流畅度模型 阶段性总结报告

# 项目背景

在长期机型项目及各种系统性能优化项目的测试验证过程中，测试方法及测试评判标准一直是：单一的手工执行流畅度用例、响应时间启动时间等指标，以及最直观但不具备数据指标说服力的直观感受等，苦于流畅度测试除了FPS和掉帧数外，没有一项可以具体量化的测试数据、测试方法、测试用例。

因此，流畅度模型的建立，初衷是建立统一的测试方法、统一的测试用例、统一的评判标准。在同一套测试体系下，能够公平且有力的以数据说明的方式，直观的反映测试结果，整个系统的流畅度表现。

# 2 实际开发成果

## 2.1 流畅度模型 整体规划及流程图

### 2.1.1 流畅度评估模型整体规划

流畅度评估模型，由于需要以统一的测试步骤，输出统一评判标准，因此实现一套可重复可执行的自动化测试，以第三方应用的流畅度表现为主，定制少部分Flyme 系统应用，可用于不同Flyme 版本间的整体系统流畅度对比，也可用于竞品间的第三方应用的流畅度表现，以反映系统整体的流畅度情况。

流畅度模型的主要流程为：

1. 以appium测试框架作为自动化测试框架，进行UI自动化用例的开发， 已实现33条测试用例，第三方应用共23条，系统应用共10条；
2. 在进行自动化操作的过程中，多线程收集关键的性能数据，主要包括FPS、掉帧率、最大掉帧数、gfxinfo等一系列性能关键数据，用于后期数据分析。
3. 最终测试结果以HTML格式展示，展示内容为：测试用例对应的 关键性能数据，每帧耗时分布图，可反映该条用例的执行耗时。同时，测试报告提供版本间对比的功能。

### 2.1.2 流畅度评估模型 流程图

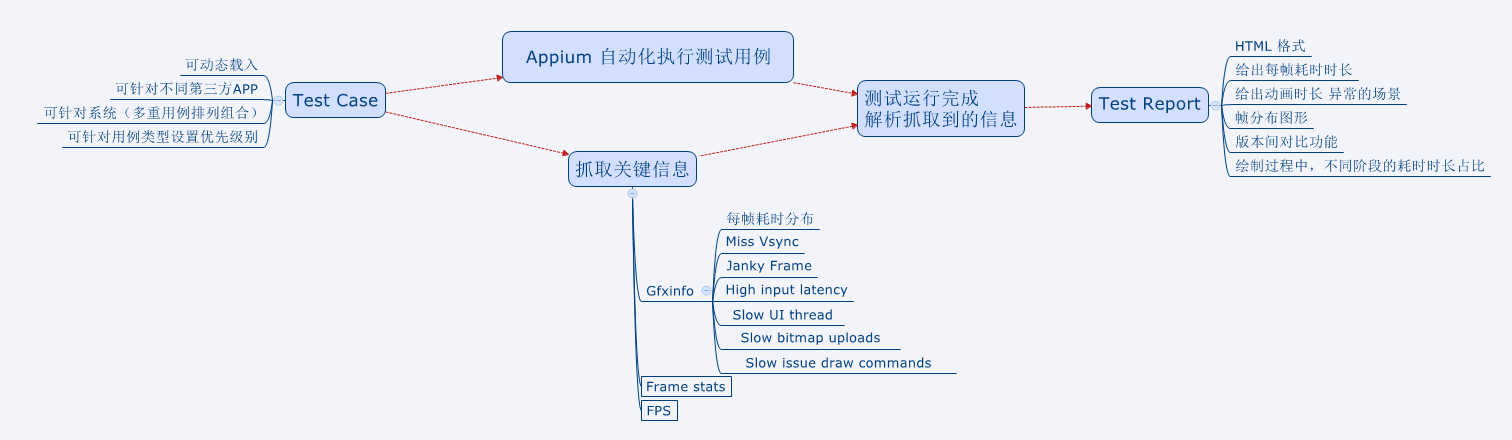


图1 流畅度评估模型流程图

## 2.2 主要功能及成果

### 2.2.1 可重复、跨平台执行UI自动化测试用例

目前已实现33条测试用例，第三方应用包含23条，系统应用共10条。

* 针对竞品机器，可针对执行第三方应用的测试用例；
* 针对Flyme版本间的相互对比，可以以系统应用的表现作为是否衰退的标准，同时以第三方应用的流畅度表现，作为系统整体的表现。
* 针对单个优化项目，可以以影响具体应用的绘制阶段的指标，作为效果验证的性能优化结果。
* 测试用例可动态载入，侧重测试需要关注的测试项

