


军用软件能力成熟度模型 及其应用

GJB 5000—2003 SW-CMM

石柱 编著

*Junyong Ruanjian
Nengli Chengshudu Moxing
Jiqi Yingyong*



 中国标准出版社
www.bzcb.com

军用软件能力

成熟度模型及其应用

石柱 编著

中国标准出版社

图书在版编目(CIP)数据

军用软件能力成熟度模型及其应用/石柱编著. — 北京:中国标准出版社,2003. 12

ISBN 7-5066-3322-1

I. 军… II. 石… III. 军用计算机—软件工程
IV. E919

中国版本图书馆 CIP 数据核字(2003)第 096227 号

中国标准出版社出版

北京复兴门外三里河北街 16 号

邮政编码:100045

电话:68523946 68517548

中国标准出版社秦皇岛印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

开本 787×1092 1/16 印张 13¼ 字数 318 千字

2003 年 12 月第一版 2003 年 12 月第一次印刷

*

印数 1—3000 定价 46.00 元

网址 [www. bzcsb. com](http://www.bzcsb.com)

版权专有 侵权必究

举报电话:(010)68533533

前 言

随着计算机科学与技术的飞速发展,应用领域的不断扩大,各种软件的需求量急剧增加。软件不再只是计算机硬件的附属品,而已成为系统中的一个独立部分,甚至是主宰系统功能的关键部分。软件在现代战争、航空航天等领域都有着广泛应用并起着极为重要的作用。

由于软件实体的复杂性,使得软件的构思描述和测试都非常困难,提高软件的生产率和质量便成为软件界所面临的最大挑战。目前,在我国国防科技工业领域,软件研制与开发仍基本上处于“初始”阶段,其质量控制还缺乏有效的组织、手段和相应的规范。

为了降低软件的风险,改进软件承制单位的软件过程能力,总装备部电子信息基础部组织有关专家,参照国内外有关研究成果,制定了国家军用标准 GJB 5000—2003《军用软件能力成熟度模型》,该标准规定了军方用于评估组织的软件能力成熟度的模型,并规定了软件整个生存周期内的主要软件管理过程和工程过程,该标准适用于第二方或第三方对组织的软件能力进行评价,也适用于组织本身对软件过程进行评估和改进。

本书主要介绍军用软件能力成熟度模型的主要内容,阐述应用该模型进行软件过程改进、软件过程评估和软件能力评价的参考方法。本书的主要内容和结构如下:

第1章 绪论。主要介绍军用软件能力成熟度模型的由来、编制思路和应用设想。

第2章 基本概念。主要介绍理解军用软件能力成熟度模型的一些基本概念,包括软件、软件与硬件的异同、软件度量、软件过程、不成熟组织与成熟组织的特征及实施军用软件能力成熟度模型的组织保证等。

第3章 军用软件能力成熟度模型概述。主要阐述军用软件能力成熟度模型框架、模型的内部结构、成熟度等级的跨越问题及模型的应用。

第4章 可重复级。主要阐述可重复级的特征和其6个关键过

程域的内容和实施要求。每个关键过程域均阐述其主要内容、目的与目标、实施基础、实施活动、实施验证、实施要点及相关标准。

第5章 已定义级。主要阐述已定义级的特征和其7个关键过程域的内容和实施要求。每个关键过程域均阐述其主要内容、目的与目标、实施基础、实施活动、实施验证、实施要点及相关标准。

第6章 定量管理级。主要阐述定量管理级的特征和其2个关键过程域的内容和实施要求。每个关键过程域均阐述其主要内容、目的与目标、实施基础、实施活动、实施验证、实施要点及相关标准。

第7章 优化级。主要阐述优化级的特征和其3个关键过程域的内容和实施要求。每个关键过程域均阐述其主要内容、目的与目标、实施基础、实施活动、实施验证、实施要点及相关标准。

第8章 军用软件过程改进参考模型。主要介绍军用软件过程改进参考模型的5个阶段(启动、诊断、建立、行动和提高)的目的、入口准则、出口准则及活动。

第9章 军用软件过程评估方法。主要介绍有关军用软件过程评估方法、最低要求和评估的参与者,详细阐述了军用软件过程评估方法的活动、时间和资源要求及后续活动。

第10章 军用软件能力评价方法。主要介绍有关军用软件能力评价方法的基本概念,详细阐述军用软件能力成熟度的评价过程和活动。

本书的主要内容曾作为航天科技集团公司一院十二所软件过程改进的内部培训教材进行了多次宣讲,在本书的编写过程中曾得到了何新贵院士、闫宇华、宋太亮、蔡愉祖、王方德、崔宗学、王伟、赵宇棋、刘继忠、陈政、潘华、刘益新等人的关心、支持和指导,在此表示感谢。

