

产品自动化测试的误区

Sundy 2008-3-18

一个实际案例

N年前某大型通信设备公司的测试部门发起一场轰轰烈烈的测试转型运动，驱动转型的动力非常简单：人手太紧了，要释放人力，当时该部门有 95% 以上的测试精力都投入系统测试上，导致其它测试，比如组网测试、协议测试一致性测试、性能测试，还有白盒测试根本顾不上。部门经理贾 XX 决心很大，先部门总动员，历经艰难，后来终于从各产品测试组抽调 20 多位骨干来研发自动测试工具。

半年过去，终于有一两个自动工具的初始版本做出来了，选测试组进行试验，为让自动测试跑起来，测试组额外多投了 20% 的精力伺候这个脆弱系统，后来工具研发人员几经努力，又用了半年时间，自动系统终于能稳定的跑起来了。大家舒了一口气，都指着这个系统能帮他们节约出几个人来，结果，两个版本测试周期过去了，人力没节省，反倒多搭进 50% 的人力写测试脚本，维护测试脚本。部门经理贾 XX 并不因此灰心，他认为这是测试自动化推行初期必然要经历的。

这次，他们调整策略，只从所有产品中选择 1/3 容易做自动化的测试组，给每个测试组指定一个测试自动化率指标，拿这个指标考核测试经理。如此又坚持了一年半，最后统计发现，这 1/3 测试组中又只有 1/3 是出效果，自动化测试有助于减少测试工作量，而见到显著效果的，却只是个别产品。

问题到底出在哪儿呢？

不适合自动化测试的产品领域

业务自动化测试是个香饽饽，看起来很诱人，尝一口苦涩得很。并不是所有产品都适合推行自动化测试，尤其是具有如下特点的产品。

1. 一次性产品

这类产品生命周期不长，客户需求比较确定，做出产品测稳定后，就很少升级，不会一个接一个的升级版本。

此类产品不宜做自动化的主要原因是，首次自动化的成本远高过投入，实现自动化后重用性差，自动测试是不大可能在前一两轮测试就产生效益的。

2. 接口原型或业务模式尚不稳定的产品

当产品业务的模型还不成熟，经常会变，或者子系统之间接口不稳定，经常变来变去。这样的系统不大适合做自动化。

3. 涉及过多物理设备交互

测试自动化的基础是设备仿真，如果被测产品与众多物理设备打交道，而又缺乏恰当的软件仿真或硬件工具仿真（缺少测试仪表或模拟器）时，自动化测试是难以为继的。

4. 项目周期很短的项目

自动化测试是一项长期投入，项目周期短往往只看到为自动化做投入，而等不到它产生效益。

5. 在用户界面操作表现复杂业务的产品

业务复杂意味着针对界面操作实现自动测试的代价很高，业务越复杂涉及的界面元素就越多，业务越复杂也意味它越容易变化，这对于自动测试，实现一次测试的自动化尚不轻松，更别说长期维护它了。

解决此类产品的业务自动测试，只一条途径：优化产品结构，进行抽象分层，让复杂的业务逻辑集中某几个规格化的接口（比如命令行接口、MML 接口，数据驱动或表格驱动接口）来处理。

