**自动化测试总体方案**

(V1.0)

Version History

| Ver. No. | Ver. Date | Revised By | Description | Reviewer | Status |
| --- | --- | --- | --- | --- | --- |
| 1.0 | 2017-04-11 | 欧仕强 | 创建 |  |  |
|  |  |  |  |  |  |

[1. 目前背景 2](#_Toc479767193)

[2. 设计目标 3](#_Toc479767194)

[3. 平台介绍 3](#_Toc479767195)

[3.1 工具介绍 3](#_Toc479767196)

[3.2 平台设计 5](#_Toc479767197)

[3.3 平台特点 5](#_Toc479767198)

[3.4 持续集成 6](#_Toc479767199)

[4. 脚本开发 7](#_Toc479767200)

[4.1 开发流程 7](#_Toc479767201)

[4.2 接口测试 8](#_Toc479767202)

[4.3 Web测试 9](#_Toc479767203)

[4.4 测试结果 11](#_Toc479767204)

# 目前背景

由于公司对软件质量的要求不断提高，对测试工作提出更高要求，而随着持续改进类项目规模扩展，回归测试范围不断扩大，工作量也不断增大。手工测试必然面临如下严峻挑战。

* + - 系统新功能模块上线，原有业务功能模块正确性验证工作量巨大；
    - 手工测试时间周期长；
    - 手工测试无法在有限版本验证周期内完成完整回归，只能覆盖新功能与少量原有功能；
    - 手工测试大量重复性劳动，人员疲惫，降低兴趣和工作积极性；
    - 手工测试质量基本以测试人员的责任心、忍耐性、注意力为基础；
    - 测试记录、报告等均靠人来管理和执行，随意性比较大。

据以上所述，测试风险将逐渐变高，软件质量也无法得到有力保障。因此，我们必须探索一条解决这一现状的道路——建立符合业务发展和易扩展的自动化测试体系保证软件质量。

# 设计目标

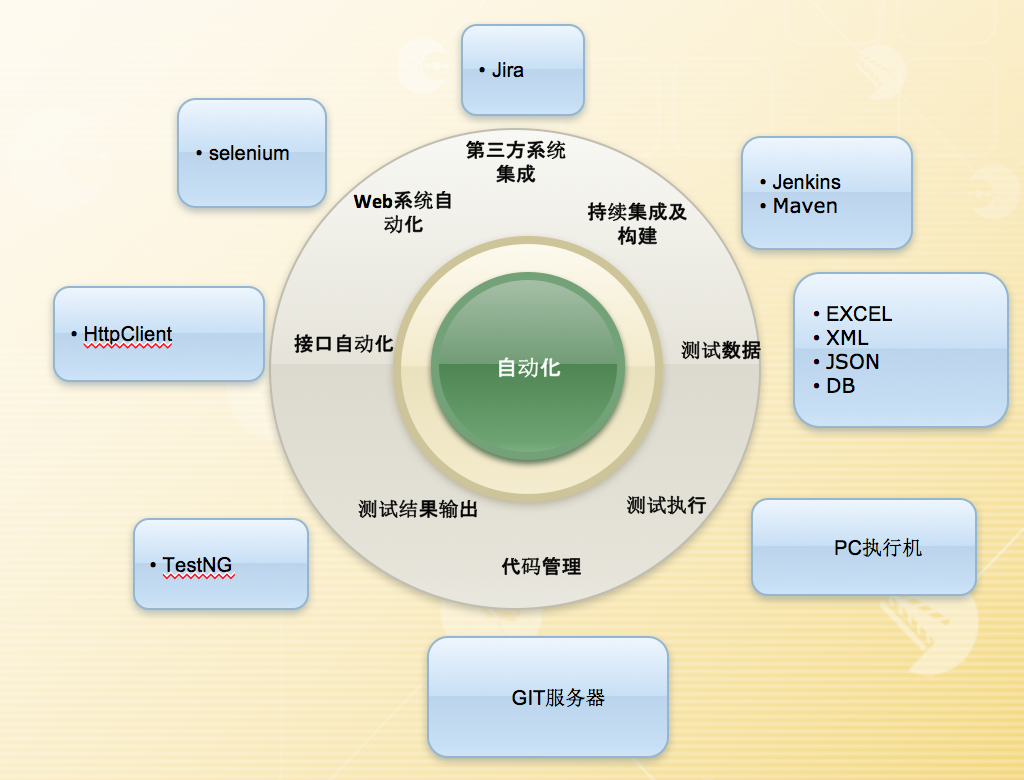
自动化测试平台提供一整套自动化测试工作流，可将多个脚本组成一个业务流程，并用不同的测试数据驱动执行业务流。通过业务流程测试方法可确保业务分支对业务流程的覆盖度，同时可提高测试脚本、测试数据可复用性。通过自动化测试平台逐步建立起符合目前测试现状的自动化测试体系。

