# 计算机软件可靠性质疑

中国航天科工集团公司三院三部研究员　关世义

**编者按**：关于大型软件开发中的可靠性与维修性指标问题，至今尚无定论，在认识和实践上还没有一个固定的模式可以遵循。本文提出的计算机软件可靠性质疑问题，可供同仁们研究和讨论，欢迎大家参与争鸣。

**文　摘**　从工程实际出发，讨论了目前一些大型软件系统开发中有关软件可靠性和维修性指标的问题，并提出了一些疑问。同时，提出了若干保证软件质量的建议。

**关键词**　计算机　硬件　软件　可靠性　维修性

## 1. 引言

近年来，随着计算机技术的迅速发展和普遍应用，计算机软件可靠性问题引起了人们广泛的注意。什么是计算机软件的可靠性？它同硬件可靠性有什么关系？用什么指标来表示软件的可靠性？怎样解决软件可靠性问题？软件使用寿命是什么意思？这些都是计算机软件的开发者和软件用户十分关心的问题。

软件的开发和测试实际上是一个容易出错的过程。不管开发者怎样努力去检查每一个细节，也不能完全排除软件设计或编程过程中存在的问题或缺陷。开发当中所犯的错误常常会引起发布以后的软件失效。软件可靠性理论是工业界预估软件失效的基本方法之一。不幸的是，对于软件可靠性理论所作的假设并没有阐明大多数软件的复杂性，结果，这种理论很少在实际中得到采用。

软件可靠性理论似乎在通讯和宇航方面得到成功应用。其原因有二：首先，政府部门基本上可以对这两个部门的产品质量进行调控；第二，通讯和宇航方面的软件通常都工作于嵌入环境，并且在许多情况下，它们同所嵌入的硬件是不可分离的。这些软件被称为嵌入式软件。由于软件与其硬件环境密切相关，因此直接把硬件可靠性理论应用于硬件起主导作用的嵌入式软件，这中间并不存在多大差距。我国现有的国军标“GJB/Z 102-97 软件可靠性和安全性设计准则” [3] 中明确规定其适用范围主要是：“武器装备嵌入式软件的需求分析、设计和实现”。

今天，情况已经发生了很大的变化。软件的准确测试不再局限于一些特殊的工业部门，对于那些“压缩包装”（shrink-wrap）的软件，这种测试特别需要。许多在网络（局域网或广域网）中运行的大型软件系统，诸如管理信息系统（MIS）、决策支持系统（DSS）、决策信息系统（DIS）、地理信息系统（GIS）、分布仿真系统（DIS）和任务规划系统（MPS），等等，同以往的嵌入式软件存在着很大的差别。网络计算的普遍采用使得软件的质量问题成为人们关注的中心。“带病”的操作系统、网络浏览器，以及用户终端机的应用都可能带来安全方面的漏洞。这些漏洞将给计算机病毒提供许多入口，并使机器不能正常工作。因此，人们期望有一种科学的软件可靠性理论产生。

笔者是搞系统工程的，而不是一个专业的软件工程师。因此，在这里不打算讨论有关软件设计的问题，以免“班门弄斧”之嫌。但是，实际工作中，我们往往遇到这样的问题，一些用户把硬件可靠性的指标原封不动地搬到大型软件系统上，如 MTBF、MTTR、使用寿命等等，并把它们作为该软件系统的技术指标。同时，像硬件设计一样，还要求把这些指标分配到下层的模块。这种做法存在很多问题，实际上是行不通的。下面打算提出我们的一些疑问，希望得到有关专家和内行的指导。

## 2. 计算机软件的特殊性

目前开发者称之为软件可靠性的观点大都是直接借用或采用了硬件可靠性中比较成熟的内容。在以硬件为主体的系统和主要关心硬件的情况下，这种影响是明显的。在测试或应用硬件时，主要考虑日历时间；由于不断使用，硬件会受到磨损；由于

运输、振动、冲击、载荷的作用，硬件的一些连接件会松动，一些零件、部件甚至会变形或损坏；由于环境（高温、潮湿、盐雾等等）的影响，一些设备会锈蚀；某些材料特性，如塑料、橡胶和一些电子元器件等会变质、老化，等等。因此，完美无缺的硬件实际上是不存在的。解决的办法是:给产品规定一个保质期,并提供一定的备份件。

