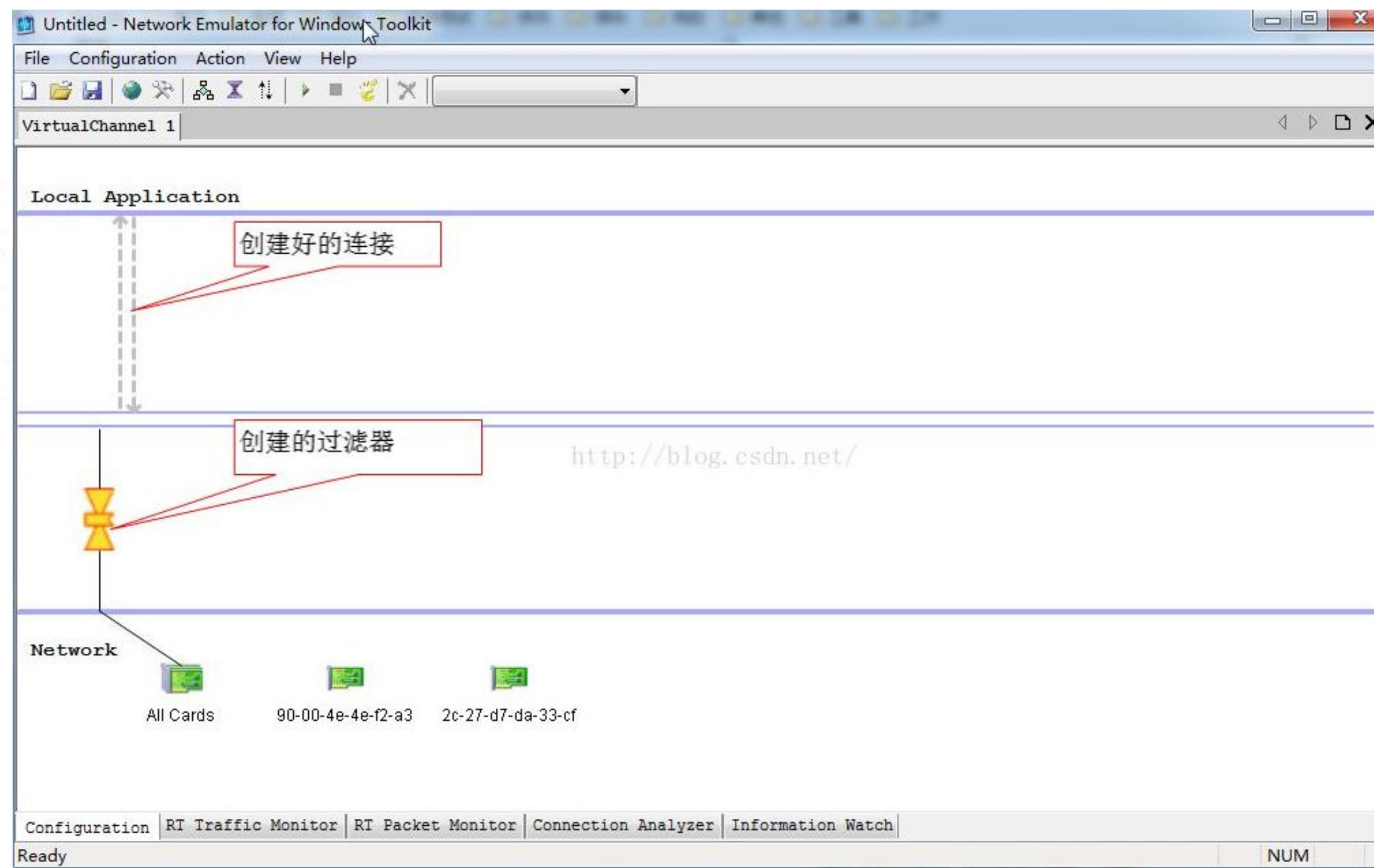
客户端内存大小			Android平台：在对应档次客户端最低配置以上，均需满足以下内存消耗指标（PSS）：
	2	2	1档机型指标：最高PSS<=550MB
	2	2	2档机型指标：最高PSS<=450MB
	0	2	3档机型指标：最高PSS<=350MB
			iOS平台：在对应档次客户端最低配置以上，均需满足以下内存消耗指标（real mem）：
	2	2	1档机型指标：消耗内存（real mem）不大于500MB
客户端包体大小	0	2	2档机型指标：消耗内存（real mem）不大于400MB
	0	2	3档机型指标：消耗内存（real mem）不大于360MB
	3	3	Android平台：不大于700MB；iOS平台不大于1500MB
客户端游戏流量			游戏核心玩法流量消耗情况（非一次性消耗）应满足以下条件：
	3	3	1.对于分局的游戏场景，单局消耗流量不超过 1MB
	3	3	2.对于不分局游戏场景或流量与局时有关的场景，10分钟消耗流量不超过3MB
客户端耗电	5	5	耗电比例15%-20%之间打分
启动加载时间	3	3	加载时间15-20秒之间打分
断线重连、游戏挂起处理能力			<a href="#">见说明页【断线重连测试说明】</a>
			客户端的核心场景必须有断线重连机制，并在有网络抖动、延时、丢包的场景下，客户端需达到以下要求：
			一. 不能出现以下现象：
	5	5	1、游戏中不能出现收支不等、客户端卡死/崩溃等异常情况；
	2	2	2、游戏核心功能（如登录、单局、支付等）不能有导致游戏无法正常进行的UI、交互问题；
	8	8	3、不能有损害玩家利益或可被玩家额外获利的问题；
		4、需要有合理的重连机制，避免每次重连都返回到登录界面。	
	1	1	二. 需要对延时的情况有相应的提示
客户端安全	1	1	1. 游戏本地和下载的数据需要加密
	3	3	2. 游戏编译完的文件需要将函数名和字符串抹除
	3	3	3. 游戏客户端服务器协议需要加密，密钥在每一次登录期间要变化
	5	5	4. 游戏内使用的脚本语言需要加密
客户端CPU占用率	3	3	Android平台：CPU占用(90%)小于60% iOS平台：CPU占用(90%)小于80%

