

# 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！

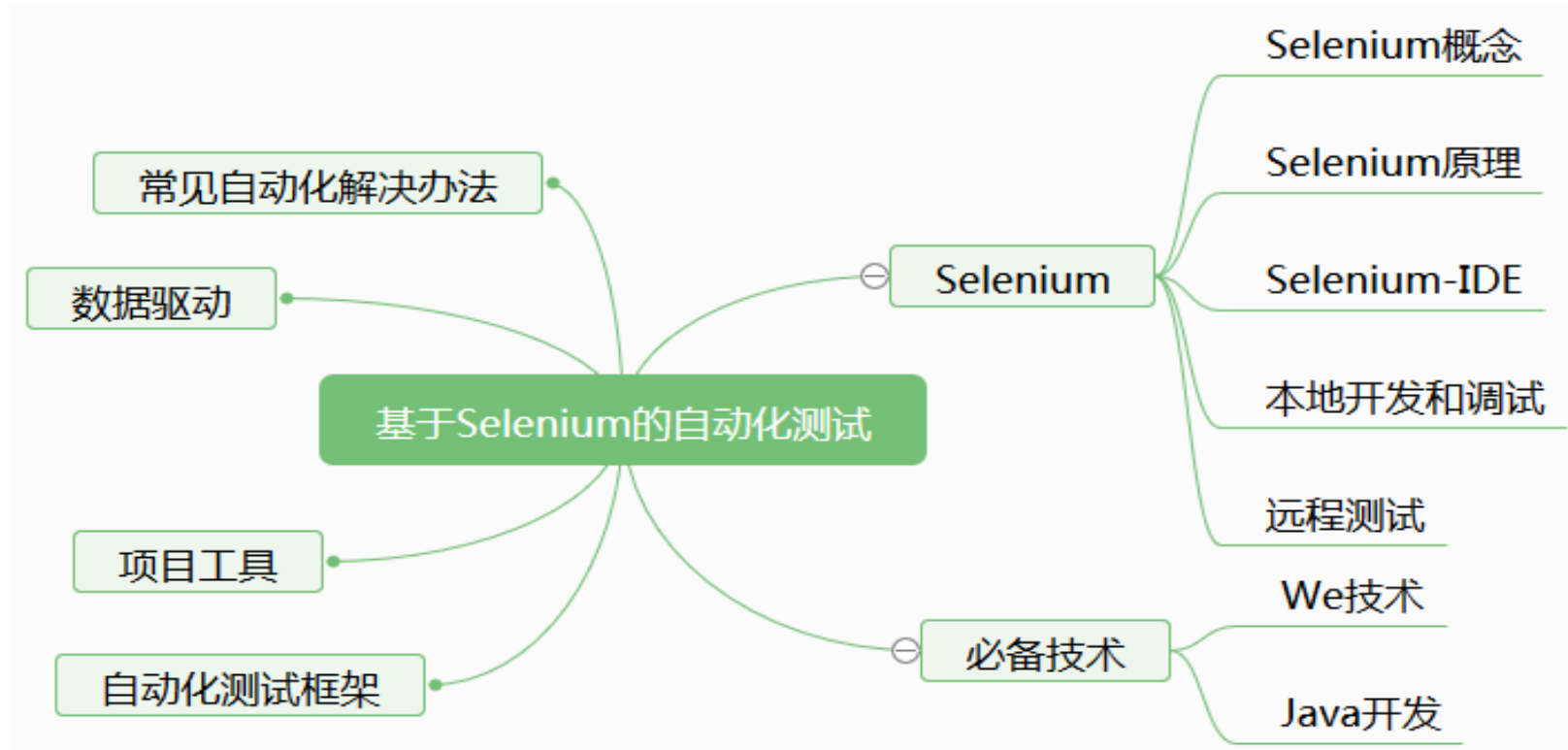
# 自动化测试

- 随着软件测试领域技术的不断创新，测试人员开始不满于仅仅使用手工测试来完成重复性的劳动，尤其是在软件领域中，我们开始关注如何使用工具或者脚本代码来让计算机“帮助”我们做一些简单的操作。
- 自动化测试就是把手工测试转化成用机器、软件、程序来测试产品的过程。通常来说，在设计了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程一步步执行测试吗，得到实际结果与期望结果的比较。这个过程会带来大量工作的重复，并且在时间紧迫的情况下会遗漏测试范围，为了节省人力、时间、资源，提高测试效率，保证测试范围，引入了自动化测试的概念。

# 自动化测试进行的场景

- 自动化测试是完全高效的吗？什么情况下选择自动化测试？
- 自动化测试并不是完全高效的，有时手工测试比自动化测试更加高效。比如应用程序自身变化频繁、项目周期短、代码不够规范，自动化脚本需要大量的修改，也没有足够的时间运行自动化脚本，甚至由于代码问题提升了编写自动化脚本的难度，需要大量时间去解决一些未知的问题，那么这时候手工测试往往更加高效。