但是，计算机软件的情况有所不同。软件或软件的某些部分是可能设计得完美无缺的，而且会一直保持这种完美无缺的性能。软件的故障或失效是由于其潜在的设计或编程缺陷在开发和测试（或检测）过程中没有暴露出来。人们会问："在暴露软件的潜在缺陷方面，时间因素会起多大作用？用两周或两个月时间测试某一软件功能，会有很大区别吗？"回答当然是："视情况而定"。两个月时间可以给我们更多的机会去测试输入和数据的变化。但是，时间的长短并不是软件完整测试的决定性因素。对于一个优秀的软件测试者，两个月时间可能太长，而对于一个能力较差的测试者，两个月时间可能还不够。

仅仅考虑时间和运行剖面（或任务剖面）两个要素对于软件可靠性而言是不完整的。影响软件可靠工作至少应当考虑以下三个因素：

●软件及其应用的复杂性：一个复杂的软件系统包括应用软件、支撑软件和系统软件。应用软件与领域专家有着密切的关系，它们可能由许多模块组成，而这些模块又可能由许多子模块、软构件组成。支撑软件大都是一些商品化的软件包、工具箱、数据库等等。虽然这些支撑软件在发布（release）之前经过了各种测试，但它们都可能存在着一些潜在的设计"bug"(缺陷或隐患)。这些"bug"是否可以视为软件可靠性问题,显然是值得商榷的。这里尚未提及系统软件问题。实践表明,即使是系统软件,它们同样可能存在着"bug"。

●软件的任务剖面（或运行剖面）：任务剖面是否完全反映了将来软件使用中的情况，将影响软件测试的有效性。因此、对于软件的测试而言，任务剖面的确定是否合理和全面是一件至关重要的事情。运行一个软件就好像一个人在丛林中漫步，只要你走在熟悉的小路上，你就不会被老虎吃掉。但是，如果你偏离了这条小路，你就可能会遇到麻烦或危险。在确定任务剖面时，不能只考虑一个基本的用户。操作系统和多个应用程序对有限的计算机资源的抢占可能导致某个应用程序的失效。考虑到软件环境的方方面面，即使一个用户按照指定的任务剖面工作，也可能引起软件失效。在这些情况下，熟悉的小路并不安全，老虎可能无处不在。这里要问，按任务剖面测试，即使进行随机输入，所解决的是软件性能问题，还是可靠性问题？看来是性能问题。同时，应当指出，软件的测试或测评只是尽可能地在所设计的任务剖面内去找出软件设计中隐藏的缺陷或错误，测评单位或部门并不能保证经过测评以后的软件完全不存在问题。由此可见，软件的测评并不是100%有效的。

●软件的使用环境：这里包括硬件环境和软件环境，即操作系统（WINDOWS/NT、UNIX 等）。不同的使用环境，对软件的工作具有重要的影响。当更换操作系统时，许多软件甚至无法运行。因此，在开发应用软件时，往往提出跨平台（cross-platform）的要求。如上所述，操作系统本身也不是十全十美的，也可能存在一定的"bug"。由于操作系统的"bug"带来的运算失效，当然不应该算到所开发的应用软件上。

**3. MTBF 适于做软件可靠性指标吗?**

文献［1］关于软件可靠性的主要观点如下：

●软件可靠性的定义：软件可靠性是在规定环境下,在规定时间内,软件不引起系统失效的概率。或在规定的时间周期内所述条件下程序执行所要求的功能的能力。注意：软件可靠性不但与软件中存在的缺陷有关,而且与系统输入和系统使用有关。

●软件的平均失效前时间：一个软件产品在规定的条件下按任务剖面投入随机使用，到出现失效为止的时间叫"失效前时间"（time to failure）(简称为 TTF)。这里的随机使用意义是按输入点的概率分布随机使用。一个软件产品在多个场合使用，是一个随机变量，它的期望叫"平均失效前时间"（mean time to failure）（简称为 MTTF），它可以定义为：软件产品在规定的条件下随机使用时，在规定的时间内，软件产品的寿命单位总数与失效总次数之比。

一个软件产品在规定的条件下按任务剖面投入随机使用到出现系统失效为止的时间叫系统失效前时间（time to system down），简称为 TTD。一个软件产品在多个场合使用，出现的 TTD 是一个随

