高等学校教材·软件工程

软件可靠性工程

徐仁佐　编著

清 华 大 学 出 版 社
北　京

内 容 简 介

本书以软件可靠性工程的一些常见问题为出发点，以编者多年来所参与的工程实践为依托，帮助读者对软件质量指标体系中的最重要的质量指标之一——软件可靠性有一个全面的理解，并具有一定的实践能力。

全书共12章，各章均附有习题，一部分是为了复习、巩固本章所学的知识，另一部分是为引导学生进行创新型思维。本书最后提供了包括最新领域发展的参考文献，供有兴趣的读者进一步阅读和学习。

本书语言流畅，结构合理，内容丰富，实例众多，着重理论与实践相结合，学以致用，适合作为高等院校软件工程、计算机及相关专业的本科生和研究生教材，也可以作为软件从业人员及一般读者的参考书。

图书在版编目(CIP)数据

软件可靠性工程/徐仁佐编著. —北京：清华大学出版社，2007.5
(高等学校教材·软件工程)
ISBN 978-7-302-14293-5

Ⅰ. 软…　Ⅱ. 徐…　Ⅲ. 软件可靠性－软件工程－高等学校－教材　Ⅳ. TP311.5

中国版本图书馆CIP数据核字(2006)第152712号

责任编辑：付弘宇　徐跃进
责任校对：李建庄
责任印制：
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　　**邮购热线**：010-62786544
投稿咨询：010-62772015　　**客户服务**：010-62776969
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185×260　　**印张**：21.75　　**字数**：541千字
版　　次：2007年5月第1版　**印　　次**：2007年5月第1次印刷
印　　数：1～　000
定　　价：　　元

出版说明

改革开放以来，特别是党的十五大以来，我国教育事业取得了举世瞩目的辉煌成就，高等教育实现了历史性的跨越，已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上，高等教育规模取得如此快速的发展，创造了世界教育发展史上的奇迹。当前，教育工作既面临着千载难逢的良好机遇，同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾，是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001 年 8 月，教育部下发了《关于加强高等学校本科教学工作，提高教学质量的若干意见》，提出了十二条加强本科教学工作提高教学质量的措施和意见。2003 年 6 月和 2004 年 2 月，教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件，指出“高等学校教学质量和教学改革工程”是教育部正在制定的《2003—2007 年教育振兴行动计划》的重要组成部分，精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间（2003—2007 年）建设 1500 门国家级精品课程，利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放，以实现优质教学资源共享，提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻

性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统。

(6) 高等学校教材·财经管理与计算机应用。

清华大学出版社经过20多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

前言

自从美国 AT&T Bell 实验室的 J. D. Musa 于 1991 年在美国得克萨斯州的奥斯汀市召开的第十三届国际软件工程会议(The 13th International Conference on Software Engineering)上正式提出软件可靠性工程(software reliability engineering)的概念,至今已过了十多年的时间。

当时,Musa 先生将软件可靠性工程仅限于"预计、测量、管理以软件为基础的系统的可靠性,以最大限度地满足用户需要的应用科学"的范畴。随着近十几年的软件可靠性工程的实践与发展,随着软件工程的进步,软件可靠性工程的范畴也在逐步地拓宽。软件可靠性工程发展到今天,伴随着软件开发的全过程,在软件生命周期的各个阶段,都发展出了一系列的技术和管理方法。纵观当今软件可靠性工程的实践,可以说,软件可靠性工程是与软件开发过程各阶段同步并行的,用于设计、保证、计量、管理软件可靠性目标的应用科学。由香港中文大学的吕容聪(Michael R. Lyu)教授编辑、IEEE Computer Society Press 出版的《The Handbook of Software Reliability Engineering》一书,收入了一系列早期有关软件可靠性工程的文献,对推动软件可靠性工程的发展起到了一定的作用。但此书中远未包括软件可靠性工程研究的全部,特别是近年来开发的方法和技术。

