|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档信息 | | | | | |
| 项目名称 |  | | | | |
| 文档作者 | 路培强 | 版本 | V1.0 | 文档完成时间 | 2019-03-14 |

## 目的

自动执行重复工作较大回归测试、兼容性测试，提高测试效率

## 自动化实施方案简介

自动化测试框架：selenium、appium

用例管理：Excel

测试数据：Excel

脚本开发：Java、Python

脚本生成方式：手工编写

脚本运行方式：testing测试框架、unittest测试框架

脚本回放：Chrome、Firefox、安卓真机or模拟器

自运行方案：Jenkins

测试报告：reporting

**工具：selenium、appium、jenkins**

## 自动化实施规范

### 1、 用例选取标准

a.该测试是否包含核心业务流程

b.该测试是否覆盖了最关键的特性路径

c.该测试的重复执行率较高

d.该测试是否定期运行，比如，经常重用，还作为回归测试或构建测试的一部分 对于手动运行这个测试是否太昂贵而不可能或是禁止的，如并行，渗透，耐力测试，内存泄漏等

e.该测试是否覆盖了最复杂的领域（通常是最有可能出错的领域）

f.使用相同步骤时，该测试是否需要许多数据组合

g.期望的结果是常数吗，比如每一次测试时都不会改变或不同？即使结果不同，是否可参数化（结果可预知）或可测出一个与期望结果的可接受的百分比（结果不可预知）

h.该测试是否非常耗时，如对[成百上千](https://www.baidu.com/s?wd=%E6%88%90%E7%99%BE%E4%B8%8A%E5%8D%83&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)的输出进行预期的分析

i.该测试是否运行在稳定的应用上

j.运行速度很慢的 case 不应该选取为自动化实现

k.自动化测试用例是否包含了手工测试的基线用例集

l.自动化的用例以正向用例为主，辅以个别重要的反向用例

### 2、定位对象识别规范

a.优先选择 By.id,By.name,By.classname,By.linkText,By.partialLinkText,By.cssSelector

b.尽量少用或不用By.xpath定位（xpath是万能的也是最不稳定的）

c.优先选择当前页面中不重名的属性

d.如遇重名属性，使用 List 返回多个元素，使用时根据元素在 List 中的位置调用

e.获取重名属性的列表数据，使用list返回多个元素集合，使用循环获取每个元素的文本值或者其他属性，符合输入的值，对这个元素进行其他操作

f.极其难以定位的元素可以考虑使用 xpath 定位（APP端可考虑坐标定位）

### 3、测试数据规范

a.字段名英文并首字母大写，使用驼峰命名，如Id

b.字段名说明应以英文括号标注，如Id(用例编号)

c.测试数据，已页为单位

c.测试数据以模块为单位，以json格式存储。如登录页的用户名和密码，表头为：Login(登录)，测试数据为：{"user": "135000000","password": "1234qwer"}

d.变量以${variableName}方式

### 4、脚本编写规范

a.常量命名规范

命名遵循驼峰命名规范，单词尽量使用全名称，如，final int num; 另外，对常量的声明必须带上类型

b.**变量命名规范**

变量命名应该简单、明确，遵循驼峰命名规范，如，String studentName

c.**参数命名规范**

参数命名应该简单、明确，见名知意。遵循驼峰命名规范

d.**方法命名规范**

见名知意，遵循驼峰命名规范

e.**代码注释规范**

注释务必做到准确简洁，能够充分表达代码实现的功能

f.**代码复用性**

代码尽量可被重复使用，避免耦合性、重复代码

### 5、脚本组织规范

理论上，一个模块一个文件(.java/.py)。一个功能一个方法，如新增：add，删除：delete

### 6、脚本断言规范

断言是测试框架提供的一种判断验证点是否通过的机制，需要说明的是：如果某个断言失败，则当前 case 会自动结束并 fail 掉，不会继续执行当前 case 的后续步骤。断言可以添加在业务层即 business 层也可以添加到 case 层，可根据 case 的实际业务选择断言的位置。