机变量，它的期望叫“平均系统失效前时间”(mean time to system down)，简称为 MTTD。

注：为照顾习惯，作者把它称为 MTBF。

●软件可靠性和可维修性指标体系：目前我国尚未建立有关标准。其实，我国目前有关软件的标准（国标和国军标）基本上都是针对嵌入式软件的，对于大型软件系统，这些标准未必适用。如上所述，影响软件出现故障或失效至少应考虑三个重要原因。但是，上述有关软件可靠性的定义并没有全面体现这些因素。

**4、计算机软件维修的含义**

软件的修理和修改是有关软件最基本的、也是目前国内外尚无定论的问题之一。一旦软件改变了，它就变成了一个完全新的相关产品。软件的修理和修改需要软件产品开发者来承担。修理和修改软件可能使其质量变得更好，也可能使其质量变得更坏，或者根本没有改变。问题是怎样在统计的意义上处理软件修理和修改？怎样处理故障和失效的复现？怎样考虑软件的回归试验？怎样保留旧的可靠性资料和添加新的资料，或者每一次对软件的修理和修改是否要从最原始的代码开始测试？如果不是通过测试的方法找到了软件失效，怎么办？可以采取与可靠性措施不同的办法吗？

问题在于：软件的开发者和用户在自己的脑海中现在还没有一幅清晰的图像，用以判明哪些参数对软件可靠性带来影响。相对于那些可以用数学方法处理的、简单的时钟转换和运行剖面而言，软件可靠性和维修性问题要复杂得多。因此，在一些大型软件系统开发中，提出软件维修性指标，也就没有任何实际的意义。

**5、关于软件寿命问题**

在一些大型应用软件系统方案中，要求开发者给出软件的使用寿命，如规定某系统的软件使用寿命为某一年限；或者要求软件寿命不低于相应硬件的寿命。这里要问：软件寿命指标是怎样提出来的？又怎样去检验它？如果规定一个大型软件的正常使用年限，那么，在该年限之后软件为什么会突然“寿终正寝”？如上所述，一个软件如果性能上不存在问题，那么，在设计允许的条件下使用，除了其载体（介质）可能变质以外，软件本身将永远不会受到“磨损”而变质。因此，我们认为，不应该像对待硬件一样，对软件寿命提出指标要求。这样作，既无依据，也无从检验。

这里需要提及软件的更新换代或所谓版本升级（更新）问题。即使不存在软件寿命问题，软件的开发者也会对现有软件不断地进行完善、改进和更新，原因是：

●改正开发者或用户在使用现有软件过程中新发现的设计缺陷和错误；

●完善和改进现有软件中某些部分（或模块）的功能；

●为现有软件添加一些新的功能模块或软构件。

实际的情况往往是，未等到现有软件使用期限的到来，该软件恐怕已经进行多次版本升级了。当然，软件版本的升级是需要付出许多劳动和代价的。因此，如果用户希望将自己现有的软件版本升级，就需要为此支付一定的费用，这一点往往反映在软件的开发或购买合同中。

**6、结束语**

综上所述，对于大型软件系统及其应用软件，套用硬件可靠性和维修性的概念、理论和方法，看来是不合理的、行不通的。对于软件可靠性而言，需要有创新思维，需要有新的可靠性概念、理论和方法。但是，目前国内外还没有这套新的东西。为了使新开发的软件具有较高的质量，应当遵循软件工程要求，采用科学的集成思想和设计方法；采用高质量的硬件、网络设备和成熟的软件开发平台；采取必要的安全保护措施；加强软件开发过程中的质量管理；加强软件开发过程中的质量检查，如贯彻“三检”（自检、互检、专检）制度；加强软件的专业性综合测评；加强用户培训和软件试用等等。实践表明，以上措施对于提高软件质量是行之有效的，应当积极贯彻之。同时，我们将以积极的态度期待着新的软件可靠性理论及其指标体系的产生。

**参考文献**

1　何国伟主编.软件可靠性.北京：国防工业出版社，1998

2　Whittaker, J.A.& Voas, J. Toward a More Reliable Theory Of Software Reliability. Computer, December 2000

3　中华人民共和国军用标准 GJB/Z 102-97.软件可靠性和安全性设计准则.1997.11.05