本书可供从事软件过程改进的技术管理人员、项目管理人员、质量管理人员、软件技术人员、大专院校本科生、研究生学习及参考。

限于作者的水平,本书难免存在欠缺和错误,敬请读者批评指正。

石柱

2003年9月8日

目

录

第1章 绪论

- 1.1 军用软件能力成熟度模型的由来..... 1
- 1.2 军用软件能力成熟度模型的编制思路及应用设想..... 5

第2章 基本概念

- 2.1 软件及其特点..... 8
- 2.2 软件与硬件的异同..... 9
- 2.3 软件度量..... 11
- 2.4 软件质量..... 13
- 2.5 软件过程..... 16
- 2.6 不成熟组织与成熟组织的特征..... 19
- 2.7 实施军用软件能力成熟度模型的组织保证..... 21

第3章 军用软件能力成熟度模型概述

- 3.1 军用软件能力成熟度模型框架..... 24
- 3.2 模型的内部结构..... 31
- 3.3 成熟度等级的跨越问题..... 37
- 3.4 模型的应用..... 38

第4章 可重复级

- 4.1 特征及包含的关键过程域..... 42
- 4.2 需求管理..... 43
- 4.3 软件项目策划..... 46
- 4.4 软件项目跟踪和监督..... 51
- 4.5 软件子合同管理..... 56
- 4.6 软件质量保证..... 62
- 4.7 软件配置管理..... 66

第5章 已定义级

- 5.1 特征及包含的关键过程域..... 72
- 5.2 组织过程焦点..... 72
- 5.3 组织过程定义..... 76

5.4	培训大纲	81
5.5	集成软件管理	85
5.6	软件产品工程	92
5.7	组间协调	99
5.8	同行评审	103

第6章 定量管理级

6.1	特征及包含的关键过程域	108
6.2	定量过程管理	108
6.3	软件质量管理	114

第7章 优化级

7.1	特征及包含的关键过程域	120
7.2	缺陷预防	120
7.3	技术更改管理	125
7.4	过程更改管理	131

第8章 军用软件过程改进参考模型

8.1	概述	137
8.2	启动阶段	138
8.3	诊断阶段	147
8.4	建立阶段	151
8.5	行动阶段	160
8.6	提高阶段	167

第9章 军用软件过程评估方法

9.1	概述	173
9.2	软件过程评估方法的最低要求	174
9.3	选择软件过程评估的参与者	176
9.4	软件过程评估活动	179
9.5	时间表和资源需求	181
9.6	后续活动	183

第10章 军用软件能力评价方法

10.1	概述	184
10.2	软件能力评价的目标与目的	184
10.3	软件能力评价方法的数据收集模型	185
10.4	软件能力评价过程	187
10.5	软件能力评价活动	189
	参考文献	204

1.1 军用软件能力成熟度模型的由来

1.1.1 软件面临的严峻问题

随着计算机科学与技术的飞速发展,应用领域的不断扩大,对各种软件的需求量急剧增加。软件不再只是计算机硬件的附属品,而已成为系统中的一个独立部分,甚至是主宰系统功能的关键部分。关于软件在现代战争中(特别是在“沙漠风暴”之后)的作用,美国国防部长威廉·S·科恩曾经指出:“信息技术为战斗机提供了军事历史上前所未有的信息广度和深度。利用这些信息提高精确打击武器的指挥控制,将为美军提供前所未有的能力。”^[1]在1994年,我国航天界就已意识到软件的重大作用,认识到软件是武器装备系统的神经中枢,是武器装备系统的威力倍增器^[2]。

但是,“在规模上,软件实体可能比任何由人类创造的其它实体要复杂,因为没有任何两个软件部分是相同的……数字计算机本身就比人类建造的大多数东西复杂。计算机拥有大量的状态,这使得构思、描述和测试都非常困难。软件系统的状态又比计算机系统状态多若干个数量级。”^[3]提高软件的生产率和质量便成为软件界所面临的最大挑战。在美国国防部,负责软件研究计划的C. J. Greene将军指出,“美国国防部为未能解决软件问题受到了巨大的惩罚。惩罚不仅是软件的时间推迟,而且使作战能力大打折扣。”^[1]在我国,总装备部殷鹤龄将军在1995年指出,“目前,在我国国防科技工业领域,软件研制与开发仍基本上处于‘初始’阶段,其质量控制还缺乏有效的组织、手段和相应的规范。”^[4]

软件面临的问题依然严峻,主要表现在软件的质量形势严峻、软件的进度难以保证和开发软件的经费难以控制等方面。

软件的质量形势严峻,其主要原因如下^[6]:

