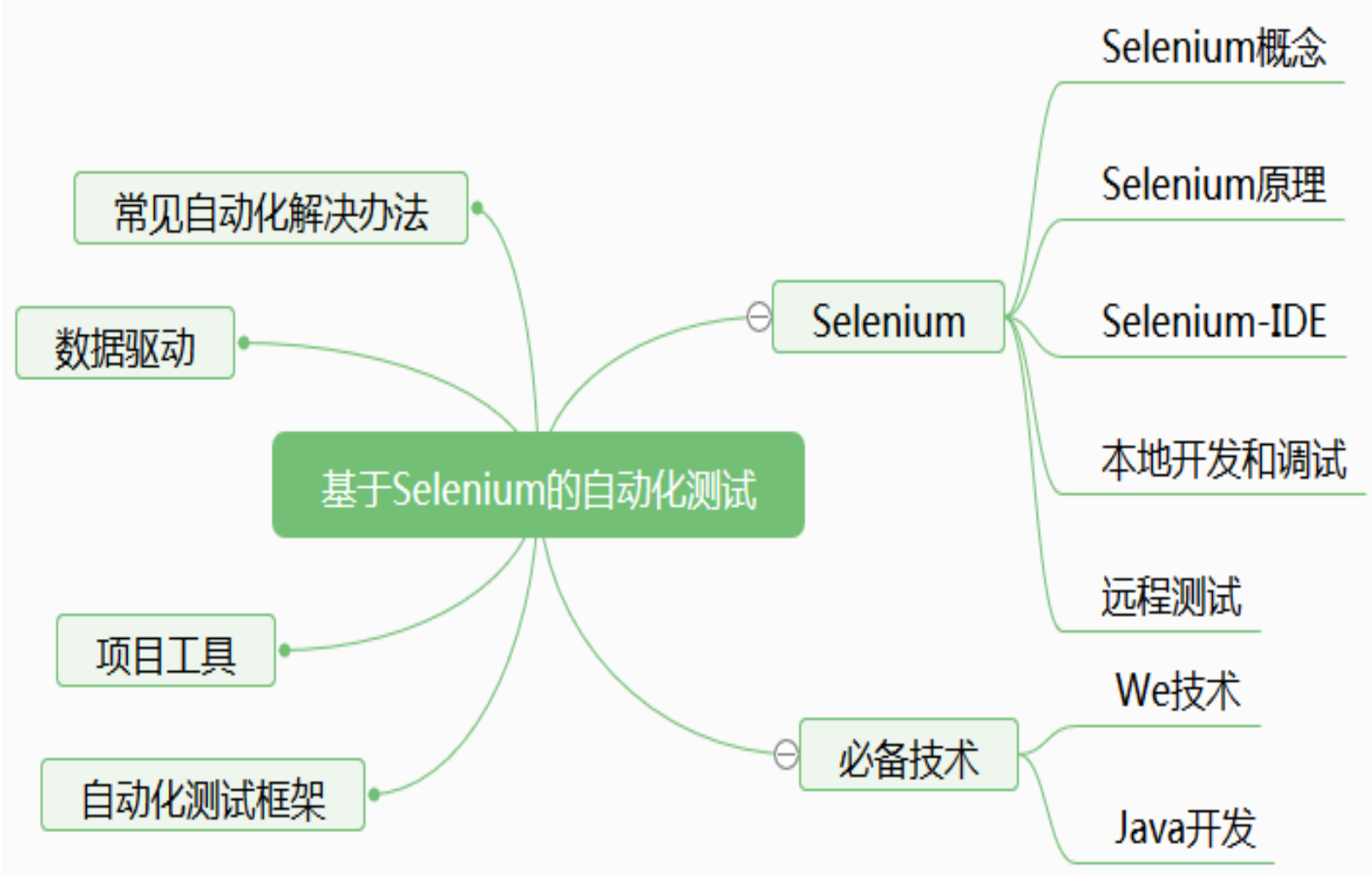
目前软件测试职业者的危机

- 单纯的手工测试人员必须思考职业转型，否则必将逐渐被时代淘汰
- 互联网测试职业一定以技术为依托，没有技术含量的测试逐渐失去竞争力
- 互联网测试不会只侧重一个层面，必须是web端、移动端、服务端甚至微信端、云端的多重维度性的综合性测试，此为未来软件测试职业人才的职业竞争力精髓所在
- 传统手工测试需要基于自身的测试从业经验，分析个人技能体系，根据期望的职业转型路线去针对性的提高自己，从此在IT行业立于不败之地！

课程内容


基于Selenium的自动化测试
常见自动化解决办法
数据驱动
项目工具
自动化测试框架
Selenium
Selenium概念
Selenium原理
Selenium-IDE
本地开发和调试
远程测试
必备技术
We技术
Java开发

