

软件工作的 科学管理

[美] C.L. 麦克卢尔 著 谢铭培 译 招兆铿 校

MANAGING
SOFTWARE
DEVELOPMENT
& MAINTENANCE

复旦大学出版社

15
4/1

TP311.5
MKL/1

软件工作的科学管理

(美) C·L·麦克卢尔 著

谢铭培 译 招兆铿 校

复旦大学出版社

Carma L. McClure
MANAGING SOFTWARE
DEVELOPMENT AND MAINTENANCE
Van Nostrand Reinhold, 1981

软件工作的科学管理

[美]C.L.麦克卢尔 著

谢铭培 译

复旦大学出版社出版

(上海国权路579号)

新华书店上海发行所发行 复旦大学印刷厂印刷

开本850×1168 1/32 印张5.625 插页0 字数160,000

1988年2月第1版 1988年2月第1次印刷

印数1—7,000

ISBN7—309—00062—5/O·011

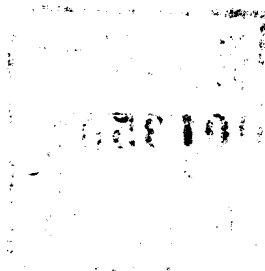
定价：1.15元

内 容 简 介

本书借助于一些成功和失败的实例,详细介绍了在软件开发和维护工作中如何组织课题组,以及在要求分析、形成说明书、设计、编程、测试和维护等软件生存期的各个阶段中如何进行科学管理的规范和方法,同时还阐述了对软件生存期的评价等问题。

本书可供从事软件开发和维护的专业人员、软件管理人员以及计算机的管理干部和用户参考,也可作为计算机和科学管理等专业的师生以及科技管理干部培训班的教材或参考书。

J5330/12



译者的话

随着国民经济的发展,我国计算机事业的形势如火如荼。计算机在各领域中的应用,目前正朝着纵深开拓。广大软件人员已经或正在着手开发许多应用软件或软件系统,形势喜人。

然而,软件或软件系统的质量如何?它能否为其今后的维护(包括进一步开发)工作提供良好的条件?软件的生存期多长?此类问题,不仅取决于软件开发人员和维护人员的技术水平,而且还取决于软件工作的管理水平。即便开发人员的技术水平较高,但若管理不善,那就仍然会使软件的质量大大降低,同时也会给维护工作造成很大困难,甚至不可能进行维护。国外在这些方面有许多经验教训。他山之石,可以攻玉;前事不忘,后事之师。对于国际上六十年代以来人们感受到的“软件危机”,我们应有所认识,并引以为戒。

本书重点阐述软件开发与维护的管理方法。全书由四部分组成。第一部分列举了软件开发中成功和失败的实例,阐明了对软件开发工作进行管理的重要性。第二部分介绍软件开发的管理方法。作者强调在开发软件时就要注意到维护,并阐述了组织好软件开发课题的方法,还介绍了如何对软件开发工作进行管理的方法。第三部分着重介绍软件维护工作的管理方法。第四部分阐述了对软件生存期的评价问题。

本书可供从事软件开发和维护工作的专业人员、软件管理人员以及与计算机应用有关的管理干部和用户参考,也可作为计算机和科学管理等专业的师生以及科技管理干部培训班学员的教材或参考书。

限于译者水平,难免有不妥之处,敬请读者指正。

周仲良同志对本书的翻译工作曾给予大力帮助,在此深表谢意。

1987年5月

序 言

本书可供从事软件系统开发和维护工作的专业人员使用, 还可供软件工作的管理人员和学习软件工程学的学生使用。

本书是根据伊利诺工学院(Illinois Institute of Technology)为研究生开设的软件工程学这门课程的教材改写而成的。这门课程的主要目标之一是使学习软件工程学的学生认识到当今软件产业中的一些问题。

本书与其他软件工程学书籍比较起来有两点不同: 第一, 本书着重于软件实际工作人员面临的管理问题, 而不是技术问题; 第二, 本书着重于软件的维护工作, 而不是软件的开发工作。

虽然人们对软件危机已在六十年代就有所认识, 但是它至今仍在阻碍着软件技术的进步。多年来, 硬件的价格一直在大幅度地下跌, 与此同时, 软件的价格却不断地上涨。以前人们认为只要软件产品的质量有所提高, 那么软件产品的价格就能得到控制, 因而为解决软件工作中存在问题的办法大多停留在改善软件开发技术方面。这种看法没有认识到既要更改软件又要维持软件质量会遇到多大的困难。

所有软件都会遇到更改的问题。为了纠正错误、改善性能、适应应用环境的变化以及满足不断变化的用户要求(也许这是最重要的), 就要更改软件。即使完全能满足用户原先要求的一个绝对可靠的软件系统, 在其整个生存期内也很可能要作多次的更改。