游戏审核标准

技术打分项目

## 弱网测试规则

- 弱网不等于限速
- UI表现符合用户的预期
- 不可让用户反复登录
- 防止消息被处理多遍
- 常用的工具





## 游戏审核标准

## 内容打分项目

- 主观性的打分项目建议让多人打分后取平均分。并且打分人是不同角度的。建议叫上运营一起打分。
- 将各种游戏打分结果积累起来分门别类。

玩家目标	1	2	玩家每日上线是否有明确的目标
UI操作及布局	3	5	UI布局美观，操作便捷，操作逻辑易懂
核心战斗玩法	2	5	玩法新意程度
核心战斗表现	5	5	打击感或AI表现
核心战斗节奏	2	2	操作疲劳度、时间
新手引导	6	8	新手引导时间长度是否合适，操作教学交代是否简洁易懂，是否有未告知的功能点
数值架构合理性	7	8	任务经验断档、升级速度、人物属性和副本难度的匹配等

## 常用工具

借助巨人的肩膀做事情

常用工具


客户端闪退



常用工具

兼容性测试

## 应用信息

	应用名称	美职篮梦之队3	应用版本	0.2.1	应用大小	142.63 MB	提测时间	2017-12-18 17:21:19
	测试类型	深度兼容测试	完成终端数	300 款	覆盖活跃用户数	20628.6万	测试状态	测试完成

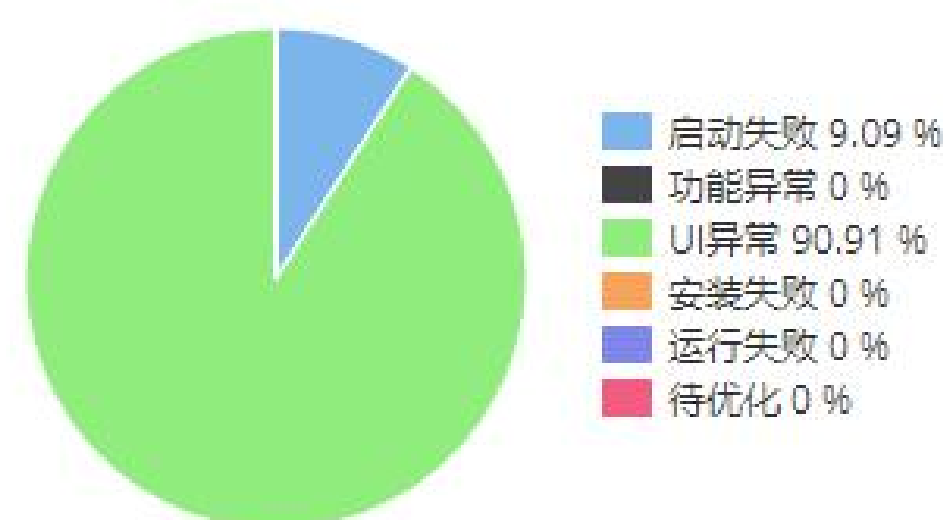
## 兼容概况



通过机型/完成机型总数



根据通过率在同行业的排名计算



不兼容机型的分布问题

Testin

TestBird

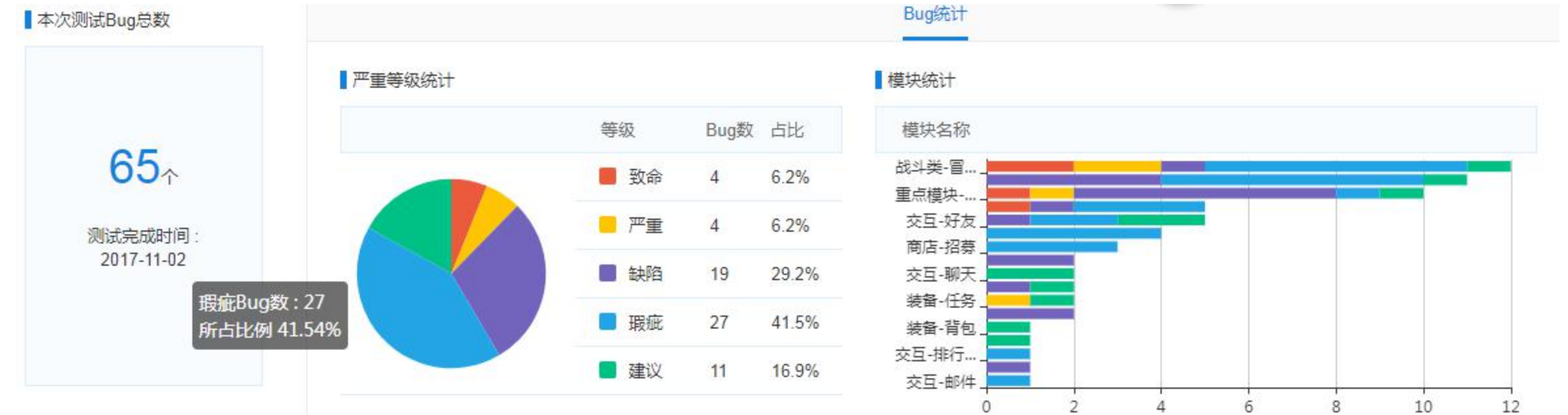
WeTest  
腾讯质量开放平台

常用工具

探索性测试

## 探索性测试的适用场景

- 测试时间紧急，测试人员不足
- 测试旧功能占多数
- 被测试功能已经测试了多遍



868980 [崩溃问题] [重点模块-新手引导] 我叫黑崎一护胜利升级, 后台运行... 专家评级 致命 您的评价: [点击置评](#)

