865

软件可靠性、安全性与质量保证

黄锡滋　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书主要介绍了软件质量及可靠性的基本概念、软件的可靠性设计、软件测试、软件可靠性预计模型、软件与硬－软件复合系统结构模型、软件系统安全性分析、程序的复杂性与可靠性分配、软件维护、软件的质量保证等。

本书不仅力求反映本学科国际上的新动态，同时也介绍了我国专家近年的一部分有实用价值的研究成果。

本书适合IT产业的高层管理人员、科技人员、质量及可靠性工程技术人员阅读，可作为高等院校相关专业的研究生、本科生的教材。

图书在版编目(CIP)数据

软件可靠性、安全性与质量保证/黄锡滋编著. —北京：电子工业出版社，2002.10
ISBN 7-5053-7990-9

Ⅰ.软… Ⅱ.黄… Ⅲ.软件可靠性—研究 Ⅳ.TP311.5

中国版本图书馆CIP数据核字(2002)第071582号

责任编辑：龚立堇
印　　刷：北京牛山世兴印刷厂
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京市海淀区万寿路173信箱　邮编100036
经　　销：各地新华书店
开　　本：787×1 092　1/16　印张：16.5　字数：420千字
版　　次：2002年10月第1版　2002年10月第1次印刷
印　　数：5 000册　定价：26.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。
联系电话：(010)68279077

前　　言

20世纪90年代是IT技术高速发展的时期，软件可靠性也开始从理论研究向实际工程应用过渡，为保证和促进IT技术的发展起到了积极作用。我国的情况亦大体类似，现在我国IT行业的技术和管理人员对软件可靠性的认识已远非90年代初启蒙时期可比拟。但是如果更换一个角度观察，我们看到整个90年代软件重大故障依然连绵不断，尤其是新旧世纪之交的“千年虫”事件，更使人类社会受到了巨大震撼，这些事实再次显示出软件可靠性的重要性，同时也表明软件可靠性发展状况仍然不能适应IT技术的需要。

1993年科学出版社出版了我编著的《软件的可靠性与安全性》一书，该书在部分高校研究生教学和信息产业系统高级技术培训中发挥了积极的作用。八年之后再来回顾，原书的部分内容已经不能反映学科发展现状了。为了给我国的软件可靠性研究和应用增添一砖片瓦，于是以原书为基础，重新编写一本适应当代软件可靠性发展水平的新书的设想油然而生。与此同时，中国电子产品可靠性与环境试验研究所和电子工业质量与可靠性培训中心为了适应和推动我国IT产业的发展，也急需一本实用的软件可靠性教材，用来提高IT产业的高层管理人员、科技人员、质量及可靠性工程技术人员的基础理论水平和工程应用能力。在中国电子产品可靠性与环境试验研究所和电子工业质量与可靠性培训中心的组织、推动和电子工业出版社的支持下，我终于能将名为《软件可靠性、安全性与质量保证》的新书呈献给读者。需要提及的是我在编写本书时不仅力求反映本学科国际上的新动态，同时也介绍了我国专家近期的一部分有实用价值的研究成果。

本书初稿完成后，曾送请中国电子产品可靠性和环境试验研究所协助审阅，该所陈昭宪、李新祥、杨伟光、文立春、李平等专家提出了许多宝贵意见，仅在此表示衷心的感谢。

黄锡滋

2002年10月

目　录

第1章 绪 论

1.1 计算机软件的“虫灾”

可靠性工程学从20世纪中期诞生后，经过近50年的发展，已经建立了一套完整的理论体系，开发了适合工程需要的应用技术和管理方法，为当代高新技术发展提供了有力的保证。但是在20世纪80年代之前，可靠性工程界关心的主要是硬件，软件可靠性没有受到足够的重视。是日益严重的软件“虫灾”，从反面使人们认识了软件可靠性的重要价值。

计算机出现和应用是20世纪的一项杰出的科学成就。现在，计算机技术已渗透到人类社会的各个领域，推动人类向信息时代迈进。软件是计算机的神经中枢，在计算机系统中起着至关紧要的作用。然而，一个不能忽视的事实是，软件从它诞生之日起，就受到“虫子”的折磨。所谓的“虫子（bug)”，是指寄生在软件中的故障，它具有巧妙的隐身功能，能够在关键的场合突然现身。这时不仅计算机的正常功能无法得到保证，还会造成资源浪费，甚至可能对人类社会造成严重危害。软件虫灾的严重后果，可以从下列两方面来说明。

