

● 赵晓华 编 ●

计算机软件可靠性 与质量管理

JISUANJI KEKAO
XINGYU ZHILIAN
GUANLI
JISUANJI KEKAO
XINGYU ZHILIAN
GUANLI

计算机软件可靠性 与质量管理

赵晓华 编

中国经济出版社

(京)新登字 079 号

责任编辑：张抒文

封面设计：白长江

计算机软件可靠性与质量管理

赵晓华 编

*

中国经济出版社出版发行

(北京市百万庄北街 3 号)

各地新华书店经销

北京朝阳区北苑印刷厂印刷

*

850×1168 毫米 1/32 13.625 印张 342 千字

1992 年 12 月第 1 版 1992 年 12 月第 1 次印刷

印数：1—3500

ISBN 7-5017-1702-8/T·30

定价：8.20 元

目 录

前 言

第 一 部 软件可靠性模型解析及评估

第一章 绪论	1
1.1 软件开发过程与开发技术	1
1.2 软件开发管理	7
1.2.1 生产管理与质量管理	7
1.2.2 软件质量及实现过程	10
1.2.3 测试与质量管理	13
1.3 开发管理计划	17
1.3.1 开发规模与开发工时的估算	17
1.3.2 开发进度预测	21
第二章 软件可靠性	31
2.1 可靠度、故障、差错和缺陷	35
2.2 可靠性度量基准 (尺度)	38
2.3 可靠性模型	44
2.4 可靠性数据	44
2.5 容错技术	47
2.6 基本可靠性概率统计	50
2.6.1 可靠性概率分布	50
2.6.2 极大似然估计法	62
第三章 不可靠率模型	67
3.1 不可靠率 (危险率)	67

3.2	指数不可靠率模型	71
3.2.1	Jelinski-Moranda 模型	71
3.2.2	Moranda 模型	76
3.2.3	Shooman 模型	79
3.2.4	Musa 模型	84
3.2.5	Xie 模型	89
3.3	威布尔 (Weibuli) 不可靠率模型	94
3.3.1	Schick-wolverton 模型	94
3.3.2	Wagoner 模型	97
3.4	巴列特 (Pareto) 不可靠率模型	102
3.4.1	Littlewood-Verrall 模型	102
3.4.2	Littlewood 模型	105
第四章	NHPP 模型	111
4.1	NHPP	111
4.1.1	NHPP 基础理论	111
4.1.2	以 NHPP 为基础建立的软件可靠度增长模型	119
4.1.3	用极大似然估计法推定模型参数	126
4.2	指数型可靠度增长模型	131
4.3	修正指数型可靠度增长模型	135
4.4	延迟 S 字母型可靠度增长模型	140
4.5	普通 S 字母型可靠度增长模型	144
4.6	对数型泊松执行时间模型	150
4.7	可靠性评价例	155
4.7.1	软件可靠性数据分析	155
4.7.2	用 SRET 进行可靠性评价例	164
第五章	多维几何可靠性模型	174
5.1	多维几何分布	174
5.2	捕捉——重捕抽样法	177

5.3	SPQL	180
5.4	测试情况与差错响应关系模型	185
第六章 二项可靠性模型		
6.1	马尔可夫过程	189
6.2	基本二项可靠性模型	197
6.2.1	可靠性评价尺度	201
6.2.2	推定未知参数	203
6.3	考虑到测前信息的二项可靠性模型	204
6.3.1	可靠性评价尺度	207
6.3.2	推定未知参数	209
6.4	可靠性评价例	212
第七章 运用阶段的软件可靠性评价		
7.1	经验方法	217
7.2	不可靠率(危险率)评价方法	219
7.3	利用 NHPP 的方法	222
7.4	可靠性评价例	226
第 二 部 软件可靠性评价与开发管理		
第八章 依存测试人力型软件可靠度增长模型		
8.1	测试人力函数	234
8.2	依存测试人力型软件可靠度增长模型的构筑	241
8.3	可靠性评价例	245
8.3.1	基本可靠性数据解析例	245
8.3.2	具有威布尔测试人力函数的 NHPP 模型适用例	251