1. 软件正成为许多关键系统的核心。由于计算机的使用具有提高效率、能取代人进行某些工作等优点,因此,计算机正日益广泛地应用于监视和控制复杂的、时间关键的物理过程和机械设备。在这些物理过程和机械设备中,一个错误或失效可能会造成人身伤亡、财产损失或环境危害。而在这些关键的物理过程和机械设备中,软件所起的作用非常关键,它是控制的中枢和灵魂。一旦软件因质量问题出现错误或失效,就会造成系统危险,甚至造成灾难性的后果。例如:

- 1) 法国气象卫星软件由于质量问题,当计算机应当给一些气象探测气球发出一个“读取数据”指令时,竟错误地发出了一个“紧急自毁”指令,从而毁坏了141个气

象气球中的 72 个,造成了探测任务的失败。

- 2) 1981 年 4 月 10 日,美国准备发射一枚空间回收装置,在离发射时间尚有 20 分钟时,计算机实时控制系统的软件突然发生故障,迫使发射延期进行。事前尽管花了数千小时进行测试和模拟,但仍未测出这个隐患。
- 3) 一个挂载在 F-18 战斗机机翼上的导弹在点火之后未能成功地从发射装置中分离,其原因是在导弹发动机产生足够的推力使导弹离开机翼之前,因计算机软件错误而锁住了导弹保持机制,从而使该战斗机严重失控。
- 4) 1989 年 9 月,苏联载人航天飞船由于软件问题无法启动返回火箭发动机,经修复后推迟两天才返回地面。
- 5) Bell 实验室曾对一个 AT&T 运行支持系统进行过统计,发现该系统 80% 的失效与软件有关。
- 6) 1979 年,新西兰航空公司的一架客机因为计算机控制的自动飞行系统发生故障而撞到阿尔卑斯山上,机上 257 名乘客遇难。
- 7) 1983 年,美国科罗拉多河水泛滥,由于计算机对天气形势预测有错,水库未能及时泄洪,以致造成了巨大的损失。
- 8) 在海湾战争期间,“爱国者”防空系统有一次未能成功地拦截“飞毛腿”导弹,造成军营被炸,28 名英军死亡,原因是其跟踪软件在运行 100 小时后出现了一个 0.36 秒的舍入误差。

2. 软件是由人开发的,人不可避免地会犯错误。而人所犯的错误会造成软件存在缺陷,这些缺陷一旦在系统运行中暴露,就会导致系统出错或者发生故障。

3. 多数软件是由没有容错能力的机器执行的。计算机从不考虑在其上运行的软件是否存在错误,只是按部就班地执行它的命令,而不管这些命令是不是安全关键的。

4. 在当前的软件开发和维护中,主要考虑的因素是费用和进度,而不是可靠性。美国“阿波罗”宇航员 Gus Grissom 曾经指出:“每当我想到所有的火箭和宇航员舱都是由要价最低的投标者制造的这一事实时,就使我思索再三。”

5. 对软件的测试是有限的,而对于一个复杂的软件系统来说,其路径状态相对来说是无限的,因此,不能保证 100% 地剔除软件中的缺陷。

软件的进度难以保证。软件项目的管理者们普遍认为,按时交付软件项目是他们最大的挑战之一。有关研究表明,失败的软件项目平均超出时间大约是原始估计的 222%^[7]。这意味着计划 100 天完成的软件项目最后需要 222 天才能完成。影响软件项目进度的主要原因有:软件项目管理缺乏严密的计划或者严格的日程安排;对可能的瓶颈问题缺乏分析并未给予相应的资源投入;软件项目开发队伍不稳定,以致许多工作不得不从头做起;软件项目资源调度或重要决策不及时,以致造成工作停顿等待;对软件项目中间环节的质量控制不严,以致因为出错而造成返工;软件项目的需求中途改变,造成大量的设计返工;软件项目的管理者未能发现进度滞后,未能采取纠正措施;事先无法预计的人力或技术困难;因项目组内成员间的交流不够而导致返工等等^{[2][8]}。

开发软件的经费难以控制。软件项目造价昂贵,常常超出预算。有关研究表明,信息系统项目的实际成本一般为原始估算成本的 189%^[7]。这意味着一个计划用 100000 元完成的软件项目,在软件项目结束时一般需要花费 189000 元。造成软件项目进度拖延的原因通

常也是造成软件项目成本超支的原因。除此之外,造成软件项目成本超支的原因还有:由于软件项目的创新性,其经费的初始估计偏低;由于软件项目承建单位的内部管理障碍,造成开发资源不能共享和有效使用,不得不重复购置;由于项目经费未按项目工作分解结构加以权衡或者目标不明,就盲目投资,以致经费分配失衡等等^[2]。