01 自动化测试基础

本章大纲

1.1 软件测试分类

1.2 分层的自动化测试

1.3 什么项目适合做自动化测试

1.4 UI自动化测试工具介绍

1.5 Selenium介绍

软件测试分类（一）

根据项目流程阶段划分软件测试：

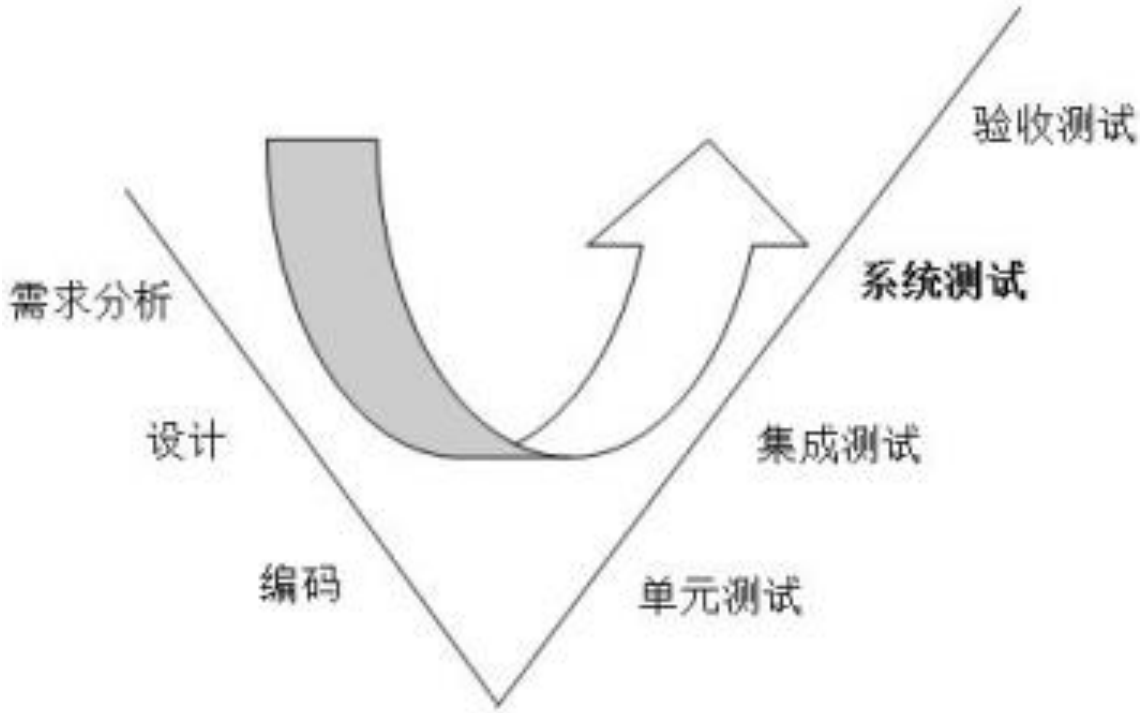
1）单元测试：单元测试（或模块测试）是对程序中的单个子程序或具有独立功能的代码段进行测试的过程。

2）集成测试：集成测试是在单元测试的基础上，先通过单元模块组装成系统或子系统，再进行测试。重点是检查模块之间的接口是否正确。

3）系统测试：系统测试是针对整个产品系统进行的测试，验证系统是否满足需求规格的定义，以及软件系统的正确性和性能等是否满足其需求规格的要求。

4）验收测试：验收测试是部署软件之前的最后一个测试阶段。验收测试的目的是确保软件准备就绪，向软件购买者展示该软件系统能够满足用户的需求。

软件测试分类（一）



软件测试分类（二）

- 白盒测试与黑盒测试，主要是根据软件测试工作中对软件代码的可见程度进行的划分。

1）黑盒测试，指的是把被测的软件看作一个黑盒子，我们不去关心盒子里面的结构是什么样子的，只关心软件的输入数据和输出结果。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

2）白盒测试，指的是把盒子打开，去研究里面的源代码和程序执行结果。它是按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常进行，检验程序中的每条逻辑路径是否都能按预定要求正确工作。

软件测试分类（二）

3）灰盒测试，介于黑盒测试与白盒测试之间。可以这样理解，灰盒测试既关注输出对于输入的正确性，同时也关注内部表现。但这种关注不像白盒测试那样详细、完整，它只是通过一些表征性的现象、事件、标志来判断内部的运行状态。有时候输出是正确的，但内部其实已经错误了，这种情况非常多。如果每次都通过白盒测试来操作，效率会很低，因此需要采取灰盒测试的方法。

软件测试分类（三）

从软件的不同测试面可以划分为功能测试与性能测试。

1）功能测试。功能测试主要检查实际功能是否符合用户的需求，因此测试的大部分工作也是围绕软件的功能进行。

功能测试又可以细分为很多种：逻辑功能测试、界面测试、易用性测试、安装测试、兼容性测试等。

软件测试分类（三）

2）性能测试。性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。软件的性能包括很多方面，主要有时间性能和空间性能两种。

- 时间性能：主要是指软件的一个具体的响应时间。例如一个登录所需要的时间，一个商品交易所需要的时间等。
- 空间性能：主要指软件运行时所消耗的系统资源，例如硬件资源，CPU、内存，网络带宽消耗等。

软件测试分类（四）

从对软件测试工作的自动化程度可以划分为手工测试与自动化测试。

1）手工测试。手工测试就是由测试人员一个一个地去执行测试用例，通过键盘鼠标等输入一些参数，并查看返回结果是否符合预期结果。

2）自动化测试。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。通常，在设计测试用例并通过评审之后，由测试人员根据测试用例中描述的规则流程一步步执行测试，把得到实际结果与期望结果的比较。

软件测试分类（四）

自动化测试又可分为：功能自动化测试与性能自动化测试。

1）功能自动化测试：它是把以人为驱动的测试行为转化为机器执行的一种过程。通过测试工具（或框架）录制/编写测试脚本，对软件的功能进行测试，并验证测试结果是否正确，从而代替部分的手工测试工作，达到节约人力成本和时间成本的目的。

2）性能自动化测试：通过性能工具来模拟成千上万的虚拟用户向系统发送请求，从而来验证系统的处理能力。

软件测试分类（五）

1）冒烟测试，是指在对一个新版本进行大规模的系统测试之前，先验证一下软件的基本功能是否实现，是否具备可测性。如果主要功能都没有运行通过，则打回开发组重新开发。这样做的好处是可以节省时间和人力投入到不可测的项目中。

2）回归测试，是指修改了旧代码后，重新进行测试以确认修改后没有引入新的错误或导致其他代码产生错误。

软件测试分类（五）

3）随机测试，是指测试中的所有输入数据都是随机生成的，其目的是模拟用户的真实操作，并发现一些边缘性的错误。

4）探索性测试，可以说是一种测试思维技术，它没有很多实际的测试方法、技术和工具，但是却是所有测试人员都应该掌握的一种测试思维方式。探索性测试强调测试人员的主观能动性，抛弃繁杂的测试计划和测试用例设计过程，强调在碰到问题时及时改变测试策略。

