

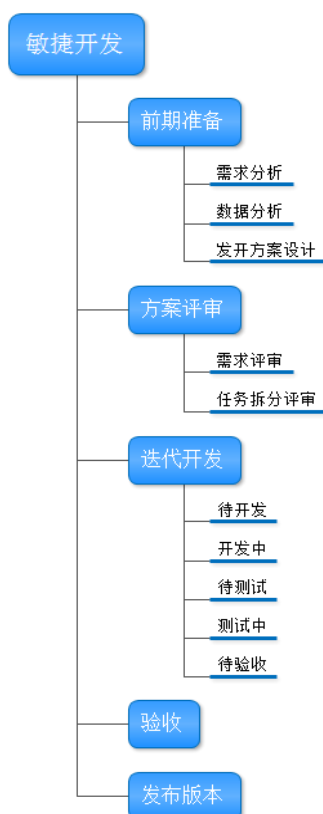
# 如何保证软件测试的质量

首先阐述下我对软件质量的理解，软件质量是软件产品对于用户的价值，这里面包含两个含义，一个是“正确的软件”，另一个是“软件运行正确”。前者是说软件要能够满足用户的需求，为用户创造价值，关系到软件的成败。后者是说软件没有 Bug 或者有极少的 Bug，性能良好，易用性高，是一个运行良好的软件，关系到软件的好坏。当二者相辅相成，达到了一定的要求，我们就可以说这个软件是一个质量良好的软件。

如何保证软件的质量，测试的重要性不言而喻。需求分析的质量可以由用户、开发人员和测试人员来判断。软件的开发质量可以由测试人员来判断。而作为内部最后一道关卡的测试，其质量又该如何保证，下面让我来阐述几条我的观点。

## 一、保证软件测试质量选择正确的开发流程很重要

现如今的大多数的软件开发面临的情况是时间紧任务重。传统的开发模式，使我们的测试人员只有在验收之前进行测试，不仅发现软件质量的问题多，而且往往很致命，导致在软件开发生命周期中，修复 Bug 的成本远远高于开发项目的成本。所以现如今的公司都抛弃了传统的开发模式，大多数选择了敏捷开发。我实习过的公司也是同样选择的是敏捷开发，所以对于敏捷开发的好处深有体会。敏捷开发面向市场变化，面向客户需求变化，采用的是迭代反馈的方式管理项目。迭代流程如图所示



敏捷的每一次迭代开发，都要将开发任务拆分为许多个 story，也就是不同的功能模块，每个开发人员负责一个或多个 story。可以从图中看出，当一个功能开发完成后提交测试模块，测试人员针对性进行测试，同时开发人员这时候可以开始进行其他 story 的开发。避免了传统开发流程需要等待测试结束才可以进行下次开发的情况，大大节省了时间和人力。整体迭代测试中虽然增加了测试时间，但是并没有延误整个的项目计划时间进度，因为在每一个迭代过程中测试过程都是提前开始的，也避免了由于开发延误导致测试时间不足的情况，所以敏捷开发无疑保证了测试人员充足的测试时间。

敏捷开发另一个显而易见的特点就是晨会。开发、测试、PM 等团队成员每天花不到半小时召开晨会，沟通各自的进展，列出项目中的问题，一起讨论解决方案，问题及时解决，而且加深团队伙伴的信任，激发工作热情。对于项目进度来说是一种助力。而以前的开发流程中，除了测试人员提出 Bug 和开发人员修改 Bug，相互的交集就比较少，开发人员只管开发的事，开发完成后修复 BUG 就可以了，但另一方面测试人员会不停的测试版本，这样带来的问题就是修改 BUG 不及时，有可能会出现开发完成某个功能很久之后测试才提出这个问题，开发再去修改，BUG 越往后修改效率就越差。相比之下，敏捷开发在安排任务时，就把开发，测试，需求无形中绑定在一起了，过程中把控的方式手段也有很多，比如我们常用的晨会、每日汇报、周报等。项目可的可视化方式也有很多，比如常用的任务图，鱼骨图，Bug 池等。