大多数软件在设计时并没有考虑到适应软件的更改, 也没有要求软件工程技术人員预先采取某种措施来适应软件可能发生的变更, 这两者的后果是: 软件一旦有所更改, 它的质量也就下降。在维护阶段, 软件质量下降的问题就更突出了。

如果将软件生存期的各个阶段加以比较的话, 那么现今大部分精

力和费用都耗散在维护阶段上。事实上,在许多组织机构中,为维护现有软件系统所花费的时间比开发新的软件系统所花费的还要多。

如果软件工程技术人员和管理人员继续将软件看作为一种抽象的结构,还一直从形式上而不是从实用性方面来评价软件的质量,那么软件的费用还会大幅度增加。今后需要维护的程序愈来愈多,新程序的规模愈来愈大而且其复杂性愈来愈高,这些势必使软件的维护问题更加突出。最终,必然会将程序设计方面的人力物力几乎全部投入到软件的维护工作上,从而就无力去开发新的软件了。

综上所述,人们对于所开发和维护的软件,不仅要求其数量更多,而且要求其质量更高。本书的宗旨是介绍一些具体的方法来满足人们这种日益增长的需要。

本书之所以要提出软件工作中的科学管理问题,目的在于制订一些管理制度,使之能用来较好地检验和协调软件工作,并改善用户和软件技术人员之间信息交流的途径,从而提高软件的可见性。本书提出的有关管理的指导思想不是要象现在许多软件工作中经常出现的那样去“救火”。我们认为,满足用户的眼前利益或者满足软件技术人员试验最新技术的愿望,并不能作为制作一个软件系统或者对某个软件系统进行更改的主要依据。相反地,应把重点放在软件的期望寿命这个长远目标上。而让用户一时满意或者使软件得到暂时改善的做法,却可能有损于这个长远目标,这不是我们所求的。

C·L·麦克卢尔

目 录

译者的话	1
序言	2

第一部分 引 论

第一章 软件技术的现状	1
第二章 软件开发的实例	7
2.1 引言	7
2.2 失败的实例	8
2.3 成功的实例	13
2.4 GIRG 课题和 SMCS 课题的比较	21
2.5 小结	24

第二部分 软件开发的管理

第三章 开发软件时就要注意到维护	26
3.1 引言	26
3.2 软件开发的原理	28
3.3 软件开发进程	33
3.4 小结	61
第四章 组织软件开发课题组	63
4.1 引言	63
4.2 课题组的组织机构	65
4.3 改进的主程序设计员组	73
4.4 小结	81
第五章 软件开发工作的管理	83

6.1	引言	83
5.2	课题分工	84
5.3	课题的管理机构	88
5.4	变更的控制	112
5.5	小结	116

第三部分 软件维护工作的管理

第六章	软件工程学用于软件维护	118
6.1	引言	118
6.2	维护	120
6.3	维护的实施	121
6.4	小结	140
第七章	软件维护工作的管理	143
7.1	引言	143
7.2	维护课题组	144
7.3	软件变更的控制	153
7.4	小结	159

第四部分 结 论

第八章	软件生存期的评价	161
8.1	软件课题的研究	161
8.2	对软件产品的评价	163
8.3	对软件生存期进程的评价	164
索引		166

第一部分 引 论

第一章 软件技术的现状

我们现今所面临的程序设计问题与六十年代所遇到的以及八十年代可以预料的问题都极为相似。大多数软件项目失败了。其进度和费用往往比预先估计的要糟；很少软件能满足用户的期望或要求；已成产品的软件系统几乎没有不出毛病的。

在六十年代中，我们开始认识到制作和维护软件的困难，后来就把这个时期称为软件危机时代。从那时以来，人们在研究软件的原理和方法方面做了大量的工作，其中大多数是针对软件生产技术的。为了要研究软件的开发和维护问题，软件工程学就应运而生了。

对程序设计本身的看法的演变，也许是至今所取得的最大收获。在五十年代，人们认为程序设计是一项简单的工作，只是将计算过程编成程序而已；在六十年代，人们将程序设计看作需要依靠个人劳动才能完成的困难工作；然而，现今人们已将程序设计看作为复杂的、需要一个集体来协同工作和解决问题的学科。结构化程序设计的提出，象征着我们在看法上的转变。结构化程序设计到底是一种简单的概念呢，还是各种方法之汇总？在这个问题上，目前虽然还有争议，但我们同意这样一种看法，即这种争议使人们注意到有必要使程序设计规范化。