本章大纲

1.1 软件测试分类

1.2 分层的自动化测试

1.3 什么项目适合做自动化测试

1.4 UI自动化测试工具介绍

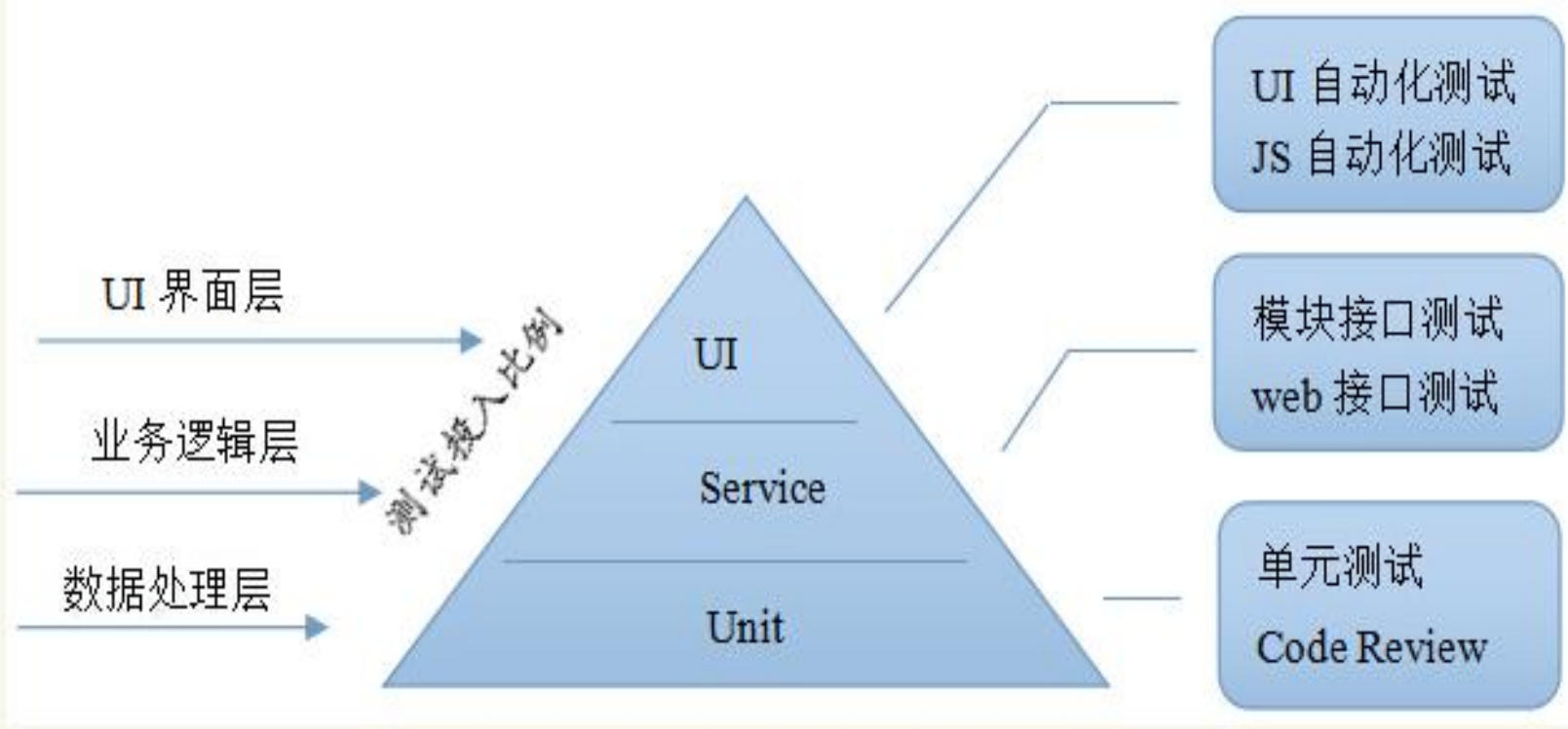
1.5 Selenium介绍

分层的自动化测试

测试金字塔的概念由敏捷大师Mike Cohn在他的《Succeeding with Agile》一书中首次提出，如图所示。他的基本观点是：我们应该有更多的低级别的单元测试，而不仅仅是通过用户界面运行的高层的端到端的测试。

分层测试

- Martin Fowler大师在测试金字塔模型的基础上提出分层自动化测试的概念



单元测试（Unit testing）

In computer programming, **unit testing** is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method.[2] Unit tests are short code fragments[3] created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing.

单元测试自动化

- 单元测试是应用程序的最小可测试部分。
- 在面向过程编程中，单元也可以是整个模块，单常见的是单个函数或过程。
- 在面向对象编程中，单元通常是整个接口，例如类，但可以是单独的方法。
- 单元测试多数情况下由程序员自己完成的
- 单元测试更强调的是程序的最小可测试单元，而模块测试更强调被测程序功能的完整性
- 单元测试框架，如java语言的Junit、TestNG，C#语言的NUnit，以及Python语言的unittest、pytest

接口测试自动化

Web应用的接口测试大体分为两类：模块接口测试和Web接口测试。

（1）模块接口测试，主要测试模块之间的调用与返回。它主要强调对一个类方法或函数的调用，并对返回结果的验证，所用到的测试工具与单元测试相同。

（2）Web接口测试又可分为两类：服务器接口测试和外部接口测试。

接口的分类

从系统的调用方式不同，又可以将接口大致分为以下三种。

1.系统与系统之间的接口（例如QQ的第三方登录）

2.下层服务上层服务的接口

3.系统内部，服务与服务之间的调用

接口测试的意义

- 更早的发现问题
- 缩短产品研发周期
- 发现更底层的问题

UI 自动化测试

UI 层是用户使用该产品的入口，所有功能都通过这一层提供并展示给用户，所以大多测试工作都集中在这一层进行。

目前主流的测试工具有UFT、Watir、Robot Framework、Selenium 等。

本章大纲

1.1 软件测试分类

1.2 分层的自动化测试

1.3 什么项目适合做自动化测试

1.4 UI自动化测试工具介绍

1.5 Selenium介绍

什么项目适合做自动化测试?