已实现的测试用例：

|  |  |  |
| --- | --- | --- |
| 模块 | 测试类型 | 测试用例 |
| 微信 | Scrolling | 快速来回滑动朋友圈 |
|  | 与朋友发送微信消息 |
| window动画 | 连续切换聊天对话窗口 |
| Scrolling | 来回滑动聊天记录界面 |
| Scrolling | 滑动公众号-文章列表 |
| Scrolling | 进入 我-相册，来回滑动自己的相册 |
| Scrolling | 滑动朋友列表 |
| Scrolling | 查看好友朋友圈 |
| window动画 | 查看聊天中的图片 |
| 微博 | Scrolling | 微博 首页向下滑动 |
| Scrolling | 微博 首页向上滑动 |
| window动画 | 查看微博详情中的图片 |
| 淘宝 | Scrolling | 淘宝 首页向下滑动 |
| Scrolling | 淘宝 首页向上滑动 |
| window动画 | 查看淘宝宝贝详情 |
| Scrolling | 向下滑动淘宝 宝贝详情页面浏览 |
| 知乎 | Scrolling | 知乎 首页向下滑动 |
| Scrolling | 知乎 首页向上滑动 |
| window动画 | 查看知乎任一回答 |
| Scrolling | 答案页面向下翻页 |
| 网易新闻 | Scrolling | 网易新闻 首页向下滑动 |
| Scrolling | 网易新闻 首页向上滑动 |
| window动画 | 进入-退出 查看新闻页面 |
| 今日头条 | Scrolling | 今日头条 首页向下滑动 |
| Scrolling | 今日头条 首页向上滑动 |
| 桌面 | Scrolling | 桌面来回滑动 |
| systemui | Scrolling | 下拉通知栏 |
| click | 点击展开WiFi列表 |
| Scrolling | 呼出任务管理器 |
| 设置 | Scrolling | 滑动设置选项列表 |
| Scrolling | 进入 无线网络，滑动WiFi列表 |
| 图库 | Scrolling | 进入图库应用，来回滑动浏览图库图片 |

