

测试中的常见问题-测试流程部分

陈绍英

MSN:tiger19790520@hotmail.com

Mail:peaksoftchen@163.com

注：本文摘自作者正在编写的《软件测试管理》一书。

1、测试人员要需要何时参加需求分析？

原则上，测试人员对需求了解得越深入对测试工作越有利，所以最好一开始就应该参加需求分析工作。这样可以带来如下好处：

- 测试人员全程参与需求分析，对需求了解很深刻，减少了很多与开发人员的交互，节省了时间。测试人员参与前期开发讨论，直接掌握了不清晰的需求点；
- 早期确定测试用例的编写思路，为测试打好了基础；
- 可以获取一些测试数据，为测试用力设计提供帮助；
- 可以发现需求不合理的地方，降低了测试成本。

测试人员主要的工作之一就是确认系统是否正确实现了需求。测试人员不参与前期的工作，就只能依赖最后形成的需求文档，甚至由开发人员来讲解需求，而这些需求可能发生了“问题”，因为这个需求是已经经过分析的需求，很多的内容可能与用户的真正要求发生了偏差。同时如果只看最后形成的需求文档，对需求也会有理解上的偏差。因此作为测试人员要尽可能的获取到“第一线”的需求资料，才能真正地了解用户的业务，从而更好的对系统进行测试。

当然，如果测试人员不能参与需求环节，一定要通过其他途径保证需求的精确性，例如和开发人员进行集中讨论需求疑问的项目会议，并且一定要加强测试案例评审，甚至于是测试需求的评审。

2、系统测试阶段低级缺陷较多怎么办？

在系统测试阶段，如果仍有很多低级缺陷，说明测试对象是不合格的，没有达到测试标准。如果系统阶段发现的简单缺陷（也就是不应该有的缺陷）较多，最好停止测试，转由开发人员进行测试，发现问题立刻修改，因为这种由测试人员进行的成本较高，反复交互还会耽误进度。

建议建立预测试制度：系统测试前对核心模块进行抽查测试，如果问题较多（例如平均每个核心模块发现 10 个以上缺陷），就可以停止本次测试，直到抽测后发现问题较少才可以启动系统测试。

3、缺陷流落到客户那里有什么后果？

如果软件缺陷被遗漏并流落到客户那里，结果就是代价高昂的电话或者现场支持费用，还可能需要修复、重新测试和发布新的产品，更糟糕的情况是产品要被召回甚至被客户起诉。这种成本付出非常高，几乎是在内部修改缺陷的几何级数倍。

质量之父 PhilipCrosby 把质量的费用分为整合费用和非整合费用两类，整合费用是指与一次性计划和执行测试相关的全部费用，用于保证软件按照预期方式进行。如果发现缺陷，经过一系列的缺陷处理流程而解决缺陷，这种费用就是非整合费用。PhilipCrosby 在自己的作品中详细论述了内部的整合费用和内部的非整合费用之和远远小于外部也就是客户引起的非整合费用。

总之，软件缺陷一定要尽可能的在内部解决，这对节约成本、提高产品知名度都大有裨益。

4、什么是冒烟测试？

冒烟测试从操作上是一个随机的测试，操作对象通常是核心业务模块。测试员任意操作，要是发现多数功能走不下去（大概 20%），那么这个冒烟测试就算是结束了。冒烟测试一般不用参照测试用例。

执行冒烟测试的目的是对要测试的产品进行一个大概的度量。如果冒烟测试不能通过，通常不会启动测试计划。因为软件缺陷较多的情况下，启动测试计划会浪费更多的人力和物力。通俗的说，对“垃圾”产品执行测试实际是测试人员抢了程序设计人员的工作，这些缺陷应该在开发阶段消灭，只有这样才可以真正的节约成本。

5、在集成测试的时候，已经对一些子系统进行了功能测试、性能测试等等，那么在系统测试时能否跳过相同内容的测试？

因为集成测试是在仿真环境中开展的，那不是真正的目标系统。再者，单元测试和集成测试通常由开发小组执行。根据测试心理学的分析，开发人员测试自己的工作成果虽然是必要的，但不能作为成果已经通过测试的依据。

