

# 软件测试之工程实践

王培华

中国电子科技集团公司第二十七研究所 郑州 450005

**摘要:** 简要介绍软件测试的基本方法和测试阶段。结合工程实践,介绍各测试阶段的主要任务、测试依据、测试方法、工作内容,以及笔者在实践中遇到的问题及解决方法、对软件测试的认识和体会。

**关键词:** 软件测试 测试方法 测试阶段 测试实践

中图分类号:TP311.52

文献标识码:A

## 1 概述

软件测试在整个软件开发过程中占有非常重要的位置,它是保证软件质量的重要手段。本文从实践的角度,介绍在工程项目应用软件开发过程中,如何进行软件测试,并讨论在实际工作中遇到的问题及解决方法。

## 2 软件测试的基本方法

对于软件测试方法,可以根据不同的测试特点加以分类。目前最常用的分类,一是根据测试过程中是否需要运行被测程序,分为静态测试法和动态测试法;二是根据测试是否针对被测程序的内部结构和内部特性,分为白盒测试法和黑盒测试法。各测试方法的特点分别如表1和表2所示<sup>[1]</sup>。

表1 静态、动态测试方法和特点

分类	方法描述	特点	常用方法
静态测试	不直接运行被测程序,通过人工或利用计算机和测试工具,对被测程序进行静态分析。	1. 不需要执行被测测试程序。 2. 不需要特殊条件,容易开展。 3. 充分发挥人的优势,行之有效。	代码走查、代码审查、技术评审、自动化的静态分析。
动态测试	必须在计算机上运行被测程序,通过输入测试用例,对其运行情况进行分析。	1. 必须使用测试用例,执行被测测试程序。 2. 能得到程序执行后的真实情况。 3. 需要为测试提供测试软件。 4. 选择测试用例是提高测试质量和效率的关键。	黑盒法、白盒法。

表2 黑盒、白盒测试方法和特点

分类	方法和特点	常用方法
黑盒测试	基于规格说明,把被测程序当作一个黑盒子,不考虑其内部结构和内部特性,只考虑其输入与输出的关系和功能。	等价类划分、边界值分析法、因果图。
白盒测试	基于程序,它不考虑程序的功能,只注重程序的内部结构。	语句覆盖、判定覆盖、条件覆盖、分支覆盖、路径覆盖。

## 3 软件测试的阶段

软件测试划分为单元测试、集成测试、配置项测试、系统测试和验收测试五个阶段。各个阶段测试的条件和主要任务如图1所示。

各测试的阶段产品有测试说明、通过测试的程序代码、测试软件、测试记录和测试报告。如果验收测试不单独进行,则不单独提供测试说明、测试记录的测试报告,通过验收测试的软件代码是可以交付使用的软件代码。

## 4 软件测试之工程实践

虽然各测试阶段要完成的任务和所采用的测试方法不同,但工作内容基本相同。它包括拟制测试计划、编写测试说明、建立测试环境、进行静态测试和动态测试、测试结果分析和评价、测试交付。