# 课程内容



# 01 自动化测试基础

# 本章大纲

## 1.1 软件测试分类

## 1.2 自动化测试的基本概念

## 1.3 分层的自动化测试

## 1.4 UI自动化测试工具介绍

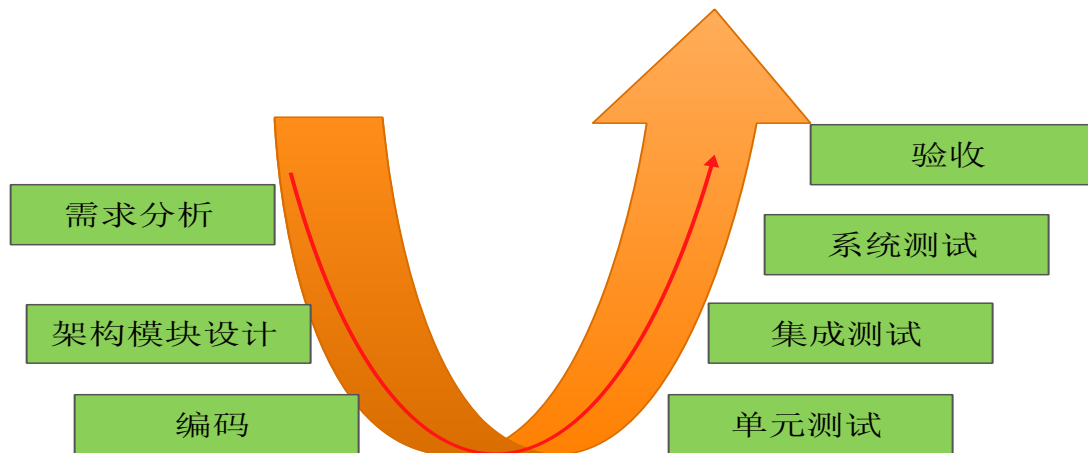
## 1.5 Selenium介绍

# 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 自动化测试的基本概念

1.3 分层的自动化测试

1.4 UI自动化测试工具介绍

1.5 Selenium介绍

# 深入比较自动化测试与手工测试

通常，软件测试的工作量很大（据统计，测试会占用到40%的开发时间；一些可靠性要求非常高的软件，测试时间甚至占到开发时间的60%）。而测试中的许多操作是重复性的、非智力性的和非创造性的，并要求做准确细致的工作，计算机就最适合于代替人工去完成这样的任务。

软件自动化测试是相对手工测试而存在的，主要是通过所开发的软件测试工具、脚本等来实现，具有良好的可操作性、可重复性和高效率等特点。

# 深入比较自动化测试与手工测试

## 手工测试的局限性：

- 通过手工测试无法做到覆盖所有代码路径。
- 简单的功能性测试用例在每一轮测试中都不能少，而且具有一定的机械性、重复性，工作量往往较大。
- 如果有大量（几千）的测试用例，需要在短时间内（1天）完成，手工测试几乎不可能做到。

# 深入比较自动化测试与手工测试

自动化测试的优势：

- 缩短软件开发测试周期，可以让产品更快投放市场。
- 测试效率高，充分利用硬件资源。
- 节省人力资源，降低测试成本。
- 增强测试的稳定性和可靠性。
- 提高软件测试的准确度和精确度，增加软件信任度。
- 软件测试工具使测试工作相对比较容易，但能产生更高质量的测试结果。
- 软件测试实行自动化进程，绝不是因为厌烦了重复的测试工作，而是因为测试工作的需要，更准确地说是回归测试和系统测试的需要。

# 自动化测试基本概念

➤ 由于自动化测试的诸多好处，那么是不是自动化终将取代手工测试呢？

➤ 那么此时引入了一个新的争论焦点：

何时选取哪种测试方法，以及何种情况下手工测试更合适或自动化测试更合适？

# 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，资源丰富。
- 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) 脚本重复利用率高。

# 哪些部分适合自动化？

- 一种观点：认为自动化测试更适合回归测试和API接口测试，手工测试更适合做验收测试和GUI测试。
- 另一种观点：应该从代码是**业务逻辑**相关还是**基础性代码**出发；业务逻辑代码对应终端用户使用的那些功能，是实际完成工作的；基础性代码则确保业务逻辑代码运行在合适的环境中。起支撑作用而彼此之间相对独立，并不存在业务关系的。
- 很明显，两种代码都要测试。手工测试更适合测业务逻辑，因为这部分人类学习起来要比交给自动化容易。

# 自动化测试基本概念

- 手工测试者最适合成为领域专家，他们可以把相当复杂的业务逻辑存在最强力的测试工具——大脑里。而且手工测试速度比较慢，测试者就有时间可以观察分析细微的逻辑问题。速度虽然慢些，但是比较容易。
- 自动化则胜在测试底层的细节。自动化可以测试错误返回值、返回码、异常和内存使用等等。速度快但是也困难些。想对业务逻辑进行自动化测试比较困难，风险也大。



