

# 软件测试质量的保证

14126169-闫峻函

随着软件测试受关注程度越来越高，如何采用技术手段有效提高软件测试质量就成了软件测试领域的一个重要课题。软件产品的质量取决于软件开发过程，软件测试作为软件生存期中的一个重要阶段，受重视程度越来越高。软件测试是保证软件质量和可靠性的关键步骤，也是用来验证软件是否能够完成所期望功能的唯一有效的方法。测试已不仅仅局限于软件开发中的一个阶段，它已开始贯穿整个软件开发过程，进行测试的时间越早，整个软件开发成本下降就越多。

软件测试是提高软件测试质量的有效途径。据统计，在软件开发总成本中，用在测试上的开销要占 30%到 50%。但是，就目前软件工程发展的状况而言，软件测试又是相对于其他研究方向而言较为薄弱的一个方面。所以有必要开展软件测试质量度量理论的研究，以使之成为选择高效软件测试方法的指导。

所谓软件测试，是为了发现错误而执行程序的过程。或者说，软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计出一批测试用例，并利用这些测试用例的运行结果来发现程序错误的过程。而测试用例实际上是对软件运行过程中所有可能存在的目标、运动、行动、环境和结果的描述。软件测试的核心是设计和执行测试用例。而测试用例的选择问题可以看作是从庞大的输入状态组合中，搜寻哪些可以发现错误的状态组合。因此需要用抽象的手段来尽量使测试更加有效。

一个成功的测试至少应包括两个方面：一是软件在所有的（或足够多的）测试用例上是正确的；二是测试用例是充分的，即软件在测试用例上的表现能够充分地反映软件在整个输入空间上的表现。因此，这种软件测试的充分性既是从软件在有限多个测试用例上的行为判断软件在整个输入空间上的行为的逻辑基础，也可以作为软件测试质量的一种度量标准。

## 如何设计好的测试用例

其一，测试用例设计的时间性问题：越早地设计和执行测试用例越好。目前很多测试模型都是V模型，就是将测试依次划分为几个阶段，进行完了一个阶段然后再展开下一个阶段的测试。这种分阶段测试最大的缺点就是定性地测试分为几个固定的阶段，其实很多测试可以提前展开，尽早发现错误。而V模型

显然有点违背了越早发现错误越有利的原则。在开发产品时，将测试用例的设计提前进行。在需求阶段，便开始考虑和实施系统测试用例；当需求规格书确定后就开始设计功能性测试用例，同时概要设计也并行进行；等概要设计好了进行详细设计时，同时展开集成测试用例的设计。这样当开始编码的时候，我们便可以开始测试准备好的单元测试用例，而且也可以执行集成测试用例和系统测试用例；这样伴随编码同步地进行这些测试用例，可以及时发现错误，得到反馈信息，用以纠止软件设计中的错误。另外一个好处就是在编码阶段就展开集成测试、功能测试和系统测试，能尽早发现错误。

其二，在结构性测试用例设计中，我们要注意的原则：尽可能利用路径覆盖方法设计测试用例来覆盖程序所有可能路径，尤其在程序本身的复杂度高的情况下。一般说来，在编码过程中，程序本身设计的复杂度也决定了测试用例设计的复杂度。

在结构性测试用例的设计中，还经常采用的方法有：

语句覆盖：就是使程序中的每个可执行语句至少执行一次。

判定覆盖：使程序中的每个判定至少都获得一次“真”和“假”值。

条件覆盖：使每个判定中的每个条件的可能取值至少满足一次。

判定 / 条件覆盖：使判定中的每个条件的可能结果和每个判定本身的判定结果均至少出现一次。

条件组合覆盖：使得每个判定中条件的各种可能组合都至少出现一次。

其三，在功能性测试用例设计中，我们常常需要：把百分之八十的精力投入到那些边界情况和失效数据作为输入的测试情况。

功能测试即黑盒测试，一般由测试人员来执行，而测试用例的设计，最好在需求阶段依据需求规格书进行设计。如果设计测试用例的人员没有完全理解需求规格书，这里面就会引起一些问题，到产品最后验收的时候，就会发现产品根本不能交付使用，大大延误了产品上市时间。测试人员依据功能测试用例来进行测试，对软件内部结构和运作并不了解。一般初期设计出来的测试用例往往又会遗漏很多边界条件。这需要在设计功能性测试用例的时候，不仅仅要考虑有效等价类，那些无效的等价类更加重要，因为这些都是人们容易忽视的，无论开发人员还是测试人员，往往有定势思维，总感觉自己的产品没有问题，他们也习惯性地