当结构化程序设计在七十年代早期刚进入软件产业时，人们将它定义为一组编程规范，其职能只是把源码的形式标准化而已。而后又将结构化程序设计描述为对一个分层有序的软件系统进行设计、编程和测试的自顶向下、逐步求精的方法。如今，结构化程序设计则是指一种科学的方法。它要求在软件生存期的每个阶段以及在软件开发课题组的组织中都要有一定的规范。

对结构化程序设计的定义在认识上的演变表明,人们已开始日益广泛地关注软件课题的管理问题和技术问题。他们不仅把眼光局限在软件的开发阶段,而且也关心软件在整个生存期内的情况。

在过去的十年中,我们已研究了软件工程学的许多方面。为了开发软件系统,已创立了一些原理和方法。我们发现,在许多课题中,遵循这些基本原理,课题就能按计划完成,还提高了软件的正确性和质量,使课题容易成功。例如,IBM 的研究报告表明:应用结构化程序设计,就可在实时、商用和系统软件等方面的课题中平均节省百分之四十的人力和物力。

然而在有些课题中,我们却发现软件工程学的原理未必能确保课题成功。我们经常看到如下的情况:

- 如果对编程标准不从管理手段上加以严格监督实施的话,单有编程标准也是不能提高程序质量的。
- 自顶向下设计是一个多义性的一般概念,难以在实践中妥善应用。
- 主程序设计员组的概念对于许多数据处理组织并不见得适用,因为主程序设计员责任重大,又需具备较高的技术水平,而能肩负此职的合格人员却寥寥无几。
- 企图只采用结构化程序设计而不搞好课题管理的话,课题不一定能成功。
- 程序设计人员的态度在很大程度上影响到课题的成败。
- 因为管理部门不知道将程序设计人员当作技术工人还是当作工程师看待,所以软件课题常常失控。
- 软件工程学没有论及用户的需求或软件维护的后援(即给软件的维护工作以支持的程序)。这就有不足之处。
- 过去搞过的软件课题所提供的资料,能够对现在的研究工作有所帮助的,为数不多。

来自软件产业的统计表明,关于软件工程学的研究,还有许多工作要做。由于绝大部分的精力和费用都耗费在软件开发之后和在软件维

护期间,因此软件的费用(包括人力、物力和经费等)一直在增加。在过去的十年中,程序设计人员的生产率增加了一个数量级,硬件的费用则下降了两个多数量级。现今软件的费用已占自动化费用的百分之七十。在1973年,美国的软件费用是二百亿美元,1985年的规划是二千亿美元。

工业方面的调查报告指出,在典型的工业应用中,如果将整个系统的各个指标分别作为一百的话,那么在软件维护期间所花的费用占总费用的七十,而劳力占总劳力的四十至九十五。来自美国国防研究部(Department of Defense Study)近期对某个惊人实例(但愿不是普遍的)所作的报道说,空军的航空电子控制系统中开发软件的费用是平均每条指令七十五美元,而该软件的维护费用却是每条指令四千美元左右。

大多数软件工程学的方法和工具都只涉及到新软件的开发技术,而忽视了另外一方面。其实,也应当研制一些软件,供软件课题的管理工作或者现有系统的维护工作作工具。但是这类软件工具却相当缺少。过去的课题和目前在费用分配上的趋势都表明,这些领域都还没有引起人们的重视,这一情况在今后应加以改变。事实上,1978年对制造业和非制造业的数据处理管理人员进行了一次调查,结果表明,管理问题比技术问题更重要,而且维护现有的系统比开发新的课题更重要。

传统的做法是将数据处理部门与整个公司分割开来。因为数据处理并不为管理部门所熟悉,所以它并不像其他部门那样受到应有的关心和重视。连数据处理人员也受到另眼看待。数据处理管理人员中受过正规管理训练的人数很少,他们对整个公司的目标漠不关心,也很少有要与整个公司打成一片的想法。数据处理管理人员中调入公司其他部门的为数不多,晋升到非技术性的高一级管理职位上的人员就更少了。然而,数据处理人员及其管理者可以经常在各公司的数据处理部门间流动。根据最近的调查,一个程序设计人员的平均任职时间是三年。这么频繁的人员调动对于保持软件开发和维护工作的连续性非常不利。由于与整个公司缺乏联系,因此数据处理人员对用户的要求

(这恰恰是本公司的要求)无动于衷,似乎用户根本就不存在一样。这么一来,数据处理人员只是将软件看作为抽象的结构,而不将它看作为满足用户需要的一个软件系统。因而就会从其形式上而不会从其实用性方面来考察软件质量的优劣。君不见人们不是在过分地强调结构化程序设计和软件工程学的技术问题吗?这就再好不过地看出其后果如何了!