重复Bug

ID	测试设备	网络	复现率	查看附件
868257	三星 SM-C9000 (Android - 6.0)	WiFi	100.00%	<a href="#">查看附件</a>
868268	索尼 E6683 (Android - 7.1.1)	WiFi	100.00%	<a href="#">查看附件</a>
868525	小米 小米6 (Android - 7.1.1)	WiFi	100.00%	<a href="#">查看附件</a>
869203	索尼 E6683 (Android - 7.1.1)	WiFi	100.00%	<a href="#">查看附件</a>
869226	索尼 E6683 (Android - 7.1.1)	WiFi	100.00%	<a href="#">查看附件</a>
869239	索尼 E6683 (Android - 7.1.1)	WiFi	100.00%	<a href="#">查看附件</a>
870213	魅族 MX6 (Android - 6.0)	WiFi	100.00%	<a href="#">查看附件</a>

Bug评语:

点击写下对此Bug的评语, 以便专家团队为您提供更加贴切的服务

测试设备: 小米 红米note 4X (Android - 6.0) 网络: WiFi 复现率: 100.00%

Bug标题: 我叫黑崎一护胜利升级, 后台运行再次进入APP, 点击跳过出现黑屏

前提条件: 由新手指导进入的挑战

测试步骤: 1. 进入游戏->新手指导-挑战->我叫黑崎一护  
2. 结束战斗胜利升级  
3. 后台运行, 再次进入APP  
4. 出现对话, 点击跳过

期望结果: 显示画面

实际结果: 胜利后点击跳过, 出现黑屏

备注: 无

附件信息:

我叫黑崎一护胜利升级... 我叫黑崎一护胜利升级... [点击下载](#)

重复Bug

## 常用工具

## 服务器压力测试



### 压力测试目录结构解析

- Config
- Script
- Modules

```
app.js //main文件
package.json //依赖
startServer.sh //驱动main文件
//数据驱动
app
  config
    env.json //配置执行script入口业务代码
  optest
    config.json //压测服务本地配置(固定)
    server.json //压测服务host.ip(每次修改)
  acc
    op_acc.json //虚拟用户账号数据
  script
    optest //对口项目请求的业务代码
      api_test.js
      demo.js
      logintest.js
      op_robot.js
  log
    .log //存储日志文件夹
  node_modules //全局npm模块依赖区域
    async //异步函数库
    body-parser //http族请求体解析中间件
    bytes //处理bytes解析
    content-type //请求头解析
    debug //调式模块
```

```
dena-client //客户端处理模块
  index.js
  optest-client //对口项目
    index.js
  lib //处理c->s协议代码
    client.js
    client备份.js
    ope.js
  node_modules //客户端处理模块npm
    request //处理返回请求npm+打包其他库
dena-robot //处理调度和前端
  index.js
  package.json
  lib
    robot.js //代理进程通信的
      agent
        actor.js
        agent.js
    common //日志&工具类
      logging.js
      util.js
    console //前端界面(css,图片+UI.js)
      css
      images
      js
        lib
        ui
    naster //调度前后端交互
      nodeclient.js
      server.js
      starter.js
      uebclient.js
    monitor //监控性能模块
      monitor.js
      stat.js
  node_modules //网络层模块(包含解析各种数据格式的)
    socket.io
    socket.io.client
```

## 常用工具

## 服务器压力测试

**Config**

Per Client Agents:  Per Agent Users:

Select Run Script:  ReReady RUNNING

Current Agents:  
[127.0.0.1:8888] [127.0.0.1:8888] [127.0.0.1:8888] [127.0.0.1:8888] [127.0.0.1:8888]

---

**Summary**

Request	Max	Min	[0-100]	(100-500]	(500-]	<500(%)	Avg	Qps	Count
QuickPlayPve	4606	13	9428846	150894	21983	1.00	32	155	9601723
ClaimFightPveChapter	3592	6	5719104	40204	5480	1.00	15	333	5764788
ClaimFightPveChapterVerify	4954	17	5642200	108157	13953	1.00	38	130	5764310
Gacha	3108	8	1897540	18951	2527	1.00	21	233	1919018
GetUserNiudanList	3081	2	1904198	12773	1898	1.00	9	531	1918869
GetNiudanVipList	3320	2	1905392	11646	1699	1.00	9	553	1918737
GetDominnerTrainingList	2704	7	380632	3317	512	1.00	18	281	384461
GetMessageCenterList	2976	13	381500	2591	344	1.00	22	232	384435
GetAchievement	3418	3	381437	2622	369	1.00	11	470	384428
ClaimRandomDomineerTraining	2729	18	377565	6307	529	1.00	51	98	384401
VerifyDomineerTraining	2745	13	378087	5517	761	1.00	30	166	384365
GetGemFightGuankaList	3609	5	380445	3126	403	1.00	16	306	383974
ResetGemChapterGuankaList	3121	25	376922	6354	675	1.00	45	112	383951
ClaimFightChapterGuanka	1875	6	380091	3349	479	1.00	17	288	383919
ResetBlackMarketItemList	1996	9	379527	3816	549	1.00	21	233	383892
FightGemChapterGuankaVerify	3897	15	378082	5211	591	1.00	33	150	383884
GetBlackMatrktItemList	2144	3	380342	3036	475	1.00	12	401	383853
BuyBlackMarketItem	2897	9	379499	3867	460	1.00	22	226	383826
GetPvePrisonList	3285	3	380356	2603	454	1.00	11	456	383413
GetPvePrisonMonster	1888	2	380211	2739	435	1.00	11	461	383385
ClaimFightPvePersionGuanka	2786	6	380032	2923	398	1.00	16	306	383353
FightPvePrisonGuankaVerify	2444	7	378905	3913	499	1.00	19	258	383317
GetItemPackBuyHist	2239	2	380473	2071	268	1.00	9	585	382812
CheckTname	220	6	2546	4	0	1.00	11	448	2550
LoginMobageAccessToken	115	11	2548	2	0	1.00	15	332	2550



