框架需求功能:

1. 跨平台
2. Case（继承自unittest.TestCase）：
   1. 复用，例如一个用例可以出现在多个suite中
   2. 每个一个测试步骤即为一个以“test\_”命名的方法，每个检查点使用assert来进行判断
   3. 每个assert支持iswarning参数，表示该assert是一个警告性质的断言，不造成整个用例的失败
   4. 支持串联步骤的测试用例
   5. 支持收集关注点，例如connect time
   6. 支持收集检查点，通过assert中的msg参数来指定检查点
   7. 每个test生成一个log文件，case的每个错误（包括代码异常或assert异常）均输出到log文件中，标注各种异常的log文件，并放到专门的文件夹中
3. Suite（继承自unittest.TestSuite）：
   1. 为每个case生成log文件，初始化case的log handler，handler中只有对满足条件的log进行输出（线程名称、用例所属模块和基础库）
   2. 从TestContextManager中获取当前测试线程所对应的TestContext对象。

1. Context：
   1. 每个线程一个TestContext对象，对象中包含当前的配置和资源
   2. TestContextManager是一个应用了单件模式的对象，每次初始化该实例均返回同一个对象，通常用于多线程系统中。它维护和管理TestContext对象。

1. Resource：
   1. 包含资源对象和资源池，资源池为单件模式对象

1. Result：
   1. 保存每个用例的测试结果记录，每个记录包括用例id、用例名称、用例状态、警告、检查点、关注点等
   2. 支持多个报告目的地（例如记录在Shelve或通过Http传输到服务器端），默认为保存在本地内存中。
   3. 支持停止、暂停和恢复操作

1. Runner：
   1. 每个runner对象为一个线程对象，支持添加多个suite，在所有的suite都完成后，可以读取其result属性
   2. 每个runner对象持有一个context对象，并将该对象交由TestContextManager对象管理

1. Loader：
   1. 根据提供的参数初始化TestCase和TestSuite

以上为可复用模块

客户端单独执行所需要准备的模块

1. Report：
   1. 读取TestResult中的测试结果，生成测试报告

1. Manager：
   1. 读取xml配置文件，
   2. 调用TestLoader对象加载TestCase到TestSuite中
   3. 添加TestSuite到TestRunner中，并执行，
   4. 执行完成后，通过TestRunner中的TestResult对象调用TestReport对象生成报告
   5. 通过smtp发送报告

服务器、客户端模式：

1. Agent对象注册xmlrpc调用，通过xmlrpc调用创建TestRunner，并添加报告的回报地址

类：

1. 类名名词，首字母大写
2. 类方法动词
3. public属性和方法以小写字母开头
4. protect属性和方法以一个\_开头
5. private属性和方法已\_\_开头