# 自动化测试基本概念

自动化测试的流程：



# 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

# 自动化测试工具的特征

- 支持脚本化语言(Scripting Language)
- 对程序界面中对象的识别能力
- 支持函数的可重用
- 支持外部函数库
- 抽象层—将程序界面中的对象实体映射成逻辑对象
- 支持数据驱动测试(Data-Driven Test)
- 错误处理
- 调试器(Debugger)
- 源代码管理
- 支持脚本的命令行(Command Line)方式

# 本章大纲

1.1 软件测试分类

1.2 自动化测试的基本概念

1.3 分层的自动化测试

1.4 UI自动化测试工具介绍

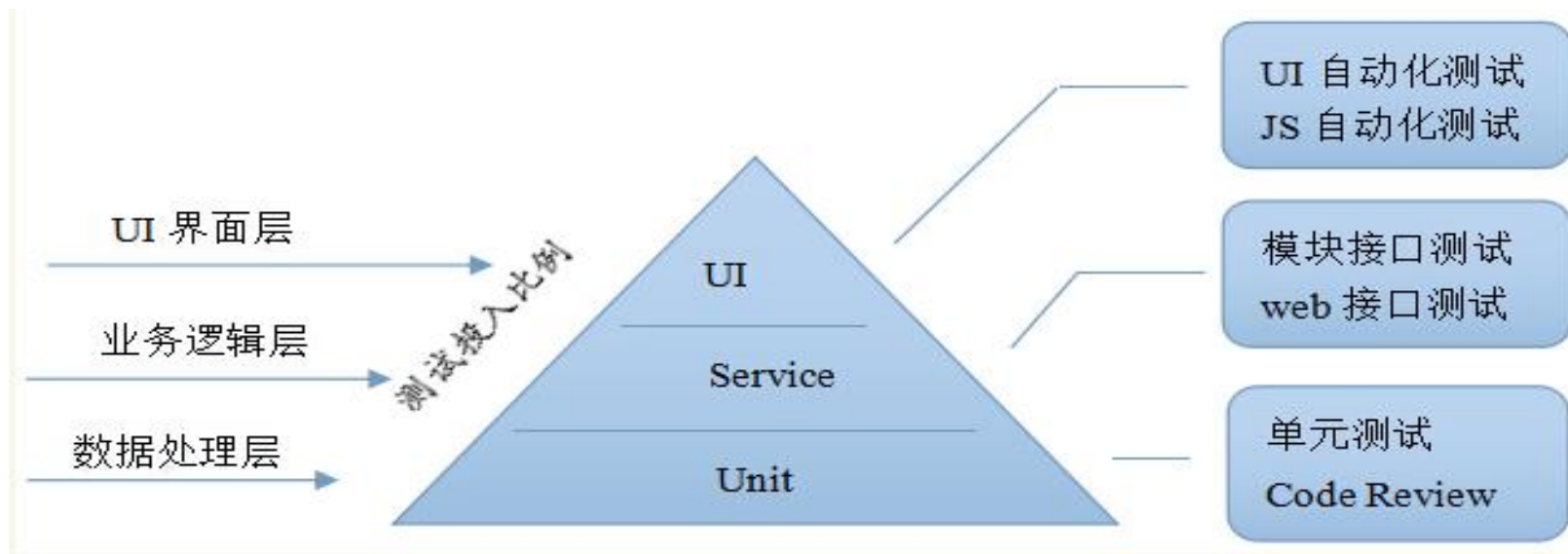
1.5 Selenium介绍

# 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 自动化测试的基本概念

1.3 分层的自动化测试

1.4 UI自动化测试工具介绍

1.5 Selenium介绍

# 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。

# RFT

RFT ( IBM Rational Functional Tester )

框架结构：

RFT 的脚本可以分别被归类为 AppObjects、Tasks 和 Testcases

AppObjects：定义页面上的元素。在测试过程中，所有用到的页面元素都定义并储存在这一层中，这一层还包含了所有页面上显示的字符串。

Tasks：定义可以单元化，可重用的任务，调用在 AppObjects 中定义的元素。

Testcases：一个 case 写成一个脚本，每个测试场景，可以写成一个或多个脚本，每个脚本只调用在 Tasks 中定义的可重用的任务。

# Watir

一款轻量级框架。开源工具，用于开发基于Web 应用的自动化测试程序。它使用Ruby 脚本语言，提供了轻量级的自动化测试程序框架和丰富的开发库，有效地加速了自动化测试程序开发。

优点：

- 1) ruby是面向对象语言，功能强大，且不用编译执行。
- 2) 轻量级，语言简单，易读易写易用。

缺点：就是既然要开源，要手动去写，又没有像python、java这样强大的支撑，索性直接去用selenium。

# Watir

```
1  # To change this template, choose Tools | Templates
2  # and open the template in the editor.
3
4  require 'rubygems'
5  require 'watir'
6
7  test_site = 'http://www.baidu.cn/'
8  # open the IE browser
9  ie = Watir::IE.new
10 ie.goto(test_site)
11 ie.text_field(:id, "kw").set("登陆")
12 ie.button(:id, "su").click
13 if ie.contains_text("生活")
14   puts "Test Passed!"
15 else
16   puts "Test Failed!"
17 end
18
19
```