1）任务测试明确，需求不会频繁变动。
2）每日构建后的测试验证。
3）比较频繁的回归测试。
4）软件系统界面稳定，变动少。
5）需要在多平台上运行的相同测试案例、组合遍历型的测试，大量的重复任务。
6）软件维护周期长。
7）项目进度压力不太大。
8）被测软件系统开发比较规范，能够保证系统的可测试性。
9）具备大量的自动化测试平台。
10）测试人员具备较强的编程能力。

自动化测试人员分工

- 测试框架开发人员
- 基于测试框架编写测试脚本的人员
- 编写需要自动化测试用例及测试框架需求的人员

本章大纲

1.1 软件测试分类

1.2 分层的自动化测试

1.3 什么项目适合做自动化测试

1.4 UI自动化测试工具介绍

1.5 Selenium介绍

UI自动化测试工具介绍

UFT（全称Unified Functional Testing）由QTP（Quick Test Professional software）与ST(Service Test)合并而来，由HP公司开发。它是一种企业级的自动测试工具，提供了强大易用的录制回放功能，同时兼容对象识别模式与图像识别模式两种识别方式，支持B/S与C/S 两种架构的软件测试，是目前主流的自动化测试工具。

Robot Framework

- Robot Framework是一款基于Python语言编写的自动化测试框架，具备良好的可扩展性，支持关键字驱动，可以同时测试多种类型的客户端或者接口，可以进行分布式测试。

Watir

Watir全称是“Web Application Testing in Ruby”，是一种基于Web模式的自动化功能测试工具。Watir是一个Ruby语言库，使用Ruby 语言进行脚本开发。

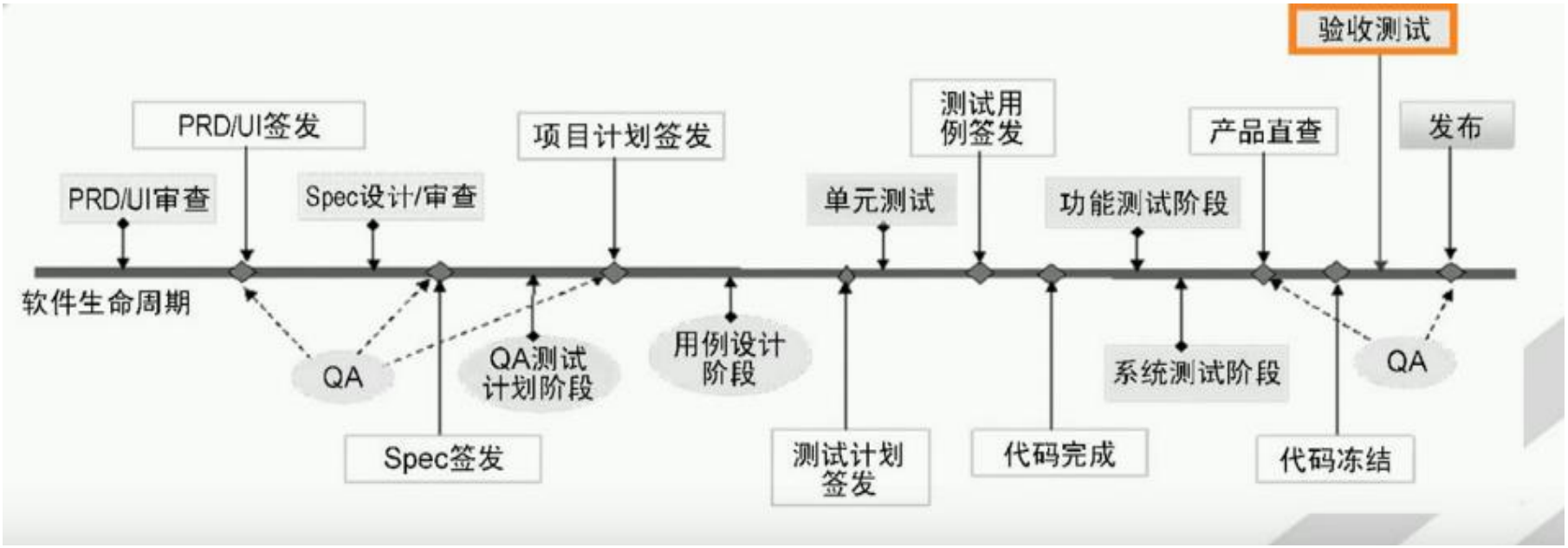
Selenium

Selenium也是一个用于Web应用程序测试的工具，支持多平台、多浏览器、多语言去实现自动化测试。目前在Web自动化领域应用越来越广泛。

http://www.seleniumhq.org/

什么是Selenium

Selenium 是专门为web应用程序编写的一个自动化验收程序（Acceptance Test）工具。



本章大纲

1.1 软件测试分类

1.2 分层的自动化测试

1.3 什么项目适合做自动化测试

1.4 UI自动化测试工具介绍

1.5 Selenium介绍

Selenium名称来源

“You can cure mercury poisoning by taking selenium supplements”

-Jason Huggins

Selenium发展史

Selenium 2.0 = Selenium 1.0 + WebDriver

在Selenium 2.0中主推的是WebDriver，可以将其看作Selenium RC 的替代品。因为Selenium为了保持向下的兼容性，所以在Selenium2.0中并没有彻底地抛弃Selenium RC。

Selenium与WebDriver合并原因:WebDriver解决了Selenium存在的缺点（例如能够绕过JavaScript沙箱），Selenium解决了WebDriver存在的问题（例如支持广泛的浏览器）。

