**浅谈什么是自动化测试框架**

无论是在自动化测试实践，还是日常交流中，经常听到一个词：框架。之前学习自动化测试的过程中，一直对“框架”这个词知其然不知其所以然。

最近看了很多自动化相关的资料，加上自己的一些实践，算是对“框架”有了一些理解，这篇博客，就聊聊自动化框架的一些事吧。

## **一、什么是框架**

框架（framework）是一个框子——指其**约束性**，也是一个架子——指其**支撑性**。是一个基本概念上的结构，用于去解决或者处理复杂的问题。

在软件工程中，框架（Framework）是整个或部分系统的可重用设计，表现为一组抽象构件及构件实例间交互的方法；

另一种定义认为，框架是可被应用开发者定制的应用骨架，前者是从应用方面而后者是从目的方面给出的定义。

上面的概念比较抽象，换另一个角度解释：

1、框架本身一般不完整到可以解决特定问题；

2、框架天生就是为扩展而设计的；

3、框架里面可以为后续扩展的组件提供很多辅助性、支撑性的方便易用的工具，也就是说框架是配套了一些帮助解决某类问题的库（libraries）或工具（tools）。

**约束性：**针对解决特定问题的软件框架会首先定义问题的边界，进而将相关的软件组件约束在这个边界内，保持框架在解决问题方面上的内聚性。

**支撑性：**框架本身不解决什么问题，但给了解决问题的相关组件一个组合底子，这个底子的科学性和易用性直接影响在此之上进一步开发的科学性和方便性。

上面的2个对框架的解释和定义，看起来也并没有很简单直白的说清楚什么是框架，下面是我的一些理解：

定义：**为解决某些特定问题而约束边界，支撑整个问题解决方案，配套了一些解决问题的组件而构成的工具**。

特定问题：什么问题？——自动化测试

约束边界：为什么约束？——明确测试范围和目的

解决方案：用什么方案解决问题？——编程语言+工具+其他

构成工具的组件：哪些组件？—— 用例、脚本、数据、日志、报告、通知

工具：特点是什么？—— 灵活性、可扩展性、高内聚低耦合

## **二、自动化测试**

**1、为什么要进行自动化测试？**

①、黑盒测试回归效率低

②、手动测试的偶然性和不确定性

③、回归的覆盖率不足

④、交付的产品质量无法保证，全靠评估

⑤、系统越复杂，问题越多

⑥、上线时间长、构件失败率高导致的蝴蝶效应（迭代快，加班多）

**2、自动化测试能解决什么问题？**

①、提高出现问题后的响应速率

②、降低回归成本

③、提高回归覆盖率

④、提高回归效率

⑤、提高回归的稳定性

**3、自动化测试的不足有哪些？**

①、无法减少成本投入，而是为了加快测试结果反馈，提升测试质量

②、自动化适用于回归和冒烟，而不是发现BUG

③、录制回放功能是鸡肋，可视化并不是一个很好的做法

④、不是所有所有系统所有功能都适合做自动化测试

## **三、自动化测试框架**

构成框架的组件，最起码应该具备以下的功能：



**Log：**日志记录和管理功能，针对不同的情况，设置不同的日志级别，方便定位问题；

**Report：**测试报告生成和管理以及即时通知，测试结果快速响应；

**Source：**配置文件、静态资源的管理，遵循高内聚低耦合原则；

**Common：**公共函数、方法以及通用操作的管理，遵循高内聚低耦合原则；

**TestCase：**测试用例管理功能，一个功能点对应一个或者多个case，尽可能的提高覆盖率；

**TestData：**测试数据管理功能，数据与脚本分离，降低维护成本，提高可移植性；

**TestSuite：**测试组件管理功能，针对不同场景不同需求，组装构建不同的测试框架，遵循框架的灵活性和扩展性；

**Statistics：**测试结果统计管理功能，每次执行测试的结果统计、分析、对比以及反馈，数据驱动，为软件优化和流程改进，提供参考；

**Continuous：**持续集成环境，即CI环境，包括测试文件提交、扫描编译、执行测试、生成报告及时通知等功能，**持续集成是自动化测试的核心**！

## **四、常见的自动化测试框架**

**1、接口自动化框架：**

①、java+testNG/Junit+Maven/Ant/Gradle+Jenkins+MySQL+testlink/redmine

②、python+unittest/pytest+Git+Jenkins+MySQL+testlink/redmine

③、python+rebot framework+unittest/pytest+Git+Jenkins+MySQL+testlink/redmine

④、jmeter+Maven/Ant+Jenkins+MySQL+testlink/redmine

**2、UI自动化测试框架**

①、java+selenium/appium+testNG/Junit+Maven/Ant/Gradle+Jenkins+MySQL+testlink/redmine

②、python+selenium/appium+unittest/pytest+Git+Jenkins+MySQL+testlink/redmine

③、python+rebot framework+unittest/pytest+Git+Jenkins+MySQL+testlink/redmine

通过上面的一些常见框架，你发现了什么？

它们都拥有共同特性：**编程语言+单元测试框架+扫描编译工具+持续集成工具+数据库+项目管理工具**。

编程语言：编写测试脚本、日志记录和输出；

单元测试框架：提供测试脚本运行、异常校验等一些列的配置；

扫描编译工具：测试文件扫描编译，一般配合持续集成工具使用效果更佳；

持续集成工具：Jenkins，经典的持续集成工具；

数据库：测试数据管理；

项目管理工具：测试结果统计管理；

**PS：**自动化测试工具太多，上面只是列举了使用率较高以及我个人还算了解的一些开源工具，具体的框架选型，需要根据具体项目特点和团队、个人技术特点来决定！

关于自动化测试框架的一些理解和思考总结，就是上面这些内容，提到的一些框架组件可能存在不合理的地方，仅供参考，如有更好的建议，请指出，不胜感激。