输入一些一定满足条件的数据进行测试，无形中掩盖了潜在的失效数据的处理。边界情况和失效等价类是设计测试用例时必须使用到的。这样才能设计出好的功能性测试用例。

### **为了提高测试质量，选择合适的测试管理模型是必要的**

模型是系统功能的形式化或半形式化的表示，支持输入状态组合的系统枚举。基于模型的测试主要考虑系统的功能，可以认为是功能测试的一种。测试模型体现了被测试系统的最本质的功能关系。而且要比系统本身更易于开发和分析。一个可测试的模型要能提供足够的信息用来产生测试用例。所以可测试的模型必须满足以下要求：

（1）必须是某种测试实现的完全准确的反映，模型必须表示要检查的所有特征；

（2）是对细节的抽象；

（3）可以表示所有事件和所有的动作；（4）可以表示系统的各种状态，以便由已知的方法来确定已达到或没有达到什么状态。

### **可复用的测试用例库同样可以有效提高软件测试的质量**

软件测试过程中产生的大量测试用例对保证软件的质量起到关键作用，测试用例的复用就显得尤其重要。在测试中应当构造出基于复用的测试用例，并采用测试用例库管理的方法来实现，这样可以大大提高软件测试的工作效率。

设计出好的测试用例是确保软件质量的大前提，有效复用现有的测试用例更能提升企业软件开发过程。目前很多软件企业，对测试用例的复用没有引起足够的重视。软件开发往往时间紧张，客观上也成了开发测试人员的借口，设计出来的软件或者测试用例，过于局限于本产品使用，依赖性非常强，根本就不利于升级或者拓展。这里我们提到的就是测试用例的复用。我们要能够将所有的测试用例有效地组织起来，使得测试用例集合里面的每一个测试用例都能够独立地运行，这样才能提高软件测试用例的复用度。结合工作中的经验，传统的测试用例设计有很多方法，看上去相互间没有统一的结构。我们所需要注意的是：能够将测试用例综合起来分析，然后让所有的测试用例组织在一起，它们具有统一的输入、输出接口，并且每个测试用例独立性比较强，这样即使以后软件运行环境发生变化，测试用例还能够继续加以使用。测试用例独立性强，采用一致的结构，是测

试用例复用度高的关键。

基于复用的目的，对所使用的测试用例进行统一的建模组织，有效地将测试用例收集到测试用例库中，并按照行业项目等进行多级合理的分类、组织、存储。对采用不同方式描述的测试用例，将分别实现不同程度的复用。

对数据库中存有的测试用例实行有效管理，通过提供有助于复用的多种查询方式，确保测试用例的复用程度，对不同类别的测试用例进行复用频率的统计，为测试人员提供有效的复用信息，在保证软件测试质量的前提下，提高了测试的效率。

对数据库中的测试用例具体实现复用，通过查询出所需的测试用例，得到测试用例信息，针对不同描述方式的测试用例采用不同的复用方式进行复用。

## 结语

软件测试用例的设计，是软件测试中比较重要的一环。只有设计出更多更好地测试用例，才可以更快更好的发现潜在的错误与失效。使用最少的测试用例，实现最大的测试覆盖，可复用度好，并且在需求分析阶段提前展开测试用例的设计，是我们软件测试的目标。只有制定出完善的测试计划，有效的测试用例，进行测试结构分析以及文档管理，才能保证软件测试的成功。以软件测试性设计为中心，配合合理的软件测试方法来实现软件测试质量的提高。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。