何国伟研究员在对我国武器装备软件的研制状况进行了深入的研究后指出,“我国国内许多软件还处于‘自编、自导、自演’的前工业革命生产方式的年代。使用部门反映:‘软件是不透明的,质量是不高的,可靠性是不知道的’。随着武器装备中使用软件越来越多,软件的质量及可靠性严重威胁武器装备系统的效能及战备完好性。”^[9]并认为,“我国软件目前的形势是严峻的。”^[9]

1.1.2 解决软件问题的出路

软件作为一种逻辑实体,具有抽象性、严密性、“一次性”、智力性、持久性、依赖性、复杂性、难以度量、易出错、必须维护和成本昂贵等显著特点^[10]。由于软件的上述特点,因而对许多组织来说,在规定的的时间和预算内开发出符合质量要求的软件是一件非常困难的事情。

为了解决上述问题,自从1968年在NATO会议上提出软件工程以来,人们一直在寻求更先进的软件开发方法和技术。每当出现一种先进的技术和方法时,就会使得上述问题得到一定的缓解。例如,在20世纪70年代提出了被Sami Zahran称为“软件行业的第一次浪潮”的结构化开发方法,它对缓解上述问题起到了一定的作用,是软件开发从“手工作坊”向大规模生产进行转换的开端,但还远远不够^[11]。在20世纪70年代中期,美国国防部的统计数据表明,在失败的软件项目中,70%的项目是由于管理不善造成的。此外,尽管在20世纪80年代提出的面向对象方法很好地解决了软件开发的完备性和软件代码的重用性问题,但仍未从根本上解决上述软件问题。因而Zahran指出,“软件界使用计算机辅助软件工程工具的经验已经证明,导致软件项目失败的主要原因几乎与技术 and 工具没有任何关系,更多的是缺乏过程规范。”^[11]从而可以得出结论:如果仅从技术的角度来解决上述问题,其效果将十分有限。

30多年来,人们试图采用新的方法和技术来解决上述问题,但结果却令人无法满意,这种现象促使人们进一步考察软件过程,从而发现,问题的关键在于对软件过程的管理不尽人意。事实表明,在无规则和混乱的管理条件下,先进的技术和工具并不能发挥应有的作用。正如Humphrey指出的那样,这就如同“在同一支球队中,一些人在踢足球,一些人在打棒球,而其它人却在打橄榄球。在这样的环境下,即使拥有最好的球员,也只能组成一支最差的球队。”^[11]人们逐步认识到,改进软件过程的管理是解决上述问题的出路。

1986年11月,卡内基-梅隆大学的软件工程研究所应美国联邦政府的要求,在Mitre公司的协助下,分别于1987年6月和9月开发出了软件能力成熟度框架和软件成熟度提问单,用于评估软件供应商的能力。通过4年的应用实践,软件工程研究所于1991年推出了软件能力成熟度模型1.0版。1994年4月,软件工程研究所举行了一个由400多名软件专业人员参加的CMM研讨会,与会代表针对软件能力成熟度模型1.0版进行了深入的讨论,提出了改进意见。软件工程研究所在广泛地征求意见的基础上,于1993年推出了软件能力成熟度模型1.1版。在经过软件能力成熟度模型1.1版大量使用和征求意见的基础上,软件工程研究所于1997年发布了软件能力成熟度模型2.0版。尽管这个版本曾有草稿A

版、B版和C版,但始终没有推广,并且在开展集成能力成熟度模型的工作时,就停止了软件能力成熟度模型2.0版的进一步工作。

软件能力成熟度模型的主要研究人员 Paulk 在为 Zahran 著作《软件过程改进——获得商业成功的实用指南》撰写的序言中指出:“如果我们打算克服软件危机的话,方法之一就是采用本书所阐述的内容——软件过程改进。在为 Watt Humphrey 的《管理软件过程》一书所写的前言中, Peter Freeman 写道,‘软件危机已经消亡了!’而 Humphrey 的这本书正是这种转变的最好标志。经过了8年的时间,现在软件组织在规定的时间与预算内能够开发出高质量软件产品的能力在不断提高,而这正说明了 Freeman 当年提出的观点的正确性——至少在系统地采用软件过程改进方法的软件组织中是正确的。”^[11] 实践的结果表明,过程改进通常可以导致更短的上市时间、更高的质量和客户更为满意,过程改进的投资收益率(ROD)大约在4.5:1至7.5:1之间^[12]。对各类软件研制单位交付的软件进行的统计表明,初始级研制单位交付的软件平均每千行源代码的缺陷数为11.95个,可重复级研制单位交付的软件平均每千行源代码的缺陷数为5.52个,已定义级研制单位交付的软件平均每千行源代码的缺陷数为2.39个,定量管理级研制单位交付的软件平均每千行源代码的缺陷数为0.42个,优化级研制单位交付的软件平均每千行源代码的缺陷数为0.32个^[6]。