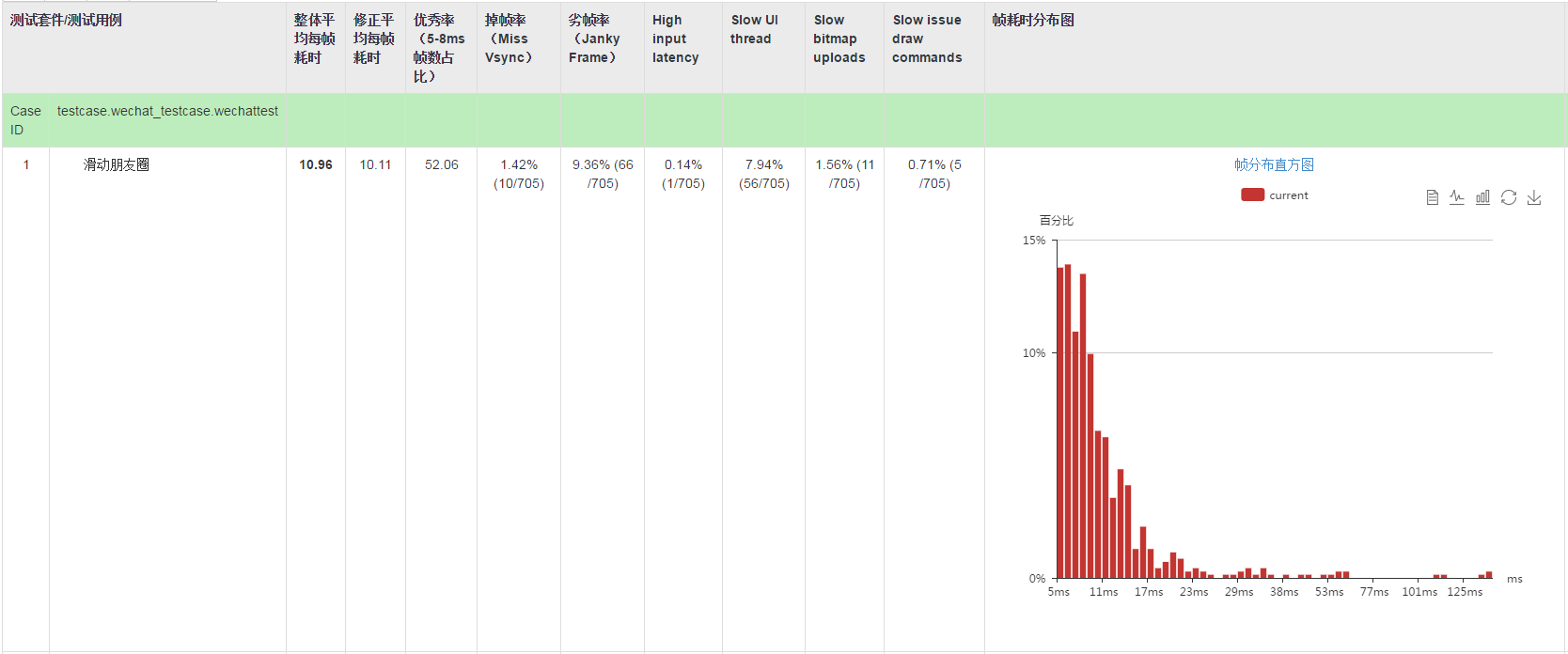
### 2.2.2 数据采集及数据展示

测试过程中同步采集测试过程中的数据，

关键数据来源：

1. adb shell dumpsys SurfaceFlinger –latency + ProcessName
2. adb shell dumpsys gfxinfo + ProcessName
3. adb shell dumpsys gfxinfo + ProcessName + framestats

经数据分析后，提取关键数据，最终展示效果如下：



### 2.2.3 打分规则制定及评判标准

经过组内会议评审，最终制定如下的评判指标。

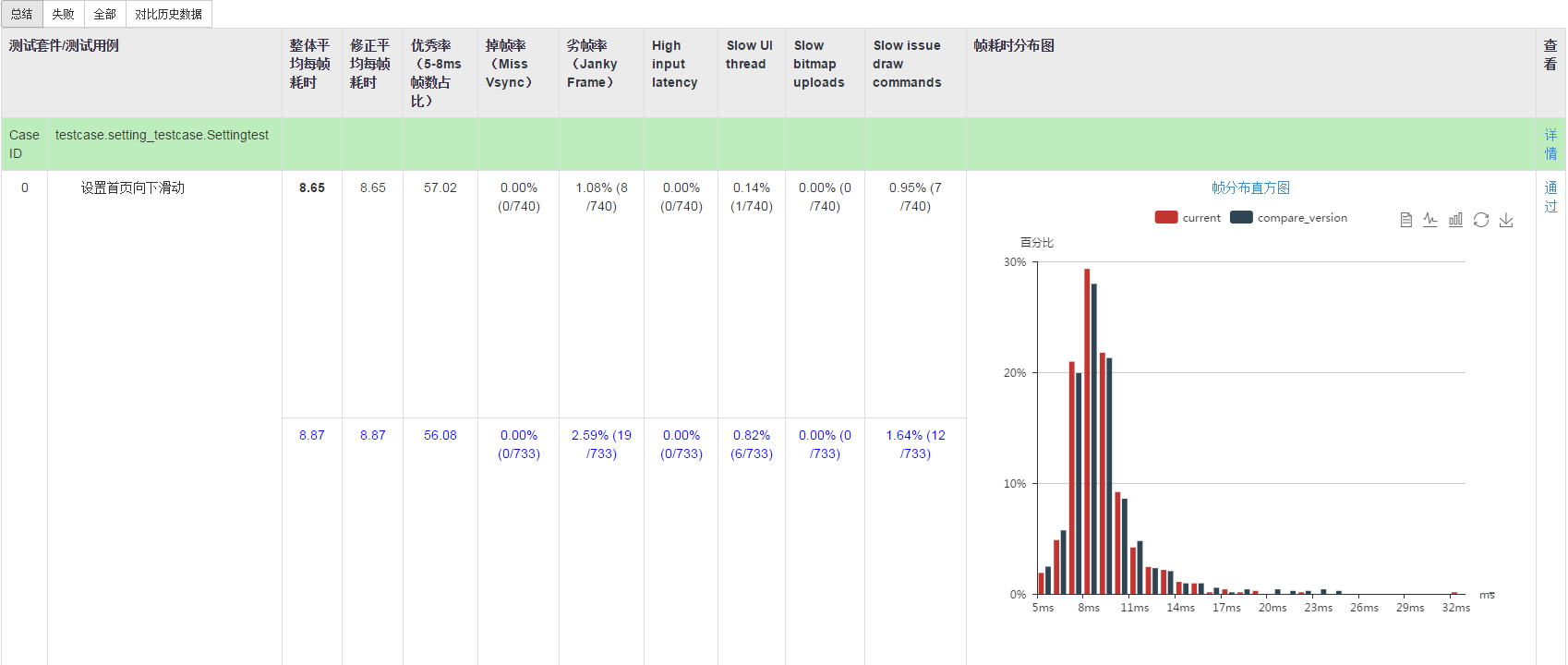
暂时未有一个总分标准，以不同维度的指标提升或衰退的比例作为评判该版本是否有提升。

