

## 第1部分

# 软件测试入门



### 本部分概要

第 1 章 软件测试概述

第 2 章 软件测试基础



## 软件测试概述



### 本章概要

本章介绍了软件测试的发展历史,软件测试技术的分类方法、测试标准、测试原则,阐述了软件测试与软件开发的关系。

### 1.1 软件测试背景

软件的质量就是软件的生命,为了保证软件的质量,人们在长期的软件开发过程中积累了许多经验方法。但是,借助这些方法,只能尽量减少软件中的错误和不足,却不能完全避免所有的错误。

由于软件是人脑高度智力化的体现这一特殊性,不同于其他科技和生产领域,软件与生俱来就可能存在缺陷。

开发大型软件系统通常需要较长时间,面对各种复杂的现实情况,人的主观认识和客观现实之间往往存在差距;开发过程中,各类人员之间的交流和配合也往往不是尽善尽美的。

如果不能在软件正式投入运行之前发现并纠正软件中的错误,那么这些错误必然会在软件的实际运行过程中暴露出来。到那时,改正这些错误不仅要付出很大的代价,而且往往会造成无法弥补的损失。

如何防止和减少这些可能存在的问题呢?那就是进行软件测试。测试是最有效的排除和防止软件缺陷与故障的手段,并由此促进了软件测试理论与技术的快速发展。新的测试理论、测试方法、测试技术手段不断涌出,软件测试机构和组织也在不断产生和发展,软件测试技术职业也随之逐步完善和健全起来。

#### 1.1.1 软件缺陷

##### 1. 软件错误案例研究

软件已经深入渗透到我们的日常生活中,在电子信息领域里无处不在。然而,软件是由人编写开发的,是一种逻辑思维的产品。尽管现在软件开发者采取了一系列有效措施,

不断地提高软件开发质量,但仍然无法完全避免软件(产品)会存在的各种各样缺陷。

下面以实例来说明。

(1) 迪士尼的狮子王游戏软件缺陷。1994年秋天,美国迪士尼公司发布了第一个面向儿童的多媒体光盘游戏——《狮子王动画故事书》(*The Lion King Animated Storybook*)。由于这是迪士尼公司首次进军儿童游戏市场,所以进行了大量促销宣传,狮子王动画故事书的销售额非常可观。然而,同年12月26日,圣诞节的第二天,迪士尼公司的客户支持电话开始响个不停。很快,电话支持技术人员们就淹没在愤怒家长的责怪声和玩不成游戏的孩子们的哭叫声中。报纸和电视新闻对此进行了大量的报道。

后来证实,迪士尼公司未能对市面上投入使用的许多不同类型的PC(Personal Computer)机型进行广泛测试。软件在极少数系统中工作正常——例如在迪士尼程序员用来开发游戏的系统中,但在大多数公众使用的系统中却不能运行。

(2) 爱国者导弹防御系统缺陷。爱国者导弹防御系统是美国里根总统提出的战略防御计划(即星球大战计划)的缩略版本,它首次应用在海湾战争中对抗伊拉克飞毛腿导弹的防御战中。尽管媒体对爱国者导弹系统赞誉的报道不绝于耳,但是,它确实对抗几枚导弹的过程中出现了失利,包括在沙特阿拉伯的多哈击毙了28名美国士兵。通过分析发现问题在于一个软件缺陷——系统时钟的一个很小的计时错误,当系统计时积累起来达到14小时后,跟踪系统不再准确。在多哈的这次袭击中,系统已经运行了100多个小时。

(3) 千年虫问题。20世纪70年代早期,某位程序员为其公司设计开发工资系统。由于他使用的计算机存储空间很小,这就迫使他尽量节省每一个字节。他将自己的程序压缩得比其他任何人都紧凑,其中一个方法是把4位数的年份,例如1973年,用2位数表示,即73。因为工资系统非常依赖日期的处理,所以需要节省大量的存储空间。该程序员认为只有在2000年以后,程序开始计算00或01这样的年份时,才会出现问题。虽然他知道程序会出现这样的问题,但是他认定在25年之内程序肯定会升级或替换,而且,在他看来,完成眼前的任务比现在计划遥不可及的未来更加重要。1995年他的程序仍然在使用,而他退休了,谁也不会想到如何深入程序中检查2000年的兼容问题,更不用说去修改程序。

估计全球各地更换或升级类似的前者程序以解决潜在的2000年问题的费用已经达数千亿美元。

(4) 美国航天局火星登陆探测器缺陷。1999年12月3日,美国航天局的火星极地登陆者号探测器试图在火星表面着陆时失踪。一个故障评估委员会调查了该事件,认为出现故障的原因可能是一个数据位被意外置位。此次事故最令人警醒的问题是——为什么没有在内部测试时发现数据位被意外置位。