何国伟研究员在研究了我国装备软件的现状和硬件的工业化生产经验后指出,“解决我国装备软件危机的出路是软件生产的工业化,即建立现代化的软件工厂,按照软件工程要求开发软件。在软件工厂内按ISO 9000建立软件质量体系,严格进行软件质量管理。加强软件可靠性设计、管理、测试及验证,不断改进软件工程过程,提高软件工程能力。”^[9]

1.1.3 军用软件能力成熟度模型的诞生

航天系统是我国最早研究软件能力成熟度模型的单位,早在20世纪90年代初期,为了保证载人航天工程等型号软件的质量,原航天工业总公司从技术和管理两个方面入手进行了系统的研究。在技术上,提出采用一些新的方法(如容错设计、避错设计等)和工具(如软件开发工具、测试工具等)进行软件开发,以提高软件的内在质量;在管理上,提出了“以软件测试促进软件工程化”的思路,建立了以航天软件评测中心、软件检测站、软件开发项目组内的软件测试人员组成的3级软件评测体系,制定并采用了以GJB/Z 102—1997《软件可靠性和安全性设计准则》^[13]为代表的一系列标准和规范,并开展了对软件能力成熟度模型的研究,力图借助该模型来提高软件研制单位的过程能力,改善软件产品的质量。

1995年9月,国防科工委质量与可靠性研究中心出版了蔡愉祖研究员翻译的《评价承包商软件过程能力的方法及实践》一书,该书由“评估承包商软件工程能力的一种方法”、“软件能力成熟度模型”、“能力成熟度模型的关键实践”和“休斯飞机公司的软件过程改进”4篇文章组成。总装备部殷鹤龄将军在该书的序言中指出,“目前,在我国国防科技工业领域,软件研制与开发仍基本上处于‘初始’阶段,其质量控制还缺乏有效的组织、手段和相应的规范,正如资料中所描述的那样,属于‘不成熟的组织’在‘低的质量保证能力水平上’。在这种情况下,首先要解决的主要问题是管理问题,而不是技术问题,这一点更值得引起重视。”^[4]

此后于1997年1月,国防科工委质量与可靠性研究中心又出版了蔡愉祖研究员翻译的《评价承包商软件过程能力的方法及实践(1.1版)》一书,该书由“软件能力成熟度模型1.1版”、“能力成熟度模型的关键实践1.1版”和“软件过程成熟度提问单”3篇文章组成^[5]。

此后,由何新贵院士领导的载人航天软件工程专家组对软件能力成熟度模型进行了深入的研究,并结合载人航天工程的实践,提出了由初始级、基本级、可重复级、已定义级、定量管理级和优化级组成的软件能力成熟度模型 CSCMM,并于 2000 年 11 月在清华大学出版社出版了《软件能力成熟度模型》一书^[14]。

2000 年 6 月,国务院下发了《鼓励软件产业和集成电路产业发展的若干政策》(国发[2000]18 号)文件,其中明确指出国家鼓励软件出口型企业按 CMM 改进软件过程,并将对其给予资金支持。文件精神在各地方产业主管部门迅速落实,同时在软件企业中引起了强烈的反响。截至 2003 年 3 月,全国共有近 50 家软件企业通过 CMM 认证,其中通过 2 级的 32 家,3 级 9 家,4 级 2 家,5 级的 4 家。

2001 年中国人民解放军总装备部电子信息基础部技术基础局下达了制定军用标准《军用软件能力成熟度模型》的课题,随后成立了以何新贵院士为编制组长的标准编制组,在广泛征求意见的基础上经过送审稿和报批稿等几个版本的修改,于 2003 年正式颁布了 GJB 5000—2003《军用软件能力成熟度模型》^[15]。

2003 年 9 月,中国人民解放军总装备部批准成立了以何新贵院士为组长的总装备部军用软件能力评价专家组,负责对军用软件承制单位实施军用软件能力评价,参加军用软件能力评价政策、程序和方法研究,并对军用软件能力评价工作提供技术咨询和技术把关。

1.2 军用软件能力成熟度模型的编制思路及应用设想

1.2.1 编制思路

为了解决军用软件研制过程中存在的问题,促使军用软件承制单位尽快提高其软件过程能力,降低军用软件的风险,我们吸取国外的成功经验,对 CMM 进行了理论研究和工程应用,并在此基础上制定了 GJB 5000—2003《军用软件能力成熟度模型》^[15]。