### 4.1 制定测试计划

制定测试计划就是对测试进行人员和时间上的安排。测试计划要细化到对拟制测试说明、进行测试和拟制测试报告等工作的安排。

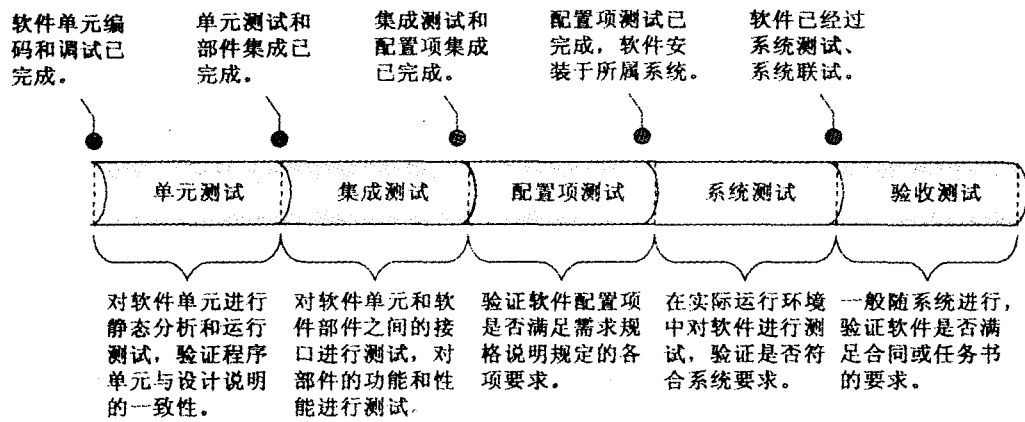


图 1 软件测试阶段及阶段性任务

合理的测试计划,对于提高软件测试效率、保证测试进度是非常重要的。因此,在制定测试计划时,要注意以下几点:

首先,要对被测软件的重要性和复杂程度、测试工作量、测试难易程度、是否需要编制测试软件等进行认真的分析。在分析的基础上,才能合理安排测试人员、人数和所需要的时间。

第二,要充分考虑与软件开发计划、硬件研制计划的协调性。

依我们的实践经验,充分考虑计划的协调性,在制定单元测试计划时尤其重要。某些软件单元一旦通过单元测试,就可以作为其它软件单元的测试程序。制定测试计划时,就应考虑将这样的软件单元先行测试,不必等到所有软件单元都编制和调试完成后再进行。

#### 4.2 编写软件测试说明

测试说明的内容包括测试所需的软硬件环境、测试方法、测试过程、测试用例、期望结果以及评判测试结果的标准。

编写测试说明的过程是对测试方法、测试过程以及测试用例等进行设计的过程,其中测试用例设计是关键,它直接关系到软件测试的质量和效率。

编写测试说明要注意以下几点:

第一,测试用例设计要发挥人的优势。尽管有些测试工具号称可以自动生成测试用例,但是就目前情况看,自动化的测试工具无法完全代替人的作用。

第二,不同测试阶段,设计测试用例的依据和方法不同。

表 3 各阶段测试依据、目标和方法

测试阶段	测试依据	测试目标	测试方法
单元测试	软件详细设计	既要对软件单元的功能和接口进行测试,也要对其内部结构和特性进行测试。	黑盒和白盒相结合
集成测试	软件概要设计	测试主要针对软件部件或软件配置项的功能和接口,不关心程序的内部结构和特性。	黑盒法
配置项测试	软件需求规格说明		
系统测试			
验收测试	合同或研制任务书		

第三,对于每个测试用例,它的测试内容和目的、输入数据(包括值域、取值、来源等)、由这些输入数据所产生的期望结果以及评判测试结果的标准,都要在测试说明中描述清楚。

需要注意的是,对期望结果的描述,应该是具体的、直观的、精确的,是通过规定的测试方法能够直接获取的。例如,某个软件单元的输出为字符串变量 A,获取测试结果的方法是在界面上显示 A 的值。对于某测试用例,运行结果为 A 等于“123”,那么,对期望结果的正确描述是“界面显示 A 的值为‘123’”。如果将期望结果描述为“A 的值为‘123’”,又没有清楚地说明获取测试结果的方法,那么测试人员就不知道从哪里去查看这个结果。

评判测试结果的标准,可以描述为与期望结果完全相同,或者是在期望结果的多少误差范围内,等等。

#### 4.3 建立测试环境

##### 4.3.1 测试环境的建立

建立测试环境即根据测试说明规定的测试条件,包括测试工具软件、自行开发的测试软件和硬件设备等,构建被测试软件运行的软硬件平

台,提供测试数据的输入手段和测试结果的输出手段。

单元测试阶段,由于软件单元是软件的最小单位,无法独立运行,因此需要设计两种测试程序,即驱动模块和桩模块。驱动模块为被测单元的上级模块,它直接调用或以某种方式驱动被测单元,它通常还具有接受并向被测单元传送测试数据和输出测试结果的功能;桩模块模拟被测单元的下级模块,由被测单元调用,通常具有测试结果输出功能。