支持的平台

支持的浏览器

支持的开发语言

C#

JavaScript

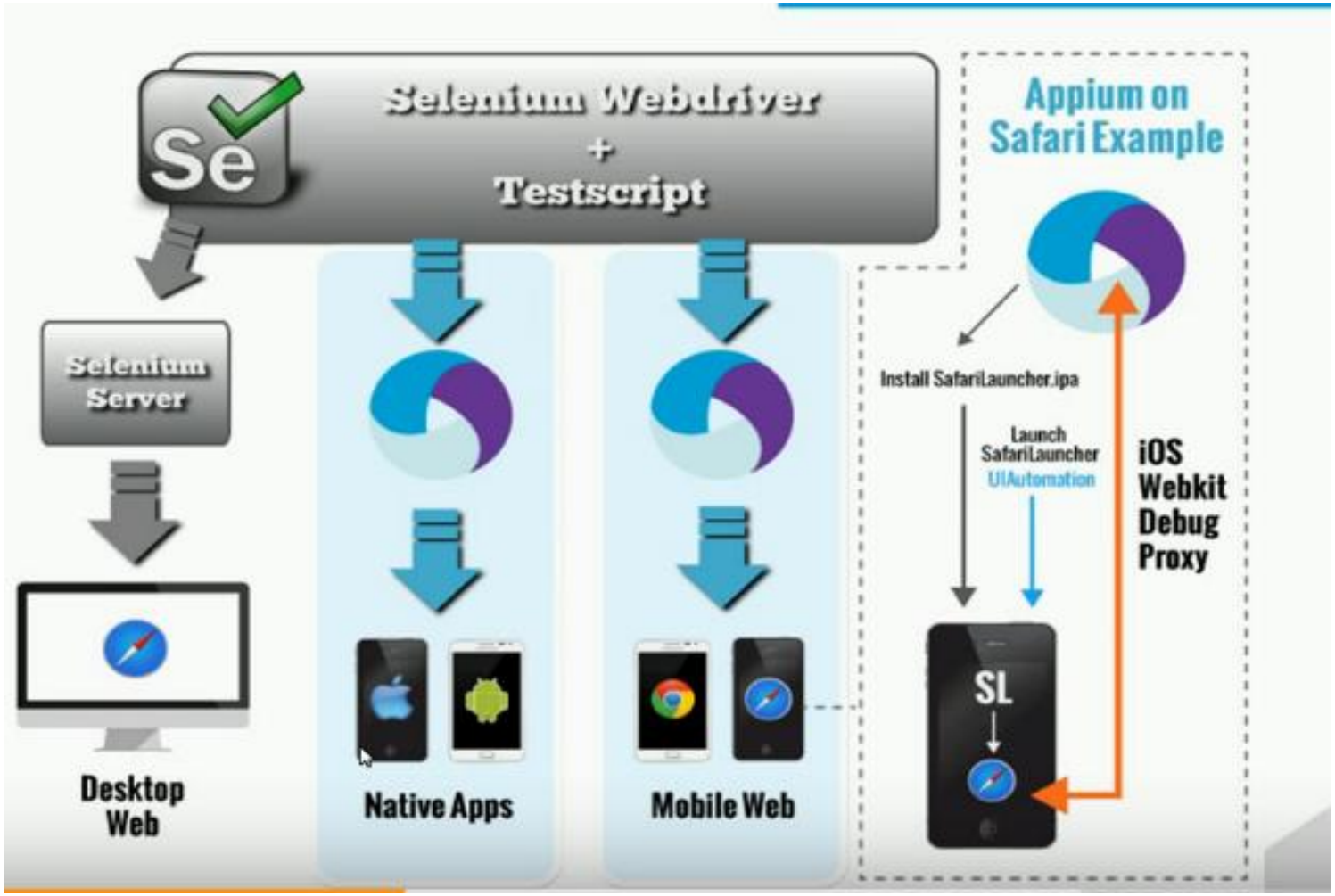
Ruby

php

python

Java

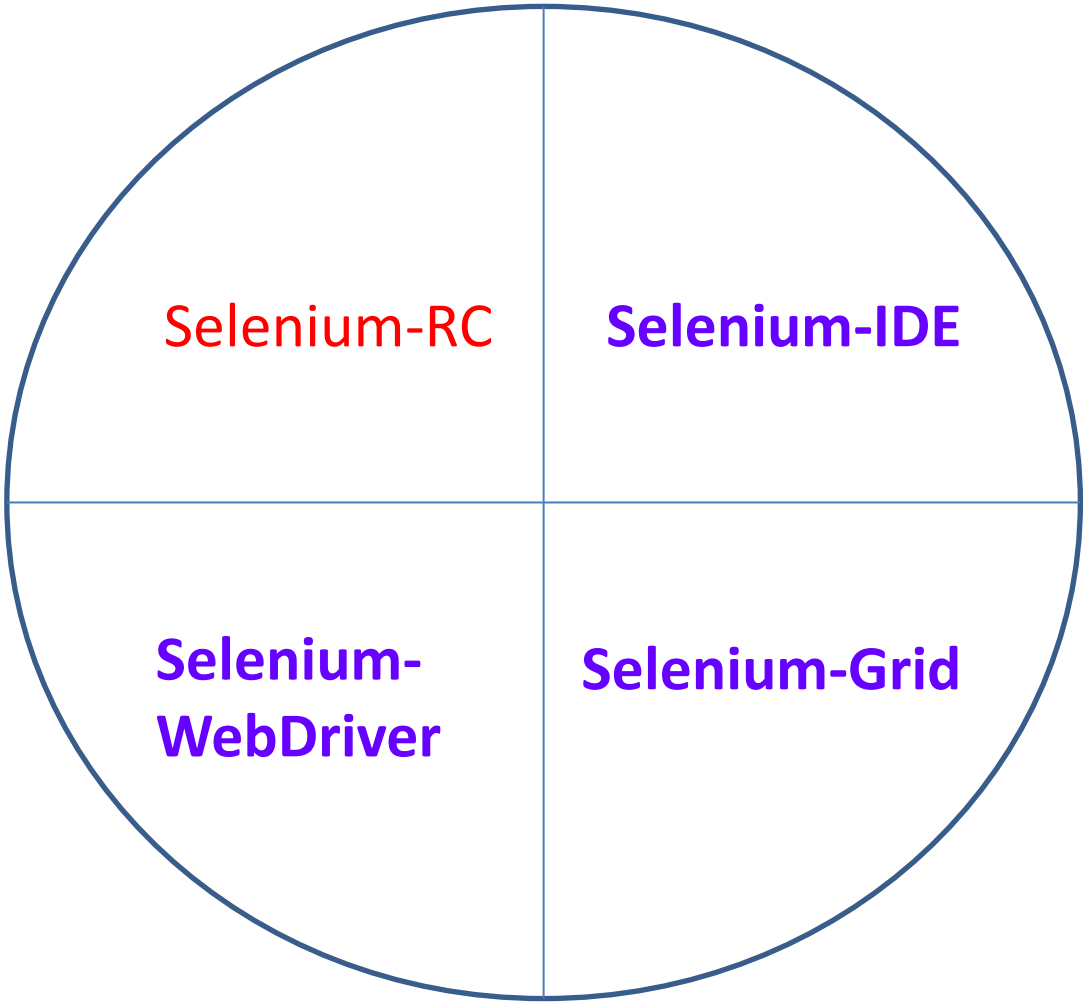
移动设备自动化测试


Selenium Webdriver
+
Testscript
Appium on
Safari Example
Selenium
Server
Install SafariLauncher.ipa
Launch
SafariLauncher
UIAutomation
iOS
Webkit
Debug
Proxy
SL
Desktop
Web
Native Apps
Mobile Web