GJB 5000—2003《军用软件能力成熟度模型》^[15]是参照美国软件工程研究所 1993 年发表的软件能力成熟度模型(SW-CMM)1.1 版制定的,规定了军方用于评估组织的软件能力成熟度的模型,并规定了软件整个生存周期内的主要软件管理过程和工程过程,适用于第二方或第三方对组织的软件能力进行评价,也适用于组织本身对软件过程进行评估和改进。

由于我国军工企业既承担着武器装备软件的研制,又承担着国家经济建设项目软件的开发,在制定本标准时,重点考虑了军工企业与国际接轨的需要,并保持与 2000 年 6 月国务院下发的《鼓励软件产业和集成电路产业发展的若干政策》的精神相一致,本标准实质内容(第 5 章至第 8 章)等同采用 SW-CMM 1.1 版中关于各等级的关键过程域的描述。GJB 5000—2003 与 SW-CMM 1.1 版主要区别如下:

1. SW-CMM 1.1 版由“软件能力成熟度模型 1.1 版”、“能力成熟度模型的关键实践 1.1 版”和“软件过程成熟度提问单”3 篇研究报告组成^{[16][17][18]};GJB 5000—2003《军用软件能力成熟度模型》共有 8 章 2 个附录。

2. SW-CMM 1.1 版由三个研究报告组成,其编写格式不同于标准的格式,其叙述为陈述语句;GJB 5000—2003《军用软件能力成熟度模型》的编写格式按照 GJB 0.1—2001《军用标准文件编制工作导则 第 1 部分:军用标准和指导性技术文件编写规定》^[19]的要求进行了规范,对 GJB 5000—2003 在格式方面的特殊要求在引言中进行了格式约定。

3. 在 SW-CMM 1.1 中,存在大量的说明性信息,如“致读者”、“如何获得更多信息”、“CMM 的未来发展方向”和“参考文献”等等;在 GJB 5000—2003《军用软件能力成熟度模型》,根据我国军用标准的编写习惯进行了处理,删除了这些说明性信息。

4. 在 GJB 5000—2003《军用软件能力成熟度模型》中,考虑到我国军用软件开发单位的实际情况,为了更顺利地采纳 CMM 理念,减少实施中的困难,真正提高软件过程能力,在模型应用方面,增加了应用软件能力成熟度模型的建议。

1.2.2 内容结构

GJB 5000—2003《军用软件能力成熟度模型》共有 8 章 2 个附录,附录 A 是规范性附录“术语”,附录 B 是资料性附录“软件能力成熟度提问单”。第 4 章“概述”从软件能力成熟度模型框架、软件能力成熟度等级、关键过程域、软件能力成熟度模型的应用、应用软件能力成熟度模型的建议等 5 个方面系统地介绍软件能力成熟度模型,为了解和应用该标准提供指南;第 5 章至第 8 章分别阐述等级 2 至等级 5 中各个关键过程域的内容。

GJB 5000—2003《军用软件能力成熟度模型》共由 5 个成熟度等级、18 个关键过程域、52 个目标、28 个执行承诺类关键实践、71 个执行能力类关键实践、150 个执行活动类关键实践、20 个测量分析类关键实践和 47 个验证实施类关键实践构成,详见表 1-1。

表 1-1 GJB 5000—2003 的构成

等 级	关键过程域	目标	关 键 实 践					实 践 总 计
			承诺	能力	活动	测量	验证	
初 始 级	0	0	0	0	0	0	0	0
可重复级	6	20	9	25	62	6	19	121
已定义级	7	17	9	25	50	9	15	108
定量管理级	2	6	3	8	12	2	6	31
优 化 级	3	9	7	13	26	3	7	56
合 计	18	52	28	71	150	20	47	316

1.2.3 应用设想

军用软件能力成熟度模型主要用于软件过程改进、软件过程评估和软件能力评价。通常将软件过程评估和软件能力评价统称为估价。有关软件过程改进、软件过程评估和软件能力评价的具体方法将分别在本书第 8 章、第 9 章和第 10 章中详细论述。

为了更好地在军用软件开发组织推行军用软件能力成熟度模型,作者认为^[20]:

1. 为了保证武器装备软件的质量,军方应对军用软件开发组织的软件过程能力有明确的要求,有要求才有压力,有压力才会变成软件过程改进的动力。

2. 应加大培训力度,尤其是加强对软件开发组织领导的培训力度,让他们了解软件过程改进的好处、重要性和难度,以便为其组织的软件过程改进提供必要的资源。目前,在总装备部电子信息基础部技术基础局的支持下已经完成了培训教材的编写工作,在 8 月份已经开展了军用软件能力成熟度模型的培训工作。