1. 导致系统设备失效，危及人身设备安全

在20世纪40年代计算机出现后的很长一段时间，计算机应用局限于科学计算领域，程序由使用它的工程技术人员自行编制和调试，软件可靠性问题只有少数人具有朦胧的警觉。到了20世纪60年代，随着计算机性能的改善，应用领域日益扩大，对系统软件和应用软件的需求激增，人们仓促地进入了设计比较复杂的软件及软件系统时期。正是从这个时期开始，计算机科学家在付出高昂代价之后，逐渐认识了软件的二重性：一方面它是设计师自己塑造的、唯命是从的工具，另一方面寄生着“虫子”的软件，顷刻间可以演变为破坏力强大的敌人。

飞向宇宙空间，探索宇宙奥秘，是全人类共同的理想和追求。技术先进的国家不惜投入巨资，建造各种先进的航天器具。鉴于宇航过程中故障将造成严重的后果，对于航天计算机软件，人们都竭尽全力进行设计和测试，不敢有丝毫松懈。但就是这个领域里，人们仍然饱尝了软件故障的苦头。20世纪60年代中期，美国的首次金星探测计划，就因为在FORTRAN语言程序的DO语句中漏掉一个逗号，惨遭失败。在随后实施的阿波罗计划中，软件故障连绵不断。例如阿波罗宇宙8号飞船的仓载计算机因为软件故障丢失了寄存的数据，在阿波罗14号宇宙飞船的飞行过程中发现了18个软件故障。这些事件曝光后，社会舆论为之哗然。1973年，美国空军杂志刊载了一篇题为《计算机——明天空军的关键》的文章，其中有这样一段精彩的评论：“为阿波罗登月飞行而制定的软件开发计划，是当代耗资最大的、精心安排的工程。这项工程吸引了许多全国最卓越的程序设计师，组成了两支相互竞争的设计队伍。专家们对软件的检查倾注了全部专业知识和技能，同时将总计高达66亿美元的巨额资金用于阿波罗计划的软件开发。然而，阿波罗计划实施中的每一个重要的故障，从误报警直至真正的灾难，几乎都是计算机软件直接造成的”。

与此同时，在软件的其他应用领域中，软件故障也频频发生，不胜枚举。

2. 导致严重的经济损失

软件故障发生，导致系统失效并造成经济损失，这是明显的事实。此外，软件在开发和维护过程中，由于可靠性问题造成的隐性经济损失也不容忽视。这种损失可以从两方面分析。首先，由于软件不符合可靠性和质量要求，无法使用而造成的损失。关于这一点，美国ALABALA大学及MARSHALL空间飞行中心在20世纪70年代发表了一个调查报告，报告中说到："在调查所涉及的680万美元的软件开发费用中，占总数95%的650万美元被浪费了。其中，47%的软件已交付给用户，但从未使用过；有20%虽支付了开发费用，但无法交付用户使用；有19%因无法使用而放弃或重新进行设计；在其余的5%中，只有小于2%的费用所开发的软件交付使用后在真正发挥作用"。在本书即将付印之际，2002年6月28日美国国家标准技术研究所（NIST）公布了一项调查结果，该项调查表明，由于软件错误美国每年的经济损失高达600亿美元。进入80年代后，由于技术和工艺的改进，在计算机系统总的开发费用中，硬件占用的比例明显降低。与之相反软件功能更加强化，结构日趋复杂，使得软件占有的比例急剧上升，软件开发不当造成的经济损失显然随之上升。其次是软件交付使用后，如果软件的可靠性出现问题，势必导致软件的维护费用增加。统计资料表明，软件生存期费用中，维护费用占有很大的比重。图1.1是软件生存期费用中各阶段费用所占比例的图形。由图1.1可见，软件的维护费用占50%的份额。在软件的维护费用中，又以改正软件中的错误和改换版本的费用居多。改换版本虽然包括了扩充功能的因素，但也含有改正错误的因素。