- 密集访问的单接口压力测试
- 基于真实数据的压力测试混合模型
- 持续72小时以上

常用工具

服务器压力测试

- 游戏测试循环有紧密的上下文联系
- 基于真实数据的压力测试混合模型

```
if (!!robot.caseData.previous) {
  robot.caseData.previous = false;
  robot.funcseries = [];

  if (robot.caseData.ratemode) {
    robot.funcmap = {
      pve_battle: {funcArray: [GetterDataBattle, StoryBattleStart, StoryBattleResult, StoryBattleFast],
        Gacha: {funcArray: [Gacha], rate: 5},
        TiGuan: {
          funcArray: [LadderRefresh, LadderBattleStart, LadderBattleResult],
          rate: 5
        },
        GetterSort: {funcArray: [GetterSort], rate: 10},
        GetterActive: {funcArray: [GetterActive], rate: 30},
        GetterDaily: {funcArray: [GetterDaily], rate: 30},
        GetterItem: {funcArray: [GetterItem], rate: 20},
        GetterRecord: {funcArray: [GetterRecord], rate: 30},
        GetterSeason: {funcArray: [GetterSeason], rate: 30}
    };

    robot.willTest = [];
```



常用工具

服务器压力测试

- 混合模式的业务搭配
- 异步控制序列化

```
robot.funcArray.forEach(function (element, index, array) {
    var func = function (callback) {
        setTimeout(element, robot.randomIntTime(), callback);
    };
    robot.funcseries.push(func)
});
// 竞技场
robot.funcseries.push(Login_pvpServer, ArenaQueueJoin, wait_S, Pvp_Begin_To_End, connect_battleServer, sen
// 东西部争霸赛
robot.funcseries.push(Login_pvpServer_EW, ArenaQueueJoin_EW, wait_S_EW, Pvp_Begin_To_End_EW, connect_battl

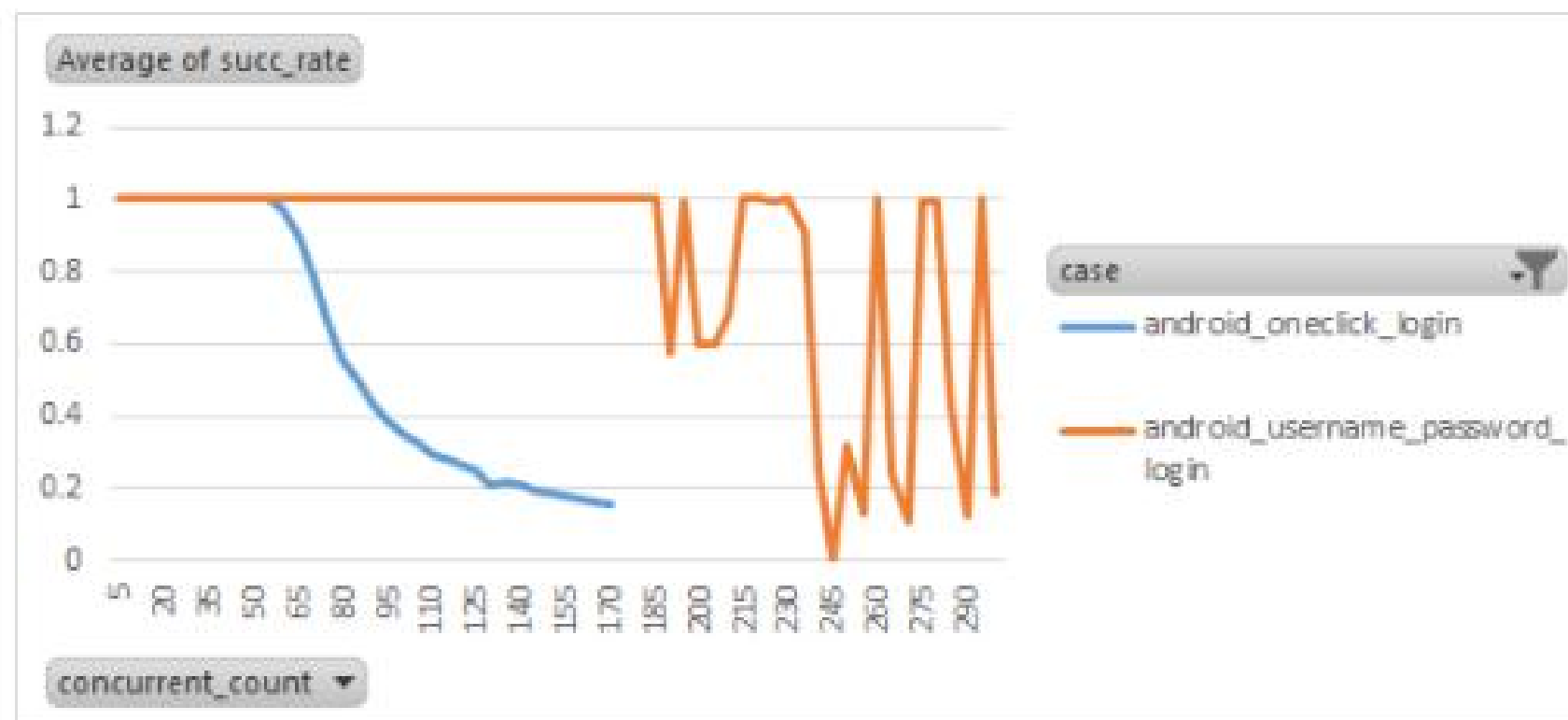
async.series(robot.funcseries,
    function (err) {
        if (err) {
            console.log("Error====>", err);
        } else {
            robot.caseData.previous = true;
        }
    }
);
}
```