#### 4.3.2 测试程序确认

如果需要自行编制测试程序,要对测试软件进行需求分析、设计、编程和调试。

自行编制的测试程序是否正确,是否满足测试需要,如何对它进行测试呢?如果象测试被测程序那样对测试程序进行测试,可能又需要编制对测试程序进行测试的测试程序,这样循环下去,就会没有尽头,也是不现实的。我们在工程实践中,采取了检查和确认的方法。主要检查测试程序功能是否完备,是否满足测试需要。对于测试程序中可能出现的错误,分析它对发现被测程序的错误是否有影响。如果对发现错误有影响,那么需要有一定的测试手段来测试它,如编制一些结构简单、功能单一的测试程序来测试它。如果测试程序的错误对测试效果几乎没有影响,只是对测试效率有一些影响,那么就可以确认它可以用于测试。例如,对于一个依据接口控制文件编制的被测程序,由测试人员根据同一接口控制文件编制接口模拟程序作为测试程序,由于测试人员和编程人员不同,他们在处理同一接口信息时出现相同错误的概率极低,因此无论是被测程序的错误,还是测试程序的错误,都能在测试结果中反映出来,不会影响测试的效果。当然,测试程序在测试过程中逐步得以完善。实践证明这种办法是行之有效的。

### 4.4 执行测试

#### 4.4.1 单元测试

单元测试分两步进行。首先进行静态分析。检查和程序代码的功能、算法、接口等与设计是否相符,是否符合编程规则、是否有资源泄漏、是否有冗余代码等等。静态分析完成后,编程人员对发现的问题进行修改,然后进行运行测试。

单元测试非常重要,它能尽早地、有效地地发现程序中的错误。实践证明,大量的错误都是在

单元测试中发现的。根据实践经验,为了保证单元测试的质量、提高测试效率,要特别注意以下几个方面。

第一,测试人员要对整个软件的需求和设计有全面和深入的了解,才能站在系统的角度,而不是单元的角度,去发现对全局有影响的错误。

第二,保持测试活动的独立性。

单元测试一般由软件项目组内部完成,开发人员和测试人员要相互独立,实行双岗制。在任务急、人员紧张,无法绝对实现双岗制的情况下,开发人员和测试人员可以有交叉,但是,要杜绝同一人员既开发又测试同一程序单元。这是因为首先,编程人员自己测试自己编写的程序,容易受到设计和编程思路的影响;第二,单元测试从设计测试用例、编制测试程序、提供输入测试数据和获取测试结果的手段,到执行测试,是一个相对复杂的过程,工作量非常大,由编程人员自己测试,特别是在工作忙的情况下,很可能回避复杂的过程,或通过简单抽测验证程序,不能保证单元测试的效果,给以后的测试工作留下很多隐患;第三,软件测试有专门的方法,并不是简单地抽测和验证程序,因此编程人员本身并不一定能胜任软件测试的任务。

第三,提高被测软件单元的代码质量。

如果编程人员提交测试的代码没有经过认真的自测试,代码中可能存在很多错误,甚至是低级错误,测试就难以进行下去。测试变成了调试,这样就会影响测试质量和进度。这是我们在实践中曾经遇到的问题,为了避免这种情况,除要加强软件工程化管理外,还应尽早开始测试工作。测试人员在单元测试前,将测试说明和测试程序,提供给编程人员,由编程人员进行认真的自测试。采取这一措施,收到了良好的效果。

#### 4.4.2 集成测试

集成测试的方法有非渐增式和渐增式两种。顾名思义,渐增式集成测试,是一个单元一个单元逐步集成,边集成边测试。非渐增式测试则是先将所有单元集成到一起,然后再进行测试。采用非渐增式测试,一旦出现问题,不易分离和定位错误。因此通常采用渐增式集成测试<sup>[2]</sup>。

