

# 软件可靠性测试及其实践

Software Reliability Testing and Practice

**【摘要】**软件可靠性测试是软件可靠性工程的一项重要工作内容,是满足软件可靠性要求、评价软件可靠性水平及验证软件产品是否达到可靠性要求的重要途径。本文探讨、研究了软件可靠性测试的基本概念,软件可靠性测试过程以及软件可靠性测试中的主要问题,还介绍了一个实际软件的可靠性测试工作。

**关键词:**软件可靠性,软件可靠性测试,软件测试,软件运行剖面

**Abstract:** Software reliability testing is an important task in software reliability engineering. It serves as the main means to achieve the software reliability requirements, evaluate software reliability levels and demonstrate whether a software product has achieved its reliability requirement. This paper investigates the basic concepts of software reliability testing and software reliability testing procedures, discusses the key issues in it. A practical work on a real software product is also presented.

**Key words:** software reliability, software reliability testing, software testing, operational profile

软件可靠性工程是指为了满足软件的可靠性要求而进行的一系列设计、分析、测试等工作。其中确定软件可靠性要求是软件可靠性工程中要解决的首要问题。软件可靠性要求可以包括定性及定量要求。

软件可靠性测试是在软件生存周期的系统测试阶段提高软件可靠性水平的有效途径。各种测试方法、测试技术都能发现导致软件失效的软件中残存的缺陷,排除这些缺陷后,一般来讲一定会实现软件可靠性的增长,但是排除这些缺陷对可靠性的提高的作用却是不一样的。其中,软件可靠性测试能最有效地发现对可靠性影响大的缺陷,因此可以有效地提高软件的可靠性水平。

软件可靠性测试也是评估软件可靠性水平,验证软件产品是否达到软件可靠性要求的重要且有效的途径。

## 1 软件可靠性测试概念

“测试”一般是指“为了发现程序中的错误而执行程序的过程”。但是在不同的开发阶段、对于不同的人员,

测试的意义、目的及其采用的方法是有差别的。在软件开发的测试阶段,测试的主要目的是开发人员通过运行程序来发现程序中存在的缺陷、错误。而在产品交付、验收阶段,测试主要用来验证软件产品是否达到用户的要求。或者说,对于开发人员,测试是发现缺陷的一种途径、手段,而对于用户,测试则是验收产品的一种手段。根据测试用例选取原则的不同,测试可分为黑盒测试方法和白盒测试方法两大类。黑盒测试方法是指按照软件需求生成测试用例对软件进行测试的方法,黑盒测试不关心程序是如何实现的;而白盒测试方法则是指根据程序的结构生成测试用例对软件进行测试的方法。

软件可靠性测试是指为了保证和验证软件的可靠性要求而对软件进行的测试。其采用的是按照软件运行剖面(对软件实际使用情况的统计规律的描述)对软件进行随机测试的测试方法。通过软件可靠性测试可以达到以下目的:

(1) 有效地发现程序中影响软件可靠性的缺陷,从而实现可靠性增长:软件可靠性是指<sup>[4]</sup>“在规定的时间内,规定的条件下,软件不引起系统失效的能力,其概率度量称为软件可靠度。”软件的“规定的条件”主要包括相对不变的条件和相对变化的条件,相对不变的条件如计算机及其操作系统;相对变化的条件是指输入的分佈,用软件的运行剖面来描述。按照软件的运行剖面对软件进行测试一般先暴露在使用中发生概率高的缺陷,然后是发生概率低的缺陷。而高发生概率的缺陷是影响产品可靠性的主要缺陷,通过排除这些缺陷可以有效地实现软件可靠性的增长。