* + - 以开源的自动化测试工具为基础；
    - 通过自主定制开发，建立起自动化测试技术框架；
    - 引入“数据脚本分离”实现数据和脚本的复用；
    - 通过统一的自动化测试平台管理测试任务；
    - 建立起与手工测试相结合的自动化测试流程和规范；
    - 通过端到端的测试任务保证测试质量。

遵循自动化测试体系标准进行实施，提高测试效率、测试覆盖率，解决工作量大、人力资源紧张、无法完整回归问题。通过测试资产库的不断积累，逐步增大自动化测试覆盖范围。

# 平台介绍

## 工具介绍

自动化测试平台是一个集成化平台，不仅包含脚本编写，还包含脚本用例的管理和执行，该平台包含的工具如下图所示，通过这些测试工具实现自动化测试和手工测试协同办公，手工测试人员完成用例编写，自动化测试人员根据手工测试人员编写的案例编写测试脚本，并上传到测试平台，测试人员通过该平台运行测试脚本，查看测试结果。

自动化测试平台工具集

主要工具说明如下：

* + - 使用Selenium负责PC端测试及页面对象识别操作。
    - HttpClient用于接口自动化测试，支持HTTP，Soket，WebServices测试。
    - 通过Maven管理项目的依赖和测试脚本执行。
    - 采用Jenkins实现分布试执行。
    - TestNG作为测试脚本运行和报表输出
    - 采用Git 管理测试脚本，测试平台自动同步Git上的测试脚本。并和手工测试用例关联。
    - Excel,DB和XML,JSON文件用于管理测试数据保存测试结果
    - Jira作为第三方用于需求，缺陷管理平台，主要用于最终项目报表展示。

## 平台设计

自动化测试框架以测试脚本为基础，服务于功能测试，提高测试效率和测试覆盖率为目标。下图中是对测试平台的整体设计。包含自动化测试脚本开发模块和自动化测试管理模块，在设计过程中保证程序高内聚低耦合，保证模块的独立性。提供一套扩展性强的自动化测试平台。该自动化平台包含两个阶段：

第一阶段要用于满足项目的回归测试任务，主要集中测试脚本的开发，采用数据与脚本相分离的设计模式。通过Excel，JSON，XML保存测试脚本数据。主要包含接口测试， PC端测试脚本编写，在开发测试脚本中不断的扩展公数库。

第二阶段主要用于自动化测试管理，开发一套完善的自动化测试任务管理平台并可以和第三方平台进行对接，测试人员登录该测试平台查看项目需求，通过需求编写测试用例，创建业务流，完成用例和自动化脚本关联，设定执行策略，管理测试设备，运行查看测试结果并通过邮件发送测试结果。