软件可靠性工程属于软件工程范畴,正如软件需求工程、软件再工程、软件复用工程、软件测试工程等一样,是软件工程的一个子领域。软件可靠性工程的发展要从属于软件工程发展的需要,反之,软件可靠性工程的发展又可以促进软件工程的发展,软件可靠性工程的方法和技术可以补充软件工程的方法和技术,使之更加丰富多彩。

软件可靠性工程是围绕软件质量指标体系中最重要的质量指标之一的软件可靠性来展开研究的,软件可靠性指标的高低,决定了软件是否能稳定、可靠地工作。软件中的错误是在软件的开发过程中,因为人的错误而引入到软件中的。开发软件的人也是社会的人,他们在开发软件的过程中,必然要受到其教育背景、工作经历、开发软件的经验的影响,同时也会受到其生活环境、周围的人们的影响,而且还会受到当时社会上各种思潮的影响。所以,尽快开展软件工程中的人的因素的分析研究工作,对于改善软件的质量有着十分重要的作用。软件可靠性工程中大量的管理工作,实际上一个重要方面就是如何做好人的工作。人的因素是根本的问题,因为从积极的方面来看,世界上所有推动人类社会进步与发展的事情都是由人做出来的。

综合以上原因，有理由认为本书讨论的软件可靠性工程问题已经远远超出了当年Musa先生提出的软件可靠性工程的概念。所以，从这个意义上来说，本书是一本讨论广义软件可靠性工程的著作。

本书是在软件可靠性工程研究课题组全体成员的长期努力下共同创造的成果。笔者的一届又一届的研究生们为此做出了持续不断的努力。如今，他们有的早已走上了工作岗位，有的还在国外继续深造，书中的许多章节就是他们的研究成果。笔者的主要工作就是将他们的这些成果用软件可靠性工程的一根主线贯穿起来，奉献给广大的读者。

在笔者普及、推广软件可靠性工程的工作过程及从事软件可靠性工程的教学过程中，经常会遇到各种各样的问题，其中最重要的问题也就是软件可靠性工程中的核心问题，即“软件可靠性工程与软件工程是一种什么样的关系?”、“应该怎样设计一个软件系统的软件可靠性指标?”、“怎样将软件系统的指标分配到各个软部件、甚至分配到各个模块上去?”、“什么样的技术可以保证软件可靠性指标的实现?”、“怎样反映和控制软件测试的进度与成本?”、“怎样控制将开发的软件产品在正确而适当的时候投放市场?”等。

上述这些问题有着深刻的工程背景，同时又涉及多方面的理论问题，要想清楚地回答这些问题是很不容易的。本书试图部分地回答若干问题，但真正的目的是以这些问题为出发点，将感兴趣的读者引入对这些问题的探讨中来，从而在国内普及软件可靠性工程的基本知识和应用，推动软件可靠性工程在我国的发展，为我国软件产业的发展壮大和国家的现代化做一点力所能及的工作。

本书第1章讨论软件可靠性工程在软件工程中的地位和作用，第2章讨论软件可靠性的基础理论，第3章讨论软件可靠性的分配问题，第4章研究软件测试的基本问题。自第5章开始，分别讨论软件的各种测试方法，最后给出有关部分的参考文献。因为本书是作为教材出版，受篇幅所限，与本书初稿相比较，将参考文献删掉了大半，主要是将那些较早的文献删除。如果书中引用的内容在书后的参考文献中未能找到，则笔者向这些作者致歉！

本书第2、3章介绍的软件可靠性基础理论、可靠性指标分配的优化问题以及第8～12章中有研究性质的内容，笔者建议作为研究生教学的内容和本科教学中的选学内容。

感谢国家自然科学基金委员会的系列资助，使我们这个研究小组得以在软件可靠性工程领域进行长期而稳定的研究工作。

参加本书编写工作的有谢旻、郑人杰等，感谢他们的辛勤劳动。感谢我的研究生周瑞、杨晓清、肖英柏、向剑文、张良平、陈波、张大帅、王果、陈斌、马若锋、郑红军、高俊鹏、刘丽娜、刘彦伸、徐剑宏、韩轶凡、黄灿、戴璐、周乙、鲜军、杨宏、任杰、张学斌、伍雁鹏、汪超、黄巍、齐大鹏、刘文、张霆、龚蓉、魏毅、伍永豪、朱州、晁冰、朱小冬。没有他们卓有成效的研究工作，本书不可能顺利完成。同样，我要感谢软件工程国家重点实验室的其他教授和他们的研究生，本书也包括了他们的许多研究成果。特别要感谢康立三教授和李元香教授领导的演化计算研究小组及他们的研究生，他们对演化计算和面向对象软件测试的研究成果是本书中重要的一部分内容，而演化计算的方法在本书讨论的许多方法中有着大量的应用。

