RUANJIANKEKAOXING
FENXIJIZHILIANGBAOZHENGYANJIU

软件可靠性分析及质量保证研究

谢 芳 卢朝晖 胡海东 编 著

中国水利水电出版社
www.waterpub.com.cn

RUANJIAN KEKAOXING FENXIJIZHILIANGBAOZHENGYANJIU

软件可靠性分析及质量保证研究

谢 芳 卢朝晖 胡海东 编 著

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书从软件工程实际出发，以为可靠软件的开发提供指南为目的，对软件可靠性及软件质量保证两大方面的内容进行深入探究。本书内容包括软件可靠性度量与测试、软件质量标准、软件质量保证方法分析、软件质量保证技术、软件能力成熟度模型 CMM&CMMI、软件测试与维护、软件全面质量管理、软件配置管理等。全书结构合理，条理清晰，内容丰富，是一本值得学习研究的著作。

图书在版编目(CIP)数据

软件可靠性分析及质量保证研究/谢芳，卢朝晖，胡海东编著. --北京：中国水利水电出版社，2013.4

ISBN 978-7-5170-0835-4

Ⅰ. ①软… Ⅱ. ①谢… ②卢… ③胡… Ⅲ. ①软件可靠性－研究 Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字(2013)第 084998 号

策划编辑：杨庆川 责任编辑：杨元泓 封面设计：崔 蕾

书 名	**软件可靠性分析及质量保证研究**
作 者	谢芳 卢朝晖 胡海东 编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址：www. waterpub. com. cn E-mail：mchannel@263. net(万水) sales@waterpub. com. cn 电话：(010)68367658(发行部)、82562819(万水)
经 售	北京科水图书销售中心(零售) 电话：(010)88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京鑫海胜蓝数码科技有限公司
印 刷	北京市登峰印刷厂
规 格	185mm×260mm 16 开本 16 印张 389 千字
版 次	2013 年 4 月第 1 版 2013 年 4 月第 1 次印刷
定 价	64.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

前　言

随着社会信息化进程的加快，计算机软件已成为支撑社会正常运行和发展的重要基础设施。人们对软件要求的提高，也从某种程度上加大了软件的复杂程度，可靠性、易用性、应变性和兼容性等逐渐成为当前人们对软件产品质量的基本要求。

软件生产的社会化决定了软件产品质量的高低，它与软件公司实施的开发过程和具体的管理水平有关。一个软件系统通常由不同软件专业公司生产的软件构件产品集成，为了保证软件质量，还要求软件行业有相关的统一标准和协议，以及按照标准进行测试和认证。

软件可靠性是软件质量的一个非常重要的指标，它的诞生标志着软件质量管理跃上了一个新的台阶，软件可靠性的分析、评价、设计和验证以及管理水平迈向了系统化、规范化、全员化的进程。软件可靠性研究的是如何应用理论知识、科学方法和工程规范来指导可靠软件开发，以达到投入尽量少的时间获得尽可能高的可靠性软件。其内容丰富而广泛，覆盖了软件可靠性预计、分配、分析、设计、评价、测试、检定，以及可维护性设计、安全性设计和可靠性工程管理等方面。

软件可靠性及质量保证既是软件工程研究与实践的必然结果，也是可靠性工程发展的必然选择，它们离不开知识与技能、方法与技术的灵活运用。基于此，本书从软件工程实际出发，以为可靠软件的开发提供指南为目的，对当前软件开发与使用现状、存在问题及其发展趋势等进行了系统分析，详细介绍了有关软件质量管理、控制、保障及度量的程序、方法和技术。考虑到软件可靠性是软件质量中最重要的因素，本书对软件可靠性模型及其应用、软件可靠性测试、软件可靠性数据收集和处理进行了深入的讨论。

本书共 10 章。第 1 章给出了软件可靠性的基本概念，分析了其研究内容及影响因素；第 2 章给出了软件可靠性模型、测试及评估方法，并就提高软件可靠性的方法和技术进行了分析；第 3 章给出了软件质量保证的基本概念；第 4 章给出了软件质量标准；第 5 章分析了软件生命过程、软件质量的度量；第 6 章按照软件开发的过程从文档编制到需求变更控制对软件质量保证技术进行了分析；第 7 章分析了软件能力成熟度模型 CMM & CMMI，同时给出了我国的软件评估体系 SPCA；第 8 章给出了软件测试的策略、过程质量度量、评审和质量保证，以及软件维护活动的实施；第 9 章给出了软件全面质量管理，一一论述了全面质量管理的概念、质量管理的战略及文化、6σ 项目管理、质量功能展开设计、dfss 流程与主要设计工具；第 10 章给出了软件配置管理的相关信息。全书内容具有理论联系实际、系统、简洁的特点；在软件开发、使用和维护方面具有较强的工程适用性。