## 常用工具

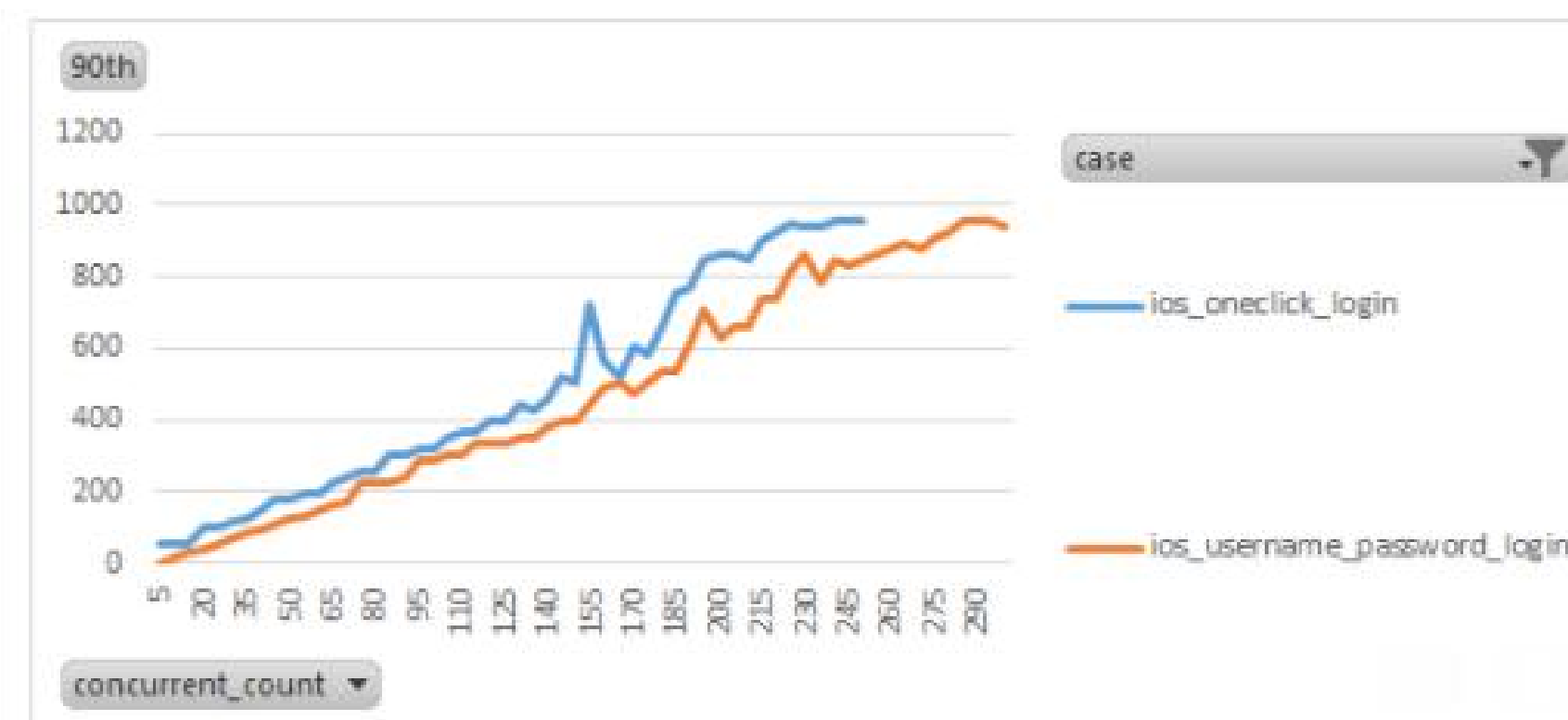
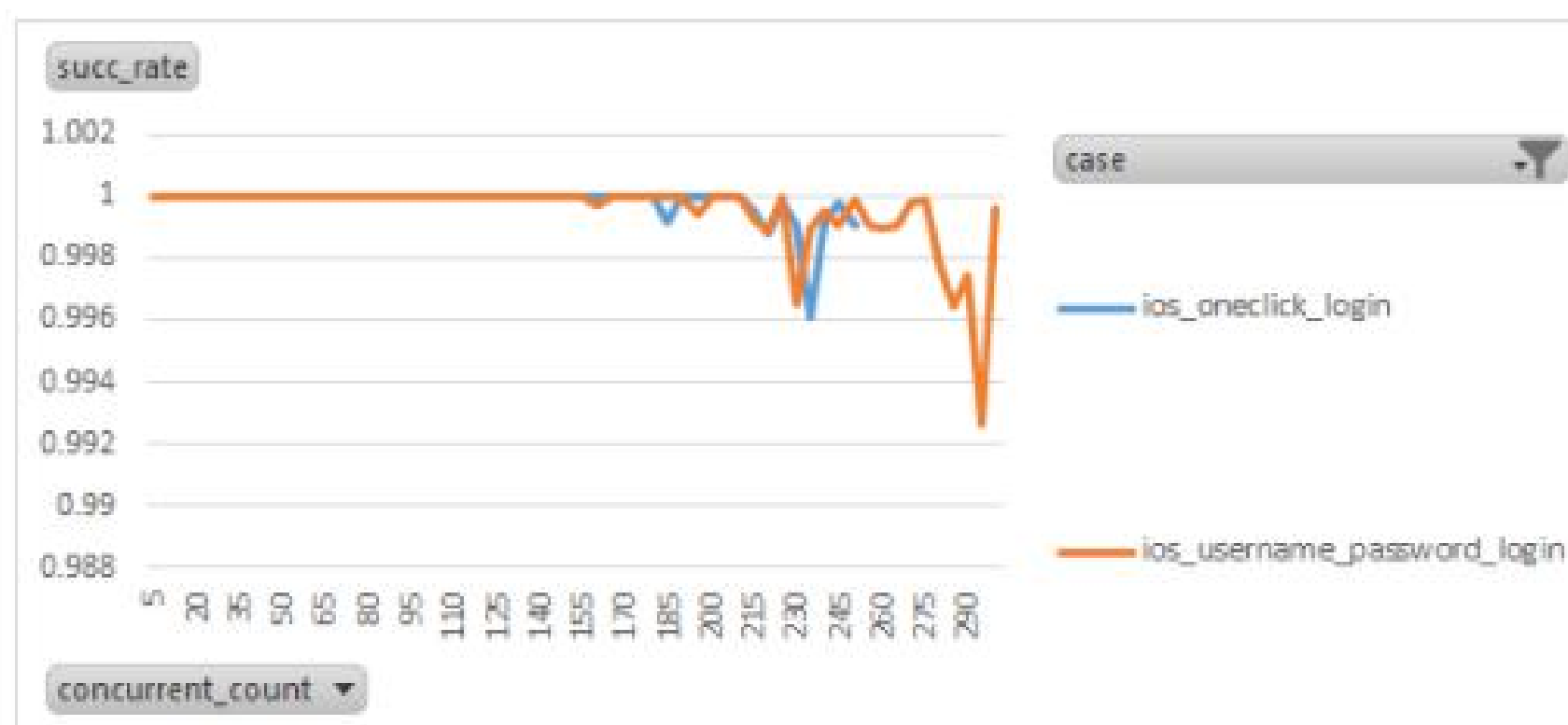
## 服务器压力测试

