X\_Key\_Work\_TestTool\_PRS

版本修改历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 版本 | 修改人 | 描述 | 备注 |
| 2017-03-12 | V0.1 | 卢磊 | 初稿 |  |

目录

[X\_Key\_Work\_TestTool\_PRS 1](#_Toc477161456)

[1. 目标群体 4](#_Toc477161457)

[2. 实现策略 4](#_Toc477161458)

[2.1. Case 解析 4](#_Toc477161459)

[2.2. 生成代码（附加） 5](#_Toc477161460)

[2.3. Case管理 5](#_Toc477161461)

[2.3.1. Case基础管理 5](#_Toc477161462)

[2.3.2. 其他Case操作 5](#_Toc477161463)

[2.4. 执行进度 6](#_Toc477161464)

[2.5. 手机状态与错误信息 6](#_Toc477161465)

[2.5.1. 手机状态 6](#_Toc477161466)

[2.5.2. 错误信息 6](#_Toc477161467)

[3. 功能 6](#_Toc477161470)

[3.1. Case解析与管理 6](#_Toc477161471)

[3.1.1. Case组成格式 6](#_Toc477161472)

[3.1.2. Case管理 7](#_Toc477161473)

[3.1.3. Case解析 7](#_Toc477161474)

[3.2. 配置文件 8](#_Toc477161475)

[3.3. UI界面 8](#_Toc477161476)

[3.4. 报告 8](#_Toc477161477)

[3.4.1. 报告要求生成各种图表 8](#_Toc477161478)

[3.4.2. Summary 8](#_Toc477161479)

# 目标群体

方便手工测试人员编写自动化测试case，工具采用关键字驱动的测试框架，使测试人员不用接触代码就可以编写自动化case，同时采用导图的模式管理Case，更为合理方便。

# 实现策略



具体实现如图所示。主题分为三个部分，UI，实现逻辑，报告，具体的技术实现策略暂时不表。

## Case 解析

见功能case解析

## 生成代码（附加）

为附加功能点，用于ci集成以及其他第三方集成方便。

## Case管理

### Case基础管理

见功能case管理

### 其他Case操作

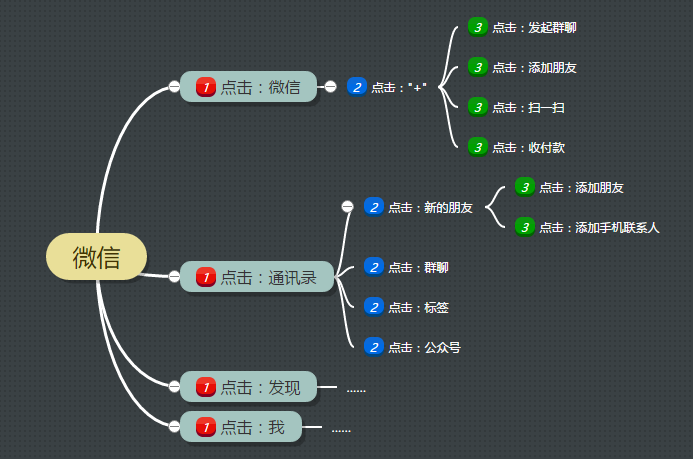
这部分主要是UI界面的Case操作具体为1、Case修改，2、深度管理，3、根据旧case生成新的Case。

#### UI Case修改

解析Case之后发现Case解析错误，可以再进行Case修改，然后生成固定的配置文件。

#### Case深度管理

首先了解Xmind解析成case之后，比如：



上图中以第一分支“点击：微信”为例，“点击：微信”为深度为1，“点击："+"”深度为2，如果执行到深度2，则需要2条case，如果是执行到深度3，则为1+1+4=6条case；所以执行深度关系到case的执行数量，case深度需要可以自定义。

#### 生成新的Case

在原有Case的基础上，进行新的Case的生成，这部分主要是UI支持。

## 执行进度

Case执行到哪一条，哪一深度，这个可以通过执行的方法或者配置文件来反馈得到。

## 手机状态与错误信息

### 手机状态

主要包括手机的CPU数据，MEM数据，以及其他相关数据，可以通过解析数据得到。

### 错误信息

手机执行过程中的ANR、Crash信息。可以通过手机Log 与其他手机信息收集得到



# 功能

## Case解析与管理

解析case是按照固定的格式进行解析，所以Case也需要按照固定的板式进行编写。

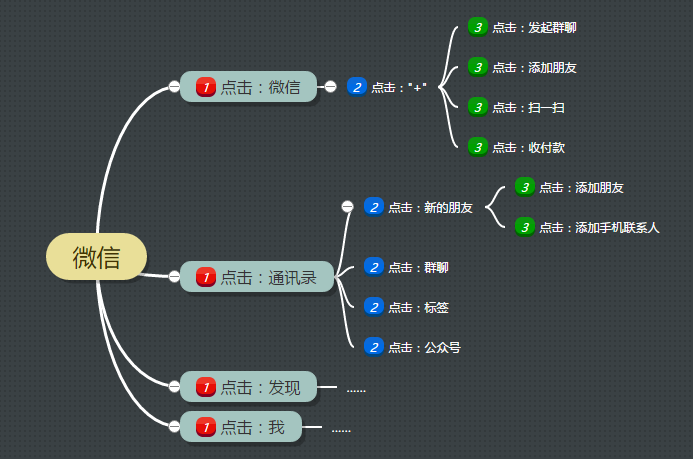
### Case组成格式

由于框架采用关键字驱动来实现，解析时，也是按照固定的“关键字：元素”格式来解析，格式可以为：Click：button1或者“点击：按键1”，long click: button2或者“长按：按键2”等“关键字：元素”的格式。

更多的关键字可以后续逐渐定义。

### Case管理

Case管理暂时采用Xmind型树形结构来管理，具体如下



红1代表微信的一级菜单，蓝2代表微信的二级菜单，绿3代表微信的3级菜单，每层Case操作有着严格的层级结构。一级菜单就是层级一，不能是层级二。

### Case解析

当按照固定格式编写好的case进行解析时，也是按照“关键字：元素”的格式来进行解析，解析成相应的配置文件，代码根据配置文件来执行，或者直接按照Case内容执行，不解析成Case；解析成case可以比较方便和第三方服务集成，后续可以作为附件功能点来做。

具体解析过程如下：



## 配置文件

不同模块间的数据传递规定采用配置文件传输，暂定‘\*.ini’，‘\*.xml’文件作为配置文件格式，具体的格式暂时不表，下一版本具体表处配置文件格式。

## UI界面

UI暂时不能用具体图例的方式标识，具体功能如下：1、解析case与展示，修改case，生成新的case，2、执行进度，当前的状态3、错误信息的展示。

## 报告

报告暂时定为Html展示，后续会选择提供Excel格式报告。

### 报告要求生成各种图表

根据各个步骤生成的数据，进行实时生成各种图表。比如饼图，柱状图，折线图等。

### Summary

#### 成功率

Case执行的成功率以及Case执行的成功与否会直观展示出。

#### 错误信息

Case执行过程中的错误信息，包括框架错误信息与Case执行过程中的ANR和Crash，错误截图。