Selenium VS QTP

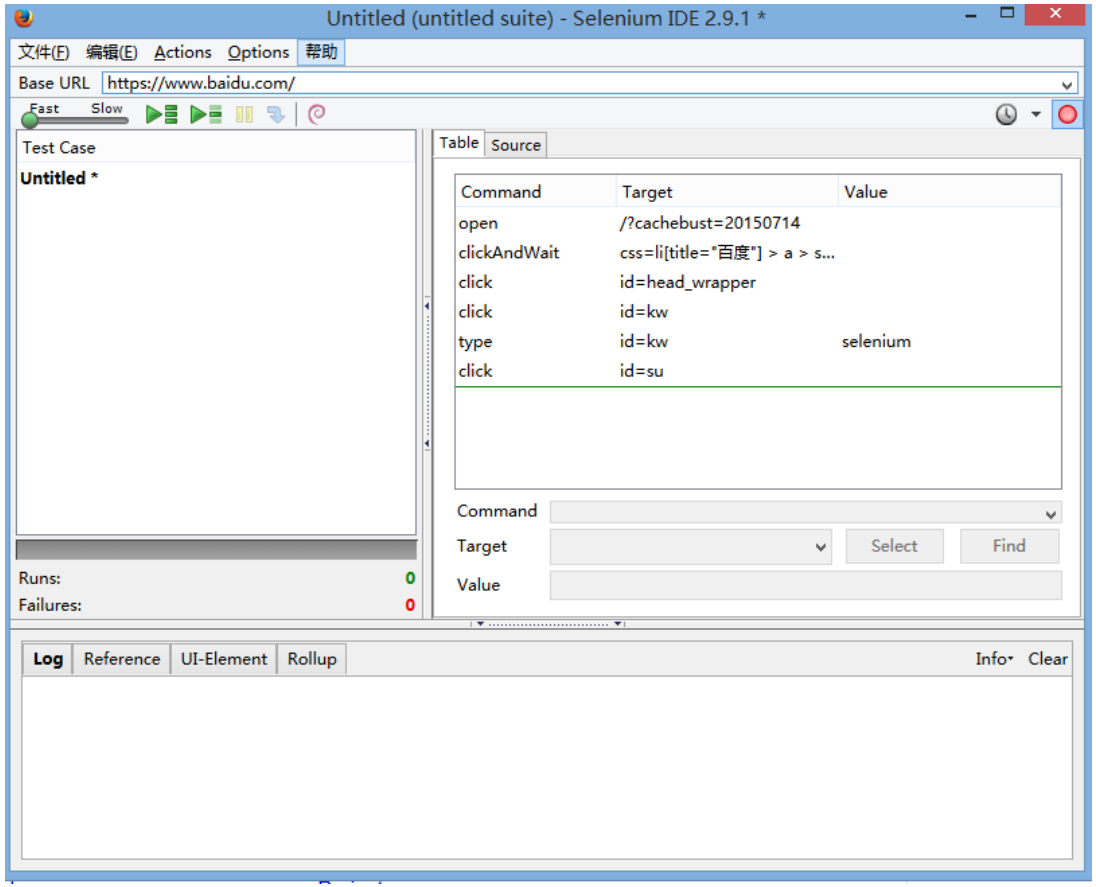
比较项	UFT	Selenium	优胜者
价钱	商用，费用昂贵	开源免费	Selenium
应用领域	基础插件支持web和windows应用，扩展插件支持.Net winform，Java Swing等应用 UFT	web	UFT
录制回放	成功率高	成功率低	Selenium
用户仿真	独占屏幕，只能开启一个独占的实例	只占用一个浏览器进程，支持同时进行多个测试	
脚本语言	VBScript	多种开发和脚本实现	Selenium
数据驱动	DataSheet实现	编程实现	UFT
对象管理	Object Repository	编程实现	UFT
浏览器支持	几乎所有的IE版本和限定版本的FF和Chrome	几乎支持主流浏览器的各个版本	Selenium
持续集成	ALM等	Jenkins等	Selenium
平台支持	Windows	跨平台	Selenium
常用发展	技术相对落后	有助于移动自动化测试	Selenium