(2) 验证软件可靠性满足一定的要求:通过对软件可靠性测试中观测到的失效情况进行分析,可以验证软件可靠性的定量要求是否得到满足。

估计、预计软件可靠性水平:通过对软件可靠性测试中观测到的失效数据进行分析,可以评估当前软件可靠性的水平,预测未来可能达到的水平,从而为开发管理提供决策依据。软件可靠性测试中暴露的缺陷既可以是影响功能需求的缺陷也可以是影响性能需求的缺陷。软件可靠性测试方法从概念上讲是一种黑盒测试方法,因为它是面向需求、面向使用的测试,它不需要了解程序的结构以及如何实现等问题。

软件可靠性测试通常是在系统测试、验收、交付阶段进行,它主要是在实验室内仿真环境下进行,也可以根据需求和可能在用户现场进行。

## 2 软件可靠性测试过程

### 2.1 软件可靠性测试活动

软件可靠性测试的一般过程如图 1 所示。主要活动包括:测试数据、测试环境的准备,测试运行,可靠性数据收集,可靠性数据分析和失效纠正。

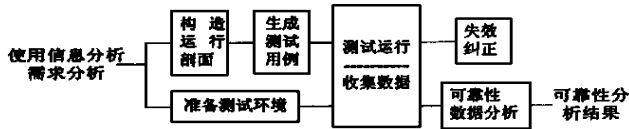


图 1 软件可靠性测试过程

(1) 构造运行剖面:软件的运行剖面“是指对系统使用条件的定义。即系统的输入值用其按时间的分布或按它们在可能输入范围内的出现概率的分布来定义<sup>[5]</sup>。粗略地说,运行剖面是用来描述软件的实际使用情况的。运行剖面是否能代表、刻画软件的实际使用取决于可靠性工程人员对软件的系统模式、功能、任务需求及相应的输入激励的分析,取决于他们对用户使用这些系统模式、功能、任务的概率的了解。运行剖面构造的质量将对测试、分析的结果是否可信产生最直接的影响。

(2) 选取测试用例:软件可靠性测试采用的是按照运行剖面对软件进行可靠性测试的方法。因此,可靠性测试所用的测试用例是根据运行剖面随机选取得到的。

(3) 测试环境的准备:为了得到尽可能真实的可靠性测试结果,可靠性测试应尽量在真实的环境下进行,但是在许多情况下,在真实的环境下进行软件的可靠性测试很不实际,因此需要开发软件可靠性仿真测试环境。比如,对于多数嵌入式软件,由于与之交联的环境的开发常常与软件的开发是同步甚至是滞后的,因此无法及时进行软件可靠性测试;有些系统中,由于交联的环境非常昂贵而无法用于需要进行大量运行的可靠性测试。

(4) 可靠性测试运行:即在真实的测试环境中或可靠性仿真测试环境中,用按照运行剖面生成的测试用例对软件进行测试。

(5) 数据收集:收集的数据包括软件的输入数据、输出结果,以便进行失效分析和进行回归测试;软件运行时间数据,可以是 CPU 执行时间、日历时间、时钟时间等;可靠性失效数据包括每次失效发生的时间或一段时间内发生的失效数,失效数据可以实时分析得到,也可以事后分析得到。数据收集的质量对于最终的可靠性分析结果有着很大的影响,应尽可能采用自动化手段进行数据的收集,以提高效率、准确性和完整性。

(6) 数据分析:主要包括失效分析和可靠性分析。

失效分析是根据运行结果判断软件是否失效,以及失效的后果、原因等;而可靠性分析主要是指根据失效数据,估计软件的可靠性水平,预计可能达到的水平,评价产品是否已经达到要求的可靠性水平。为管理决策提供依据。

(7) 失效纠正:如果软件的运行结果与需求不一致,则称软件发生失效。通过失效分析,找到并纠正引起失效的程序中的缺陷,从而实现软件可靠性的增长。

### 2.2 软件可靠性增长测试过程

软件可靠性增长测试是为了满足用户对软件的可靠性要求、提高软件可靠性水平而对软件进行的测试。是为了满足软件的可靠性指标要求,对软件进行测试—可靠性分析—修改—再测试—再分析—再修改的循环过程。软件可靠性增长测试过程如图 2 所示。