从理论上讲,探测器着陆的计划是这样的:当探测器向火星表面降落时,它将打开降落伞以减缓探测器的下降速度。降落伞打开几秒钟后,探测器的三条腿将迅速被撑开,并锁定位置,准备着陆。当探测器离地面1800米时,它将丢弃降落伞,点燃着陆推进器,缓缓地降落到地面。

美国航天局为了省钱,简化了确定何时关闭着陆推进器的装置。为了替代其他太空船上使用的贵重雷达,他们在探测器的脚部装了一个廉价的触点开关,在计算机中设置一

个数据位来控制触点开关关闭燃料。探测器的发动机需要一直点火工作,直到脚“着地”为止。

遗憾的是,故障评估委员会在测试中发现,许多情况下,当探测器的脚迅速撑开准备着陆时,机械震动也会触发着陆触点开关,设置致命的错误数据位。设想探测器开始着陆时,计算机极有可能关闭着陆推进器,这样火星极地登陆者号探测器飞船下坠 1800 米之后冲向地面,撞成碎片。

结果是灾难性的,但背后的原因却很简单。登陆探测器经过了多个小组测试。其中一个小组测试飞船的脚折叠过程,另一个小组测试此后的着陆过程。前一个小组不去注意着地数据是否置位,这不是他们负责的范围;后一个小组总是在开始复位之前复位计算机,清除数据位。双方独立工作都做得很好,但合在一起就不是这样了。

(5) 金山词霸缺陷。在国内,“金山词霸”是一个著名的词典软件,应用范围很广,对使用中文操作的用户帮助很大。但它也存在不少缺陷,例如输入 cube,词霸会在示例中显示  $3^3=9$ ;又如,如果用鼠标取词 dynamically(力学,动力学),词霸会出现其他不同的单词“dynamite *n.* 炸药”的错误显示。

(6) 英特尔奔腾浮点除法缺陷。在计算机的“计算器”程序中输入以下算式。

$$4195835/3145727 \times 3145727 - 4195835$$

如果答案是 0,说明计算机没问题。如果得出别的结果,就表示计算机使用的是带有浮点除法软件缺陷的老式英特尔奔腾处理器,这个软件缺陷被烧录在一个计算机芯片中,并在制作过程中反复生产。

1994 年 10 月 30 日,弗吉利亚州 Lynchburg 学院的 Thomas R. Nicely 博士在他的一个实验中,用奔腾 PC 机解决一个除法问题时,记录了一个想不到的结果,得出了错误的结论。他把发现的问题放到因特网上,随后引发了一场风暴,成千上万的人发现了同样的问题,并且在另外一些情形下也会得出错误的结果。万幸的是,这种情况很少见,仅仅在进行精度要求很高的数学、科学和工程计算中才会出现错误。大多数用来进行税务处理和商务应用的用户不会遇到此类问题。

这件事情引人关注的并不是这个软件的缺陷,而是英特尔公司解决问题的方式。

① 他们的软件测试工程师在芯片发布之前进行内部测试时已经发现了这个问题。英特尔的管理层认为这没有严重到要保证修正,甚至公开的程度。

② 当软件缺陷被发现时,英特尔公司通过新闻发布和公开声明试图弱化这个问题的已知严重性。

③ 受到压力时,英特尔公司承诺更换有问题的芯片,但要求用户必须证明自己受到缺陷的影响。

一时间,舆论哗然。互联网新闻组里充斥着愤怒的客户要求英特尔公司解决问题的呼声。新闻报道把英特尔公司描绘成不关心客户和缺乏诚信者。最后,英特尔公司为自己处理软件缺陷的行为道歉,并拿出 4 亿多美元来支付更换问题芯片的费用。现在英特尔公司在 Web 站点上报告已发现的问题,并认真查看客户在互联网新闻组里的反馈意见。

## 2. 软件缺陷的定义

从上述的案例中可以看到,软件发生错误时将造成灾难性后果或对用户产生各种影响。

在这些事件中,软件显然都未按预期目标运行。作为软件测试员,可能会发现大多数缺陷不如上面那些缺陷明显,而一些简单细微的错误,通常难以区分哪些是真正的错误,哪些不是真正的错误。

软件存在的各种问题称为软件缺陷或软件故障,在英文中人们喜欢用一个不贴切但已经专用的词 bug 来表示。

软件缺陷即计算机系统或者程序中存在的任何一种破坏系统或程序正常运行的问题、错误,或者隐藏的功能缺陷、瑕疵。缺陷会导致软件产品在某种程度上不能满足用户的需要。对于软件缺陷的准确定义,通常有以下五个方面的描述。