全书由谢芳、卢朝晖、胡海东撰写，具体分工如下：

第 1 章～第 3 章、第 9 章：谢芳（红河学院工学院）；

第 4 章、第 6 章第 2 节～第 8 节、第 7 章：卢朝晖（海南师范大学）；

第5章、第6章第1节、第8章、第10章：胡海东(内蒙古科技大学)。

作为一门新兴的学科和学术探讨，本书不是软件可靠性及质量保证工程研究和实践的终点，而是一个新的起点。必须看到的是，软件工程作为一项必须多人协同才能完成的工作，以有限的时间和个人认知探索软件可靠性及软件质量的无限奥秘是极其困难的，还需要业界和同仁的无私支持。由于作者水平有限，加之软件产业的发展非常迅速，书中难免存在疏漏和不妥之处，恳请有关专家提出宝贵意见。

作者

2013年3月

目　录

第1章　软件可靠性概述

1.1　软件可靠性发展史

“软件可靠性是软件质量特性中重要的固有特性和关键因素。软件可靠性反映了用户的质量观点”——SEI CMU。

软件可靠性工程的理论、技术和方法或来源于硬件可靠性，或借鉴于硬件可靠性，或受硬件可靠性研究与实践成果的启发得以产生和发展；自然，它们之间必然存在着相同或相似之处。软件的固有特性及其与硬件之间的本质差别又决定了它与硬件可靠性之间的差别。

软件可靠性与硬件可靠性之间相互支持、相互补充。二者之间的相似之处为：

①依靠开发设计过程保证产品的固有可靠性。

②简单就是可靠，结构越简单其可靠性就越容易得到保证。

③标准化、系列化、组合化是可靠性保证的重要途径。

④具有避错设计、查错设计、纠错设计、容错设计等可靠性设计思想、理论和技术支持。

⑤可靠性水平的高低受人为因素和开发人员素质的影响。

⑥可靠性设计和管理受可靠性历史数据的影响和制约。

⑦开发设计工具的先进性、有效性是可靠性水平高低的决定因素之一。

⑧利用概率论和统计学原理进行可靠性建模、度量、分析和过程控制。

⑨可靠性的预计和分配是可靠性设计的基础。

⑩可采用故障树分析、失效模式及影响分析等方法进行可靠性分析。

⑪有效的可靠性工程管理是保证。

二者的不同之处可见表1-1所示。

表1-1　软件与硬件可靠性的主要差别

序号	软件	硬件
1	逻辑实体，不会损耗，不会自然变化，只是其载体可变	物理实体，每件同规格产品的质量特性之间有散差，随时间和使用环境等的变化而老化、磨损以至失效
2	开发过程主要是脑力劳动过程，本质上无形，不可见，不透明，难以测控	研制过程不只是脑力劳动过程，还有体力劳动过程，过程有形，可跟踪，可测控

续表

序号	软件	硬件
3	不可靠问题主要是由于开发过程中的人为差错造成的缺陷和错误所引起的	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
4	软件是程序指令集合，即使每条指令都正确，但由于在执行时其逻辑组合状态千变万化，最终软件不一定正确	元器件、零部件及其组合故障均可能导致系统失效
5	系统的数学模型是离散的，其输入在合理范围内的微小变化都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，有先兆
6	应在开发的全过程采取措施防错、检错、纠错和容错，而在批量复制过程中，软件本身不会变化	除了开发过程外，生产过程对产品的影响也很大，均需加强控制
7	精心设计测试用例，执行严格的测试，查出错误并加以排除	建立适当的环境应力条件，进行环境应力筛选，剔除缺陷
8	采用冗余设计时，应确保冗余软件间的相异性。否则，相同的冗余软件不仅不能提高可靠性，反而增加了复杂性，降低其本有的可靠性	相同的部件之间是自然独立的，适当的冗余可以提高其任务可靠性
9	在使用过程中出现故障后必须修改原软件以解决问题，若在修改时未引入新的缺陷或错误，那么其可靠性就会增长	在使用过程中出现故障后无需修改原产品，只需更换或修复失效的零部件，使产品状态恢复，其可靠性一般不会提高
10	某处的修改会影响它处，错误会扩散和蔓延，维护时必须考虑这种影响	维修某处一般不会影响它处
11	失效率随故障的排除而下降	失效率的变化呈浴盆曲线
12	可靠性参数估计无物理基础	可靠性参数估计有物理基础