Selenium四大组件

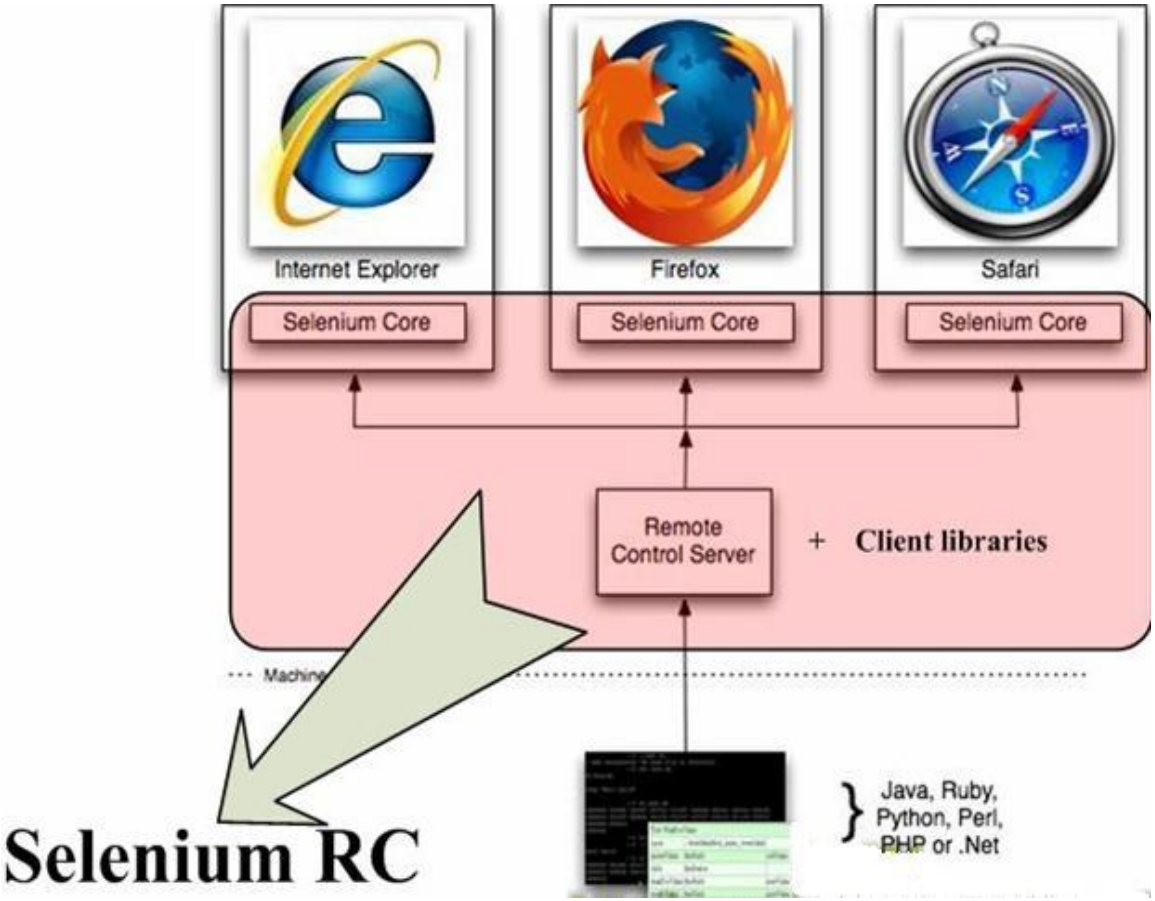


Selenium IDE

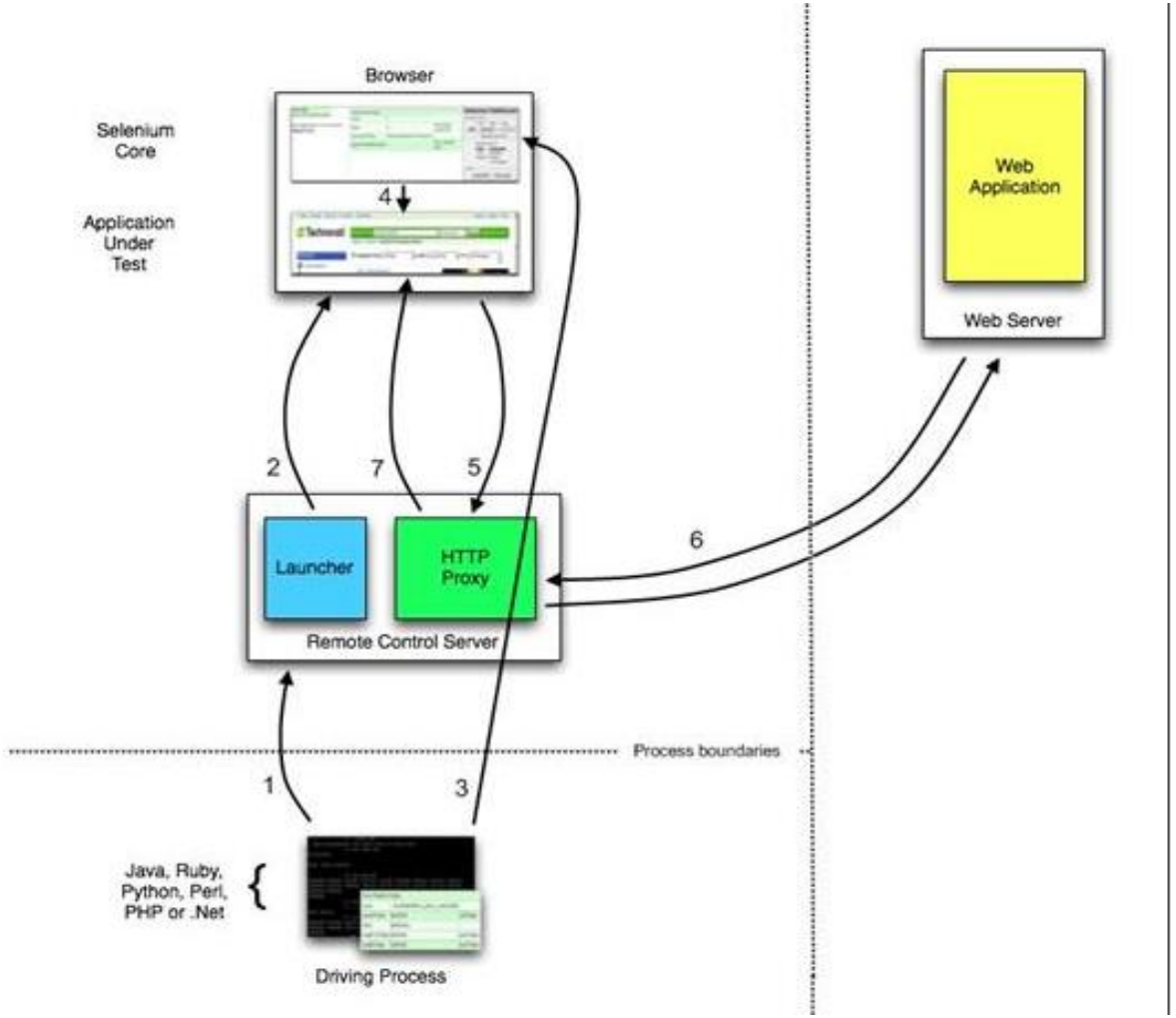
Selenium IDE 是嵌入到Firefox浏览器中的一个插件，实现简单的浏览器操作的录制与回放功能，也可以将录制的脚本导出成java、Python、Ruby、C#。