借此机会，我要向我的妻子王旭莹表示衷心的感谢。正是由于她的鼓励和鞭策，本书才得以完成。

我还要感谢所有对我们的研究工作给予过大力指导和帮助的专家学者、组织机构、合作者和朋友们。感谢清华大学出版社的编辑、工作人员对本书的出版所做的大量工作和努力。

受编者水平的限制，书中会有不少的问题和错误。这些问题和错误一律由编者负责。请广大读者提出宝贵的批评和意见。如果在本书的使用中遇到任何问题，请与责任编辑联系：fuhy@tup.tsinghua.edu.cn。

徐仁佐

于武汉　香格里·嘉园

2007年2月

第1章

软件可靠性工程与软件工程

随着计算机的应用日益广泛，人们对软件质量的要求也越来越高，作为软件质量最重要的一项内容——软件可靠性，自然也受到人们的重视。在这几十年的研究过程中，人们也日益认识到，要保证软件的质量，特别是软件的高可靠性，必须要求软件行业全体人员的积极自主的工作和广泛的协作。于是，在1990年前后，终于形成了软件可靠性工程(SRE)的概念。现在，已发展成每年举行一次国际软件可靠性工程学术会议，以交流本领域的发展与成就。

在软件工程中应用软件可靠性工程方法可以做到：

- 分析、管理和提高软件产品的可靠性；
- 在用户对竞争价格、时效及可靠产品的要求之间求得平衡；
- 确定软件质量何时达到发行要求，将发行软件存在严重问题的风险降至最低；
- 避免因过度测试而来不及开拓市场的悲剧。

上述几项具体的收益，也同样是对最近兴起的软件工程经济学的贡献。

概括起来讲，软件可靠性工程主要包括三个方面的工作：

生命周期阶段	开发步骤	SRE的活动
可行性和需求分析	可行性分析 需求分析　开发设计	确定功能概图 对失效进行定义和分类 确定用户的可靠性要求 进行平衡关系研究 建立可靠性指标
设计和实现	设计 实现	在组间分配可靠性指标 按可靠性指标进行设计 根据功能概图集中配置资源 控制错误的引入和传播 度量现成软件的可靠性
系统测试和现场试运行	系统测试 现场试运行	确定操作概图 进行可靠性增强测试 跟踪测试进展 规定必要的额外测试 证明可靠性指标得到满足
交付和维护	运行　维护	规划发行后的人员需求 监视现场可靠性是否达到指标 跟踪调查用户的满意程度 通过监视可靠性安排新功能的引入时间 用可靠性度量指导产品及工程过程的改进

图1.1　软件产品生命周期中的可靠性工程活动

① 对软件可靠性的设计、分配和定量量测；

② 管理软件可靠性工作；

③ 软件可靠性的保证技术的选择与使用。

图 1.1 说明了围绕软件产品生命周期所进行的可靠性工程活动。该过程的各个阶段不必遵循严格的顺序，可靠性工程的各种活动也是如此，其中有很多交叉和重复。应该将每个可靠性工程活动只放在要求工作量最多的阶段完成。

1.1 软件的问题

软件是人的大脑逻辑活动的产物，因为人的教育程度、工作经验、思维习惯、认识能力、工作、敬业精神等多方面的差异，致使软件出错在所难免。

软件出错的事例报道有很多。20 世纪，美国范登堡空军基地导弹试射的失败的原因就是计算导弹飞行轨道的公式中一常数丢失小数点；欧洲阿里亚娜火箭的失事等的教训，使得人类认识到软件如同硬件一样，有着可靠性的问题，而且，其中的问题，比硬件可靠性还要严重得多。