- (1) 软件未实现产品说明书要求的功能。
- (2) 软件出现了产品说明书指明不会出现的错误。
- (3) 软件超出实现了产品说明书提到的功能。
- (4) 软件实现了产品说明书虽未明确指出但应该实现的目标。
- (5) 软件难以理解,不易使用,运行缓慢或者终端用户认为不好。

为了更好地理解每一条规则,以计算器为例进行说明。

计算器的产品说明书载明其能够准确无误地进行加、减、乘、除运算。当用户拿到计算器后,按下“+”键,什么反应也没有,根据第1条规则,这是一个缺陷。假如计算器能够计算,但得到错误答案,根据第1条规则,这同样是一个缺陷。

若产品说明书声称计算器永远不会崩溃、锁死或者停止反应,但当任意进行按键操作时,计算器停止接收输入,根据第2条规则,这是一个缺陷。

若用计算器进行测试,发现除了加、减、乘、除之外,计算器还可以求平方根,而说明书中并未提到这一功能,根据第3条规则,这是软件缺陷。软件实现了产品说明书未提到的功能。

在测试计算器时,发现电池没电会导致计算不正确,但产品说明书未指出这个问题。根据第4条规则,这是个缺陷。

第5条规则是全面的。如果软件测试员发现计算器的某些地方不方便使用,无论什么原因,都会被认定为缺陷。如“=”键布置的位置使其不好按下;或在明亮光线下,显示屏难以看清。根据第5条规则,这些都是缺陷。

美国商务部国家标准和技术研究所(NIST)进行的一项研究表明,软件中的缺陷每年给美国经济造成的损失高达595亿美元。说明软件中存在的缺陷所造成的损失是巨大的,这从反面证明了软件测试的重要性。如何尽早彻底地发现软件中存在的缺陷是一项非常复杂,需要创造性的工作。同时,软件缺陷是软件开发过程中的重要属性,反映了软件开发过程中需求分析、功能设计、用户界面设计、编程等环节所隐含的问题,也为项目管理、过程改造等提供了许多信息。

## 3. 产生软件缺陷的原因

软件缺陷的产生是不可避免的。可以从技术问题、团队工作和软件本身确定容易造

成软件缺陷的原因。

1) 技术问题

- (1) 算法错误。
- (2) 语法错误。
- (3) 计算和精度问题。
- (4) 系统结构不合理,造成系统性能问题。
- (5) 接口参数不匹配出现问题。

2) 团队工作

- (1) 在系统分析阶段,不了解客户的需求,或者和用户的沟通存在一些困难。
- (2) 不同阶段的开发人员对软件功能理解不一致,例如,软件设计人员对需求分析结果的理解偏差,编程人员对系统设计规格说明书中某些内容重视不够或存在误解。
- (3) 设计或编程上的一些假定或依赖性,没有得到充分的沟通。

3) 软件本身

- (1) 文档错误、内容不正确或拼写错误。
- (2) 数据考虑不周全引起强度或负载问题。
- (3) 对边界考虑不够周全,漏掉某几个边界条件造成的错误。
- (4) 对一些实时应用系统,需要保证精确的时间同步,否则容易因时间上的不协调、不一致而出现问题。
- (5) 没有考虑系统崩溃后在系统安全性、可靠性方面存在的隐患。
- (6) 硬件或系统软件上存在的错误。
- (7) 软件开发标准或过程上的错误。

#### 4. 软件缺陷的组成

软件缺陷是由很多原因造成的,如果把它们按需求分析结果——规格说明书、系统设计结果、代码等进行归类,可以发现规格说明书是软件缺陷出现最多的地方,如图 1-1 所示。

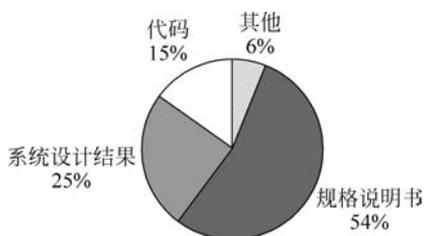


图 1-1 软件缺陷构成示意图

软件产品规格说明书是软件缺陷存在最多的地方的原因主要有以下几种。

- (1) 用户一般是非计算机专业人员,软件开发人员和用户的沟通存在较大困难,两者对要开发的产品功能理解不一致。
- (2) 由于软件产品还没有设计、开发,完全靠想象去描述系统的实现结果,有些特性还不够清晰。
- (3) 用户需求变化的不一致性。用户的需求总是在不断变化,这些变化如果没有在