# Sikuli

Sikuli 是由 MIT 的研究团队发布的新型图形化编程技术。它以图像检索技术为基础，提供了一套基于 Jython（python语言在java中的完整实现）的脚本语言以及集成开发环境。使用者可利用**屏幕截图直接引用 GUI 元素**进行编程，完成交互操作。Sikuli 一词取自墨西哥 Huichol Indian 土著语，意为“上帝之眼”，正如其开发者所说——Sikuli 让电脑能像人一样“看”这个“真实世界”。

# Sikuli

## Sikuli 脚本

Sikuli 的脚本编写遵循 Python 语法规则，其本身提供了多种自定义类及其自定义方法，其详细介绍可参见其 官方网站文档。由于 Sikuli 基于 Jython，其核心代码由 Java 编写，可在用户自定义的 Java 工程中将其作为 Java 标准类库进行引用，其官方网站亦提供了 JavaDoc 供参考。

# 其他自动化测试工具

```
4 click()
5 click()
6 wait()
7 click(getLastMatch().right(300))
8 wait(1)
9 type("www.gmail.com\n")
10 wait(, 10)
11 click()
12 type(str_username+"\t")
13 type(str_pw)
14 click()
15 wait(1)
16 print("Done")
```

# 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。

- 提供了很多插件，如：.NET的，Java的，SAP的，Terminal Emulator的等等，分别用于各自类型的产品测试。默认提供Web，ActiveX和VB。
- UFT支持的脚本语言是VBScript，这对于测试人员来说，感觉要“舒服”得多。VBScript毕竟是一种松散的、非严格的、普及面很广的语言。

# UFT新特性

## 1 ) Insight智能图像识别

图像识别一直是我們做自动化测试的阻碍之一。包含游戏自动化、flash动态的一些自动化。

## 2 ) 多脚本调试

## 3 ) PDF文本验证点

## 4 ) 支持开源CI

## 5 ) 支持移动设备

# 主流自动化测试工具

## 为什么选择Selenium ( Webdriver )

- 开源免费
- 使用灵活、简单
- 后期用例易于维护
- 支持多种语言
- 容易与单元测试框架结合
- 可支持多浏览器同时，支持远程启动其他服务器
- 高度复用性
- 代码可自主掌控，对于搭建框架、平台等有不可替代的优势

# 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/>

# Selenium与UFT

用与UFT比较的方式来了解二者的特征和优劣。

先从大方向上进行比较：

## 1 ) 价格

Selenium 是开源的自动化测试工具，但是QTP是商业版的，而且价格不菲。



## 2 ) 应用领域：

如果价格是Selenium的绝对优势的话，那么应用领域又是UFT的绝对优势。Selenium支持web页面的测试，而UFT不光支持web界面的测试，而且还支持Client方面的测试。

## 3 ) 功能方面：

UFT的录制回放成功率很高，但是Selenium的录制回放成功率就非常的低，所以我们在测试过程中也不会使用录制功能。

脚本的编辑功能。在这个方面很难做出评判，如果是熟悉java、python等的人，会喜欢Selenium的，但是如果熟悉的是VBScript可能就会喜欢UFT。

## 4 ) 框架处理的能力：

在数据驱动方面，UFT的支持很灵活，可以通过简单的设置就可以完成数据驱动的自动化脚本。

Selenium要用编程来实现才可以。但是就这点并不能说明Selenium在框架的处理能力上就比UFT差。Selenium提供了更加开阔的框架处理能力给用户本身，用户可以根据自己的实际情况开发出更适合自己的自动化测试脚本。

# Selenium与UFT

更加细节的比较：

## 1.用户仿真：

Selenium在浏览器后台执行，它通过修改HTML的DOM（文档对象模型）来执行操作，实际上是通过javascript来控制的。执行时窗口可以最小化，可以在同一机器执行多个测试。QTP完全模拟终端用户，独占屏幕，只能开启一个独占的实例。