目前，我国的软件形势严峻：透明度差，用户对软件不信任，软件危机日益严重。目前，我国的许多软件企业还没有认识到软件的质量对企业发展的重要性，还没有认识到软件的可靠性对国民经济发展的重要性。

解决之道在于软件生产的工业化——软件工程水平的普及与提高，软件可靠性工程的实施与进步。

软件工程的目的就在于提高软件生产率，改进软件质量。

1.2 与软件质量有关的基本概念

1. 概念

(1) 特性(characteristic)

帮助识别和区分各类实体的一种属性，包括物理、化学、外观功能或其他可识别的性能。

(2) 质量(quality)

反映实体满足规定和潜在需要能力的特性之总和。

(3) 需求(requirement)

需求应明确规定以下内容：

① 性能。

② 实用性(usability)。

③ 广义可靠性(ISO 文件称为可信性，即 dependability)，用于表示可用性、可靠性、安全性、机密性、完整性、维修性等的综合概念。

2. 质量

下面的基本概念从各个不同的方面用以表示软件的质量。

(1) 可靠性(reliability)

产品在规定的条件下和规定的时间内,完成规定功能的能力,其概率度量称为可靠度。

(2) 软件可靠性(software reliability)

在规定环境下及规定时间内,软件不引起系统失效的概率。它与软件中的缺陷、系统输入和系统使用有关。

(3) 维修性(maintainability)

产品在规定的条件下和规定的时间内,按规定的程序和方法进行维修,保持或恢复到规定的状态的能力。

(4) 可用性(availability)

产品在任一随机时刻需要和开始执行任务时,处于可工作或可使用状态的程度。

(5) 安全性(safety)

将伤害或损坏的风险限制在可接受水平内的能力。

(6) 机密性(confidentiality)

避免未经许可泄露信息的能力。

(7) 完整性(integrity)

避免不适当的变更的能力。

(8) 经济性(economics)

通常用寿命周期费用(life cycle cost,LCC)表示软件开发过程中的成本开销。它可以定义为:在产品的寿命周期内,在产品设计、研究和研制、投资、使用、维修及产品保障中发生的或可能发生的一切直接的、间接的、派生的或非派生的与其他有关费用的总和。

国外的部分统计数字可以说明软件开发费用的变化情况,从中也可以略窥其中的端倪。

1955 年:计算机成本中,硬件占 80%,软件开发占 10%,软件维护只占 10%。

1985 年:计算机成本中,硬件占 10%,软件开发占 30%,软件维护占到 60%。

1991—1992 年度 DOD 国防关键技术报告:在其年度国防开支中,软件占 10%(即 3000 亿美元的 10%=300 亿美元),其中,软件开发占 90 亿美元,软件维护占 210 亿美元。

美国软件水平、软件工程水平比我国高出许多,尚且有如此沉重的软件维护费用,我国更应重视软件可靠性。

1.3 软件质量的 6 个特性

1. 功能性(functionality)

指软件应正确实现的功能,包括以下几点。

- 适用性(suitability) 完成规定任务及规定任务的一组功能的协调性。
- 准确性(accurateness) 软件处理的结果应达到的精确度。
- 互操作性(inter-operability) 软件与规定系统交互作用的能力。
- 一致性(compliance consistency) 软件不应存在前后矛盾的地方。
- 保密性(security) 在未授权情况下,软件不会被获取、修改等。

2. **可靠性**(reliability)

- 成熟性(maturity)　对于应用而言,系统应该是足够成熟的。
- 容错(fault-tolerance)　系统应该具有容错的能力。
- 可恢复性(recoverability)　系统应具备自动恢复的能力。

3. **可用性**(usability)

- 可理解性(understandability)　系统应是容易理解的。
- 易学性(learnability)　系统的使用应是很容易学习和掌握的。
- 易操作性(operability)　系统的操作应该是简单明了的。

4. **效率**(efficiency)

- 时间(time behavior)　时间复杂性应该小。
- 资源(resource behavior)　系统对资源的要求应越小越好。

5. **可维护性**(maintainability)