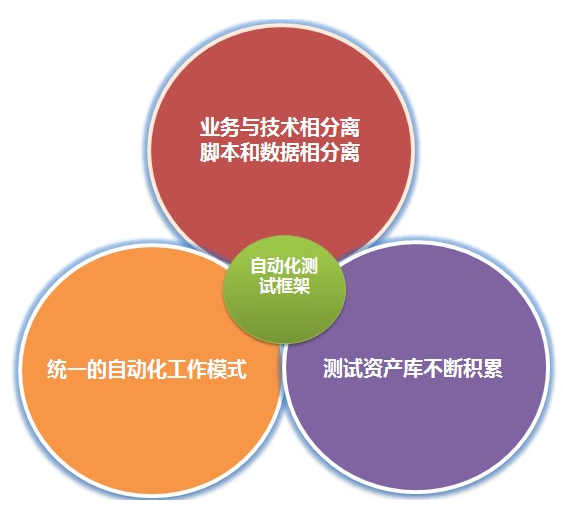


自动化测试管理平台功能模块

## 平台特点

该自动化测试平台只要用于自动化测试管理和自动化测试脚本编写。通过业务与技术相分离脚本与数据相分离、统一的自动化工作模式、测试资产库的不断积累为设计思想来设计自动化测试平台，该平台与市面上主流的测试工具相比，该主要包含如下特点。

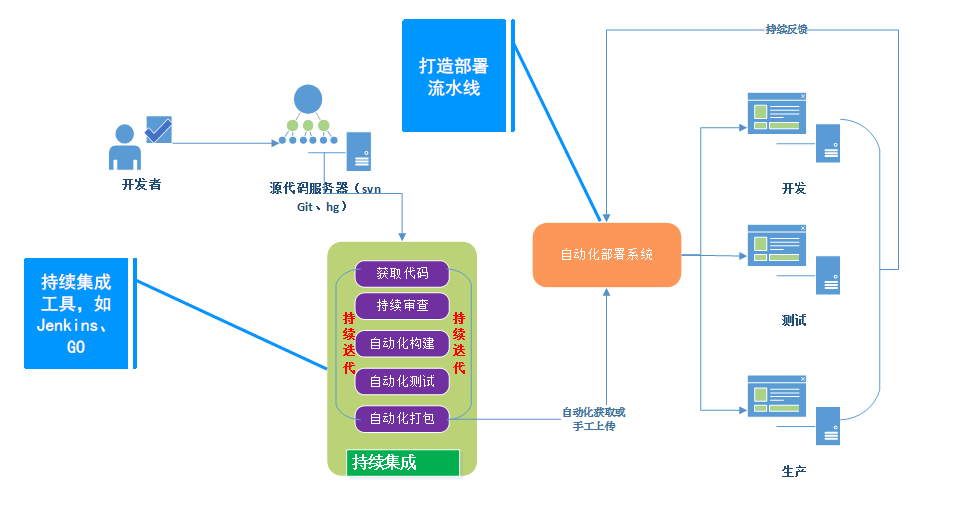
* 具有统一的工作模式，从用例设计、脚本设开发，到调度和执行，都有固化的模式和规范
* 一套平台支持接口测试和Web端测试脚本编写。
* 统一的工作平台、业务与技术分离、脚本与数据分离。
* 丰富的函数库，提高测试脚本编写效率。
* 为测试脚本开发提供统一的功能接口。
* 采用业务流程分析方法，保证测试脚本的独立性
* 手工测试和自动化测试协同工作，并发挥各自优势。
* 执行方便，测试报告展示美观，易于开发测试人员定位问题。



自动化测试测试特点

## 持续集成

该自动化测平台是一套完整的解决方案，应该融入持续集成中，开发和测试通过源代码服务器为界限，开发人员只需把代码提交到Git服务器，自动化测试平台就会获取到源代码，然后对原代码进行审查，查看代码是否符合规范，然后执行自动打包，编译并发布到测试环境，触发自动化测试脚本，执行自动化测试，执行完成后并发送邮件到相关人员，实现持续集成。