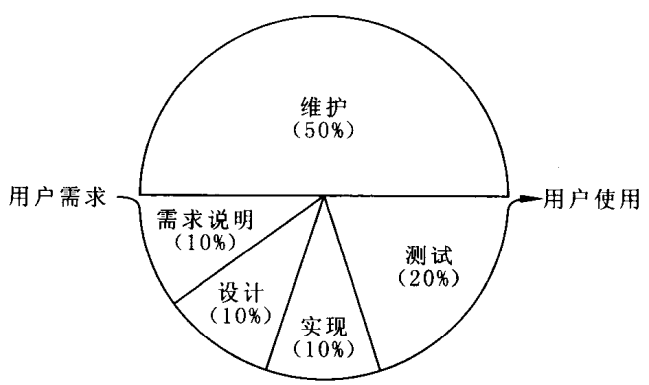

图1.1　软件生存期费用比例图

由于上述的原因，在80年代以后，国外对软件可靠性研究的投入明显加大，进入20世纪90年代，软件可靠性已经成为科技界关注的一个焦点。但是软件可靠性问题并没有丝毫缓解，软件可靠性技术仍然滞后于科学技术的发展，软件故障导致的重大事件在90年代依然屡屡出现，下面是几起有代表性的案例。

(1) 在电子和通信设备方面。1995年据Murthy报道，数字设备公司在欧洲配置2000多套系统中发生问题，主要是来自软件需求分析和接口的问题。

(2) 在医疗设备方面。90年代中期，美国Therac 25型放射治疗仪2号治疗模式（X射线模式）发生的54号故障，多次产生超计量辐射，造成了两人死亡和多人受伤的重大医疗事故。在此之前，旧式的放射治疗仪Therac 6型，使用机械安全互锁装置，从来没有发生过类似故障。Therac 25是Therac 6的改进型，使用软件安全互锁装置，岂料超计量辐射的严重故障反而连续不断。

(3) 在军事装备方面。1991 年海湾战争中，美军使用爱国者导弹，在拦截伊拉克飞毛腿导弹中，出现过几次拦截失败事件。经查明是软件计时系统的累积误差所致。

(4) 在航天技术方面。1996 年欧洲航天局首次发射阿丽亚娜 5 号火箭失败，直接损失 5 亿美元，还使耗资已达 80 亿美元的开发计划推迟了近三年，事故的原因是火箭控制系统的软件故障。

(5) 90 年代后半期，“千年虫”问题震惊世界，各国投入了大量的人力和物力，耗资数千亿美元，虫害才基本上得到控制。“千年虫”灾害本质上是一种特殊的软件故障。

由此可见，除虫的斗争，任重而道远，今天我们有充分的理由为战胜了千年虫，安全地进入 21 世纪而庆幸，但是明天必然有新的害虫出现，切不可掉以轻心。还需要指出，由于 Internet 的迅速发展，网络安全、信息安全已经成为国际社会共同面临的重大问题，网络软件中存在的“虫”，可能给网络黑客留下入侵的缺口，给病毒的制造者提供制造和传播的良机。2001 年 7 月中旬，一种名为“红色代码”的病毒及其变种对各国的网络服务器发动了疯狂的攻击，短短的十几天，就有包括我国的一些知名网站在内的几十万个网络服务器中毒落马。这个事件的起因是 2001 年 6 月 18 日，微软公司宣布在 IIS 网络服务器软件中发现一个漏洞，而 IIS 软件是架设网站最基础的软件之一，这正是黑客梦寐以求的软件缺陷，从而引发了黑客对这一软件“缺陷”的高度重视和利用。面对这样严峻的局势，软件设计师、软件可靠性工作者要有长期防虫抗灾的准备，承担起研究虫害发生和发展的规律，控制和减少灾害的影响，直至消灭虫害的历史重任。

1.2 软件可靠性发展史

软件可靠性的发展是与软件工程、可靠性工程的发展密切相关的，尤其与软件工程的关系更加紧密。可以说它是软件工程学派生的新分支，同时又合理地继承、利用了硬件可靠性工程的理论和方法。因此需要根据软件工程的发展线索，来了解软件可靠性的发展过程。软件工程的发展大体上可分为下列四个阶段。

