

## 如何在软件频繁改变时测试-----谈回归测试的策略

就像一句古老的谚语中说得：“唯一不变的是改变”。

在软件开发模型中，曾被认为最优秀的瀑布模型的一个缺陷就是假定很少或者没有改变，而现实世界是每天都在改变的。也因为如此，其他的开发模型比如“快速应用开发（Rapid Application Development, RAD）”逐渐发展成可以接收改变，并通过计划好的迭代过程利用这些改变来改进软件的开发模型。

虽然 RAD 可以帮助软件开发人员加快开发的速度，但是却也让测试人员非常头痛。因为每一次改变都有可能产生新的缺陷，而要想找到新的缺陷只有一个办法——进行全面的回归测试——对所有原来已经测试通过的部分再次测试，对返回值进行比较并找出不同的地方。

这里有两个问题需要注意：

### 1) 在软件频繁改变的时候，可能进行全面测试吗？

实际上这是不可能的。不过，这个问题本身就有问题^\_^，因为很多时候甚至都不可能在一个完全稳定的环境中测试软件。这个问题其实是想问：“在软件频繁变化的时候，能否进行有效的测试？”我们能否期望通过更好的使用人力和其他资源来完成这种测试？我们能否找到所期望的那么多缺陷？

通过对使用 RAD 方法的项目的观察，我发现软件测试过程对于发现缺陷是非常重要的，不同的过程会表现出不同的效果。因为大多数时候我们的开发过程都不是简单的重复，所以在 RAD 环境中就不能象其他环境那样——尝试着用各种方法到处看看能不能找到一些缺陷。

### 2) 在软件频繁改变的时候，有哪些策略可以使用？

应该花些时间学习怎样在不同的环境下开展工作，不过在软件频繁改变的时候有些一般的策略还是可以参考的。

.首先，你必须接受这个事实——你不可能有 6 周的时间对一个每天都在变化的软件进行测试——或者说你的老板不会允许你在每次软件改变后都用这么长的周期来进行测试。唯一的办法是你定义一个可以快速有效的完成测试任务的过程。

.进行风险评估。能够区分不同测试对象的风险级别是非常重要的，因为这样你就可以通过对不同的测试对象排列优先级，在那些很简单的问题上只花费较少的时间，而对更高的风险则给予更高的优先级和更多的时间以及其他资源。

.必须有一个确定的工作版本（基线版本），以便于你在将来进行测试的时候可以进行比较。

.自动化测试。使用捕捉/回放工具可以借助一些自动化特性帮助你来对软件进行回归测试。应该考虑花些时间和资金把一些工具融入到你的团队中，让大家都学会如何使用这些工具会对你的工作有所帮助。对于一个不愿引入新技术的组织——比如自动化测试工具——是很难在软件频繁改变期间完成测试的。这就像盖一座房子，手头上必须要有些合适的工具才行。

.自动化工具只能对你的操作进行记录和回放，这是不够的。你必须明确业务需求，设计测试用例和测试过程，制定测试计划。另外，如果人们想在长期工作过程中获得比短期工作更多的好处，就需要考虑测试用例和测试脚本。

在软件频繁改变的时候进行测试不是不可能的，但是需要快速的响应、努力工作和维护对改变的跟踪。

在软件频繁改变时进行测试同样需要一个有创新思维的团队和过程，工具自己不会工作，只有在工作中由最优秀的人员在合适的时机使用才会产生最好的效果。使工具、人员和过程达到一个最理想的结合是一件非常有挑战性的事情。

参考资料：“Testing During Rapid Change” By Randy Rice, CQA, CSTE

---

**【郑重声明】**

本人依法保留对本文的所有权利，未经本人允许，任何人不得擅自将本文中内容用做任何盈利性的用途。为尊重作者的劳动成果，如需转载或引用，请注明文章出处及作者。

[moc.liamtoh@nahciekcaj](mailto:moc.liamtoh@nahciekcaj)