伴随软件业的快速发展，规模与复杂度逐步增加，人们也经常遇到软件可靠性问题。自20世纪70年代中后期以来，软件工程高速发展，加上传统可靠性工程技术和方法，软件可靠性工程得以产生并取得了长足的进展，各种软件可靠性模型相继推出并得到不断改进和优化，模型验证和使用一度成为软件可靠性工程的热点，直到今日也依旧是研究重点。

软件可靠性设计与测试技术得以开发并逐步应用于工程实践；软件可靠性分析、评估方法不断完善，并在一些特殊的或重点工程项目中得到应用；软件可靠性工程管理技术的开发备受推崇，相应的管理方法被实践所验证，软件业界已充分认识到，绝大多数软件问题是由管理不善所引起的，所以，以过程改进、组织性能改进、管理模式改进、软件开发人员管理为重点的管理体系和管理机制得以产生并日臻成熟；软件可靠性标准化工作也取得了很多成果。

直到目前为止，软件可靠性工程已经得到了广泛研究、实践并取得了显著的成绩，但开发足够可靠的软件并测试和验证其可靠性，仍是非常困难的问题。复杂软件不管是对大工程系统还是小工程项目都越来越显示出它是一个薄弱环节，即便是通过完备测试与合格验证的软件也常常受到错误的困扰。与此同时，一个前所未有日益增长的需求是：软件应具有检定合格的可靠性，例如，核安全控制系统等无不对软件可靠性提出了前所未有的高要求。即使是在工业和日常生活中一般应用程序的开发与销售，市场对其可靠性要求也越来越高。

21 世纪是个注重质量的世纪，人们要求高质量的生活，企业在质量的大潮中接受考验。软件的质量正在向着不同的深度和广度发生着深刻的变革。作为软件最重要的质量特征，软件可靠性已经成为开放式技术社会中企业的最后一道防火墙与最重要的市场竞争武器。

1996 年 6 月 4 日，耗时 10 年研制的 ARIANE 5 火箭，首次发射升空飞行 40s 后，由于其攻角大于 20°，引起了极高的气动载荷，导致火箭的助推级与芯级分离，不得不启动自毁装置引爆火箭，造成巨大的损失，直接经济损失达到 5 亿美元，还使耗资 80 亿美元的开发计划推迟了近三年。究其原因有两方面：一方面，ARIANE 5 火箭的主惯性参考系统在将 64 位浮点数转换成 16 位有符号整数时，数字转换超界，没有将正确的姿态数据传送给运载火箭的箭载计算机所致；另一方面，ARIANE 5 重用了同样存在这一错误的 ARIAN 4 火箭的飞控软件，但在重用此软件后未进行完备测试，致使错误残存下来。当今社会，软件可靠性工程的研究和实践已不再只局限于航空航天、军事等特殊领域，而是广泛地应用于包括一般应用软件在内的几乎所有领域。货架上的商用软件必须赢得用户的信任，才能为开发商带来利润。无处不在的网络计算机已将软件可靠性关注的焦点推向前台。bug 成堆的操作系统、Web 浏览器以及客户端桌面应用系统，都可能导致严重的漏洞，与此同时，伴随着消费者需求的不断变化和提高，软件可靠性正在成为以网络计算机为手段的政府部门、企业以及个人关注的焦点。

这一现实强力地推进着软件可靠性工程理论研究与实践的进展。软件可靠性工程的研究和实践具有如下重要意义：

- 转变观念，有效扭转只重视硬件，忽视软件可靠性的现状。
- 推进软、硬件可靠性工程的均衡发展，提高系统可靠性水平。
- 推进软件的可靠性设计、测试及其可靠性工程管理能力的改进和提高，确保并不断改进软件可靠性，提高企业诚信，增加顾客满意度，超越顾客期望。
- 指导软件可靠性分析、评估、验证与鉴定，为软件验收（验证）提供依据，增强用户信心。
- 推进软件工程的不断丰富和发展，促进软件工程管理水平及软件过程能力的改进和提高。

1.1.1　软件可靠性模型的发展

1. 软件可靠性模型概述

随着软件工程理论、技术、方法的成熟和应用，面向过程的软件开发方法和可靠性保证得到了前所未有的重视。只要严格遵循软件工程原理，建立和维护标准软件过程，并自觉运用过程方法，通过对标准软件过程的剪裁生成项目软件过程，并以此来指导和约束软件开发，即可