- 可分析性(analyzability)　在维护的过程中,对于系统的分析应该简单易行。
- 易修改性(changeability)　对于已查明的系统中的错误,应该易于修改。
- 稳定性(stability)　对于系统中的错误进行修改以后,系统的表现应该是稳定的。
- 测试性(testability)　在进行维护的过程中,对于系统的测试应该易于进行。

6. **可移植性**(portability)

- 适应性(adaptability)　在新的运行环境中,被移植的系统应表现出良好的适应性。
- 可安装性(installability)　在新的运行环境中,被移植的系统应具有良好的安装性。
- 规范性(conformance)　在新的运行环境中,被移植的系统应遵循原有的规范。
- 可换性(replaceability)　在新的运行环境中,被移植的系统应有良好的替换性。

1.4　软件可靠性工程的研究范围

软件可靠性工程的研究范围主要是:

- 软件可靠性定量评测
- 软件可靠性的设计与管理
- 软件可靠性保证技术

1.5　软件可靠性的基本概念

1. **软件的缺陷、故障(失效)**

根据国际标准 IEEE Std 610-12-1990(1990.9.28),缺陷(fault)可定义为计算机程序中

的一个不正确的步骤、过程或数据定义，英文同义词有 error、bug；而故障(failure)可定义为一个系统或部件不能完成特定的性能需求所要求的功能。

在软件寿命周期的各阶段，人的行动会使软件在一定条件下不能或将不能完成规定的功能，例如：

(1) 使用方提出的用户需求不完整、有遗漏。

(2) 开发者误解了使用方的用户需求。

(3) 开发者没有完全实现使用方需求或潜在的需求。

(4) 概要设计没有完全实现需求规范，详细设计没有完全实现概要设计规范，编码没有完全实现详细设计规范。

(5) 用了不完整或不正确的算法或判别逻辑。

(6) 人为疏漏，如抄录错误等。

缺陷是因，是一个静态的概念，它确实存在于软件之中；故障是果，是一个动态的概念，它必须在软件的执行过程中才能表现出来。软件可能存在若干缺陷。缺陷不一定每次运行都能碰到，若碰到而使软件不能完成规定的功能，则称为失效。失效的发生与软件当时运行时的各种内、外状态的组合，软件的输入，软件的运行环境都有关。

容错(fault-tolerance)设计：软件在出现有限数目的故障的情况下，仍可提供连续正确执行的内在能力。此时，软件可以出现故障，但不失效。但对无容错设计的软件，发生故障即失效。

但在进行容错设计时，应注意人类的共同错误行为的影响。

2. 操作概图(operational profile)

1) 定义

“对系统使用条件的定义。系统的输入值都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”。①

若一个平台的输入是三个加速度表及三个陀螺的测量值，只考虑这六个量构成的六维空间作为输入，就很不完整。同时应考虑每个输入出现的概率。

软件接收一组输入(输入也可以是一个过程)，按照任务书的处理法则，给出一组输出(也可是一个过程)。软件的一组输入用输入空间 S_I 的一个点表示(高维)，一组输出用输出空间 S_O 的一个点(高维)表示，软件执行一次规定的任务，就是进行一次从 S_I 到 S_O 的映射，如图 1.2 所示。

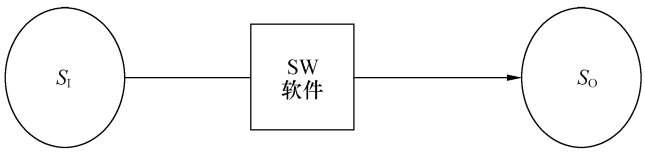

图 1.2 从 S_I 到 S_O 的映射

内外干扰也属于一种输入(计算机的瞬间断电、空间宇宙高能粒子干扰使计算机寄存器

① ESA(欧空局)标准 Std PSS-01-21(1991)，“ESA 空间系统软件产品保证要求”。

的0→1,1→0,如“风云二号”卫星所曾遭受的那样),包括实际可能发生的所谓“不希望事件”(undesired event)。

人的操作失误有:

(1) 不该动作时动作——高速飞行时的飞机,襟翼不能放下,如误按了“放下”开关;该动作时不动作——战机试飞中经常会出现的很多情况。

(2) 动作时机正确,但动作本身错误。

国内存在一种普遍现象:任务书不完整。解决办法:采用特别小组(ad hoc group)的组织形式,充分发挥集体的智慧,集思广益,尽量使用户的需求充分、完全。

2) 操作概图(operational profile)的形式化描述

在软件输入空间 S_I 上定义一个出现概率 $f(p)$(实际为概率密度函数),对 S_I 的一个子集 G 而言,出现在此子集 G 内的输入点的概率为

$$p=\int_G f(p)\mathrm{d}p$$

这个 $f(p)$客观存在,但多数情况下无法确切给出,至少要在软件投放使用之后的相当长一段时间后,经过统计,才能比较准确地估计它们的值。现在的做法是在工程中凭经验,凭借以前相类似的系统的经验数据加以推断。

记$\{S_I,f(p)\}$为软件的输入特性,称为软件的操作概图。

1.6 软件寿命的指数分布规律

软件寿命的重要特征是:

根据软件运行剖面做随机输入(或实际使用)时,如软件不做更改且是无容错设计的,则软件的寿命为指数分布。此时软件出故障则失效,故障率即失效率。

软件的一个缺陷 D 只影响 S_I 的一个子集合 G(它可以是非单连通的)。输入点 P 落入 G 时,软件的缺陷 D 起作用,使软件出故障。以下只考虑按运行剖面的随机输入。

设软件中有 N 个缺陷,第 i 个缺陷使软件出故障的概率为 p_i,设 N 个缺陷彼此独立,则有 $q_i=1-p_i$,软件的可靠性为

$$q=\prod_{i=1}^{N}q_i=q_1q_2\cdots q_N$$

又设软件的任务时间为 k(单位:分、时、天、周……),则单位时间的故障出现概率为

$$\lambda_i=\frac{p_i}{k}\qquad(i=1,2,\cdots,N)$$

考虑软件从 $t=0$ 起随机运行执行任务,到时刻 t 时还未出现故障的概率为 $R(t)$,即寿命 T 超过 t 的概率,这时有

$$P(T>t)=R(t)$$

令 T 的累计分布函数为 $F(t)=P(T\leqslant t)$,则

$$F(t)=1-R(t)$$

设输入是随机的,软件到时刻 t 还未出现故障的情况下,在$[t,t+\Delta t\,]$内出故障的概率,是一个条件概率,记为 $\lambda(t)$——瞬间故障率。

$$\lambda(t)=\frac{R(t)-R(t+\Delta t)}{R(t)}=\frac{R'(t)}{R(t)}\Delta t$$

取 Δt 为单位时间 1，则

$$\lambda(t)=\frac{R'(t)}{R(t)}$$

对第 i 个缺陷而言，$\lambda(t)=\lambda_i$，故

$$\frac{R'(t)}{R(t)}=-\lambda_i$$

用初始条件 $t=0$，$R(t)=1$ 解得

$$R(t)=\mathrm{e}^{-\lambda_i t}$$

于是整个软件的 $q(t)$ 为

$$q(t)=\prod_{i=1}^{N}q_i(t)=\prod_{i=1}^{N}\mathrm{e}^{-\lambda_i t}=\mathrm{e}^{-(\lambda_1+\lambda_2+\cdots+\lambda_N)t}$$

令 $\lambda=\sum_{i=1}^{N}\lambda_i$，则有 $R(t)=\mathrm{e}^{-\lambda t}$，即寿命 T 为指数分布。

如一个非容错设计软件，发生失效也不予纠正，仍继续使用，则与电子产品的失效规律一致。

此时可用平均故障间隔时间(mean time between failure，MTBF)表示，它的英文定义是：The expected or observed time between consecutive failures in a system or component.