## 二、保证软件测试质量自动化测试是很重要的手段

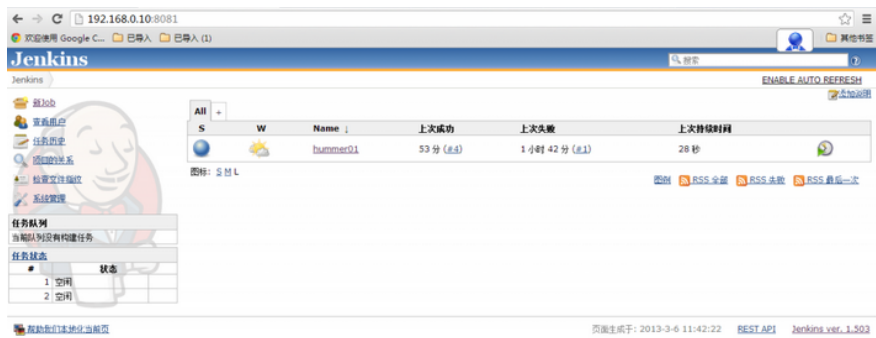
每一次代码的提交，都会产生不可避免的范围效应，一个测试人员纵然再厉害也难以同时发现很多问题。这时候，自动化测试无疑是我们最好的选择，自动化测试能帮助去覆盖那些还没有来得及跑或者根本不会有时间去跑的测试，并发现可能的问题。比如说，新功能的引入很可能会给旧的功能带来 Bug。自动化测试不但帮助测试人员检查测试质量，而且一定程度上保障了测试的质量。

在这里我要提一个持续集成的概念，即团队开发成员需要经常集成他们的工作，通常每个成员每天至少集成一次，也就意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽快地发现集成错误。许多团队发现这个过程可以大大减少集成的问题，让团队能够更快的开发内聚的软件。

持续集成可以在任何时间发布可以部署的软件。利用持续集成，可以经常对源代码进行一些小改动，并将这些改动和其他的代码进行集成。如果出现问题，项目成员马上就会被通知到，问题会第一时间被修复。不采用持续集成的情况下，这些问题有可能到交付前的集成

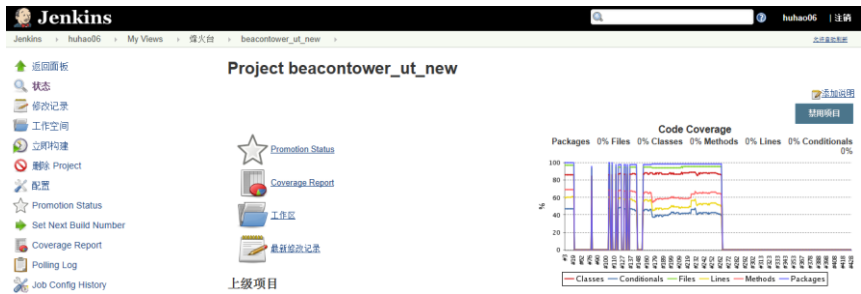
测试的时候才发现，有可能导致延迟发布产品，而在急于修复这些缺陷的时候又有可能引入新的缺陷，最终可能导致项目失败。

根据我的经验，达到持续集成，需要合适的工具。而我个人使用最多的是 Jenkins，它用于监控持续重复的工作。比如当 SVN 或者 Github 中的代码有变动时，可以自动监测到，并主动去编译项目，执行脚本。减少重复的过程可以节省时间、成本和工作量。通过自动化的持续集成可以将这些重复的动作都变成自动化的，无需太多人工干预，让项目人员的时间更多的投入到更高价值的事情上。Jenkins 软件如图所示。