自动化测试融入持续集成

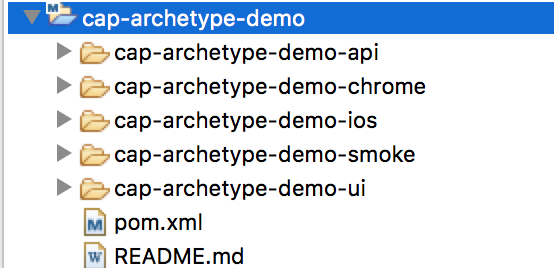
# 脚本开发

## 开发流程

在测试脚本开发过程中，按照如下流程来实施自动化测试，首先了解被测系统，确定是否开展自动化测试，确定自动化测试的范围，根据手工编写的用例来筛选自动化用例，然后开发自动化测试脚本和组件，并对测试脚本进行调试和优化，并把编写好的脚本上传到代码服务器，开始运行自动化测试脚本。

脚本开发流程

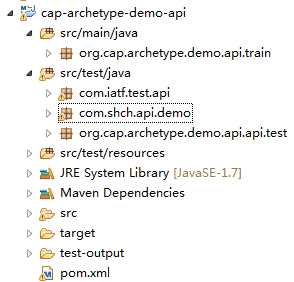
测试脚本采用Maven来管理，用Eclipse作为脚本开发工具，结合了TestNG，HttpClient，Selenium, FreeMark等基础工具，在这些基础工具上进行二次开发，提高测试脚本开发的效率。下图中是脚本的整体结构，包含了接口测试，PC端Web测试覆盖了自动化测试各个方面。



自动化测试框架脚本结构

## 接口测试

接口测试采用HTTPClient实现POST，GET，Delete，PUT等请求，结合Jackson Json包完成Json数据的解析，通过DOM4J解析XML数据，测试数据可以存放在Excel，JSON和XML中，通过FreeMarker实现数据的自动生成。接口测试结构如下:

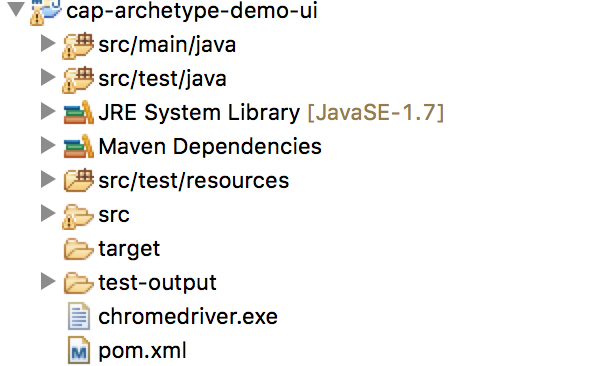


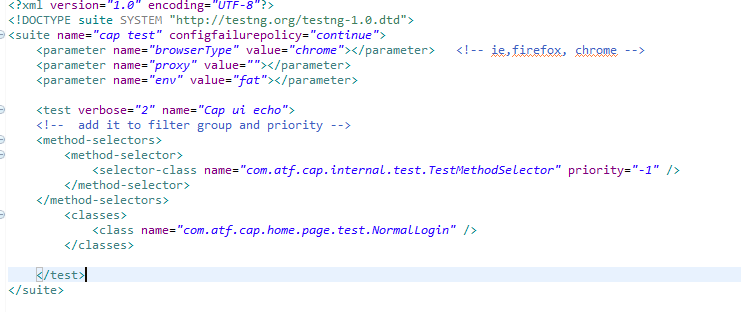
在编写接口测试脚本只需要基础APITest这个接口，就可以完成接口测试任务。如果有配置参数可以放到TestNG的配置文件中。



## Web测试

PC端的Web测试采用Selenium作为对象的识别和操作，支持IE，Chrome，Firefox等浏览器，脚本开发采用PageObject模式开发测试脚本。编写Web测试脚本只需要继承UITest接口，就可以完成PC端测试，并提供TestNG配置文件来设置运行参数。PC端测试架构预览：