3. 对一些技术条件好、愿意按照军用软件能力成熟度模型改进其软件过程的软件开发组织进行试点,在取得经验的基础上进行推广。目前,总装备部已选择了航天科技集团公司一院十二所、信息产业部十五所等四家单位作为首批试点单位,按照 GJB 5000—2003 进行软件过程改进试点。

4. 成立军方自己的软件能力评价机构,具体负责各项评价事宜。该机构应严格保持中立,不与被评价组织发生经济利益关系,以免缺乏公平公正。

5. 培训并培养自己的软件评价(估)师队伍,评价(估)师应精通军用软件能力成熟度模型,熟悉软件工程业务,熟悉软件管理业务,善于分析问题、提出问题,并具有公正、超脱和任劳任怨的职业操守。

6. 建立必要的奖励机制,对采用军用软件能力成熟度模型进行软件过程改进,并对取得成效的单位和个人进行精神和物质奖励。

7. 如果一个尚未建立基本软件过程的组织准备采用军用软件能力成熟度模型进行软件过程改进工作,其高层管理者应首先确定软件工程化的方针,即要求各软件项目彻底改变过去非工程化的软件开发方式,创造必要条件,使各软件项目按照软件工程要求实施软件开发,并考虑软件生存周期过程,将软件开发工作分为便于管理和控制的若干阶段,进行开发和管理。

8. 按照 GJB 2786—1996《武器系统软件开发》^[21]等有关标准建立基本的软件过程,为基于军用软件能力成熟度模型的软件过程改进打下基础。

9. 为了有效地执行 GJB 2786—1996《武器系统软件开发》^[21]等标准,应根据项目的实际情况对标准加以剪裁,建立起适用的基本软件过程,并注意如下问题:

- 1) 在项目中,将软件产品与硬件产品同等看待,例如列入产品配套表,按软件研制程序单独安排足够的软件开发时间、经费和其它资源;
- 2) 制定软件研制程序时应选择合适的软件生存周期模型,将软件开发过程分为可控制的若干阶段,按阶段制定计划,进行监督和跟踪;
- 3) 软件实现阶段与其前后阶段的工作应由不同人员分工合作,不宜由同一人承担;
- 4) 明确各阶段应产生的文档和验证要求;
- 5) 建立基线和版本控制规程;
- 6) 建立计划进展和质量保证方面的记录,并妥善管理。

2.1 软件及其特点

从构成软件的基本要素来看,软件是与计算机系统的操作有关的程序、规程、规则及任何与之有关的文档^[22]。软件的概念是逐渐发展起来的,在早期,软件即指计算机程序,此后将文档也包括在软件之中,再进一步发展包含了程序、规程、规则和文档的定义,并强调文档是软件的重要组成部分。

程序是“按具体要求产生的、适合计算机处理的指令序列”^[22]。程序是软件的重要组成部分,但绝不是软件的全部。

规程是“为解决某一问题而采取的动作的经过的描述”或“每次完成某一任务时要遵循的一组手工的步骤”^[22],主要描述在软件生存周期中应如何实施有关政策、规则和标准。例如,测试规程,用于描述进行软件测试时应遵循的测试步骤。

规则指软件开发人员在开发软件时应共同遵守的准则和法规。

文档指一种数据媒体及其记录的数据,它具有永久性并可以由人或机器阅读,通常仅用于描述人工可读的内容^[22]。例如,技术文件和设计文件等。

从软件的组织结构来看,软件是一个由计算机软件配置项(CSCI)、计算机软件部件(CSC)和计算机软件单元(CSU)构成的层次结构^[21]。

计算机软件配置项是为独立的配置管理而设计的、且能满足最终用户功能的一组软件。

计算机软件部件是计算机软件配置项中的一个明确的部分。它可以进一步分解成其它计算机软件部件和计算机软件单元。

计算机软件单元是计算机软件部件设计中确定的、且能单独测试的部分。

软件作为一种逻辑实体,具有下述几个显著的特点^[6]:

1. 抽象性:软件产品不是实物产品,其可见性差。
2. 严密性:软件工程是一个严密的逻辑工程。软件产品无正品和次品之分,不存在误差,误差即是错误。
3. “一次性”:任何软件的研制都是一个新的开发过程,即一次性、“创造性”的劳动,而软件的成批生产只不过是简单的复制。
4. 智力性:软件研制主要靠人的脑力劳动。软件研制的绝大部分工作都是靠人来完成的。
5. 持久性:软件产品的质量与使用时间的长短无关,即软件产品无磨损性,因此软件的

故障不能用普通产品更换零部件的方法来解决。

6. 依赖性:软件产品常常受限于具体的计算机系统,为了减少软件产品对计算机系统的依赖性,应提高软件产品的可移植性。

7. 复杂性:有人认为,计算机软件是人类能够创造的最复杂的产品之一。软件的复杂性既来自它所处理的实际问题的复杂性,又来自程序逻辑结构本身的复杂性。因此,软件技术的发展落后于人们对软件的需求,并且随着时间的推移,这种差距还在日益加大。