达到失效率在 10^{-3} 数量级这一可靠性要求。如果当软件的可靠性有更高要求时，在严格实施软件工程的同时，必须采取专门的可靠性工程技术和方法，如避错设计、容错设计、纠错设计、查错设计等。

软件可靠性工程过程模型根据软件开发过程将可靠性工程技术与方法联系起来，反映现有理论和实际应用中的差别，在对象特性和过程分析的基础上，为用户提供合适的裁减过程模型及其接口，规范可靠软件的开发，改进软件过程。根据 ISO 9000 标准和软件能力成熟度模型(Capability Maturity for Software，CMM)的过程管理思想，遵循过程原理，使用过程方法，以软件可靠性需求和指标体系为目标，在软件可靠性工程框架下，按照软件生命周期过程活动及其要求，建立如图 1-1 所示的面向过程的软件可靠性过程模型。

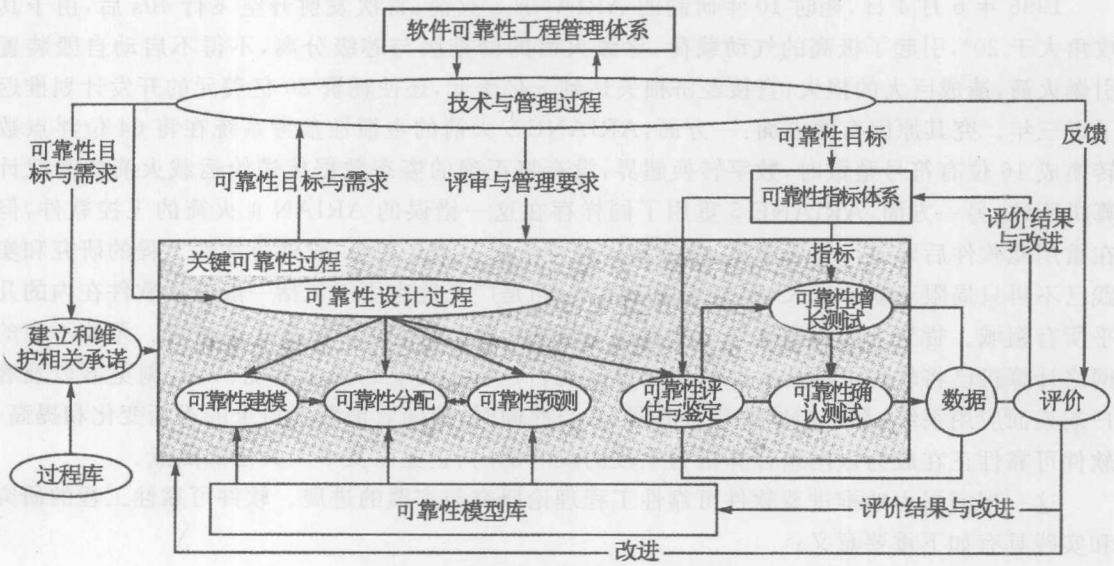

图 1-1　软件可靠性过程模型

软件可靠性工程过程模型是对软件可靠性工程框架的细化和工程实践的程序化，它不仅描述了软件可靠性过程及其活动的关系，而且能够为软件过程优化和能力改进提供指南。软件可靠性工程过程模型的目标是在软件可靠性的建模、度量、分析、预计、分配、预测、设计、测试、验证及其软件可靠性工程管理的基础上，以过程方法为手段，在模型库和可靠性指标体系、度量体系的支持下，实施对软件可靠性过程活动的有效控制与管理，实现软件可靠性的控制和过程能力的改进。

20 世纪 80 年代之前，软件可靠性工作主要侧重于模型的研究和建立。到目前为止，推出了 100 多种软件可靠性模型，且新模型还在不断推出，从而导致了所谓的“模型战”。最早的软件可靠性模型是由 H. K. Weiss 于 1956 年提出的一系列公式，但由于它们太复杂，这些公式对后来软件可靠性模型的建立几乎没有产生什么影响。

对软件可靠性模型发展起着重要或奠基作用的模型分别是 M. L. Shoomafi 于 1971 年发表的 Shooman 模型以及 Z. Jelinski 和 P. B. Moranda 发表的 J-M 模型。这些有着惊人的相似之处。Shooman 在他稍晚的一篇文章中指出，存在着一个简单的参数转换集可以将 J-M 模型