6. 涉及过多界面美观，图像质量、音频质量等可用性与易用性的产品

针对可用性与易用性测试，业界没有太多手段将它自动化起来，而且，测试结果通不过，大家都是见仁见智。给一幅图像，甲说它是色情图片，乙可能就说它没问题。

自动测试的误区

现在我们回头看看文章开头提到的案例，问题到底出在哪个地方？

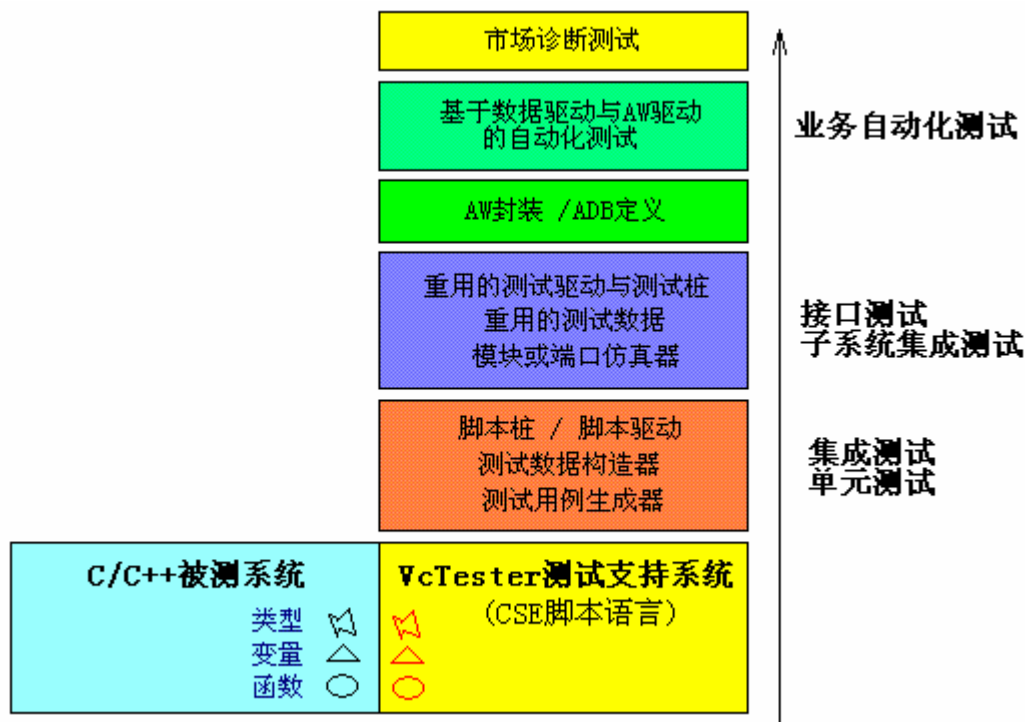
首先，测试自动化的第一个理解误区：期望自动化测试完全替代手工测试。过高追求系统测试自动化率是不现实的，也是徒劳无益的。全部门全员上马搞自动化，显然劳民伤财，如前面所提，有不少产品实际上根本不适合做自动化，或者说，能够实现自动测试，但得不偿失，没必要去实施。而且，即便面对同一易于做自动化的产品，也不该一刀切，不必追求所有测试项都自动化起来。

其次，测试自动化第二个误区：期望自动化测试发现大量新的缺陷。这个误区大概每个初次尝试自动测试的项目组都会犯。根据我们的经验，测试自动化的主要价值不是指望它发现问题，而是指望它解决重复测试问题，版本修改后用它来确保已测试通过的项目不再有问题。实际操作中，最能发现问题常常是在首次将手工测试转为自动测试的时候，这时的测试通常是半自动的，或者换一句话说，实施自动测试的项目组，在调试自动用例时最容易发现问题。

其次，测试自动化第三个误区：不关注前期测试设计，而单纯指望引入商用工具就能做好自动测试。自动测试是需要事先规划的，如果产品设计不恰当，必然导致后期自动测试难以进行，尤其是产品分布式架构、抽象逻辑分层等。如果没有事先规划，等版本出来后就很难去调整了。

最后，自动化测试还有一个最大误区：以为自动测试是后期的工作，前期单元测试、集成测试做不做，做得深不深入与之无关。对于大部分没有经历过从前期拉通测到后期的人来说，

进入这个误区是必然的。缺省对前期测试工作的重用与继承，实施自动化测试会觉得很不自
然，一种典型的自然重用形式如下图。



实际上，众多推行自动测试的项目组不见效果，或付出很大代价才推动一点点，都可以从这一点上找出原因，典型原因比如：前期单元或集成测试的成果没重用，端口难以仿真，测试数据构造困难，等等。