**[测试经验分享](http://wiki.primeton.com/pages/viewpage.action?pageId=10847481)**

     测试工作是软件质量把关的一道重要关卡，今天我主要总结自己在之前两个项目中做的测试工作和体会，根据自己的所做和所感，给以后的工作作一些指引，希望以后的工作自己能做得更到位、更有价值。

**1、工行敏捷开发模式**

我在工行主要的工作内容：测试方案设计、测试用例设计、自动化测试用例设计、以及功能测试和集成测试和系统回归测试。

      工行项目组实行的开发模式是敏捷开发模式，整个测试的工作也是依着敏捷开发模式来展开的。敏捷开发模式是一种以人为核心、迭代、循环渐进的开发方法，需要 RDT(需求、开发、测试)团队的三个角色的人相互协作，迭代前进才能对变化的需求做出快速的反应，及时地调整工作的方向，很好地完成任务。RDT重要角 色之一就是测试。

**2、敏捷开发模式对测试的要求和体现**

       敏捷开发模式中对于测试工作的一个要求:与需求、开发同步。以往的模式普遍认为测试是在开发完成之后才开始进行的，这样不仅在项目的开发过程中出现了人员 安排的搁置，而且测试的推迟介入，也会给软件的质量带来一定的风险。因此，测试的尽早实行，能更好的展开测试工作和预防bug的产生。我个人认为好处体现 在下面几点：

      (1).测试与需求、开发同步，讨论需求的时候同时与客户交谈，三种角色的人可根据需求同步完成各自的任务，需求完成需求文档，开发完成设计方案，测试完 成测试方案等等。需求变更时，各自可以快速的做出反应、调整和修改。如此每个人的工作明确，不存在一部分人在忙碌时，而其他人在等待的局面。

      (2).在把握客户的需求时，测试可以运用测试独有的思维，站在客户的角度，给需求文档和开发设计方案的形成提供比较中肯的意见，让需求和设计把握的更准确，防止彼此之间理解有偏差。

      (3).将隐患的缺陷扼杀在摇篮中。开发人员在理解需求的时候，更多的时候是站在设计的角度去想的，有时会有所偏差以及忽略一些用户的体验，测试在review设计方案时，可尽早发现缺陷，把即将出现的缺陷给扼杀住。

      (4).将70%的软件缺陷消灭在开发阶段。功能开发的过程当中，借助持续集成框架，每天集成版本，依据测试方案和用例进行功能测试，尽早的发现缺陷，大大减少软件系统测试期的压力和风险，使软件的质量更有保障。

      敏捷开发模式，强化了RDT小组的适应能力和凝聚力，很好地将风险控制在软件开发的每个阶段。然而，如果让敏捷开发的威力更大，自动化测试就必须展现应有的威力，这样敏捷开发模式才能实行的游刃有余。

**3、自动化测试**

实行自动化测试，测试人员也必须具备开发功底，根据需求和功能设计的要求，分析出可能出现的不同场景，将每一个场景用代码实现成一个个测试用例，在每天集 成版本后，运行自动化测试，根据生成的报告，判断当前版本的稳定性，做出相应的修复，控制软件质量的风险。自动化测试确实可以大大提高测试的工作进度，给 敏捷开发模式的实行也提供了有力的支持，对于原有的一些场景，自动化测试用例做出了有力的保证，减少了测试人员许多重复的工作，而投身到新需求的评审和新 功能的验证上。但自动化测试不是万能的，它也只能确认已经分析好的场景，对于没发现的场景和缺陷，还必须结合人工的测试才能更好地软件的质量。

**4、在上海银行的测试感悟**

       在上海银行的项目中，我的主要工作是编译环境搭建和交易监控功能开发。工作更多是偏向于功能开发，所以我的测试也只是属于自测，行内俗称：吃自己的狗粮。 开发某块功能之前，自己也会把实现场景理清，把功能拆分成小模块，分析每一个模块需要到哪方面的技术，有哪几种实现方式，哪种才能把代码写得更强壮些，功 能更完善。最终我选择了一种我认为最合适的方式进行开发，待开发工作完成后，总感觉开发前做了那么多评估，并且功能实现的方式是认为最好的，自己就比较容 易沉浸当前的实现方式中，很少去怀疑过代码可能出现的缺陷。结果当其他同事给我review时，却发现自己的代码还是存在许多的不足，总结之后，才发现: 自己对当前自以为最好的实现方式缺少了测试特有的一种怀疑态度。

**5、总结**

      1. 预防胜于治理。代码的好坏，更多决定在开发人员设计的阶段，而不是说一个功能完成后，才投入大量的测试改出来的。如果一块功能在前期设计不够好，不管是开 发和测试，都必须要投入很大的成本，才能挽救之前的漏洞。所以，测试的要求不能总是停留在发现bug，更应该花时间去预防bug。

      2. 怀疑的态度。经过两个项目的磨练，由测试向开发的角色转变，站在测试的角度去看，确实体会到测试思维在软件开发周期中的重要性，必须严格把关。站在开发人 员角度来说，在设计前都会进行各种场景的考虑和最优实现方案的筛选，可往往实现完成后，多少会对自己的功能缺少一种怀疑的态度。然而，任何事物都不可能完 美，所以就必须要具备测试的怀疑态度和测试方法才能让软件更趋向于完美。