产品规格说明书中得到正确的描述,容易引起前后文、上下文的矛盾。

(4) 对规格说明书不够重视,软件开发方在规格说明书的设计和写作上投入的人力、时间不足。

(5) 没有在整个开发队伍中进行充分沟通,有时只有设计师或项目经理得到比较多的信息。

### 5. 软件缺陷的修复费用

软件通常要靠有计划、有条理的开发过程来实现。在软件开发的问题定义、需求分析、软件设计、程序编码、软件测试各阶段,再到软件的公开使用过程中,都有可能发现软件缺陷。

软件缺陷的修复费用呈指数级增长,也就是说,随着时间的推移,软件修复费用呈十倍地增长。如果在早期编写软件产品说明书时,发现缺陷并修复,只要花费1美元甚至更少。同样的缺陷,如果直到软件编写完成,开始测试时才发现,可能要花费10~100美元。如果是客户发现的缺陷,可能需要花费数千甚至数百万美元。

举一个例子来说明软件缺陷的修复费用问题,比如前文的迪士尼狮子王实例。问题的根本原因是软件无法在流行的PC平台上运行。假如早在编写产品说明书时,有人已经研究过什么PC机流行,并且明确指出软件需要在该种配置上设计和测试,那么迪士尼公司付出的代价将小得几乎可以忽略不计。如果没有这样做,还有一个补救措施,软件测试员去搜集流行PC样机并在其上验证。他们可能会发现软件缺陷,但是修复费用要高得多,因为软件必须经过调试、修改、再测试。开发小组还应当把软件的初期版本分发给一小部分客户进行试用,叫作BETA测试。那些被挑选出来,能够代表庞大市场的客户可能会发现问题。然而实际的情况是,缺陷被完全忽视,直到成千上万的光盘被压制和销售出去。而迪士尼公司最终支付了客户投诉电话、产品召回、更换光盘所需的费用,以及新一轮软件调试、修改和测试的费用。可见,如果严重的软件缺陷是客户发现的,修复软件缺陷所花费的费用足以耗尽整个产品的利润。

## 1.1.2 软件测试技术的发展历史和现状

### 1. 软件测试技术的发展历史

随着计算机的诞生,在软件行业发展初期就已经开始实施软件测试,但这一阶段还不是传统意义上的软件测试,更多的是一种类似调试的测试。测试没有计划和方法,测试用例的设计和选取由测试人员的经验来决定,大多数测试的目的是为了证明系统可以正常运行。

20世纪50年代后期到60年代,各种高级语言相继诞生,测试的重点也逐步转入到使用高级语言编写的软件系统中来,但程序的复杂性远远超过以前。尽管如此,由于受到硬件的制约,在计算机系统中,软件仍然处于次要位置。软件正确性的把握仍然主要依赖于编程人员的技术水平。因此,这一时期软件测试的理论和方法发展比较缓慢。

20世纪70年代以后,随着计算机处理速度的提高,存储器容量的快速增加,软件在整个计算机系统中的地位变得越来越重要。软件开发技术日臻成熟和完善,软件规模越来越大,复杂度也大大增加。因此,软件的可靠性面临着前所未有的危机,给软件测试工作带来了更大的挑战,很多测试理论和测试方法应运而生,逐渐形成了一套完整的软件测

试体系,培养和造就了一批出色的测试人才。

如今,在软件产业化发展的大趋势下,人们对软件质量、成本和进度的要求也越来越高,质量的控制已经不仅仅是传统意义上的软件测试。传统的软件测试大多是基于代码运行的,并且常常是软件开发的后期才开始进行。但大量研究表明,设计活动引入的错误占软件开发过程中出现的所有错误数量的50%~65%。因此,越来越多的声音呼吁,要求有一个规范的软件开发过程。而在整个软件开发过程中,测试已经不再只是基于程序代码进行的活动,而是一个基于整个软件生命周期的质量控制活动,贯穿于软件开发的各个阶段。

## 2. 软件测试技术的现状

在我国,软件测试可能算不上一个真正的产业,软件开发企业对软件测试认识淡薄,软件测试人员数量与软件开发人员数量配置比例失调。在一些发达国家和地区,软件测试已经成为一个产业。例如,微软的开发工程师数量与测试工程师数量的比例是1:2,而国内一般公司的这一比例是6:1。很多人认为导致这种现象产生的原因与我们接受的传统教育和开发习惯有很大关系。软件行业相对于其他一些行业来说起步较晚,软件开发过程包含需求管理、分析、设计、测试和部署等工作。由于软件行业的发展历程短暂,而且一般人认为,软件开发周期前面的工作没有完善之前,比较难于考虑到后面的工作。因此,软件开发大部分的精力都投入到了需求管理、分析、设计3个阶段,从而促使这些方面方法论的快速发展,忽视了测试工作。