(1) 1950 年 ~ 1958 年。在这段时间里，计算机几乎完全用于科学和工程计算。计算机编程语言只有机器语言和稍晚些时候出现的简单的汇编语言。那时没有专职的程序员，程序是由应用计算机的科学家和工程师自行编制的，没有公认的规则可供遵循。他们集科学家（工程师）、程序员和用户的职能于一身。因此，编制程序成了物理学家、数学家、机械工程师及各专业工程师的一项高超的智力活动。经过这些业余程序员的精雕细刻，确实编制出一些高明的精品，但是这些程序一般仅供个人一次性使用。在软件发展过程中的这个原始阶段，完全没有软件可靠性的概念。正是由于上述原因，以可靠性工程奠基性文件著称的 1957 年的美国 AGREE 报告，完全没有提及软件和软件可靠性的问题。

(2) 1959 年 ~ 1967 年。在这段时间内，软件领域开始出现高级语言，应用也日趋广泛。操作系统开始应用于计算机，使脱机操作得以实现。计算机的功能除单一的科学、工程计算之外，增添了一个重要的数据处理功能，使计算机的应用范围扩展至银行、商业、交通管理等领域。软件设计、编码、操作及系统分析等职能逐渐分离，并出现了相应的专业人员，用户也逐渐与设备和程序脱离。这个阶段在开发高级语言方面尤其受到关注，其目的在于减轻用户在编程上的困难。“虚拟机”概念也随之形成。与这阶段计算机硬件和软件技术飞快发展形成强烈对比的是，软件可靠性问题没有引起重视。这种状况很快就受到了软件的惩罚。

在那个时代 IBM 360 系列机是一项杰出的技术成就。然而，IBM 360 系列的操作系统 OS 360 却连续不断地发生故障，最终使该系列陷入困境。OS 360 系统的开发依靠了一批优秀的专家，投入了 5000～10000 人年工作量和每年平均 5000 万美元的费用，这个挫折引起了有关方面非常强烈的反响。在计算机应用的其他领域，问题也大量暴露。所以人们将这段时期称为软件危机时期。

（3）1968 年～1978 年是软件工程学建立和发展的时期。这个阶段以集成电路为主体的第三代电子计算机问世，各种小型机逐渐得到应用并开始配备上较为满意的系统软件，FORTRAN 等高级语言也可直接在小型机上使用，在硬件技术迅速发展的前提下，以及在软件危机的刺激和推动下，软件工程学得以建立和发展。软件的可靠性问题是软件工程学得以建立的一个强大推动力，也是软件工程界力图解决的一个重要目标。软件工程学的理论和技术为可靠软件的设计、测试和管理提供了指南和工具。但是，它没有能力满足系统开发中软件可靠性定量评估的要求。于是，一些著名的软件工程专家开始致力于利用和改造硬件可靠性工程学的成果，使之移植到软件领域，从而迎来了软件可靠性开创时期。这个阶段，在美国召开了多次软件可靠性国际学术会议，吸引了各界人士的关注。软件可靠性的数学模型开始涌现，著名的 JELINSKI－MORANDA 模型、SHOOMAN 模型、NELSEN 模型、MILLS－BASIN 模型都是在这个阶段推出的。此外，软件失效数据的积累和分析工作有初步发展。另一方面，硬件技术的快速发展，也给软件工程带来了消极影响。小型机可以作为控制元件应用的优点，吸引了一大批电子工程师投入这个领域，成为集工程开发、程序设计、使用于一身的非专业软件工作人员，他们中的许多人不了解软件工程的基本准则，随心所欲地设计了大量的实时处理嵌入式软件。20 世纪 70 年代的小型机还不具备“虚拟机”的基本特征，在这种背景之下，软件的质量和可靠性问题根本无法解决。