### 2.3 软件可靠性验证测试过程

软件可靠性验证测试是为了验证在给定的统计置信度下,软件当前的可靠性水平是否满足用户的要求而进行的测试,即用户在接收软件时,确定它是否满足软件规格说明书中规定的可靠性指标。一般在验证过程中,不对软件进行修改。软件可靠性验证测试过程如图 3 所示。

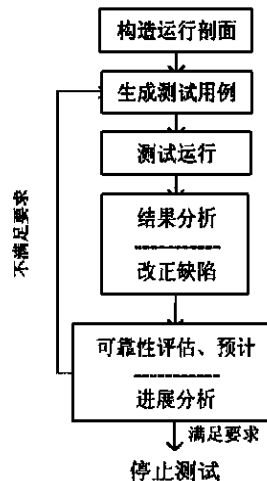


图 2 软件可靠性增长过程

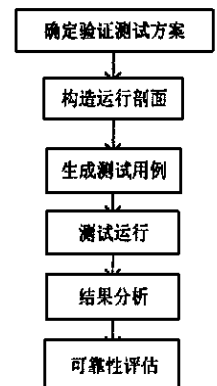


图 3 软件可靠性验证过程

## 3 某故障诊断专家系统软件可靠性测试

《××飞机起落架故障诊断专家系统》软件是一个使用“专家系统”方法对××飞机起落架进行故障诊断的软件。该软件是用 VC++ 开发的,源代码近一万条。我们在该软件调试的后期对其进行了软件可靠性测试,主要工作包括,运行剖面的构造和测试用例的生成,测试运行及数据收集,可靠性数据分析。

### 3.1 运行剖面的构造及测试用例生成

根据被测软件功能的说明,结合软件的有关文档,以及对相关概率的估计,可构造软件的系统模式剖面、功能剖面 and 运行剖面。

运行剖面的构造是一个自顶向下的层次结构。通

过不断细化被测软件的输入空间,即从划分系统模式剖面到功能剖面,直到各功能输入变量的取值区间在概率空间的划分,形成最终的运行剖面。这里一个运行如下规定:运行是由完成某一功能的一系列输入变量的某一取值区间的有序组合。各运行在概率空间的划分构成运行剖面。

测试用例是根据运行剖面生成的,完成对某一功能进行测试,按顺序输入到被测软件的一系列输入变量值的有序组合。

由于运行剖面描述了完成某一功能输入变量的取值区间,通过两次随机抽样,可以得到一个测试用例。第一次抽样选择运行,第二次抽样在每一个输入变量取值区间内随机抽取输入变量的具体取值。将其按照测试过程中的输入顺序组合起来形成测试用例。一个测试用例的具体形式如下:

测试用例序号	输入变量名称	输入变量的具体取值
1	var1	menu-1-1
...		

### 3.2 测试运行及数据收集

按照上述方法生成了 400 个测试用例。在一台配置为 Pentium 586 at 133 MHz,内存 16 MB,操作系统为 Windows 95 中文版环境的计算机上,通过手动方式将测试用例输入到被测软件,利用一个为配合这种软件可靠性测试方法而开发的数据辅助收集软件,采集测试运行的时间与失效信息,包括测试用例序号、测试日期、测试开始时刻、测试结束时刻/失效发生时刻、测试运行时间、累计运行时间、失效现象等。通过测试记录下了 60 次失效。收集到的失效数据如表 1 所示,数据从左至右,从上至下,为每次失效发生的累计运行时间(执行时间)。