为了保证测试的客观性，应当由机构的独立测试小组来执行系统测试。

6、什么是测试策略？

测试策略描述测试工程的总体方法和目标。描述目前在进行哪一阶段的测试（单元测试、集成测试、系统测试）以及每个阶段内在进行的测试种类（功能测试、性能测试、覆盖测试等）。

测试策略的制定主要包含三个方面的内容：

- （1）确定测试过程要使用的测试技术和工具；
- （2）制定测试启动、停止、完成标准；
- （3）进行风险分析和应对方案。例如测试与外部接口或者模拟物理损坏、安全性威胁。

测试计划最关键的一步就是将软件分解成单元，按照需求编写测试计划。

7、代码会审是如何进行的？

在研发小组将所开发的程序经验证后，提交测试组后，测试实施工作基本开始了。这个时候，测试人员要仔细阅读有关资料，包括规格说明、设计文档、使用说明书及在设计过程中形成的测试大纲、测试内容及测试的通过准则，全面熟悉系统，编写测试计划，设计测试用例，作好测试前的准备工作。为了保证测试的质量，我们一般测试过程分成几个阶段，即：代码审查、单元测试、集成测试和验收测试。

代码会审是由一组人通过阅读、讨论和争议对程序进行静态分析的过程。会审小组由组长，2~3 名程序设计和测试人员及程序员组成。会审小组在充分阅读待审程序文本、控制流程图及有关要求、规范等文件基础上，召开代码会审会，程序员逐句讲解程序的逻辑，并展开热烈的讨论甚至争议，以揭示错误的关键所在。实践表明，程序员在讲解过程中能发现许多自己原来没有发现的错误，而讨论和争议则进一步促使了问题的暴露。例如，对某个局部性小问题修改方法的讨论，可能发现与之有牵连的甚至能涉及到模块的功说明、模块间接口和系统总结构的大问题，导致对需求定义的重定义、重设计验证，大大改善了软件的质量。

代码会审尽管需要一定的成本，但是在大型软件中，是必不可少的。

8、回归测试中未解决的缺陷如何处理？

软件的后期测试就是一个反复回归的工作，有些问题可能修改多次才能解决，尤其是那些在开发环境下不存在的问题，这些问题很容易被程序员忽视，拖到最后才解决。因此大部分回归测试就是和开发人员反复配合解决那些上次测试中没有解决的缺陷。

这里重点讨论的是最后一次回归测试后，仍然发现有些缺陷没有解决时测试经理应该如何做。在管理不规范的组织中，由于进度或者其它方面的压力，开发工作已经停止，通常会将这些问题置之不理。正确的做法时把这些没有解决的问题形成一个未解决缺陷报告，然后

召开项目会议进行讨论，对不同的问题采取不同的处理方式：

- (1) 严重性的问题：有些问题较难解决，往往会被拖到最后，如果这类缺陷导致软件功能发生障碍，则必须解决，这也是质量控制的职责所在；
- (2) 功能性的问题：可以考虑升级时解决；
- (3) 一般性问题：不影响使用，可以不解决或者升级解决。

这类项目会议通常需要技术总监或者更高级别的人来参加。最后，需要对最终讨论没有解决的缺陷列表进行签字并存档，形成一个基线。特别要注意的某些缺陷是否修改不能由程序员或者测试人员来决定，这样有可能带来严重的后果——导致缺陷失控，最终形成没有人对质量负责的局面。