Jenkins 强大而且支持插件，测试人员可以在这里部署项目，并为项目量身定制一套执行计划。比如制定执行时间和频率，每次执行编译完项目后，主动执行单元测试脚本，将所有单元测试跑一遍，并将执行结果以本地文件或者 Email 的形式发送给测试人员。这样就节省了很多人力，测试人员也不用实时关注单测，通过控制台的输出信息一目了然。一旦有问题，及时发现，及时修复。

Jenkins 的插件功能极其强大，以往很多的输出信息可以通过图表的形式展现出来。比如说单测覆盖率，行覆盖率，增量迭代覆盖率等，通过配置插件都可以直观的显示出来。如下图中的单测覆盖率显示。



这样一来，通过每次直观的展现，测试人员就可以检查项目是否健康，以及项目的各项参数因素波动趋势，提前预测问题的发生。

### 三、保证软件测试的质量测试人员测试能力至关重要

测试人员最基本的技能就是需要保证测试设计和用例的质量，测试人员负责测试设计，

完成用例设计，进行测试的相关评审，请需求、开发人员、项目负责人进行一次评审，告知项目人员将进行哪些测试。每个功能点测试流程，重点关注哪些问题等等。如果大家觉得有漏掉点，需要询问具体遗漏点，进行相应补充查漏补缺编写一份详细的测试用例提高测试质量。详细的测试用例应该完全覆盖了逻辑的所有分支。有这样的一份测试用例在测试人员手中，都能高质量的执行测试过程，因为测试用例完全覆盖所有分支，测试人员就不会因为不熟悉业务而遗漏需要测试的需求。

有了测试用例，测试人员还需要很好的执行测试工作。完成测试的工作质量很大程度上要依赖自身必备的测试技能。所以测试人员不但要具备基本的测试技能，还需要不断学习以提升测试技能，从而提高软件测试的质量。

测试人员提高测试能力，首先提高自己的测试理论基础。比如，等价类划分，边界值，条件覆盖等等测试理论。这些方法所有的人都知道，但是未必所有的人都能说明白。首先要理解测试的理论基础，使理论与实践相结合，做起工作来事半功倍。其次，需要懂得运用各种测试工具，比如 Selenium（WebDriver 脚本模拟 Web 行为），Jmeter（压力测试工具）等等，懂得这些工具的运用，可以使得工作效率提高，并且简化了很多繁杂的测试流程，为测试工作提供了便捷的方法。最后，重中之重的是测试人员要有测试的素质。必须具备一种对于软件质量的意识，要充分认识到质量对于软件成败的一种决定性因素，只有心中有质量的这种观念和想法，才能在日常工作中严谨的对待发现的每一个软件的缺陷。

#### 四、保证软件测试质量的辅助方法

##### 1) 投入更多的测试人员提高软件测试质量

更多人进行软件测试，每个人从自己关注的角度使用软件会发现不同的问题，弥补人手不够造成的测试 BUG 被忽视的问题。但是人多花在交流沟通的时间成本就会增加，需要权衡利弊。但是如果应用得当，会从一定程度上提高软件测试的质量。

##### 2) Beta 测试

Beta 测试通常是由开发团队之外的人员进行的，且通常他们的领域知识比开发团队更丰富，因此他们往往能发现测试人员“不知道自己不知道”的风险。如果测试人员的测试方法和思路偏离实际使用太远，在这个环节中就会暴露很多问题。虽然这个环节的测试也有些偏后了，但亡羊补牢未为晚矣，至少这是部署到生产环境之前的一个再次检验和补救的机会。

#### 五、小结

以上是我个人对测试质量保证的一些看法，说的并不完全，只是对我自己比较感兴趣的几点做了详细的阐述。软件的开发流程配合合理的软件质量保证方法，可以提高软件的质量。

但是值得说明的是：软件测试人员的水平高低会严重的影响测试质量，这需要我们在实际工作中特别加以注意。