总之,与一些发达国家相比,国内的软件测试工作还存在一定差距。主要体现在对测试意识、测试理论的研究,大型测试工具软件的开发以及从业人员数量等方面。其实,这与中国整体软件的发展水平是一致的,因为我国整体的软件产业水平和软件发达国家水平相比有一定的差距,而作为软件产业中重要一环的软件测试,必然也存在着差距。但是,我国在软件测试实现方面并不比国外差,我们拥有国际上优秀的测试工具,对这些工具所体现的思想也有深刻的理解,很多大型系统在国内都得到了很好的测试。

## 1.2 软件测试的基本理论

软件测试在软件生存周期中横跨两个阶段。通常在编写出每个模块之后就对它进行必要的测试(称为单元测试),模块的编写者和测试者是同一个人,编码和单元测试属于软件生存周期的同一个阶段。在结束这个阶段之后,对软件系统还要进行各种综合测试,这是软件生存周期中另一个独立的阶段,通常由专门的测试人员来完成这项工作。

目前,人们越来越重视软件测试,软件测试的工作量往往占到软件开发总工作量的40%以上。在特殊情况下,测试那些重大的软件,例如核反应堆监控软件,其测试费用可能相当于软件工程其他步骤总成本的三倍到五倍。因此,必须高度重视软件测试工作,绝不能认为写出程序代码之后软件开发工作就完成了。

### 1.2.1 软件测试定义和目标

#### 1. 软件测试的定义

人们对于软件测试的目的可能会存在着这样的认识——测试是为了证明程序是正确

的。实际上,这种认识是错误的。因为如果为了表明程序是正确的而进行测试,就会设计一些不易暴露错误的测试方案,也不会主动去检测、排除程序中可能存在的一些隐患。显然,这样的测试对于发现程序中的错误,完善和提高软件质量的作用不大。因为程序在实际运行中会遇到各种各样的问题,而这些问题可能是在设计时没有考虑到的,所以在设计测试方案时,应该尽量让它能发现程序中的错误,从而在软件投入运行之前就将这些错误改正,最终把一个高质量的软件系统交给用户使用。

通常对软件测试的定义有如下描述。

软件测试是为了发现程序中的错误而执行程序的过程。具体说,它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,并利用测试用例来运行程序,以发现程序错误的过程。

正确认识测试的目的十分必要,只有这样,才能设计出最能暴露错误的测试方案。此外,还应该认识到:测试只能证明程序中错误的存在,但不能证明程序中没有错误。因为,即使经过了最严格的测试之后,仍然可能还有没被发现的错误存在于程序中,所以说测试只能查出程序中的错误,但不能证明程序没有错误。

## 2. 软件测试的目标

软件测试工作非常必要,测试的目的是在软件投入运行之前,尽可能多地发现软件中的错误。软件测试是对软件规格说明、设计和编码的最后复审,是保证软件质量的关键步骤。

实现软件测试目的的关键是如何合理地设计测试用例,在设计测试用例时,要着重考虑那些易于发现程序错误的方法策略与具体数据。

综上所述,软件测试的目的包括以下三个方面。

(1) 测试是程序的执行过程,目的在于发现错误,不能证明程序的正确性,仅限于处理有限的情况。

(2) 检查系统是否满足需求,这也是测试的期望目标。

(3) 一个好的测试用例在于发现还未曾发现的错误;成功的测试是发现了错误的测试。

### 1.2.2 软件测试标准

软件测试的标准是站在用户的角度,对产品进行全面测试,尽早、尽可能多地发现缺陷,并负责跟踪和分析产品的问题,对不足之处提出质疑和改进意见。

软件测试标准如下。

(1) 软件测试的目标在于揭示错误。测试人员要始终站在用户的角度去发现问题,系统中最严重的错误是那些导致程序无法满足用户需求的错误。

(2) 软件测试必须基于“质量第一”的思想去开展各项工作。

(3) 事先定义好产品的质量标准。只有建立了质量标准,才能根据测试的结果,对产品的质量进行分析和评估。

(4) 软件项目一旦启动,软件测试也就开始,而不是等程序写完,才开始进行测试。

(5) 测试用例是设计出来的,不是写出来的,所以要根据测试的目的,采用相应的方法去设计测试用例,从而提高测试的效率,更多地发现错误,提高程序的可靠性。