8. 难以度量:目前对智力劳动尚无有效的度量方法,而软件研制又是新开发的智力产业,因此就更难于对它进行度量。

9. 易出错:软件生产过程涉及一系列的“信息转移”,在有信息转移的环节都有可能发生信息转移的错误。

10. 必须维护:软件产品在交付使用后还可能经常需要经常更改,因而软件维护是软件工程的一个必不可少的阶段。

11. 成本昂贵:软件产品的研制需要投入大量的、复杂的、高强度的脑力劳动,其成本是非常高的。在国外,软件已占整个计算机系统成本的70%以上。据有关统计,美国普通民用软件的开发费用约为每行源代码10美元,而宇航级软件的开发费用约为每行源代码1000美元。

2.2 软件与硬件的异同

软件与硬件的主要不同点如表2-1所示^[10]。总的说来,软件与硬件的本质区别是智力产品与物理产品之间的区别。

表 2-1 软件与硬件的差异

软 件	硬 件
逻辑实体,始终不会自然变化,只是其载体可变	物理实体,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、磨损以至失效
其接口是不可见的	其接口是可见的
尚没有标准件	有标准的零部件
研制过程主要是脑力劳动过程,在本质上是无形的,不可见,难以控制	研制过程不只是脑力劳动过程,还有体力劳动过程,其过程有形,便于测控
其不可靠问题基本是由于开发过程中的人为差错所造成的缺陷而引起的	硬件失效通常是由其零部件或其结合的故障所引起的
程序是指令序列,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,不一定完全正确	其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
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

表 2-1(续)

软 件	硬 件
对生产过程进行严格的控制,可得到完全一致的产品	对生产过程进行严格的控制,可将产品的容差控制在可接受的范围
需在研制的全过程采取措施防错、检错、纠错和容错	除了研制过程外,生产过程对产品的影响也很大,均需加强控制
精心设计测试实例,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,进行筛选,排除缺陷
冗余设计应确保冗余软件相异,否则,不仅不能提高可靠性,反而会增加复杂性,降低可靠性	相同部件之间自然是独立的,适当的冗余可以提高可靠性
在使用过程中出现故障后,必须修改原产品以解决问题;若在修改时未引入新问题,其可靠性就会增长	在使用过程中出现故障后,一般只需更换或修复失效的部件,使产品恢复良好状态,其可靠性一般不会提高
软件维护通常涉及软件更改,软件更改通常会对其它部分造成影响	硬件维修通常涉及零部件更换,零部件更换一般不会对其它部分造成影响
维护过程复杂,通常需要专业软件人员进行	维护过程相对简单,通过一定培训的人员即可胜任
本身没有危险,但会对系统的安全性造成影响,应从系统的角度考虑软件的安全性	本身可能有危险,可以也必须单独对安全关键部件进行分析和评估
某处的修改会影响它处,维护时必须考虑这种影响	维修某处一般不会影响它处
失效率随故障的排除而下降,如图 2-1 所示	失效率的变化类似于浴盆曲线,如图 2-2 所示
可靠性参数估计无物理基础	可靠性参数估计有物理基础

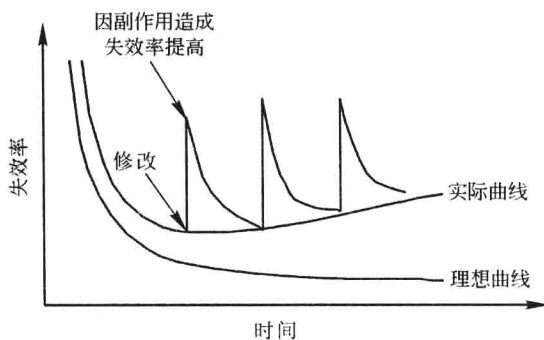


图 2-1 软件的失效率曲线

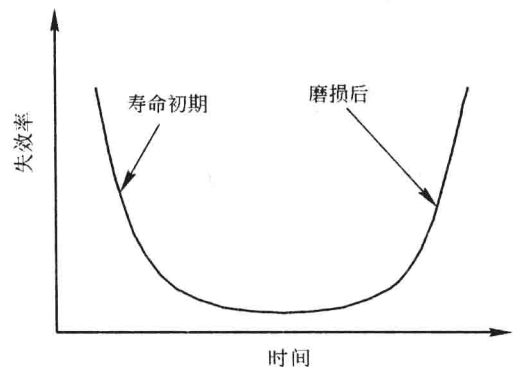


图 2-2 硬件的失效率曲线