转换成 Shooman 模型。他们都假设：

- 软件中的初始错误数为 N(N≥0)。
- 软件故障与软件中的剩余错误个数成正比。
- 错误一旦被发现，立即排除且不引入新的错误。

另外，J-M 模型还假设故障的风险率为分段常数，在首次纠错过程中，它由一个常量来改变，但在两次纠错过程之间保持常数。Jelinski 和 Moranda 应用极大似然估计来估计软件中总的错误数量、剩余错误数量与风险率之间的比例常数。

1972 年，B. Littlewood 和 J. L. Verrall 发表了第一个 Bayes 模型，假设：故障间隔时间服从含参数 λ_i 的指数分布，且 λ_i 服从先验的 Γ 分布。据此，由标准的 Bayes 过程获得 $t_{n+1} | \{t_1, t_2, \cdots, t_n\}$。同年，G. J. Schick 与 R. W. Wolverton 提出了类似的模型，它与其它型的主要区别就在于：他们假定连续的故障之间的纠错时间服从 Rayleigh 分布。

1973 年，W. L. Wagoner 发表了类似模型，但假设了风险函数服从 Weibull 分布。其特例就是 Schick-Wolverton 模型。

1975 年，J. D. Musa 发表了执行时间模型，引入了一系列新的显式函数：

- 测试压缩因子，使用测试条件和测试用例，比使用用户的输入数据有着更高诱发故障的概率。
- 初始 MTTF。
- 执行时间。

同年，P. B. Moranda 发表了几何 Possion 过程模型，该模型假设故障率随着时间的增加呈几何级数下降，且下降过程出现在每次故障的纠正期间。因此，在第 i 个时间段内的错误数满足含参数 λK^{i-1} 的 Possion 分布。

也是这年，K. Trividi 和 M. L. Shooman 发表了第一个 Markov 过程模型 i 他们将系统状态区分为“UP”和“DOWN”两个状态，并假定“UP”状态和“DOWN”状态之间的转换概率相同。模型输出结果是一个概率集合，集合的每个元素为：P{在[0,f]内找出 k 个错误}。

1983 年，由 Yamada、Ohba 和 Osaki 联合发表的 NHPP 模型，给出了呈 S 形的可靠性增长曲线的均值函数。由 K. Okumoto 和 J. D. Musa 提出的对数 Possion 过程模型，具有一个初始错误率 λ，以及对应于 λ 的一个递减率。模型的日历时间部分则与执行时间模型的日历时间部分相同。

T. Bendell 将试探性数据分析方法(Expedition Data Analysis，EDA)引入软件可靠性建模，使用时间序列分析方法和成比例的风险分析函数建立软件可靠性模型。

1989 年，Y. Tohma、K. Tokunaga、S. Nagase 和 Y. Murata 提出了超几何分布模型，用于估计软件中的剩余错误数量。

Tsu-Feng Ho、Wah-Chun Chan 和 Chyan-Goei Chung 提出了基于软件模块结构的模型，通过每个模块的可靠性来估计软件系统的可靠性。另外，他们还结合信息论方法，建立了软件可靠性模型。之后，随着软件构件技术的发展，基于构件的软件可靠性模型得到了深入研究和迅速发展。

G. Pucci 应用恢复块结构技术，对软件可靠性等软件质量进行估计，根据错误对软件系统所产生的可观察到的后果进行分类，使得将测试数据应用于模型参数的估计成为可能。T.

Downs 针对现有大多数软件可靠性模型在测试阶段将软件处理成黑盒子的做法，考虑对测试过程直接建模。

K. w. Miller 等人开发出以软件的黑盒子模型为基础的理论分析方法，以解决：

· 随机测试过程中，当观察到的故障次数为 0 时，如何估计当前版本软件的故障率。

· 使用分布与测试分布不匹配时，对估计的故障概率进行调整。

· 故障率估计时，将随机测试的结果与其它信息结合起来进行分析。

N. Karunanith，Y. Malaiya 和 D. Whitley 应用神经网络系统理论预测软件可靠性。他们利用前向神经网络对三个数据集合进行软件可靠性估测，结果发现：

· 神经网络方法在软件可靠性估测中显示出良好的一致性，这是一般现有软件可靠性模型所无法做到的，神经网络方法对于提高估测精度具有显著的效果。

· 由神经网络方法估计出的软件中的错误数量，始终较使用现有其他软件可靠性模型所估计的结果要少。

总而言之，软件可靠性模型是用来评估软件可靠性、预测产品中可能存在的缺陷数的一套方法。依据软件失效间隔时间、失效修复时间、失效数量、失效级别等数据，选择并建立适当的可靠性模型，从而得到系统的失效率及可靠性变化趋势，指导软件可靠性评估和预测。