软件的故障一旦出现，就应查找、排除，此时 λ 将不断下降，从而使 $R(t)$ 不断上升，此过程称为软件可靠性增长过程。

软件在随机输入下投入运行，到出现失效的时间即“失效前时间(time to failure)”，这是一个随机变量，其期望值为 MTTF(mean time to failure)，记为 θ：

$$\theta=\frac{1}{\lambda}$$

1.7 软件故障率的规律

为了找到软件故障率的规律，Adams E. N. 将 9 个主要的 IBM 软件产品按失效发生频率分类，将它们分为 8 档，每档的 MTTF 代表值(单位为年)如表 1.1 所示。

表 1.1 每一档的 MTTF 代表值

档次号 i	1	2	3	4	5	6	7	8
MTTF/年	1.58	5	15.8	50	158	500	1580	5000
$r_i=d_i/d$ (%)	0.4	1.0	2.5	5.2	10.6	18.7	28.2	33.4
$f_i=\lambda_i/\lambda$(%)	30.1	23.7	18.7	12.3	7.9	4.4	2.1	0.8

第 i 档的总缺陷数为 d_i，

$$d=\sum_{i=1}^{8}d_i,\qquad r_i=\frac{d_i}{d},\qquad f_i=\frac{\lambda_i}{\lambda}$$

第 i 档的故障率为 λ_i，软件的总故障率为

$$\lambda = \sum_{i=1}^{8} \lambda_i$$

从表 1.1 中可以得出如下的重要结论：

(1) 软件缺陷相应的 MTTF 相差可达 3～4 个数量级。MTTF 值大的缺陷数多，MTTF 值小的缺陷数少。

(2) 极少数故障率高的缺陷占有故障率的大部分。

从上述数据看，占总缺陷数不到 4%(3.9%)的 1、2、3 档缺陷，占总故障率的 72.5%；而占总缺陷数 80.3%的 6、7、8 档缺陷，只占了总故障率的 7.3%。

在这样的情况下，缺陷数与软件的总 MTTF 或总 λ 没有很直接的联系。

如果软件的开发者告诉用户："在我们开发的这个软件中，还有 10 个缺陷未发现。"用户会有什么样的反应呢？他们肯定马上要问："它们是属于第几档的缺陷？"

如果是属于第一档的缺陷，那就意味着它们可能在使用后的一两个月内就可能出现，用户对此多半不会接受；但如果是属于第七、八档的缺陷，也许用户就会满意地接受。

1.8 风险函数 $\lambda(t)$ 与 $R(t)$ 的关系

风险函数(hazard function)$\lambda(t)$：软件正确地运行到时刻 t 时，单位时间内软件发生故障的概率(实际上，$\lambda(t)$ 应该是概率密度，真正的概率应该是 $\lambda(t)\Delta t$)。

如以 T 表示从 0 开始使用软件，到软件发生故障为止所经过的时间。对于不同的运行，T 的值显然不同，所以，T 是一个连续型的随机变量。于是可以写出在区间$[t,t+\Delta t]$内软件发生故障的概率为

$$\lambda(t)\Delta t = P_r\{t < T \leqslant t+\Delta t \mid T > t\}$$

因此可以写成

$$\begin{aligned} R(t+\Delta t) &= P_r\{T > (t+\Delta t)\} \\ &= P_r\{(T>t) \wedge \text{在区间}[t,t+\Delta t]\text{内软件不发生故障}\} \\ &= R(t)[1-\lambda(t)\Delta t] \end{aligned}$$

将上式关于 t 求微分，得到

$$\mathrm{d}R = -\lambda(t)\mathrm{d}t$$

导出下面的微分方程

$$\frac{\mathrm{d}R}{R} = -\lambda(t)\mathrm{d}t$$

解之得

$$R(t) = \exp\left[-\int_0^t \lambda(x)\mathrm{d}x\right]$$

在实际的软件可靠性工程中，如果已知 $\lambda(t)$ 的函数形式(它一般是一个时间 t 的升函数)，则软件可靠性建模的任务就很简单了。但是，正是因为人们到目前为止，对它的函数形式仍然无法确定，所以，对于它有着形形色色的假设；也因此存在着许许多多的软件可靠性模型。