（4）1978 年至今是软件工程日趋成熟时期。在这个阶段，由于大规模集成电路的出现，导致计算机向大型化和微型化两个极端的方向发展，小型机在 1981 年 PC 机面市后逐渐被微型机所取代。这种两极发展的势头，对软件工程学的发展产生了深刻的影响。在大型化的一端，20 世纪 80 年代各种类型的大型机都具有了各种高级语言的编译程序。它的分时系统能力很强，可以配置数百个终端，同时还有批处理系统和实时系统的完善的通信软件，可以沟通与卫星站的联系，具有与其他计算机系统联网的能力。大型机系统的程序支持环境还能为其他应用软件的开发提供工具。庞大的应用软件为不同领域的用户提供了极大的方便。大型机的硬件功能也飞速地改进，到了 2000 年，IBM 的 ASCI White 系统，其计算速度已达每秒 12.3 万亿次。此外，比大名鼎鼎的“深蓝”计算机强大 1000 多倍，能在 1 秒钟内进行超过 1 千万亿次运算（1 petaflop）的“蓝色基因”已经开始研制。在微型化的一端，时至今日，体积小巧、环境适应性强，CPU 主频达到 1.5GHz，内存为 256MB 以上，硬盘 50GB 以上的 32 位微型机已经成为网络终端、办公室和家庭的标准配置。这些微机的功能，已经达到和超过了旧时大型机的水平。2001 年 10 月全球最大的芯片制造商英特尔公司声称，该公司已研制出一种新型半导体包装技术，利用这种技术可以在未来 6 年之内生产出装有 10 亿只电子晶体管，运行速度高达 20GHz 的微处理器。此外新型的 64 位的 CPU 正在开发中。作为计算机微型化的极端，各种小巧的嵌入式计算机芯片，广泛地配置在各种类型的设备中，承担着自动控制的功能。计算机在生产、武器装备的自动控制和社会生活的各个方面，扮演了极为重要的角色。成为信息社会的一个重要支柱。在上述的各种场合，软件的重要性更加突出，软件工程学的准则得到科技界的一致认同，软件工程学已经成为大学计算机专业的重要课程。

对软件可靠性的研究在这一阶段出现了旺盛的势头；软件的设计、测试技术都有所提高。各种改进后的软件可靠性模型相继推出，模型的验证和试用受到重视；软件可靠性管理技术的开发列入日程，软件可靠性标准化工作开始起步。国际电工委员会的 TC56 技术委员会在 1985 年成立了软件可靠性工作组，开始制定软件可靠性和维修性标准。软件安全性问题受到了特别的重视，硬件可靠性和安全性分析中所采用的故障树分析法、故障模式效应分析法和潜藏回路分析法，已在软件安全性分析中使用，并取得了令人鼓舞的成果。在这样的背景之下，为了应用软件可靠性的成果，1988 年 AT&T 贝尔实验室，编写了“软件可靠性工程”内部教材，此后“软件可靠性工程”这个术语很快为科技界所接受，软件可靠性从此步入了从理论研究向工程应用的过渡时期。

综观软件可靠性 20 多年的发展过程，进展是明显的。然而与硬件可靠性工程的发展水平相比，仍存在很大差距，尤其在工程应用方面，远不能满足计算机科学和信息社会发展的需要。我们希望，在新的世纪，软件可靠性会有更快的发展。

参考文献

[1] Marco, A., and Buxton, J., The Graft of Software Engineering, Addison Wesley, Workengham, 1987.

[2] Uber, J. G., Goverment Report N89 – 21754, USA, 1989.

第 2 章　软件质量及可靠性的基本概念

2.1　软件及软件工程

2.1.1　软件的定义

在现代社会，“硬件”、“软件”是两个常用词汇，其含义远远超出计算机科学的范畴。为了明确本书的研究对象和研究范围，需要对这两个名词的含义加以说明。在本书中，“软件”一词是计算机软件的简称，“硬件”一词是指计算机系统中的主机及外围设备，有时则泛指所有具有物理构形的设备及元器件。例如一个计算机控制系统，包括了受控设备、传感器、电缆及计算机，这时计算机仅是系统中的一个硬件。

根据我国国家标准 GB/T－11457 的定义，软件是“计算机程序、过程、规则及与这些程序、过程、规则有关的文档，以及从属于计算机系统运行的数据”。我国国家军用标准 GJB－437 则定义软件是“计算机的各种程序，相应的数据和文档的总称，并包括固件中的程序和数据”。这两个定义的内涵是一致的，但在 GJB－437 中明确地提到了固件中的程序和数据。有人常常把计算机用的磁盘、磁带称为“软件”，这种称呼比较含混。应该把软件和存储软件的物理介质区别开，存储程序和数据的有形物，如硬盘、软盘、卡片等，属于硬件范畴。

在软件的定义中还涉及到计算机程序、数据、文档、固件等术语，这些术语 GJB－437 作出了如下的解释。

计算机程序：指一系列可以被计算机设备所接受的指令或语句，这些指令或语句可以使计算机执行一种或多种运算。

数据：数据是用能被计算机设备接收、翻译和处理的结构形式所表示的事实、概念或规则，它们可以存储于计算机内，也可以用计算机可读的形式存在于设备之外。

文档：指技术资料，这些资料记载着计算机程序的需求、设计、实现和其他细节，也包括用可供人们阅读的形式制备的程序列表及打印输出，这些资料还为计算机程序的使用及维护提供指南。