目前，软件可靠性模型依然是软件可靠性工程研究的热点，基于构件和体系结构的软件可靠性建模正在掀起软件可靠性模型研究与应用的新一轮高潮。与此同时，模型的比较和选择正在得到较深入的研究，一系列客观标准得以产生，为模型的选择提供了支撑；模型中所使用的统计方法更趋适用，许多经典模型的一致性被提取出来并且得到了有效验证，强有力地推进了软件可靠性工程的进展。

2. 可靠性模型分类

按照模型所需要搜集数据的来源差异，可靠性模型可分为动态模型和静态模型。动态模型主要统计数据的来源是缺陷数统计分布，如依据软件生命周期中被发现的缺陷数变化趋势可作为预测可能潜伏在软件中的缺陷数参考依据。静态模型的统计数据的来源是项目其他属性或程序与模块的分析数据，如依据模块的复杂性、项目的规模等。

(1)确定预测参数。确定模型要预测的参数，也就是要明确建立模型的目的。

(2)数据搜集与分析。依据预测目标收集并研究数据的属性、单位、数据的准确性与相关性，可绘制成点分布图来观察其变化趋势。

(3)模型选择。依据上两步的工作结果，选择适合模型。在实际运用当中，可能需要选择一个或几个模型以适用于不同的情形下。模型的选择主要是看第一步研究的数据与所选模型的拟合程度，模型一旦选择，需要替换上期望预测值，得到相应的模型。这个阶段也叫模型的假设。

(4)模型测试与评价。实际运用当中，所选的模型不一定能很好的与实际情况相吻合，可能的情况有：搜集的数据不够准确，数据研究时参数的分析单位不合理(如时间，间隔太小可能观察到的变化趋势性很不规则，如果间隔调大一些，变化趋势可能会更合理一些，当然这要依据实际情况)。

(5)模型的确定。如果所选择的模型测试通过，就可以依据模型进行可靠性预测了。

例如，如表 1-2 示为某一项目的软件缺陷几种统计表所对应产生的变化趋势图，用来简要说明模型建立时选择不同的单元划分所产生的不同效果。

表 1-2　每天产品缺陷数

日期(单位:天)	缺陷数	日期(单位:天)	缺陷数
3 月 1 日	3	3 月 8 日	18
3 月 2 日	18	3 月 9 日	9
3 月 3 日	15	3 月 10 日	12
3 月 4 日	8	3 月 11 日	9
3 月 5 日	10	3 月 12 日	8
3 月 6 日	6	3 月 13 日	4
3 月 7 日	15	3 月 14 日	0

图 1-2 所示，为对应的趋势图。从图中可知以 1 天为单位时，软件缺陷的变化似乎规律性不是很强，存在好几个波折。

如果我们基于同样的数据，但以两天为一个时间单位来考察其变化趋势，如表 1-3 示，对应趋势图 1-3 所示。以两天为单位时曲线的变化趋势相对平滑一些。一般新项目的开始，被发现的缺陷数会急剧上升，上升到一定程度后，随着缺陷不断地被修复，产品的缺陷便逐步的减少。所以建模时适当的选择单位更有助于快速有效的建立适合产品的可靠性模型。

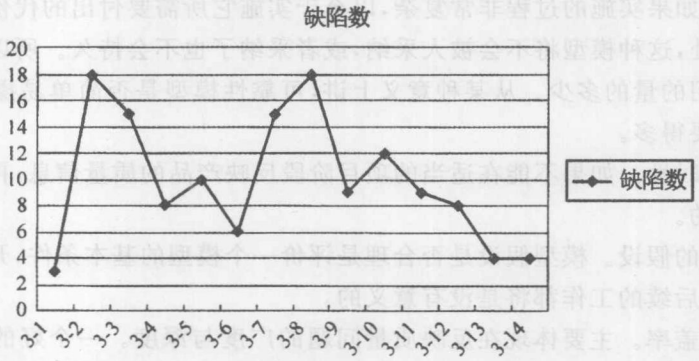

图 1-2　缺陷数变化趋势图

表 1-3　天产品缺陷数

日期(单位:2 天)	缺陷数	日期(单位:2 天)	缺陷数
3/1～3/2	21	3/9～3/10	21
3/3～3/4	23	3/11～3/12	17
3/5～3/6	16	3/13～3/14	8
3/7～3/8	33		