第九章 软件测试工程管理	264
9.1 测试工程进度管理问题	264
9.1.1 进度管理方式	264
9.1.2 测试工程进度管理具体推算例	270
9.2 测试人力的最优分配问题	272
9.2.1 最优人力分配法 (1)	272
9.2.2 最优人力分配法 (2)	277
9.2.3 最优测试人力分配问题具体例	279
第十章 软件发行方略	284
10.1 依据可靠性评价尺度的最优发行问题	285
10.1.1 最优发行时间决定方法	285
10.1.2 最优发行问题应用例	288
10.2 依据费用评价基准的最优发行问题	290
10.2.1 最优发行时间决定方法	290
10.2.2 最优发行问题实用例	295
10.3 考虑到软件工期的最优发行问题	296
10.3.1 最优发行时间决定方法	296
10.3.2 最优发行问题实用例	302

第 三 部 软件质量管理实践

第十一章 SWQC——软件全面质量管理	307
11.1 SWQC 的诞生	307
11.2 SWQC 的工具与环境	311
11.2.1 SWQC 管理者的责任	311
11.2.2 SWQC 的七大工具	313
11.3 SWQC 与智能软件工厂 (ISF)	327
11.3.1 智能软件工厂的目的	327

11.3.2	软件工程结构 SEA/I	330
1.	SEA/I 的特长	331
2.	STEPS 开发标准	334
3.	CASE 工具	345
11.4	SWQC 的展望	348
第十二章	软件质量管理方法实例集	353
12.1	日本 Omro 公司的软件质量保证方法	353
12.1.1	开发工程体系与质量管理措施	354
12.1.2	质量保证的主要作法	355
12.1.3	效果及评价	362
12.1.4	问题与对策	364
12.2	东芝工程公司的软件质量管理方法	366
12.2.1	基本概况	366
12.2.2	开发项目管理和软件质量管理	366
12.3	日立微机工程公司的软件质量管理方法	373
12.3.1	基本概况	373
12.3.2	设计阶段的高质量管理	376
12.3.3	确保程序制作与测试阶段的质量	380
12.4	东芝公司的软件质量评价方法	385
12.4.1	基本概况	385
12.4.2	质量评价例	386

附 录

附录 4.A	K-S 拟合优度检验法	392
附录 8.A	推算测试人力参数程序例	393
附录 8.B	推算可靠度增长参数程序例	397
附录 9.A	关于测试工程进度管理问题解法的程序例	399
附录 9.B	最优人力分配 [算法 1] 程序例	402

附录 9.C	最优人力分配 [算法 2] 程序例	404
附录 10.A	运用依存测试人力型可靠度增长模型解最优 发行问题的程序例	406
附录 12.A	石井潜在差错数预测法	409
附录 12.B	程序质量管理及程序可靠性评价	415

第一部 软件可靠性模型解析及评估

第一章 绪 论

在现代信息化社会，或称信息网络社会，计算机系统在一切领域中发挥着重要的作用。换言之，由于信息与经营的一体化，使各行各业对计算机系统的依存性越来越大。这就造成了对计算机软件的需求量逐年剧增，开发周期愈来愈短（短工期化）的发展趋向。而计算机系统愈是深入各行各业，它的系统形态愈变得多样化，复杂化和大规模化。在这样的背景下，如何在明确的开发目标指导下，系统地管理好整个开发项目的全过程，以确保软件开发的生产效率和用户满意的软件质量就成了软件工程学的中心命题。一般软件开发管理的主要对象是质量 (*quality*)、费用 (*cost*)、和工期 (*delivery*)。要想开发出高质量的软件产品，必须在正确预计开发规模和工时的基础上，合理运用开发资源，在不同阶段切实推行有效的质量管理，费用管理和工程管理。

本章将概述一般软件的开发过程和开发技术，并讨论开发项目管理（即 *project control*）的基本思考方法。另外还将介绍和软件可靠性密切相关的质量管理机能、软件质量及其实现过程，以及软件测试与质量管理的关系。

1.1 软件开发过程与开发技术

计算机系统之所以渗透到现代社会的各个领域，这和硬件价格低廉化，硬件产品多样化是分不开的。今天，不仅是各式各样的计算机独立设备，电话交换机、汽车、各种家庭用品也都大量地引入电脑控制系统，这就使软件在各种各样领域里的需求与日剧增。