压力测试承载量、稳定性的同时，还要注意性能的衰减

### • android平台



### • ios平台



## 常用工具

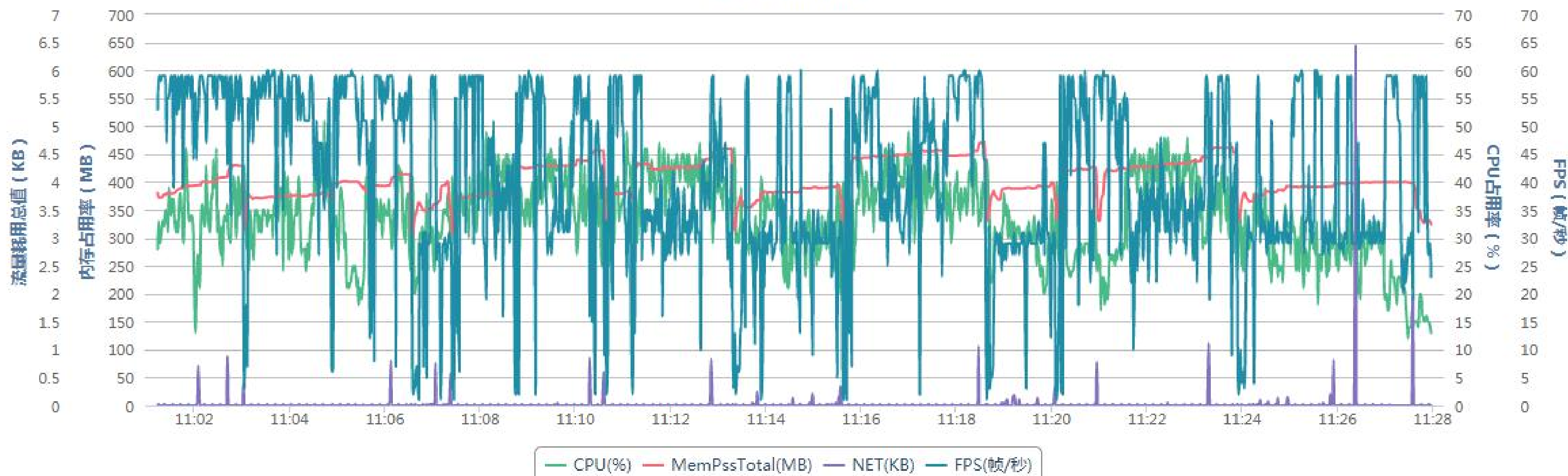
## 客户端性能测试

WeTest UPA

- 操作简单，适合做初步筛查

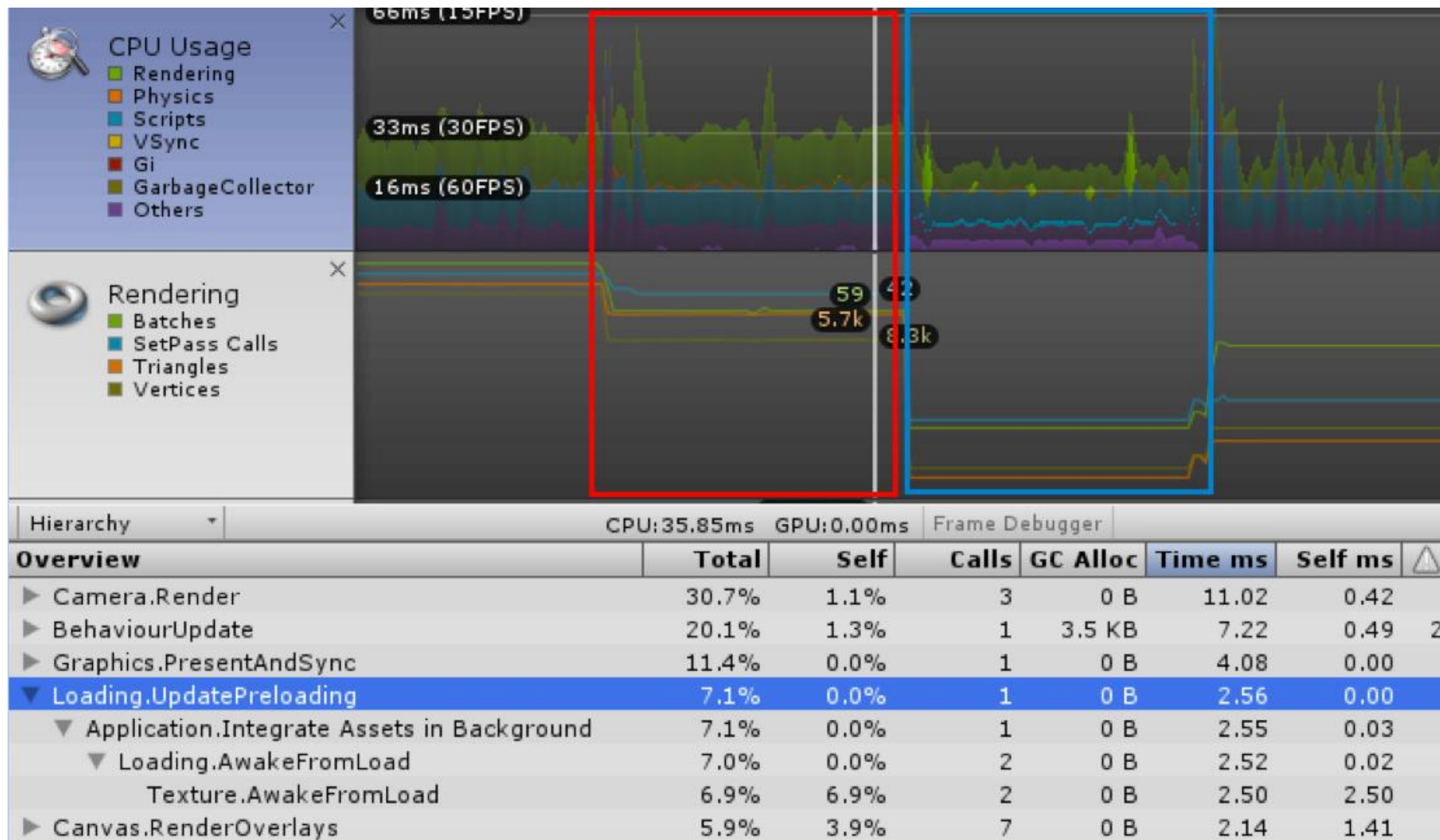


性能数据时间轴



## 常用工具

## 客户端性能测试



<https://blog.uwa4d.com/archives/2006.html>

## 常用工具

## 脆弱性测试

## 脆弱性诊断

- 协议测试优先
- 除了修改包体包含的参数外，还要注意隐含的参数
- 制定好测试规则，防止遗漏
- 既要检查修改发包，又要检查修改回包

G. 未攻略ステージのskip(危険度：Medium - リリースブロックとはならないが対応推奨または不正の監視が必要)	未通關的关卡可以直接扫荡
■概要 スキップ機能で未攻略のステージを指定してもスキップを行い報酬を得ることができません。	国服已修复
■再現方法 該当機能：スキップ機能 リクエスト例： 14 07 2C 00 00 05 10 D9 4F 18 08 リクエスト改ざん例： 14 07 DC 00 00 05 10 A1 51 18 08 上記の様にステージに該当する箇所を10201(D9 4F)から、未攻略の10401(A1 51)に行っても処理され報酬が入手できました。	
■対策 未攻略のステージを指定された場合はエラーとしてください。	