APP测试，有时toast展示时间比较短导致获取失败。从而导致断言失败。可以使用递归或者测试框架重试机制对此用例进行重试。即，断言或者脚本运行失败进行重试

### 7、结构分层规范

a.Control 层

框架底层，定义web page的基本元素类型（含元素识别、属性、方法），勿轻易修改

b.Util 层

对一些常用的方法进行重定义与封装、自定义通用方法，第三方服务等，可根据需要自行添加

c.Data 层

数据驱动层，定义case层中的测试方法所用到的数据

d.Config 层

配置定义，含浏览器驱动配置、配置文件读取等，勿轻易修改。

e.Page层

已模块为单位，定义页面上的元素识别方式、元素属性值

e.Business 层

一个模块对应一个java/py文件，定义该页面上的所有基本事务（查询、新增、删除等），定义执行流程。元素信息来源于page层（必要时，可以加入适当的断言）

e.Case 层

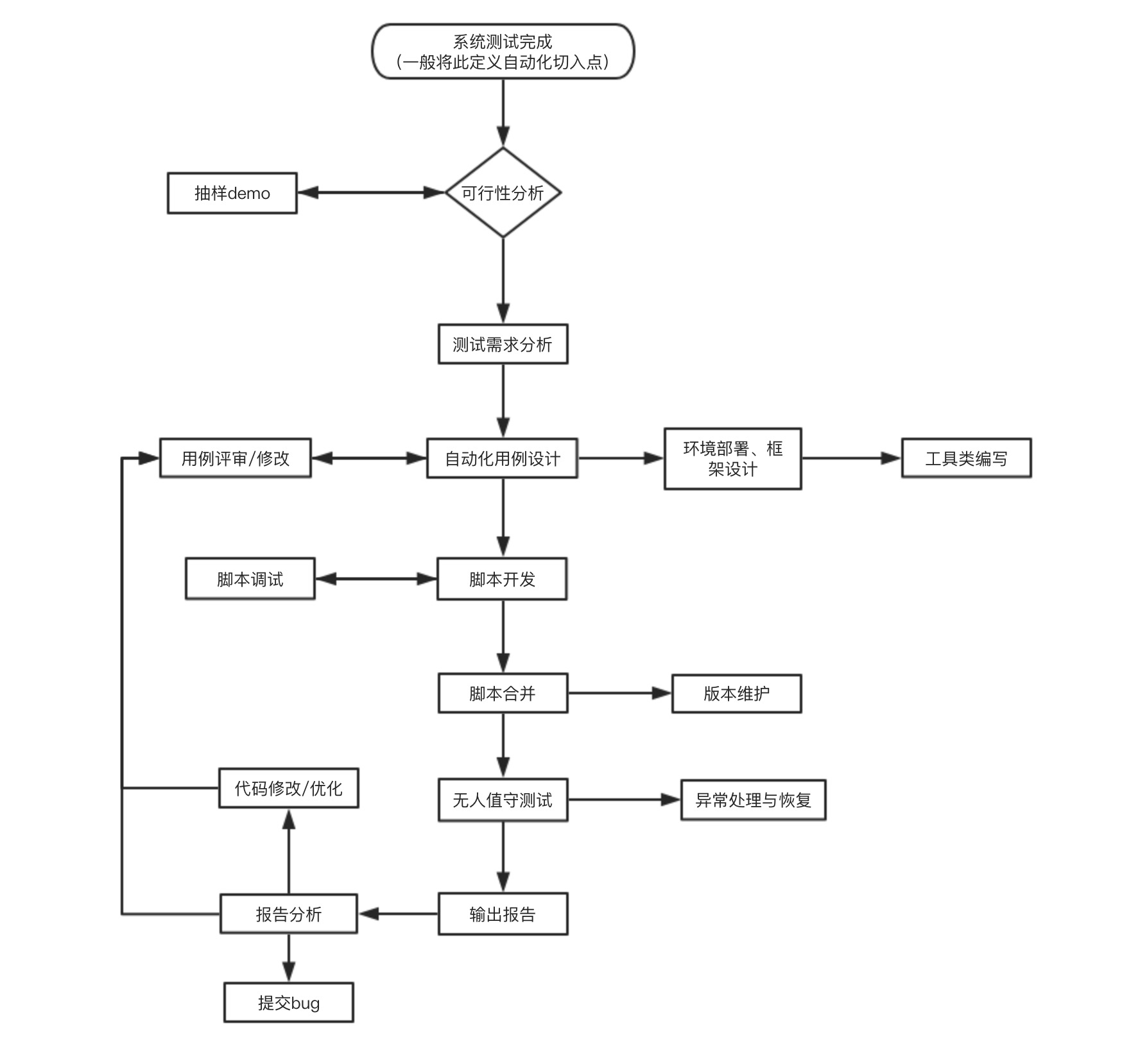
Case实现层，是business层、page层所定义对象的组合操作，并加入适当的断言（验证点）。

Case组织方式请参看“用例组织规范”

f.Listener

监听整个测试流程，case执行失败进行截图；可加入重试机制，失败后进行重试

## 自动化流程规范



## 其他说明