在初期的计算机系统开发工程中，硬件组成是研究的中心，对软件的研究比重很小。随着 IC 制造技术和大批量生产技术的发展，硬件的性能和可靠性获得了日新月异的巨大进步。现在，软件开发成了计算机应用技术的中心课题。计算机也从单性能机→多目的型机→集中处理型系统，计算机的处理内容也从科学技术计算→数据处理→信息处理。

1970 年以后，人们开始调查统计用手工作业的软件开发方法生产的软件发生错误和系统缺陷的原因，并且着手研究和实验消除软件隐患的办法。做为提高质量和生产效率的方法论，产生了强调在软件开发的各阶段加强质量、费用和工期管理的软件工程学 (*software engineering*)。

软件工程学主张对软件开发过程和包括运用阶段的维护在内的整个软件生存周期都要进行科学的管理。一般的开发方法都将开发过程分为几个阶段 (即工程, *process*)，每个工程结束时都要完成文档 (*document*) 的制作工作以后再转入下一个工程。这种开发模式叫瀑布方式 (*waterfall model*)。如图 1.1 所示，经由用户需求规格定义工程→设计工程→编码工程→测试工程等系列的开发阶段生产出软件产品。进入运行阶段的同时也相应进入维护阶段，最后在一定时限内成为废弃物。这个瀑布式的软件开发过程的各工程概要如下：

① 需求规格定义 (*specification*)

明确软件使用目的和外部环境，在充分理解用户要求的条件下，设定需求规格，并以严密的定义语言制成需求规格说明书。

②设计 (*design*)

以使用目的和外部条件为基础、设计满足需求规格的软件功能、性能及详细处理内容。设计工程分基本设计 (*preliminary design*) 和详细设计 (*detailed design*)。基本设计按照需求规格将软件系统按功能划分成尽可能相互独立的模块，确定模块间的连接方法，设计实现它们之间的联系的控制结构。详细设计根据基本设计定义实现模块功能的算法，设计模块的输入输出方式和数据操作方法。所有的设计内容最后都写入基本设计书和详细设计书里。

③编码 (*coding*)

以特定的程序语言将设计结果记述下来，使其成为可以在计算机上执行的软件产品。用程序语言记述的结果称源代码 (*Source Code*)，源代码编译后得到最后的成果目标代码 (*object code*)。

④测试 (*test*)

检查和确认经编码而制成的程序是否满足需求规格或设计内容。这时将发现和修正截止编码工程的开发过程中潜入软件产品的大量缺陷和差错。

1970年以来出现了软件生存周期 (*life cycle*) 的概念，即把软件从开发到运用、维护，直至废弃的全过程当做软件的生存周期。在这个过程的前期——开发阶段有一些问题是不确定的。例如，截止编码工程以前的制作成品在实际上是否能够实现需求规格尚不明确；仅仅根据需求规格说明书、设计书之类的文档是很难确定软件的动态特性的，况且文档中的矛盾和缺陷也很难发现。为了解决这些问题，软件工程学者们多年来一直在从理论与实践上试图制定出更科学的软件开发规范 (*paradigm*)。例如1988年日本大场充提倡的“原始规范法 (*prototyping*)”，1986年美国 B.W. Boehm 提倡的“螺旋方式 (*spiral model*)”等等。原始规范