|  |  |
| --- | --- |
| **指标** | **指标含义** |
| 整体平均每帧耗时 | 未做特殊处理的原始平均每帧耗时 |
| 修正平均每帧耗时 | 去掉可能的异常数据（>100ms）后，计算出的平均每帧耗时 |
| 优秀率 | 5-8ms帧数占比 |
| 掉帧率（Miss Vsync） | 应用miss vsync的次数，该指标可反映应用确实掉帧并导致卡顿 |
| 劣帧率 （Janky Frame） | 每帧时长超过16.6ms，但不一定造成可感官的卡顿 |
| High input latency | 具体Input命令超出Google标准，input延迟大于2ms的帧数占比 |
| Slow UI thread | 主线程measure layout ondraw耗时较长 |
| Slow bitmap uploads | 加载bitmap耗时的帧数占比 |
| Slow issue draw commands | 绘制命令耗时较长的帧数占比 |
| 每帧耗时分布图 | 根据 每帧耗时所占帧数/总绘制帧数 - 每帧耗时 绘制的图标 |

### 2.2.4 版本间对比功能

减少人工干预成本，加入测试报告对比功能。

一键导入后，各项数据的前后版本对比信息方便对比查看。同时，将前后两个版本中的帧分布图合并，更加直观的反映哪个版本的数据更加优秀。



# 3 项目验收成果

截止2-7为止，流畅度评估模型已在不同平台上进行实际使用。可以在不同的测试项中，利用流畅度评估模型的优势，评估出不同版本的流畅度性能表现差异。

主要利用在以下三种场景中：

* **竞品对比：**

1. Flyme 7 M1793 /一加 5T 的流畅度模型测试
2. 1891 对比 小米Mix 2 系统流畅度对比
3. 小米Mix 2/ 华为Mate 10/ 一加5T 流畅度模型用例输出对比数据

* **Flyme 版本间对比：**

1. 利用流畅度评估模型对比、输出 P10机型 M1621新春版本 对比 6.2.0.0A版本 流畅度提升效果
2. M1792在老化后，流畅度评估模型测试中发现老化过程中，cpu调频存在异常的问题
3. Flyme 6 对比Flyme 7，是否存在性能衰退

* **功能点/策略 流畅度效果验证：**

1. 利用流畅度评估模型，验证picker 在系统操作场景下对性能影响
2. M1712、M1891、M1621等机型， 不同绘制buffer数量，对系统流畅度表现的验证
3. 滑动冻结策略验证整体系统流畅度表现
4. 微信HC SDK集成后，系统流畅度验证
5. 验证TraceDump 在不同状态下对系统流畅度的影响

# 4 后续规划

1. 输出《流畅度评估模型-使用手册 v1.0》
2. 搭建多个测试环境，能多次、重复、循环执行流畅度评估模型
3. 增加窗口掉帧数据
4. 丰富版本对比功能，输出简单统计信息，两个版本间的大致优劣情况
5. 导出pdf功能
6. 同步丰富精简测试用例
7. 增加不同测试前提，CPU重负载情况、内存重负载情况、后台进程数重负载情况、IO重负载情况、存储系统重负载情况等等，可进一步挖掘
8. 丰富性能测试过程中的数据项收集，例如：CPU/GPU频率、CPU/GPU使用率、耗电指数、内存占用情况等等，进一步丰富成性能评估模型、性能功耗评估模型
9. 自动可持续集成，自动下发任务，自动上传测试报告，钉钉或邮件提醒结果接收方

备注：

此部分是个人对项目后期发展的初步计划，还未落地。后续会逐项评估可行性，并落实计划安排。