常用工具

脆弱性测试

脆弱性诊断



- 内存修改类
- CHARLES FIDDLER
- WPE



常用工具

脆弱性测试

代码加固

- 应对渠道包的打包策略
- 优化加固和打包速度
- 可以与打包后台对接
- DLL中的函数名一定要抽掉
- iOS防护
- 做充分地兼容性测试

梆梆安全  
BANGCLE

爱加密



果核数位



网易云

易盾

INCA  
Internet China

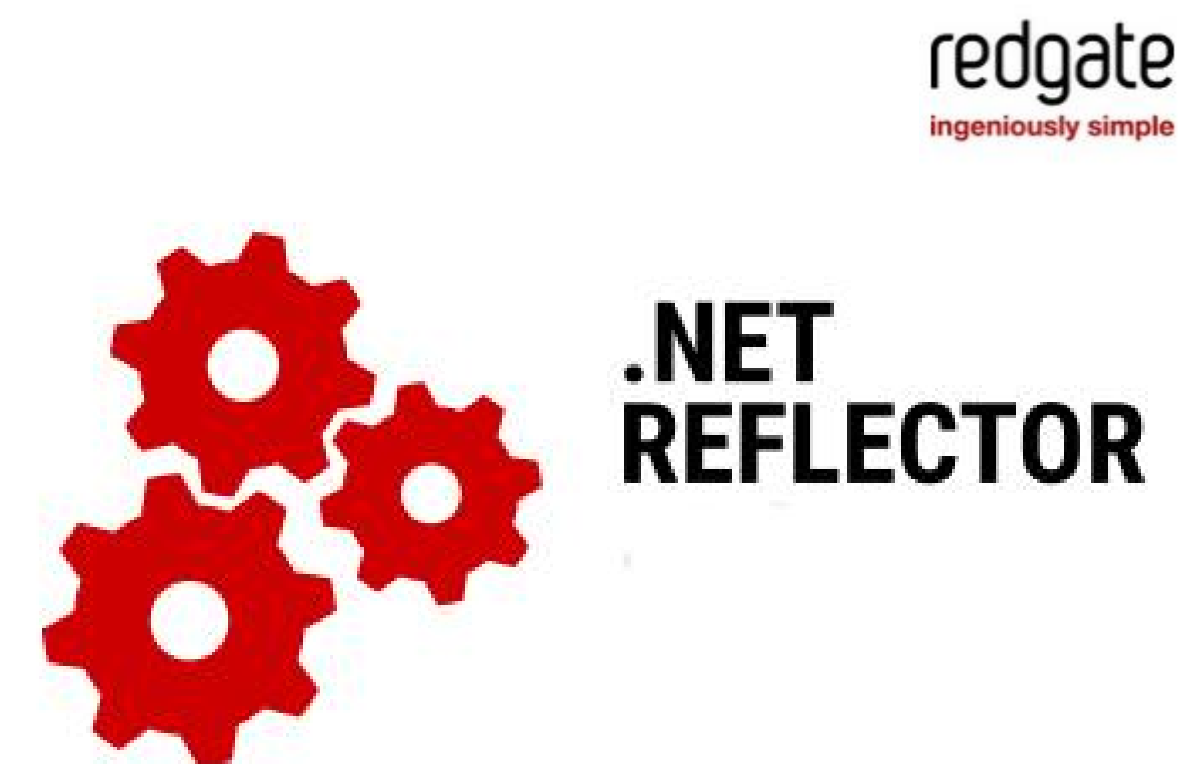