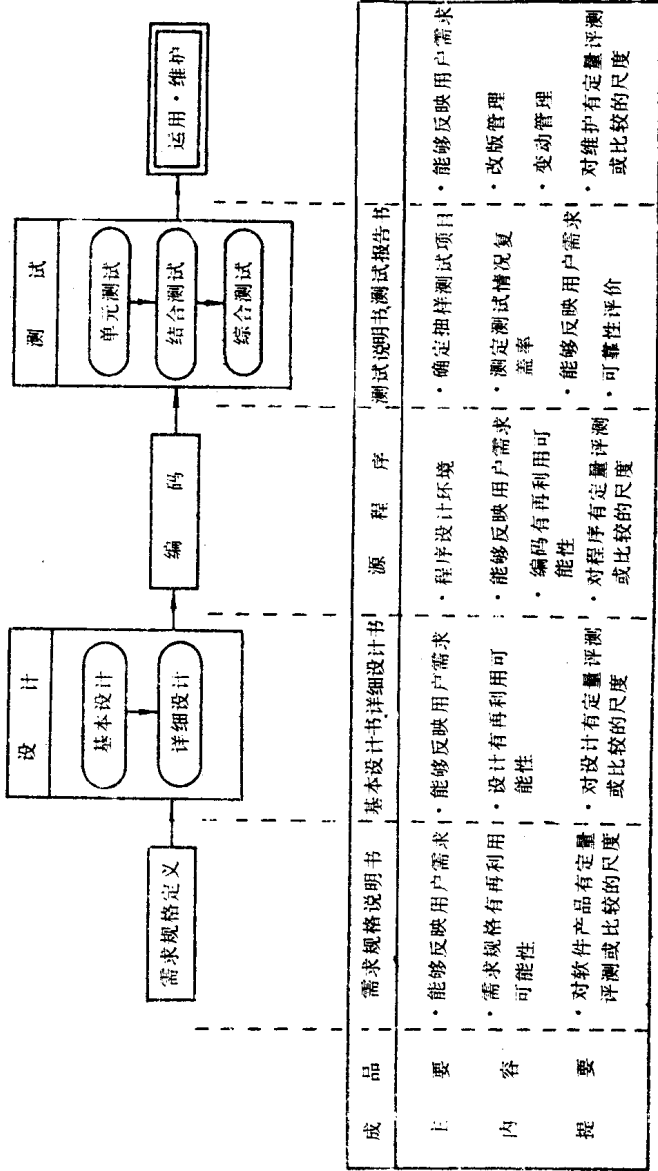


图 1.1 软件的开发过程

法以抽象程度较高的设计方法实现基本处理，以便尽早作成可以在计算机上动作的软件，然后通过反复试运行进行评价和修改，从而逐步接近用户需求的规格。螺旋方式是把“瀑布式”开发方法与“原始规范法”结合起来的一种方法。但是从七十年代起就普遍使用的“瀑布法”至今仍然是最通用的开发方法。对软件开发的规范方法的实用性研究和推广还有待进一步的发展。

软件工程学是研究如何高效率地、系统地、经济地生产出能够在计算机上正确运行的软件的技术的工程方法学。这些软件生产技术包括

- 提高软件的生产经营效率的方法
- 软件可靠性保证和软件质量评价方法
- 软件维护技法
- 软件开发人员资源的保证和合理调配方法等。

软件生产技术，如图 1.2 所示，可以分为软件固有技术与软件管理技术两大类。前者是指与构成软件的程序及文档的制作和维护工作直接相关的技术。后者虽然与软件开发、维护工作间接相关，但是对于在有限资源、时间和费用条件下生产高质量软件来说也是必不可少的技术。软件固有技术主要指下列内容：

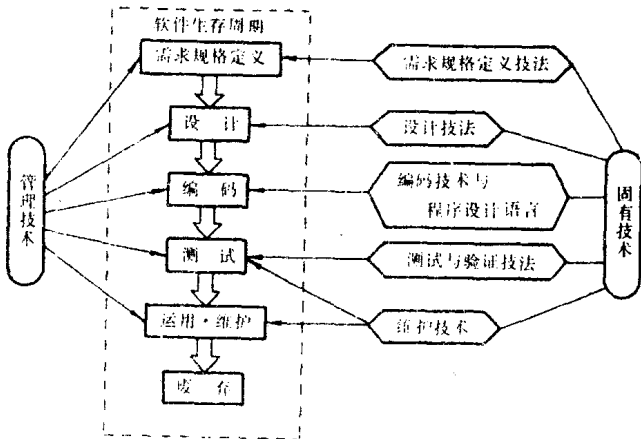


图 1.2 软件工程学的固有技术与管理技术

①软件需求规格定义技法

正确而无遗漏地记述需求规格的技术和方法。

②软件设计技法

分析需求规格，决定与程序结构相对应的处理方式的系统设计技术与方法。

③编码技术与程序设计语言

程序编码作业的标准化技术和高效率的程序设计语言。

④测试与验证技法

检验软件成品是否符合用户需求规格，检出错误和缺陷并进行修正的技术和方法。

⑤软件维护技术

运用阶段的软件规格变更、修正、改进技术（也包含测试阶段的部分变更、修正）。

⑥软件的文档化技术

将开发过程的各阶段工作结果写成文字资料的技术。这些文字资料，即文档（*document*）也是软件成品之一。

软件管理技术包括：

①软件性能和经济价值评价技法

评价软件“性能价格比”的技术和方法。

②软件质量及可靠性技术

关于如何提高软件质量，改善与分析评价可靠性的技术。

③软件开发项目（*project*）管理技术

关于软件开发项目的计划与管理的技术。

在软件开发过程中，为了更有效地使用上述软件生产技术，还应该合理地使用软件工具（*software tool, programming tool*）（详见本章参考文献2），4）。由于本书主要是讨论关于提高软件质量的可靠性技术的，所以在涉及软件项目管理的讨论时，只限于研究与可靠性密切相关的工程管理，质量管理及费用管理。

