

软件质量存在的问题及改进方法

于秀山, 张恒喜

(空军工程大学 工程学院, 陕西 西安 710038)

摘要:阐述了软件质量的度量方法,并据此对软件质量存在的问题进行了分析和分类。通过对软件质量的度量,总结了软件在易使用性、容错性、易安装性及文档资料方面容易出现的31类问题,给出了对软件进行测试的实例。实践表明,这种软件质量缺陷分类方法可操作性强,对软件开发人员有很大的帮助,有利于提高软件的质量。

关键词:软件质量;软件测试;软件错误;计算机软件

中图分类号:TP311.5 **文献标识码:**A **文章编号:**1009-3516(2001)02-0019-04

随着信息时代的到来,计算机几乎已渗透到生活的各个方面,人们对软件的质量越来越关心。毫无疑问,软件开发人员的技术水平直接关系到软件的质量。然而,由于各种原因,开发人员在软件设计时存在各种各样的问题,这些问题突出表现在只注重功能实现,忽视了与软件质量息息相关的其它方面。作者于1999年3月至2000年4月参加了某工程软件测试工作,共测试14个不同类型的软件,发现各类错误358个。本文结合对软件的测试实践,从软件质量的角度,把软件测试中发现问题进行了分析和分类,给出了测试实例,供开发借鉴,进而推动软件质量的全面提高。

1 软件质量度量

软件质量是指软件产品满足给定需求的特性及特征总体的能力。如何对一个软件的质量进行评价和度量呢?对此国内外有不同的标准和规范^[1-2]。

1.1 ISO/IEC 9126-91

ISO/IEC 9126-91将软件质量分为六个特性及21个子特性。其中的六个特性为功能性、可靠性、易使用性、效率、可维护性、可移植性。

1.2 McCall 质量模型

McCall质量模型是由McCall于1977年提出的。其结构的顶层为11个质量特性,每个特性又分解成许多质量特征。一个特定的质量特性可以与许多其它质量特征相关连,因此其分层结构不再是树形结构,而是网状结构。每一个特性都是由一定的标准来确定,为了进行定量评估,要对每一个特性进行度量。但是这样做是有困难的,有的是不可能的。因此通常可以用以下方法定义每个特性的回归公式^[3]

$$QC = r_1 \times m_1 + r_2 \times m_2 + \dots + r_n \times m_n$$

式中, QC 是指每个特性; r_i 是回归系数; m_j 是度量。

2 软件错误分析及分类

综合上述的软件质量度量方法,对作者在多次软件测试中发现的开发人员容易忽视的问题进行分析后,分类^[4-5]如下:

2.1 易使用性

收稿日期:2000-05-31

作者简介:于秀山(1962-),男,山东栖霞人,高级工程师,博士生,主要从事软件可靠性研究。

易用性方面的错误主要发生在用户界面的设计上。常见的错误如下:

1) 窗口的设计风格不一致。如窗口的布局、字体的大小、字体的颜色等各不相同。

2) 相同功能的按钮标识不一致。如实现“退出”功能的按钮,在有的窗口上为“退出”,而在其它窗口上则为“返回”。

3) 功能提供的不一致。在一个窗口中具有某种功能在另一个窗口中却没有。如对数据库进行检索的功能,在一个窗口中有“上一个”、“下一个”、“到记录首”、“到记录尾”按钮,而在另一个执行相同操作的窗口中却没有提供相同的按钮。

4) 消息框和弹出窗口中的信息表达不清或令人困惑。

5) 没有给出错误提示或错误提示过于模糊。

6) 没有给出必要的帮助信息。

7) 没有给出标准的输入格式。

8) 没有给出正确的缺省值。

9) 要求用户进行多余的、不必要的操作才能达到预期的窗口的导航。

10) 要求用户输入本来应用程序可以提供的数据。

11) 强制所有系统中的用户使用相同的选项设置。

2.2 容错性

容错性是软件可靠性的重要体现之一。它是用来衡量软件在出现错误或软件界面出现非法输入的情况下,软件维持规定性能的能力。测试数据的统计结果表明,容错性是软件开发人员最容易忽略和最容易出错的地方。常见的错误如下:

1) 对于文本字段,输入框没有对可输入的最大长度进行限制,从而用户可输入无限长度的数据,导致提交软件时出现错误;有的软件不能处理空字段。

2) 对于数字字段,没有对该字段进行充分的限制,可输入字符数据、非整数数据、负数数据;有的软件不能处理最大、最小和精度最高的数据。

3) 对于日期字段,没有对该字段的类型进行检查;没有对月份应 ≤ 12 ,天数 ≤ 31 (30,28,29)进行检查。

4) 对一些可输入很长内容的字段,在该字段为空或为最大值时,软件不能正常显示,同时也不能正常打印,甚至死机。

5) 没有对非空字段即必须输入数据的字段进行限制。

6) 没有对同名字段或文件给出重名提示信息。

7) 在文件不存在或设备不存在、设备出故障时,软件没有进行处理。

8) 没有考虑在磁盘空间不足时,软件应如何处理。

9) 没有考虑在业务量超过正常情况时,软件应如何处理。

2.3 效率

达到所要求的效率指标同完成要求的功能一样重要。如果性能未达到设计要求或是达到可被用户接受的程度,那么无论系统是多么的复杂和达到了怎样的容错水平,用户和管理人员也将认为系统是失败的。由于不同的软件有不同的效率指标,所以无法进行统一。以下给出在进行有关数据库设计时要注意的几个问题。