# Selenium与UFT

## 2.UI组件支持：

Selenium 支持主要的组件，但是某些特殊事件、方法和对象属性支持不够，UFT提供了良好的支持，通过收费的插件，提供了对各种组件的支持

## 3. 对象识别

UFT做到所见即所得，用SPY插件、对象库的方式获取，Selenium提供的是各种对象识别接口来识别

## 4. 支持的平台

Selenium支持多种语言，可以跨平台。UFT只适用于windows。

## 5. 脚本创建

UFT似乎更容易一些，而Selenium略难，需要一定的代码知识能力。

# 本章大纲

1.1 软件测试分类

1.2 自动化测试的基本概念

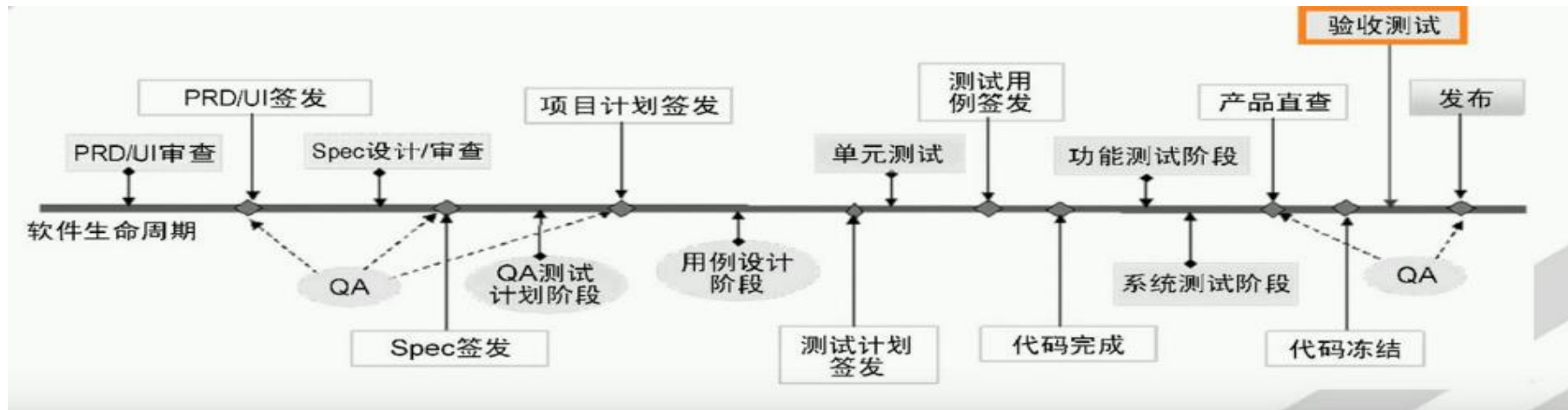
1.3 分层的自动化测试

1.4 UI自动化测试工具介绍

1.5 Selenium介绍

# 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。





# Selenium名称来源

"You can cure mercury poisoning by taking selenium supplements"

-Jason Huggins



# Selenium 简介

- 开源免费软件
- 支持主流浏览器：FireFox, Chrome, IE
- 跨平台：windows, linux, Mac
- 多语言：java, Python, Ruby, Php, JS
- 对Web支持良好，丰富简单的API