Selenium RC的实现原理



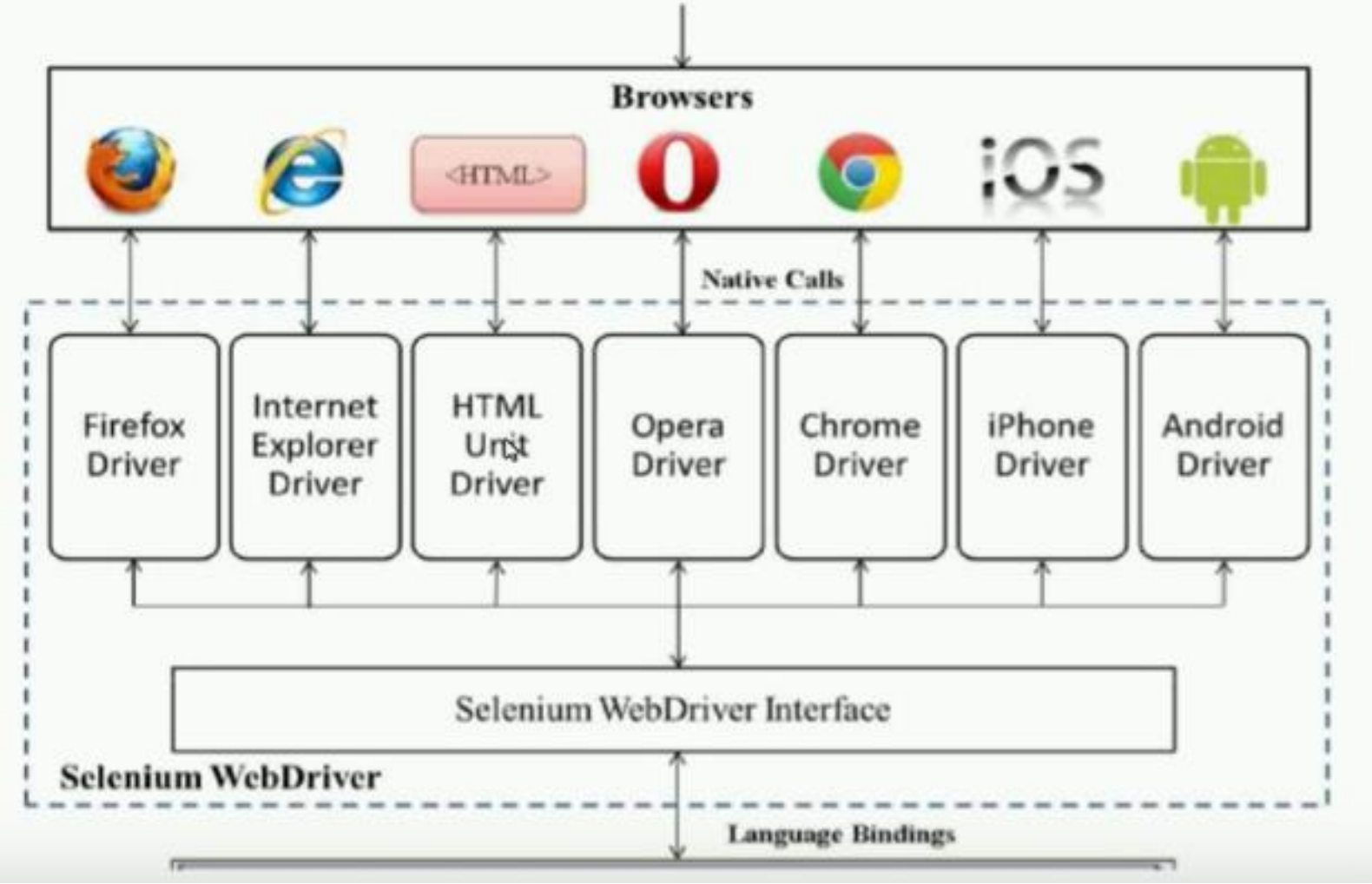
Selenium RC代理运行模式



Selenium RC代理运行模式

(1)测试案例（Testcase）通过Client Lib的接口向Selenium Server发送Http请求，要求和Selenium Server建立连接。
为什么要通过发送Http请求控制Selenium Server而不采用其他方式呢？从上文可以看出，Selenium Server是一个独立的中间服务器（确切地说是代理服务器），它可以架设在其他机器上！所以测试案例通过发送HTTP请求去控制Selenium Server是很正常的。
(2)Selenium Server的Launcher启动浏览器，把Selenium Core加载入浏览器页面当中，并把浏览器的代理设置为Selenium Server的Http Proxy。
(3)测试案例通过Client Lib的接口向Selenium Server发送Http请求，Selenium Server对请求进行解析，然后通过Http Proxy发送JS命令通知Selenium Core执行操作浏览器的动作。
(4)Selenium Core接收到指令后，执行操作。
(5)浏览器收到新的页面请求信息（因为在(4)中，Selenium Core的操作可能引发新的页面请求），于是发送Http请求，请求新的Web页面。
由于Selenium Server在启动浏览器时做了手脚，所以Selenium Server会接收到所有由它启动的浏览器发送的请求。
(6)Selenium Server接收到浏览器的发送的Http请求后，自己重组Http请求，获取对应的Web页面。
(7)Selenium Server的Http Proxy把接收的Web页面返回给浏览器。
因为浏览器存在同源策略，所以Selenium RC中的Selenium Server需要以这种代理模式运行。

Selenium Webdriver工作原理



Selenium Webdriver工作原理

webdriver是按照server－client的经典设计模式设计的:

- server端就是remote server，可以是任意的浏览器：测试脚本启动浏览器后，该浏览器就是remote server，它的职责就是等待client发送请求并做出响应
- client端（测试代码）：测试代码中的一些行为，比如打开浏览器，转跳到特定的url等操作是以http请求的方式发送给被server端（被测浏览器）server接受请求，并执行相应操作，并在response中返回执行状态、返回值等信息

Selenium Webdriver工作原理

1. 启动浏览器后，selenium-webdriver会将目标浏览器绑定到特定的端口，启动后的浏览器则作为webdriver的remote server，接受测试脚本的命令。
2. 客户端(测试脚本)，借助ComandExecutor发送HTTP请求给sever端（通信协议：The WebDriver Wire Protocol，在HTTP request的body中，会以WebDriver Wire协议规定的JSON格式的字符串来告诉Selenium希望浏览器接下来做什么事情）。
3. Sever端需要依赖原生的浏览器组件，转化Web Service的命令为浏览器native的调用来完成操作。

Selenium Grid

Selenium Grid是一种自动化的测试辅助工具，Grid通过利用现有的计算机基础设施，能加快Web应用的功能测试。利用Grid可以很方便地实现在多台机器上和异构环境中运行测试用例

Selenium Grid