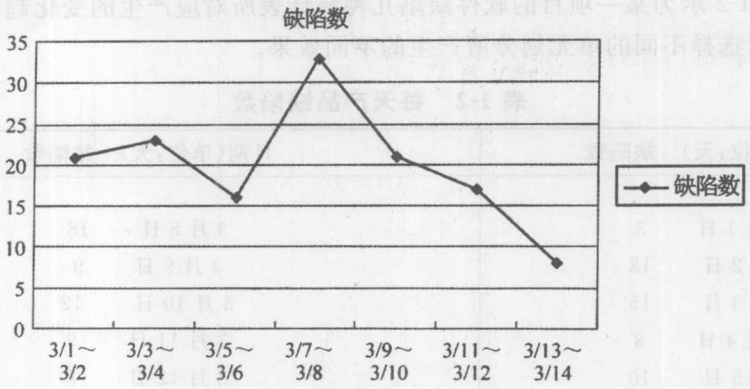

图 1-3　缺陷数变化趋势图

3. 可靠性模型评价标准

软件可靠性模型通常假设失效之间是相互独立的。失效的产生需要两个条件：错误引入和错误被输入状态激活。这两个条件都是随机的，根据对实际项目的调查，失效之间没有发现很强的关联性。

通过对可靠性模型的概念与影响因素的了解可知对模型的评价标准如下。

(1)模型实现的可操作性。好的模型并不是越复杂越好，模型的理论基础很好，推理与逻辑都很严密精准，如果实施的过程非常复杂，以至于实施它所需要付出的代价已经超出模型的使用所带来的益处，这种模型将不会被人采纳，或者采纳了也不会持久。所以应更多的关注于模型可以计算有用的量的多少。从某种意义上讲，可靠性模型是否简单易操作比模型的其他评价标准都要重要得多。

(2)预测的及时性。如果不能在适当的项目阶段反映产品的质量信息，再好的模型也是没有多大实际意义的。

(3)基于合理的假设。模型假设是否合理是评价一个模型的基本条件，开始建立的模型如果是错误的，那么后续的工作都将是没有意义的。

(4)预测的覆盖率。主要体现在反映质量问题的广度与深度。一个好的模型主要体现在可以预测有用的量的多少与准确性，如果可以预测的量极少，可能还需要同时使用多个模型才可以满足产品可靠性预测的需求，无形中会浪费更多的项目资源。

(5)预测的有效性。能够给系统的失效行为以很好的预测，依据模型所产生的质量数据要符合产品的实际情形才能确保预测的有效性。同时，模型本身也应该涵盖常用的对产品的未来状态进行预测的方法。

4. 常见的模型分类

软件可靠性模型有很多，但广泛应用的不多，这里介绍的可靠性增长模型、指数模型就是目前应用较多的两种模型，二者同属于可靠增长模型不同的类别。

(1)可靠性增长模型

可靠性增长模型有两个主要类别，一个是基于故障时间间隔时间模型（Time Between Failure Model），另一个是基于特定时间间隔内故障或失败数目（或标准化后的比率）。

①时间间隔模型。变量为故障之间的时间间隔。当缺陷从软件产品中清楚后，故障时间间隔即变得越来越长。此模型假设第 n－1 个缺陷和第 n 个缺陷的时间遵循一种分布：参数与 n－1 个故障后遗留在软件产品中的潜伏缺陷数有关。此种分布，随着缺陷被检测出来并从产品中清除能够反映出产品可靠性的提高。同时依据此种分布也可以从故障之间观察到的时间间隔来预测下一个故障的平均时间。

②故障数目模型。通常以 CPU 执行的时间或日历时间为一个特定的时间间隔为标准，观察此时间间隔内的缺陷或故障数目。以此作为随机变量，当有缺陷被检测并从软件产品中移除时，依据此模型，意味着每单位时间所能检测到的故障数目将会减少，并可以此估算出保留下来的缺陷或故障数目。

（2）指数模型

指数模型可以说是可靠性增长模型的最基本形式。尽管软件可靠性模型数量繁多，众多的专业媒体上展开过专业的讨论，但事实上对于这些模型的实用性、有效性和局限性都没有完整的实践验证。大多数模型最终并没有得到广泛的应用的原因有很多，如模型所要求的数据采集成本过高、模型本身不易理解、更有甚者的是模型根本就是一个空洞的假设，预测与实际出现反差，不实用。