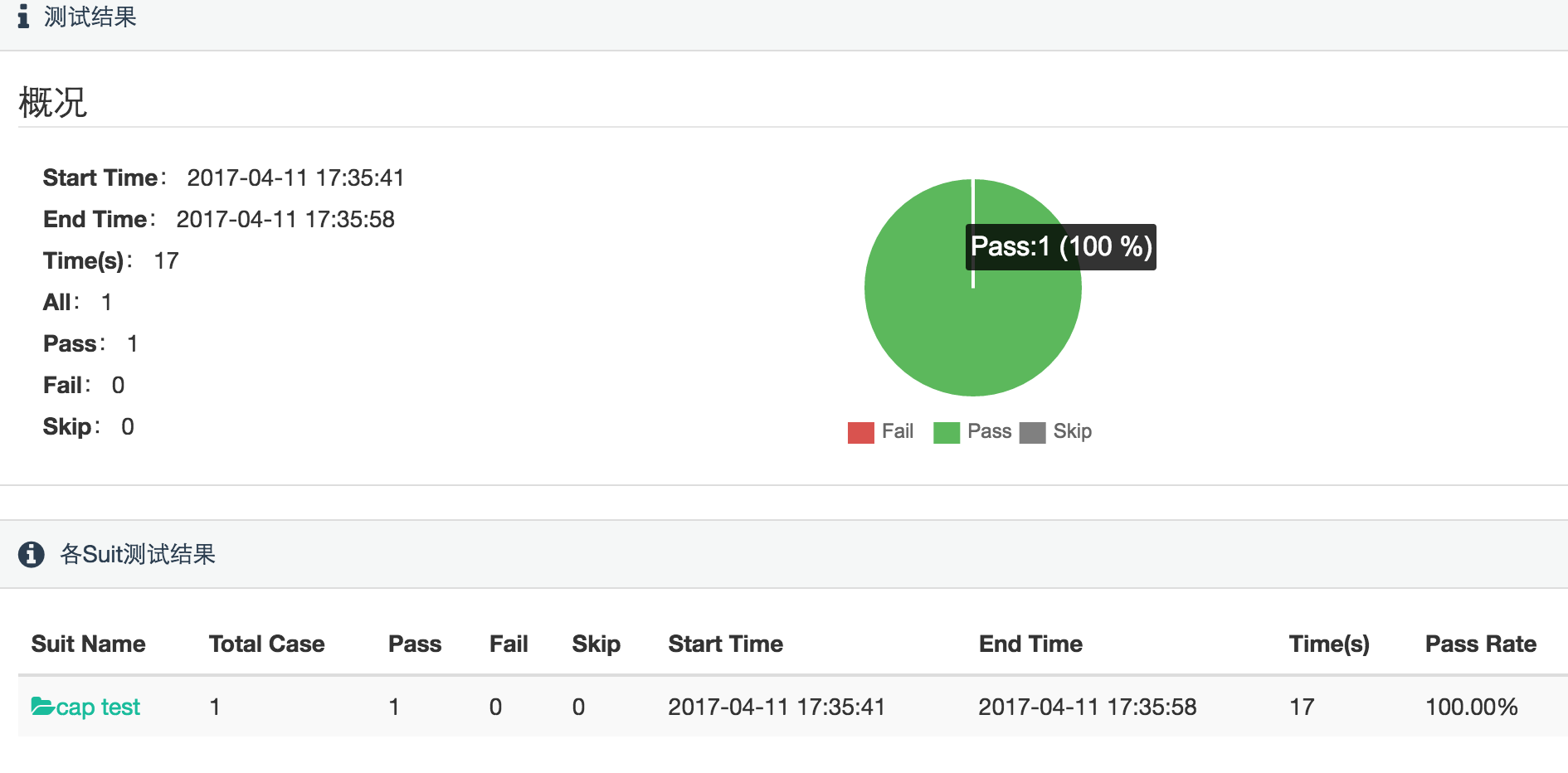


测试脚本如下:



## 测试结果

开发完成测试脚本后，可以通过Maven Test运行测试脚本，也可以和Jenkins进行集成， 下图是运行后的测试报告，该报告包含失败用例数，成功用例数，总用例数和运行时间等等。



测试报告总揽

点击用例详情，可以查看用例运行的详细报告，包含断言，截图，失败原因，用例详情如下图所示。

测试报告详情



测试报告详情截图