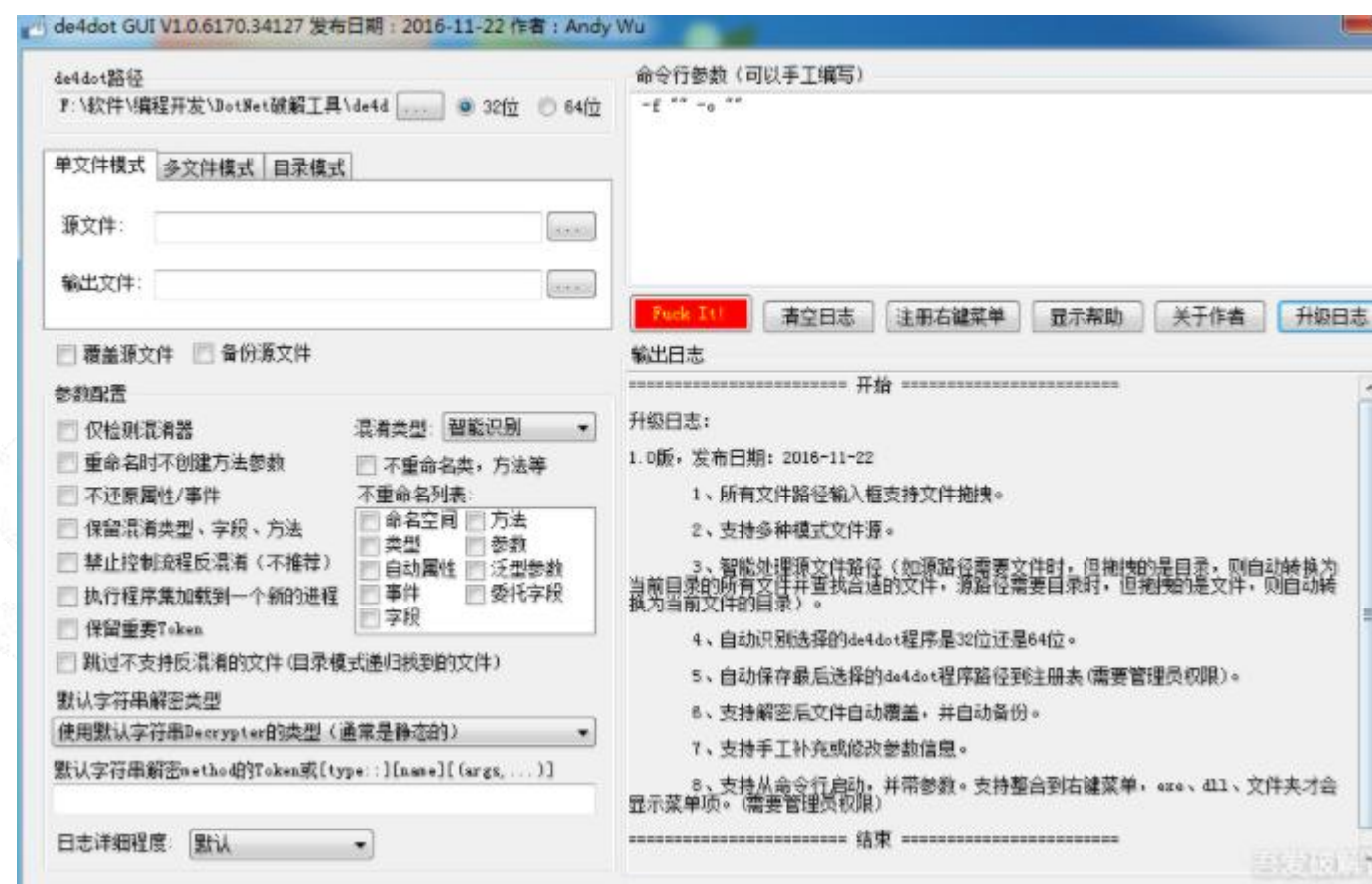
常用工具

脆弱性测试

代码加固

## 检查方式

- De4Dot
- Redgate Reflector
- IL ASM





## Q & A





MTSC2017

第三届中国移动互联网测试开

谢谢! TesterHome



# MTSC2018

第四届中国移动互联网测试开发大会

TesterHome

IT大咖说

# TesterHome