指数模型是 Weibull 系列的一个特例，其形状参数为 1。适合于单一衰减速为渐进的统计过程。其累积分布函数 CDF 和概率分布函数 PDF 为：

$$\text{CDF}: F(t) = 1 - e^{-(\frac{t}{c})} = 1 - e^{\lambda t}$$

$$\text{PDF}: f(t) = \left(\frac{1}{c}\right)e^{-(\frac{t}{c})}$$

C 是尺度因子，t 是时间，λ 是 C 的倒数。软件可靠性的应用中，λ 是指故障检测率也称故障率，曲线图如图 1-4、图 1-5 所示。

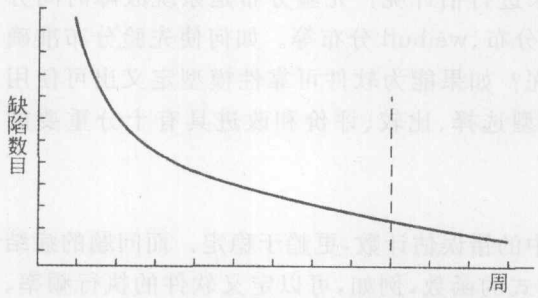

图 1-4　指数模型－密度分布

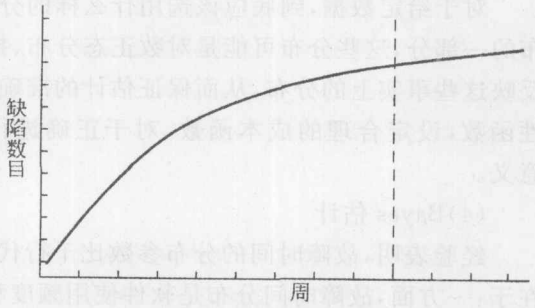

图 1-5　指数模型－累积分布

在软件可靠性领域中，指数模型是较为著名的一个模型，经常被认为是其他软件可靠性增长模型的基础。如 Goel-Qkumoto（1979）非齐次泊松过程模型（Nonhomogeneous Poisson Process Model，NPPM）中的平均值函数（CDF）实际就是指数模型。

1.1.2 软件可靠性工程的发展

20世纪90年代以来，软件可靠性工程的研究和实践最终脱离可靠性模型，在软件可靠性设计、测试、分析与评估以及软件可靠性工程管理等方面进行了比较系统的研究和比较广泛的实践，取得了显著的成效。

软件可靠性工程的发展是硬件可靠性工程发展的必然结果，是系统可靠性工程发展的历史要求，是软件工程高速发展的必然产物，其发展是高速渐进式的。软件可靠性工程根植于软件工程和硬件可靠性工程，其形态与体系的形成虽然经历了一个漫长的过程，至今还远滞后于硬件可靠性工程，但其发展势不可挡。可以预测，在今后的一段时间内，软件可靠性工程将得到更广泛的关注并将得以高速发展。

1. 软件可靠性模型

(1)假设验证

由于软件可靠性模型需要描述软件的测试策略和过程、运行剖面、错误分布等而使用大量假设，虽然这些假设中的大多数都以经验作为基础，但假设的正确性、合理性没有且难以得到有效验证。因此，对现有模型中使用的典型假设的验证，以及在此基础上提出能测试、能验证的共同假设，是软件可靠性模型开发的关键和主要方向之一。

(2)软件测度

为了提高软件可靠性模型的准确性，软件可靠性模型应将更多的知识包含进来，例如，软件开发过程、设计评审、软件测试、运行剖面以及当前的位置及其问题域的描述等知识。这些描述如果能以一种更准确的方式进行测量，那么它们将只要用“是”或“不是”的方式予以表达即可，将大大简化模型的数学表示及其应用。

(3)决策理论

对于给定数据，到底应该选用什么样的分布来进行估计呢？先验分布是系统故障时间分布的一部分，这些分布可能是对数正态分布、指数分布、weibull分布等。如何使先验分布准确反映这些事实上的分布，从而保证估计的准确性呢？如果能为软件可靠性模型定义出可使用性函数，设定合理的成本函数，对于正确进行模型选择、比较、评价和改进具有十分重要的意义。

(4)Bayes估计

经验表明，故障时间的分布参数比千行代码中的错误估计数，更趋于稳定。而问题的症结在于：一方面，故障时间分布是软件使用频度和方式的函数，例如，可以定义软件的执行频率、执行路径发生的概率等，而它们至少可以部分地被看作人类共性的一个结果，并且因此使得分布参数相对地趋于稳定；另一方面，软件错误数是问题域、系统历史和生命周期各阶段的函数，它受许多可变因素的影响，所以表现出更多的不稳定性。

(5)统一模型

种类和数量繁多的软件可靠性模型，导致了理论研究和实际应用的困难，对目前这种三国争雄的混乱状态进行澄清是非常必要的。模型统一是这一问题最有希望的解决办法。它试图