采用渐增式方法,集成测试可以和单元测试结合起来进行,将两个测试阶段合并为一个阶段。在单元测试阶段,已经测试的单元,可以作为桩模块或驱动模块,用来对其它单元进行测

试,软件单元在单元测试过程中被逐步集成。这样不仅能够及早发现软件单元接口间存在的问题,也使软件单元在测试中不断经受考验,逐步得到完善。

渐增式集成测试的次序,可以是自顶向下或自底向上,也可以从软件自身的特点出发,选择其它合适的次序集成。根据实践经验,集成次序选择合适,能起到事半功倍的效果。确定集成次序,要从软件本身和测试两方面考虑。从软件本身考虑,要找出重要和关键的单元,该单元的错误可能导致整个软件结构的更改,应该先测试。从测试方面考虑,就是要便于集成、便于测试。

集成次序在计划阶段就要加以考虑,它与软件开发进度密切相关,需要与软件开发计划协调。

#### 4.4.3 配置项测试

配置项测试可以由承制单位进行,也可以根据要求委托第三方进行。如果由承制单位完成,所有测试活动应该由独立的测试组承担。

配置项测试主要检查配置项在功能、性能、可靠性、可维护性以及需求规格说明所规定的其它方面是否满足要求。

如果所属系统硬件和其它软件已经完成,可以利用所属系统的软硬件进行测试,这样可以简化测试工作,但是系统能够满足被测软件的正常运行条件,却不一定完全满足配置项测试的要求。象在实际运行过程中极少出现的状态、某些性能测试所需要的输入条件,通过所属系统进行测试就难以实现。针对这部分测试,还需要根据测试说明构建软硬件测试环境。

#### 4.4.4 系统测试

软件配置项只是所属系统中的一个元素,系统测试就是将被测软件与所属系统中其它元素结合起来,进行系统确认测试。

系统测试要注意避免重复无效的测试。与配置项测试相比,测试环境有所不同,测试内容却有很多相同。对于与测试环境相关性不大的某些功能,可以采取抽测的办法来避免重复无效的劳动,应该将测试重点放在与系统环境密切相关的部分。

要充分利用系统联试过程。对一些在单元测试和配置项测试时无法进行充分测试的项目,在系统联试过程中,应重点进行检查。

#### 4.4.5 验收测试

软件配置项的验收测试可以不单独进行,而

随系统验收测试进行。

#### 4.5 测试结果分析和评价

测试结束后,要对测试结果进行分析和评价,提供测试报告和问题报告。依据测试说明中描述的期望结果和评判标准,分析测试结果是否正确,判定测试用例执行是否通过。对错误的结果,要分析是软件本身有错误,还是测试方法、测试过程、测试环境等有问题。对软件本身的错误,提出问题报告,供编程人员修改,对修改后的错误要进行回归测试。

测试报告的主要内容包括测试时间、测试人员、测试对象的名称和标识、静态和动态测试情况、是否全部执行了测试说明规定的测试用例、对未执行规定测试用例的合理解释、测试结论等。

测试评价是指对软件测试质量的评价,测试覆盖率是一项重要指标。测试覆盖包括对需求、设计的覆盖和对程序代码的覆盖。对程序代码的覆盖率统计,在执行测试时,可借助于软件测试工具进行。对需求和设计的覆盖程度,只有靠人工检查和分析。笔者在过去的软件测试实践中,只是在设计测试用例的时候,考虑到对需求和设计的覆盖。至于如何进行需求和设计的覆盖率统计,这项工作该由谁来完成,希望能够与读者进行探讨。

#### 4.6 测试交付

测试结束后,将测试文档(包括测试计划、测试说明、测试记录、测试报告)、通过测试的程序代码、测试程序、测试数据等,提交给软件配置管理人员。

## 5 结束语

软件测试是很重要、很有意义的工作,同时也具有挑战性。为了提高软件质量,软件测试工作应该受到足够的重视,测试人员要不断提高自己在软件设计、编程和测试方面的技术水平,不断总结软件测试经验,完善测试方法。

#### 参考文献

- [1] 郑人杰. 计算机软件测试技术. 清华大学出版社 1992 年
- [2] 张克东, 庄燕滨. 软件工程与软件测试自动化教程. 电子工业出版社, 2002 年