# 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



iOS

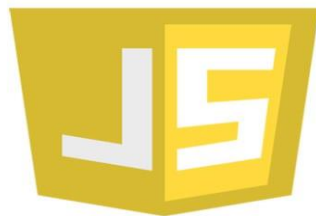


# 支持的浏览器



# 支持的开发语言

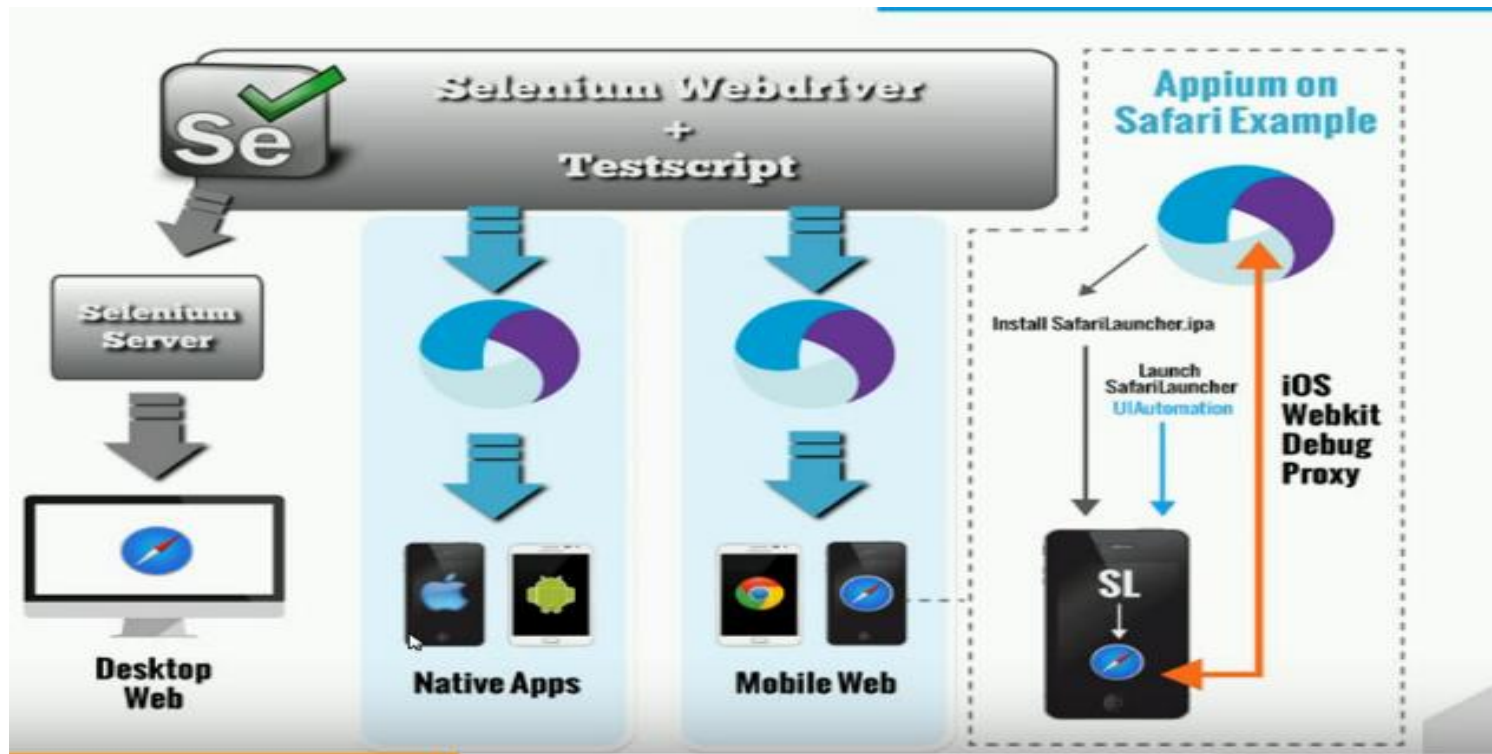
C#



JavaScript



# 移动设备自动化测试

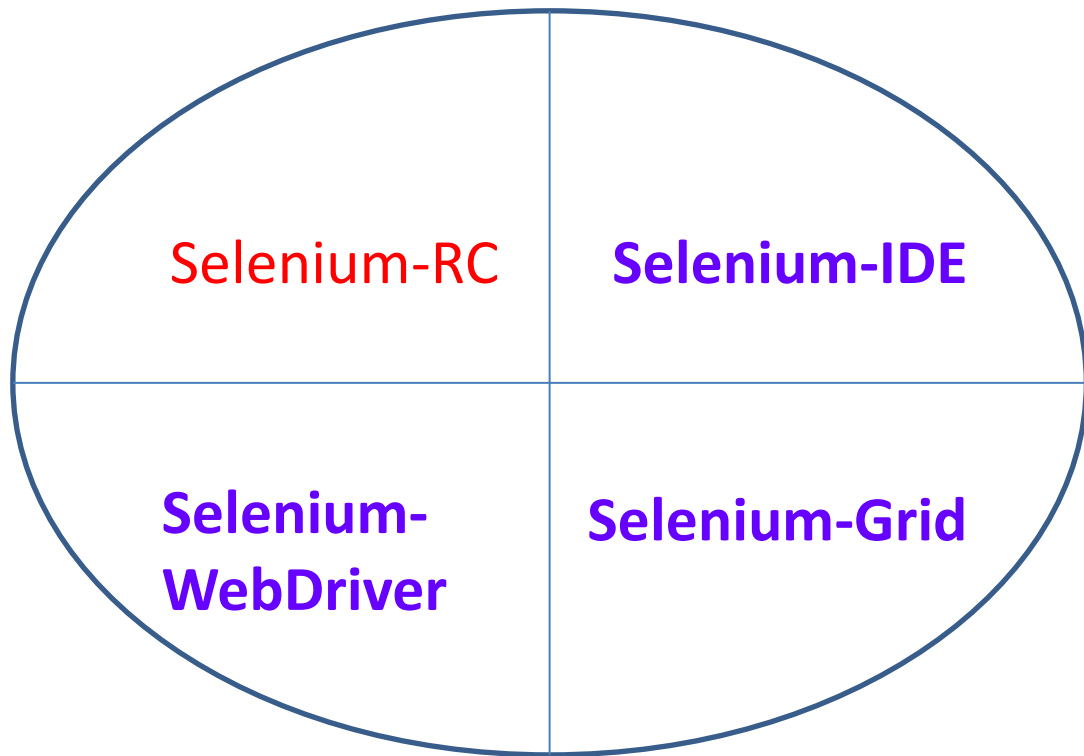


# Selenium VS QTP

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

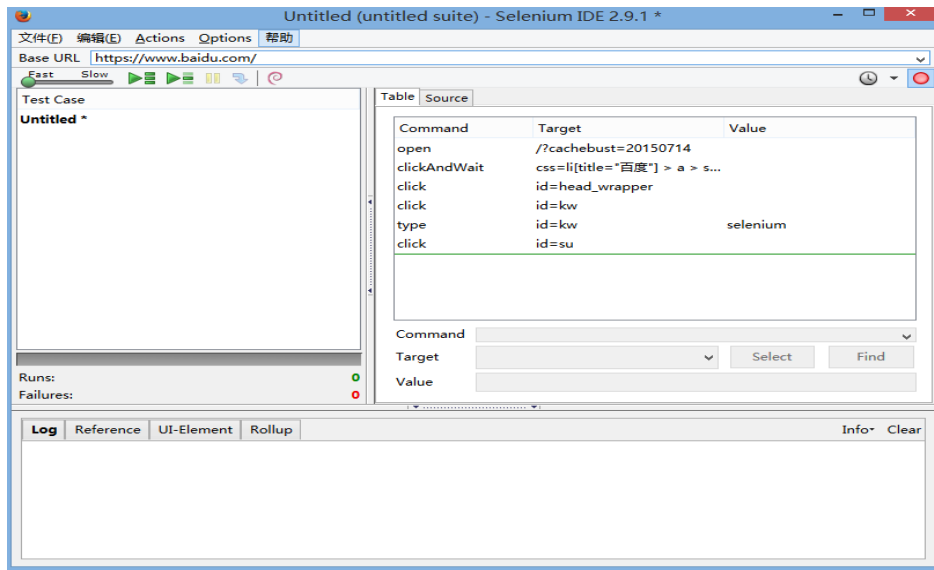


# Selenium四大组件

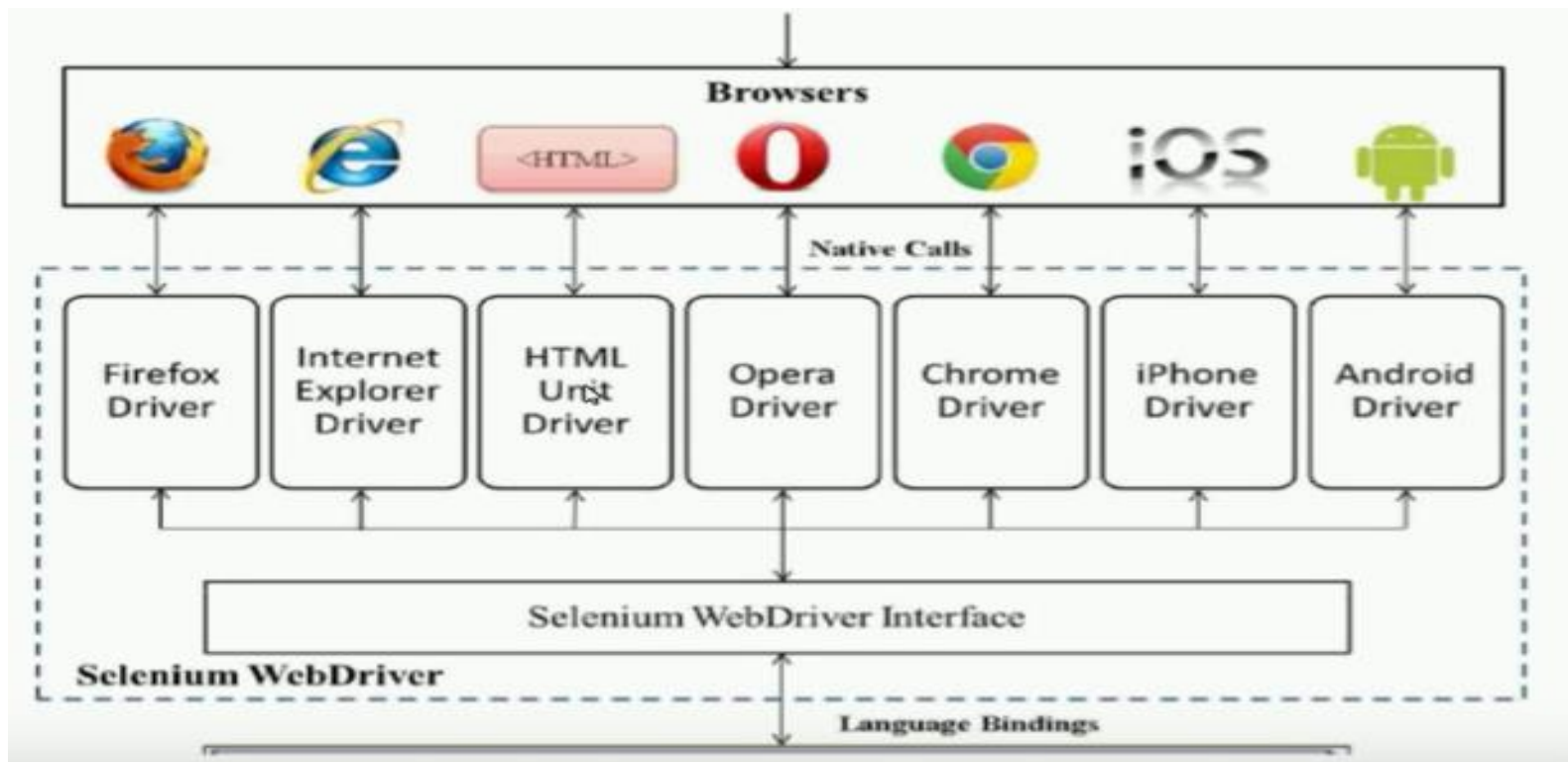