固件：指由程序、数据及具有记忆力功能的特殊硬件共同构成的功能实体，这些程序和数据在正常运行的条件下是固定的和无法改变的。

按照软件的性质和功能，可将软件区分为如下四种类型。

(1) 支持软件。指用于帮助和支持开发的软件，它包括编译程序、汇编程序、连接和装配程序、库管理程序及相应的文档，还包括调试软件、模拟软件、数据析取和归约软件、开发工作管理软件、配置管理软件、设计工具软件等。

(2) 应用软件。指用于解决各个专用领域的特殊问题的软件。应用软件种类繁多，凡是以科学计算、工程设计、过程控制、事务管理为目的而设计的软件都属于这个类型。

(3) 系统软件。指管理计算机系统资源的软件，在软件开发期和程序运行期使用的操作

系统和数据库管理系统都属于这一类型。

(4) 测试和维护软件。指用于故障诊断、错误隔离、系统调试及检验设备和系统可靠性的软件。

2.1.2 软件工程

软件工程学是在软件危机的背景下诞生的。1968 年在联邦德国 Garmush 召开的计算机软件技术、管理和维护讨论会上，首次以“软件工程”一词来标志会议的主题。此后，这一术语很快为科技界所接受。

对“软件工程”一词，曾经出现过若干种定义。

1972 年，Baner 定义软件工程是“建立和运用完善的工程原则，以获得经济上划算的、又能可靠地工作的软件”。1983 年，IEEE 计算机学会定义软件工程是“用于软件开发、运行、维护和报废的一套方法”。1985 年，Fairley 定义软件工程是“开发和维护软件产品的、一套系统的技术和管理规则，能使软件产品的生产和维修以预定的经费按预定的时间完成，其首要目的在于改善软件产品的质量”。1987 年，Macro 定义软件工程是“建立和运用完善的工程和管理原则，开发适用的工具和方法，在已知的及合理的资源条件下，获得满足规定要求的高质量的产品”。1989 年，我国国标 GB/T - 11457 采用 IEEE 计算机学会的解释方法作为软件工程的定义。以上这些定义尽管表达方式不同，但是其内涵是相似的。从这些定义可以看出，软件工程的首要目的是获得质量满意的可靠的软件。其次，合理的经费开支和准时交付使用也是软件工程的重要目标，因此可以把软件工程的目标归结为“质量、费用和时间”。

软件工程学的建立，使程序设计由个人的艺术创作转变为工程开发，为提高软件产品的质量和可靠性作出了重大贡献。但是，软件工程还需要继续发展，在软件工程既定框架内，对软件可靠性的重视程度和研究深度明显不能满足需要。

2.1.3 产品生存期

任何一项产品，从构思开始，经过设计、制造、使用直至废弃为止的整个期间，构成了产品的生存期。“产品”一词具有广泛的含义，它既用来表示由硬 - 软件综合系统构成的大型工程项目，也用来表示设备，乃至单个的元器件。产品生存期的概念与人的生存期的概念有类似之处。人生的各个阶段，由于生理及心理特征不同，采用的保健措施是不同的。产品在生存期的各个阶段，也需要采取不同的技术、管理措施。按照产品开发的客观规律，正确区分生存期的各个阶段，明确各个阶段的任务和要求，对产品不同的阶段实施规范化的管理，是软件工程的一个基本要求，也是获得高质量、高可靠性产品的保证。在硬 - 软件复合系统中，软件是系统的一个组成部分，软件的开发必须与系统的开发过程相适应，软件的功能及可靠性必须服从系统的功能和可靠性的要求。

国际电工委员会标准 IEC - 60300 将产品的生存期划分为五个阶段。

(1) 构思和定义阶段 (Concept and definition phase)。在这个阶段，使用方和承制方对产品的需求形成共识，就产品的基本特征和性能做出双方认可的定义，并用产品说明书的形式予以规定。

(2) 设计和开发阶段 (Design and development phase)。这个阶段的任务，是按照产品说明书设计制造出符合规定功能要求的产品样品，对样品进行实验室模拟试验或现场试验，拟定