1.2 软件开发管理

所谓软件生产，指生产出程序和文档两种产品的过程。这个过程可以举出如下的特点。

①软件生产一般都需要由一个技术人员集体上下协作来完成。

②软件生产的产品是非直观的，是脑力活动的产物。

③软件生产是复杂而长时间的工程过程。

④在软件生产中，参加开发的技术人员集体中的成员必须相互交流。

这就是说，要想在一定的资源，时间和费用条件下制成符合需求规格的高质量软件，就必须采用适当的软件管理技术。这个管理技术包括管理实行前对开发规模，开发工时，开发费用的估算，以及以这个基本估算为依据所展开的各项管理技术。开发规模是对产品程序的大小的描述，用行数或语句数 (*step* 数) 表示，分基本语句数 (*statement step* 数) 和命令语句数 (*command step* 数) 两种表示方式。开发工时是对软件工程技术工作人员工作时间的描述，用 (人·日) 或 (人·月) 等来表示。开发费用指开发所需要的各种费用，一般包括人工费 (工时数×单价)，计算机机时费 (计算机使用时间×单价)，以及其它消耗。软件开发管理也称开发项目 (*project*) 管理。即将有关软件开发的各项活动统一称为“开发项目”，在一定的目标、计划、进度和预算指导下推行开发项目内容的实施。在这个过程中，和所有的工业产品生产过程一样，必须重视工程管理，质量管理和成本管理。

1.2.1 生产管理与质量管理

一般的生产管理指对生产出产品的过程的管理，它由质量管理、工程管理、成本 (费用) 管理、及人员管理构成。对于软件工程学来说，这个生产管理与软件开发管理是同义词。软件开发

管理的关键问题是如何对软件生存周期内的质量，时间，费用三要素进行计量化，如何通过质量管理、工程管理和费用管理活动对这三要素进行分析和评价。

质量管理的目的在于保证开发成品的最终质量满足用户对质量的要求。为此，需要设定质量目标，质量保证基准，确立质量管理制度和措施。特别是当然质量 (*must-be quality*) 是可靠性的重要因素。

工程管理的主要工作包括确立日程计划，规定各开发阶段(工程)的定义和成品，设定工程管理重点——“里程碑 (*milestone*)”，制作工程作业网络图，掌握各工程进度，及时发现工程延误和提前，并及时调整相关部署，等等。在实际中，由于软件产品是非直观的，判断生产过程中的准确进度常常是很困难的。通常的办法是事先将开发进度状况绘制成图表，在适当的时机设定工程管理点(里程碑)，及相应的评价项目。在执行中进行到这些里程碑处时，对进度和相应项目进行检查和评价。

费用管理有两方面的工作，一是正确估计用于开发和维护的工时、人工费和计算机机时费等费用预算，一是按此预算管理生产过程中投入的费用。具体包括随时掌握预算与支出的相对状况、安排与调度人力、分析预算与实际的距离、制定标准单价、使费用内容具体化、分析损益收支情况、设定生产指标、改善生产效率等等。

表 1.1 是一个软件开发管理技法的例子(详见本章参考文献 3))。为了有效地进行开发管理，一定的管理措施和标准化方法是很重要的。图 1.3 表示了一个叫 *PDCA* 的管理措施的例子。这个循环管理过程由计划 (*Plan*)、实施 (*Do*)，检查 (*Check*) 和改善 (*Action*) 四措施构成。标准化 (*standardization*) 的意图是为了使软件开发作业工厂化，从依赖于开发人员个人能力的手工业式生产转向软件工程学指导下的工业化生产。软件开发标准化分为作业本身标准化和管理活动标准化两大类。前者指在软件开发和维