# Selenium IDE

**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 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

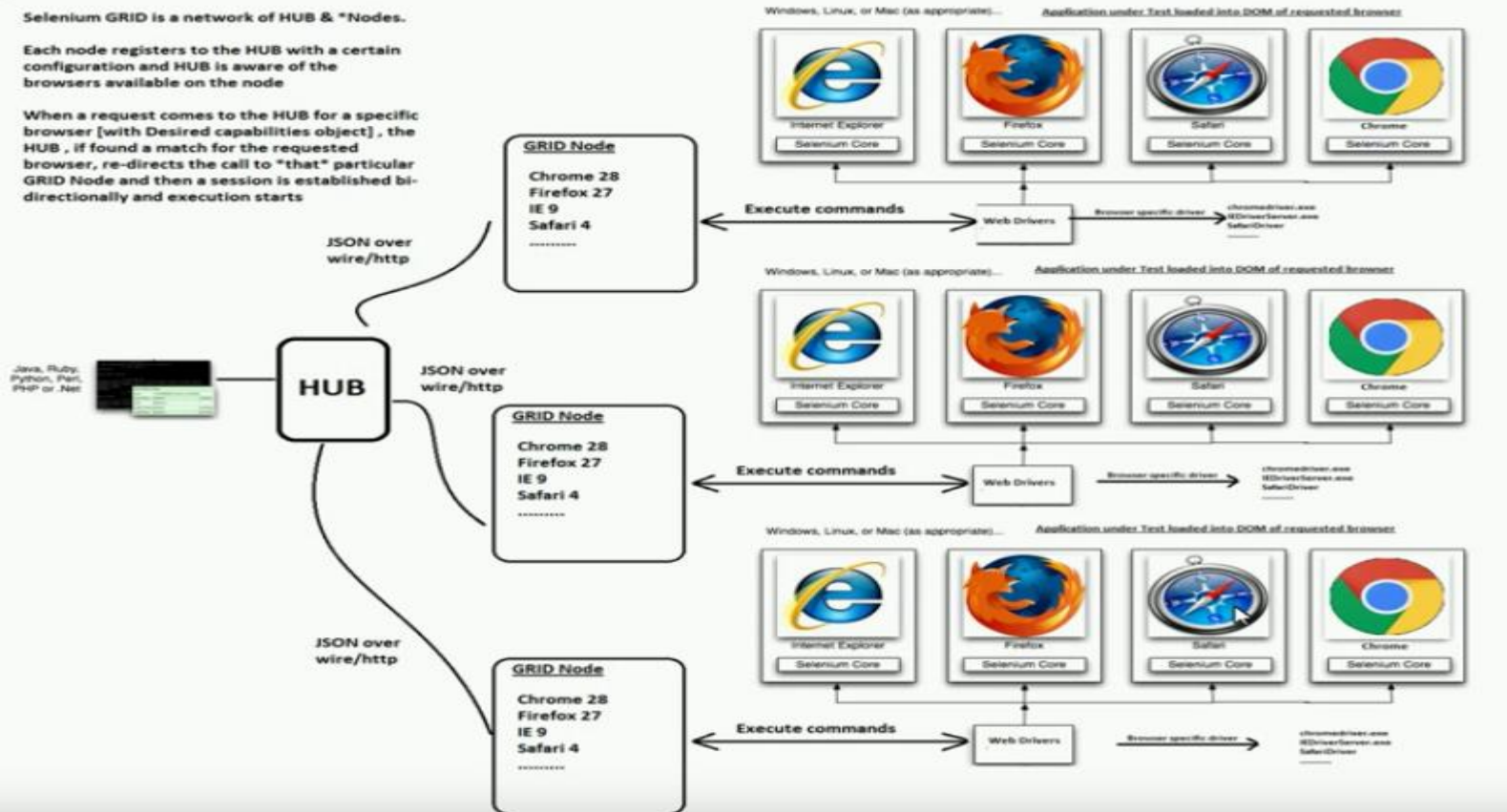
**Selenium Grid**是一种自动化的测试辅助工具，Grid通过利用现有的计算机基础设施，能加快Web应用的功能测试。利用Grid可以很方便地实现在多台机器上和异构环境中运行测试用例。

# Selenium Grid

Selenium GRID is a network of HUB & \*Nodes.

Each node registers to the HUB with a certain configuration and HUB is aware of the browsers available on the node

When a request comes to the HUB for a specific browser [with Desired capabilities object], the HUB, if found a match for the requested browser, re-directs the call to "that" particular GRID Node and then a session is established bi-directionally and execution starts



# Selenium使用的三个层次

- 基本脚本设计阶段
- 框架脚本设计阶段（数据驱动，关键字驱动，分层架构的思想）
- 平台设计阶段