# 自动化测试

1. **自动化测试技术**
2. **自动化测试技术背景**

为弥补手工测试的不足，提高测试效率，自动化测试应运而生。自动化测试是把以人为驱动的测试行为转化为机器执行的一种过程。自动化测试解决了重复的测试步骤对人力过多占用的问题。自动化测试主要应用于以下特点的软件开发项目：

1. 软件需求或基本架构模块变动不频繁

测试脚本的稳定性决定了自动化测试的实施成本。自动化测试最大的挑战就是需求的变化，如果需要频繁的修改自动化脚本去适应新的功能，投入产出比太低，那么自动化测试也失去了其价值。折中的做法就是选择相对稳定的功能模块进行自动化测试，变动较大的部分使用手工测试。

1. 项目周期足够长

自动化测试的需求分析，框架搭建，脚本开发与调试都需要时间，项目周期越长，自动化测试脚本的利用率越高。

1. 测试过程中需要执行大量重复测试步骤或遇到测试场景排列组合问题

机器天然适合执行重复的任务。如项目需要定期构建，测试验证或比较频繁的回归验证，则提高自动化测试脚本重用性。

自动化测试的优缺点：

优点：

避免重复的测试过程占用人力

提高测试效率

更好的利用无人值守的时间

保证每次测试的一致性和可重复性性

进行一些手工无法进行的测试

缺点：

开发维护成本较高

测试门槛较高，需要一定技术基础

不适应频繁的模块界面变动

界面元素问题容易导致测试失败

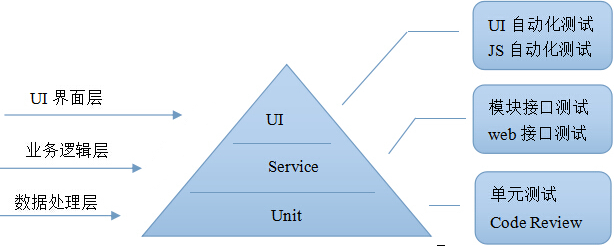
自动化测试的分类

自动化包括一切通过工具（程序）的方式来代替或辅助手工测试的行为都可以看做自动化，除了功能测试工具，还包括性能测试工具（loadrunner、jmeter）,或自己所写的一段程序，用于生成1到100个测试数据。狭义上来讲，通工具记录或编写脚本的方式模拟手工测试的过程，通过回放或运行脚本来执行测试用例，从而代替人工对系统的功能进行验证。

通常而言，大家所说的自动化测试是指：基于产品或项目的UI层的自动化测试。

* 自动化测试
* 功能自动化测试
  + Web自动化测试
  + App自动化测试
  + 性能自动化测试
    - Web自动化测试
    - App自动化测试

而从测试阶段来说，在产品的不同阶段都可进行自动化测试：



**2.自动化测试过程**

自动化测试过程与手工测试过程本质上是一样的，不同点在于，自动化测试在前期需要根据项目特点和测试需求选择合适的自动化测试方案和测试工具，搭建自动化测试环境，必要时，还需要对测试工具进行二次开发。

1. 自动化可行性分析

讨论项目开展自动化测试的可行性，明确自动化测试目标，如手工测需求分析及案例是否适合自动化测试及自动化测试覆盖率等。

2）自动化测试需求分析

当测试项目满足实施自动化的前提条件，并确定在项目中使用自动化时，就可进行自动化测试需求分析。此过程需要确定项目业务需求与技术需求，并参考手工测试用例确定自动化测试的覆盖范围。

3）自动化测试架构的搭建

初次实施自动化测试时，需要根据前期制定的自动化测试方案搭建自动化测试环境。该环境应当从技术上满足所有测试步骤的执行需求，且能方便的组织管理测试脚本、测试数据，能自动生成测试报告。

4）自动化脚本的编写

自动化测试脚本是对具体测试步骤的转换，应当实行脚本与数据的分离。组织编写脚本时构时，应考虑脚本的最大化复用。

5）自动化测试的运行与维护

自动化测试执行失败时，应排除测试环境与被测系统的因素，且应根据测试报告手动复现错误。在项目的生命周期中，被测软件难免会有模块和页面的变动，脚本开发人员应及时修改脚本。