表 1 某软件可靠性测试失效数据 单位:s

897.8	977.4	1 419.9	1 539.0	1 615.8
1 711.1	1 923.3	1 997.0	2 070.3	2 708.2
3 067.9	3 460.5	4 078.2	4 471.8	5 695.8
5 807.8	6 120.8	6 444.8	6 704.1	6 879.7
7 002.5	8 688.7	8 947.7	9 134.9	9 687.3
10 229.2	10 954.5	11 084.0	11 387.5	12 086.7
12 329.4	12 951.9	13 084.6	13 608.8	13 707.2
13 975.7	14 068.3	14 451.1	14 552.2	14 634.0
14 800.3	15 398.8	15 455.5	15 694.3	15 804.5
15 897.5	15 971.8	16 090.6	16 618.0	17 820.9
18 622.4	18 731.6	18 966.8	20 356.5	20 510.6
21 116.0	21 248.0	21 320.6	21 655.8	21 783.7

### 3.3 可靠性数据分析

表 1 所示失效数据是一组完全失效数据,首先用自行开发的软件可靠性分析工具(SRAT)对数据进行了趋势分析,分析表明,软件的可靠性呈现稳定趋势,即软件具有不变的失效率,失效时间服从指数分布。因此可计

算失效率和平均失效间隔时间分别为:

$$= \text{总失效数} / \text{总运行时间} = 60 / 21\ 783.7 = 0.002\ 75 \text{ (失效数/s)}; \text{MTBF} = 1 / \quad = 363 \text{ s}.$$

事实上,在测试中,每次失效发生后,并没有对软件进行失效纠正,因此,失效率应该是不变的,数据分析结果也验证了这一点。分析结果表明,该软件的可靠性尚需进一步提高。事实上,上述失效数据中,许多失效是由相同的缺陷造成的。如果对相同的失效只考虑首次发生的失效,亦即首次发现就加以纠正的话,软件的可靠性将得到很大的提高。需要强调的是,该分析结果是在给定的运行剖面下、在给定的运行环境下进行测试得到的分析结果。不同的运行剖面,不同的运行环境(如不同的机器速度)会得到不同的可靠性估计。另外,所收集的失效时间数据的类型也会影响数据分析的结果。

## 4 结论

软件可靠性测试是面向可靠性要求的测试,是软件可靠性工程中的一重要工作,它能有效地发现影响软件可靠性的缺陷,通过软件可靠性测试,可以有效地实现软件可靠性的增长,估计软件的可靠性水平,验证软件可靠性是否达到要求。但是对于软件可靠性测试的困难和局限性,也应有充分的认识。

(1) 软件可靠性测试是一项高投入的测试工作:进行软件可靠性测试必须要了解软件过去的历史,或估计可能的使用,构造软件的运行剖面,准备测试环境,且要进行大量的测试运行。

(2) 软件可靠性测试不能代替其他测试和验证方法:从有效发现缺陷角度出发,软件可靠性测试可能不是最有效的方法,必须结合其他的测试和验证方法、手段发现软件中存在的各种缺陷。

(3) 难以验证具有极高可靠性要求的软件:对于有极高可靠性要求的软件,如失效率为  $10^{-9}$ ,用软件可靠性测试的方法进行验证所需的时间是不切合实际的,必须采用如形式化验证等方法来加以解决。

### 参考文献

- 1 Michael L, ed., Handbook of Software Reliability Engineering, McGraw Hill and IEEE Computer Society Press, 1996
- 2 陆民燕. 软件可靠性增长测试的研究. 航空学报, 1995, 16
- 3 陈雪松. 软件可靠性测试技术研究: [硕士论文]. 北京航空航天大学, 1997
- 4 GB/T 11457 - 89: 中华人民共和国国家标准——软件工程技术语
- 5 ESA 空间系统软件产品保证要求. 欧空局标准 PSS - 01 - 21, 1991

作者简介:陆民燕,女,1963年生,北京航空航天大学可靠性工程研究所副教授,工程硕士。曾赴英国城市大学的软件可靠性中心进修,参加了欧共体信息技术研究计划(ESPRIT)中的基础研究项目“可预计的可信的计算系统”的研究工作。现主要从事软件可靠性管理、度量及测试技术的研究和应用工作。在国内学术刊物、会议上发表论文 10 余篇,现为 IEEE 会员。

(收稿日期:2000 - 03)