1) 并发控制。由于并发控制要在多用户环境下才能够验证,所以开发人员有时难以开展此项工作。若软件存在并发控制问题,软件的响应时间将变慢,甚至被锁死。大多数的 RDBMS 提供了能帮助识别并发问题的小工具,利用这些工具检查那些已经开始但却停顿下来的 SQL 任务,其状态信息将显示它们正等待数据库解锁。

2) 事务处理。设计不当或有错误的事务将使软件的响应时间变慢,甚至被锁死。通常事务在执行时,包含一个排它锁和若干个数据库表。在事务完成之前,其他的用户是无法访问这些表的。如果程序中有错,那么用户可能会使一个事务执行无限长的时间。正确的程序应能避免这种情况,也就是不允许将其事务处于永远打开的状态。

3) 查询优化。不同的 SQL 查询语句其执行效率差别很大。一条 SQL 语句可能引用了多余的数据表,使用了不必要的子选择,或者包含了会减慢系统速度的系统函数,也可能是在 where 子句中包含有不必要的条件。要通过进行代码检查或使用有关评测工具找出低效的 SQL 语句。

4) 成组处理与按行处理。关系数据库可以成组处理数据。在数据库查询中,无论是 select 语句、update 语句还是 delete 语句,都可以改变满足它们的 where 子句中条件的一组记录,而不止是一行记录。如实现把月工资等于 1 000 元的职工工资提高到每月 1 200 元这样一个操作,可以写出许多 SQL 语句来实现。下面给出按组处理和按行处理的两个实现方法:

```
UPDATE employee SET salary = 1200 WHERE salary = 1000;
```

在此方法中,只用了一条 SQL 语句就实现了相应的功能。另外,如果该软件是运行在分布处理的环境下,则所有工作都是在服务器上完成的,在整个过程中不需要通过网络传送记录信息。

```
SELECT employee _ id = employee _ id, salary = salary FROM employee
{
    IF salary = 1000 THEN
        UPDATE employee SET salary = 1200 WHERE employee _ id = employee _ id;
    ENDIF
}
```

此方法是逐行地把表中的数据读入到局部变量中,然后检查其中的 salary 字段是否等于 1 000,然后把那些等于 1 000 的行进行修改,这种修改也是一次只改动一项记录。如果该软件是运行在分布处理的环境下,则所有数据必须通过网络传送到本地计算机,显然,这将加重网络负担,最终使得响应时间达不到指标要求。

2.4 易安装性

易安装性是可移植性中的一个重要指标。常见的错误如下。

- 1) 没有详细的安装向导。
- 2) 在磁盘空间不足、缺创建目录权限、存在同名目录的情况下软件处理不正确。
- 3) 没有提供卸载功能。

2.5 文档资料

文档资料是可维护性中的一个重要指标。但在实际的软件开发过程中,这方面的工作往往被忽视。具体表现在:

- 1) 没有文档资料或文档资料没有被及时更新。
- 2) 文档资料不全、不详细。
- 3) 源程序中没有或只有少量的注释。
- 4) 没有对软件版本进行控制和管理。

3 软件自测实例

开发人员在软件的开发过程中通常对软件进行了多次测试,但这些测试大部分是基于正常的操作,其充分度远远不够。表 1 给出一个对登录屏幕应进行的常规测试内容,以期对开发人员能有所帮助。

表 1 登录屏幕所需测试

测试事例	预期结果
不输入 ID 和口令,单击确定按钮	将有一个错误信息显示出来
输入有效的 ID 并单击确认按钮	将有一个错误信息显示出来
输入有效的口令并单击确定按钮	将有一个错误信息显示出来
输入无效的 ID 并单击确定按钮	将有一个错误信息显示出来
输入无效的口令并单击确定按钮	将有一个错误信息显示出来
多次输入无效的信息	将有一个错误信息显示出来。当登录次数超过一定的值时,系统将会自动中止
单击取消按钮	应用程序将被中止
以大写、小写、字母、数字或混合输入	根据应用程序的设计而定
输入有效的 ID 和口令并单击确定按钮	成功登录

4 结束语

本文依据国内外有关软件质量的度量标准,通过对不同软件进行测试,总结归纳了有关软件质量方面容易出现的问题,这些问题在软件开发人员中具有一定的普遍性。实践表明,通过对这些问题进行分类,开发人员可以清楚地知道在软件设计中容易出现的问题,能够及时采取相应的措施,推动软件质量的全面提高。

参考文献:

- [1] GB/T 16260 - 1996, 信息技术-软件产品评价-质量特性及其使用指南[S].
- [2] GJB 439 - 88, 军用软件质量保证规范[S].
- [3] 罗圣仪. 计算机软件质量保证方法和实践[M]. 北京: 科学出版社, 1999.
- [4] Pressman Roger S. Software Engineering[M]. Beijing: China Machine Press, 1999.
- [5] Kaner Gem, Fahl Jack, Hung Quoc Nguyen. Testing Computer Software[M]. New York: Van Nostrand Reinhold, 1993.

The Problems in Software Quality and Its Improving Method

YU Xiu-shan, ZHANG Heng-xi

(The Engineering Institute of the Air Force Engineering University(AFEU.), Xi'an 710038, China)

Abstract. We describe the measurement of software quality, analyze and classify the problems found in software quality. We then summarize 31 kinds of problems in usability, fault-tolerance, efficiency, installability and document. Finally, the instance of software testing is provided. It proves that this kind of classification can be easily manipulated and is very helpful to software developer. In particular, the method is favourable to the improvement of software quality.

Key words. software quality; software testing; software error; computer software