**3.自动化测试的技术需求**

根据自动化测试特点，一个自动化测试项目的实施，需要解决以下技术上的需求：

1）测试驱动工具

不同于手工测试，自动化测试需要通过测试工具来实现对手工测试步骤的模拟，因此不同的自动化测试需求，不同的软件类型，需要不同的测试工具。根据项目特点选用合适的自动化测试工具，是自动化测试的关键。

2）持续集成与版本管理

测试脚本与测试数据需要统一的组织管理与版本管理，方便维护与执行。

3）测试任务调度管理

为达成提高测试效率的目的，自动化测试一般使用专门的执行机采取远程调度的方式提交自动化测试任务，因而需要工具组织管理测试任务的调度执行。

1. 自动化测试报告生成

为方便查看测试结果，定位测试失败原因，自动化测试环境应能生成自动化测试报告与详细的测试日志。

1. **UI自动化测试**

UI层是用户使用产品的入口，UI自动化测试是使用测试工具，模拟用户在界面上的操作来执行测试步骤。根据软件架构，分为C/S和B/S两种类型。而根据使用终端，分为PC端与移动端。对于这几种类型的软件有不同的自动化测试工具。

主流自动化测试工具：

QTP：

企业级自动化测试工具，支持VBScript语言，可用于C/S和B/S架构的软件自动化测试，为收费软件。

Selenium:

一款开源的Web自动化测试工具，支持Java，Python，Ruby，C#等多种编程语言，支持多种浏览器。有很强的扩展性，适合二次开发。

Appium：

一款主流的App自动化测试工具，支持Android和iOS平台的App测试。Appium继承了SeleniumDriver protocol，也支持多种编程语言。

**1.Web自动化测试**

Web系统使用通用的浏览器作为与用户交互的接口，因此Web自动化测试工具需要通过控制浏览器来执行测试任务。浏览器根据HTML和CSS渲染页面，且执行Javascript脚本，测试工具利用浏览器原有的元素定位方法定位操作元素，还可以直接执行Javascript脚本模拟页面的操作。

Selenium是目前主流的Web自动化测试工具，提供了自动化测试所需的页面元素定位方法和操作方法，配合驱动来操控浏览器模拟用户操作。Selenium WebDriver本质上是一个函数库，可由编程语言直接调用，因而具有很高的灵活性和可扩展性，但需要测试人员有一定的编程能力。

浏览器驱动

测试代码

Selenium库

IE

Firefox

Chrome

Safari

1. **APP自动化测试**

与Web自动化测试类似，APP自动化测试需要通过中间媒介来控制手机的行为，不同的是手机的连接和获取控制权限更为复杂，测试工具需要取得手机的完全控制以便安装、卸载APP，启动APP，对屏幕进行操作，截图等等。APP自动化测需要测试人员懂得一定的APP开发知识。

Appium是目前主流的APP自动化测试工具，支持iOS和Android平台。Appium核心是一个Web服务器，提供了一套接口，当收到客户端连接后，会创建一个Session，根据接收到的命令在移动设备上执行。

同样的，Appium继承了WebDriver协议，提供了类似于Selenium WebDriver一样的函数库供编程语言调用，因此也具有很高的灵活性和可扩展性，要求测试人员有一定的编程能力。

1. **自动化测试框架与持续集成**

无论哪种自动化测试，都不能局限于本地的测试开发与调试，因此自动化测试需要一个完整的体系框架来实现以下功能：

1. 分布式测试，能同时在多台机器上并行自动化测试
2. 实现关键字或数据驱动

管理测试集与测试数据，通过关键字或数据来控制测试的业务流。

1. 自动化测试用例执行
2. 持续集成，定时自动构建和执行测试任务
3. 测试报告与日志生成

对于小型自动化项目而言，一般采用现有的工具完成上述功能，如：Git/SVN、Maven、Jenkins、TestNG、Robotframework等等。这些工具的部署对人员技术要求较高，如果需要大规模多系统应用自动化测试，一般公司都会选择在开源工具的基础上，进行二次开发，建立统一的自动化测试平台，平台集合了以上功能，降低了工具部署和开发难度，便于实际应用和项目管理。

1. UI自动化测试
   1. Web自动化测试

前端定位

SeleniumWebDriver

* 1. App自动化测试

自动化测试框架

数据驱动

关键字驱动

RobotFramework

Java+selenium+maven+git+jenkins