9、状态为已经修改的缺陷没有修改怎么办？

首先要对这类缺陷进行分析：

- (1) 有些问题在开发环境下没有重现，而开发人员迫于进度压力，往往会把它标记为已经修改。这种条件下测试人员应该和开发人员进行直接沟通；
- (2) 有些问题测试人员没有描述清楚，开发人员认为问题不存在也可能把问题标记为已经修改（正确的做法是标记问题为商讨或者不存在状态）。测试人员应该清晰的描述问题，减少这类问题的发生，尤其要描述清楚运行环境以及缺陷的重现步骤；
- (3) 第三类情况纯属个人行为：迫于进度压力，开发人员来不及修改问题，会故意把一些问题标记为修改，这样就可以在下次测试后进行修改。解决这种问题方法就是统计缺陷的修改次数，分析出那些反复修改的缺陷归属于那些开发人员，然后和绩效考核结合起来。（测试人员也可以这样做，把一些未验证的缺陷标记为“重新打开”，让开发人员来帮忙“验证”，我们仍然可以统计这种问题的次数，最后根据时间进行分析。）

而解决这种问题的根本方法就是加强项目管理，提高项目执行能力，一旦资源较充裕时，测试人员和开发人员就会更加投入的一起解决缺陷，共同来提高软件质量。

10、产品测试完成后产品谁来发布？

很多时候产品经过最后一次测试后由开发人员来发布，或者由质量管理部来发布，这样做都是不合适的。

开发人员发布产品常常会导致缺陷解决不彻底。一种较常见的现象是最后一次回归测试后，开发人员修改完成最后几个缺陷直接从开发环境发布产品，13.1.3 节中的案例就是这样的一种情况，这种条件下实际是缺陷一次测试，因为修改缺陷通常会引入新的缺陷，甚至是严重的缺陷。

测试人员发布缺陷也是不和流程的，测试人员的职责是报告软件质量情况。而且测试人员发布缺陷容易带来版本管理混乱。

正确的做法是产品经过最后一次测试后，把产品和缺陷修改情况存入产品基线库，形成一个可以发布的版本。这样发布产品的一个前提是每次产品提交测试前都要有一个预备发布版本，测试或者回归测试后如果有问题需要修改解决，开发人员对该预备版本进行修改。如此反复多次后，直到最后一次测试，所有缺陷都得到修改或者审核，同时开发人员此次测试后没有对产品经过任何修改，我们就可以把这个最后一个预备发布版本存入基线库。

进行了上面正确的版本控制后，我们可以通过配置管理库进行产品发布管理。对外部发布产品时，直接从配置管理库中提取就可以了。详细的内容，读者可以参考配置管理方面的书籍。

11、性能测试什么时候开展最为合适？

大多数情况下，性能测试在系统测试的最后阶段进行阶段进行。这主要是由于性能测试是一种综合性的测试，只有在功能测试通过后，谈系统测试才会有较大意义。但下列两类情

况比较特殊，性能测试一般进行的较早，几乎伴随着单元测试同步进行：

第一类是系统软件，例如开发操作系统或者数据库，等系统开发完成后再进行性能的唯一作用就是进行一个综合评估。如果在最后发现性能问题，很有可能推翻整个系统。

第二类是对性能要求较高的应用软件。例如银行、电信的系统，对系统的性能要远远高于一般的办公自动化系统。这类系统软件最后测试时发现性能问题，往往是系统架构或者一些关键算法、重要功能模块的原因，这个时候会带来较大的改动，甚至可能报废整个系统。

对于上面两类软件，性能测试应该贯穿着整个软件开发过程，大致要经过下面几个测试过程：

- (1) 单元性能测试阶段。上面这两类软件的性能测试最后从单元测试阶段就应该介入，具体做法就是安排性能测试工程师对一些重要算法进行测试，保证这些算法能够满足性能要求。这样做的好处是把问题尽早解决，可以大大提高整体性能。
- (2) 组成/集成性能测试阶段。这个阶段的性能测试是前面的深入，相当于把系统测试阶段的组合业务性能测试提前进行，可以把一些性能问题在集成测试阶段发现并解决。
- (3) 系统测试阶段的性能测试。这个阶段的性能测试是一个全面的性能测试，有了前面的基础，这个时候发现的问题很更加容易解决。

总之，性能测试是十分重要而且投入较高的测试，开展性能要根据具体的软件属性以及其它实际情况来制定测试策略。性能测试的具体设计方法可以参考 10.2.2 节中的内容。