1.9 软件与软件可靠性工程

1.9.1 软件及其研制过程的特点

为了讨论软件开发的特点，下面先介绍软件、硬件的特点和它们之间的区别。

表 1.2 将软件和硬件进行了对比。

表 1.2 软件与硬件的对比

软　件	硬　件
是逻辑实体，始终不会自然变化，只是其载体可变	是物理实体，每件同规格产品的质量特性之间有散布，会随时间和使用而老化、磨损以致失效
研制主要是紧张的脑力劳动过程，本质上无形，看不见，难测控	不只是脑力劳动，过程有形，便于测控
不可靠问题基本是开发过程中人为差错造成的缺陷所引起的	不可靠问题不只是设计问题，在生产和使用过程中也会产生新的故障
程序是指令序列，即使每条指令本身都正确，但在执行时其逻辑组合状态变化万千，不一定完全正确	硬件失效总是由其零部件或其结合的故障所引起
系统的数学模型是离散型的，其输入在合理范围内的微小变化可能引起输出的巨大变化，故障的形成无物理原因，失效的发展取决于输入值和运行状态的组合，无前兆	系统在正常工作条件下其行为是渐变的，故障的形成和失效的发生一般都有物理原因，有前兆

1.9.2 软件可靠性工程

从软件特性和用户需求分析开始，软件可靠性工程(SRE)贯穿于软件生命周期的全过程，主要包括下面三个方面的内容：

(1) 为满足用户对软件可靠性的要求，必须给出关于软件可靠性的规格说明(specification)，在做软件可靠性设计时，甚至要在其他软件质量指标间做出某些折中(trade off)，如可适当增加开发成本，延长测试时间等以求得更高的软件可靠性。

(2) SR 的量测与分析技术，这是必要条件。

(3) 软件工程中一整套保证软件开发质量的方法和技术。

这里仍然沿用 Musa J. D. 的定义。

定义 软件可靠性工程是预计、测量和管理以软件为基础的系统的可靠性，最大限度地满足用户要求的应用科学。

随着人类生产活动的发展和科技水平的进步，许多系统包含了更多、更复杂的硬件和软件，它们成为了可靠性研究的主要对象。于是有人用下面的词来称呼这样的系统。

产品(X-件，X-ware)：即含有硬件、软件的综合系统。

表 1.3 中列出软件可靠性工程在软件的生命周期各阶段所涉及的人员及其任务。

表 1.3 软件可靠性工程在软件的生命周期各阶段所涉及的人员及其任务

人　员	生 命 周 期			
	需 求 分 析	设计/实现	测试/域试用考核	投放/维护
产品经理	√		√	√
项目经理	√	√	√	
开发经理		√	√	
可靠性工程师/分析师	√	√	√	√
系统工程师/分析师	√	√		
软件系统设计师		√		
软件系统设计员		√		
程序员		√	√	
质量保证工程师/分析师		√	√	√
测试经理	√	√	√	
测试设计师		√	√	
测试员		√	√	
装配与运行经理				√
顾客/用户	√			√

SRE 与软件生命周期各阶段的关系如图 1.3 所示。

在图 1.3 中，出现在图的左边的是软件生命周期各阶段的任务，右边的则是 SRE 的工作任务。

对照图 1.3，下面分别讨论 SRE 各阶段的任务、涉及的人员。

1. 可行性研究/需求分析阶段软件可靠性目标的确定

$$R_S = R_{HW} \times R_{SW}$$

其中，R_S 表示系统可靠性，R_{HW}表示硬件可靠性，R_{SW}表示软件可靠性。

确定之后，将 R_{SW}分配到各部件上，要求做到：

- 了解顾客的需求。
- 了解软件发生故障的后果的严重程度，是否采取容错设计。
- 了解顾客的支付能力，研究和确定市场营销策略。
- 正确把握最佳投放市场的时间。

1) 选定故障强度目标(failure intensity objective，FIO)

$$C_{LCR} = \text{Min}\ \{(C_{RGT} + C_{OF}) \mid \text{FIO}\}$$

C_{LCR}：生命周期可靠性成本；

C_{RGT}：可靠性增长测试成本；

C_{OF}：运行期间的故障损失成本。

它们之间的关系，用图 1.4 和图 1.5 中的两个曲线图表示。