人们习惯上将某个软件系统的维护工作与该系统的开发工作截然分开。通常,都是安排有经验的程序设计人员去承担开发新软件系统的工作,而又安排另一批数据处理人员去承担该系统的维护工作。这些维护人员中没有一个是该系统的开发课题组成员,对原先的开发工作不了解,他们的程序设计经验也比不上开发人员。这两部分人员之间平时又互不通气,一直到课题移交使用时才碰面。在数据处理界,很多人看不起维护工作,认为干维护工作没有出息,学不到最新的技术,甚至认为软件工程学不适用于维护工作。但是,需维护的复杂系统愈来愈多,而又很少采用先进的软件工具,这样势必使维护问题更加突出而且使维护费用日益增多。

本书的宗旨是将软件工程学的基本原理用于现有软件系统的维护工作。在课堂上和学术会议上,当讨论到软件生存期时经常有人提到在开发软件时就要注意到维护问题的方针。这无疑是正确的。但它实际上意味着什么?系统开发人员和系统维护人员怎样在实践中实现这些方针?我们用以下几点来回答这些问题。

- 将在新系统的开发中用得成功的规范和措施也用于软件的维护后援上。
- 在软件开发进程中,就要着手解决软件维护后援的问题。
- 在软件开发人员、维护人员与用户之间明确通信的途径,使这三部分人员在开发过程中以什么方式来相互交流信息以及交流哪些信息等,在管理体制上确定下来。

本章末尾列出了软件开发的一般规范。该表罗列了三个类别:技术规范、产品规范和课题规范。后面几章将讨论这些规范,并将它们引

用于维护环境。已经有许多书籍专门介绍软件工程学的技术，所以我们的重点就放在第二类和第三类规范上。

本章的第二部分将回顾软件工程学和软件生存期的基本原理，以供讨论维护环境时应用。还有，我们将对软件生存期的概念加以推广，把软件后援所需的准备工作也包括在软件生存期中。我们将不把软件的开发和维护看作截然分开的两件事，因为它们具有共同的原理和技术，并且在人员方面也可互相交换。

在第三部分中，我们将介绍软件维护工作，把重点放在软件维护的管理方面。为履行维护的职能，维护管理人员应做些什么准备工作？怎样筹建维护课题组以及如何与用户经常取得联系？在这部分中会提到这些内容。

最后，第四部分将讨论软件生存期的最后阶段，以及根据在课题开始阶段就确定的目标来评价课题和产品的成果。

软件开发的规范

1. 技术规范

- 1.1 应用自动化的软件工具。例如，性能良好的操作系统，预处理程序，程序包，相互对照引用的生成程序，文献的生成程序，测试的生成程序，资源管理系统，等等。
- 1.2 应用结构化程序设计。
- 1.3 应用自顶向下和自底向上两者混合的设计方法。
- 1.4 在设计时就要制订一个测试规划。
- 1.5 应用较高级的语言。
- 1.6 尽量做到程序与语言标准一致，而且与所用机器的具体型号无关。
- 1.7 选用性能良好的机器，更要挑选精干的“人”。
- 1.8 保留公用模块区。
- 1.9 研究已成功的软件工程学技术并加以采纳。

2. 产品规范

- 2.1 在课题的初始阶段,就要将系统分为确定要求、制订测试计划以及里程碑(即具有阶段性象征的必做的重要内容)等几部分。
- 2.2 设计时就应考虑到如何应付非标准输入。
- 2.3 设计时就应注意到维护问题。
- 2.4 力争简化而不求十全十美。
- 2.5 定期地审查课题,对原先公认的论证手续以及必须检查的控制点加以实施。
- 2.6 让用户也参加评审。
- 2.7 在课题的早期就形成用户文献。

3. 课题规范

- 3.1 制订课题计划并以此来管理这个课题。
- 3.2 明确地确定课题的目标和优先次序。
- 3.3 形成简明扼要的用户要求陈述书,并定期与用户一起检查,而且要愈来愈细致。
- 3.4 参加开发课题组的人员要精干。
- 3.5 做到每个课题组成员的职责分明,使每个人易于考核自己的工作。
- 3.6 确定每个人的业务方向、工资级别和福利待遇,以奖励工作成绩突出的人。
- 3.7 不要因使用结构化程序设计而取消良好的管理。
- 3.8 为课题的管理工作建立程序库。
- 3.9 对照课题的目标来评价课题是否成功。

第二章 软件开发的实例

2.1 引言

为什么某些软件课题成功了,而另一些却失败了呢?我们对软件产业中许多失败的软件进行了总结,发现它们共同的教训主要在于管理上的问题,技术方面的问题还是次要的。

课题要获得成功,应考虑三个基本要素:

1. 技术工具; 2. 技术专长; 3. 管理技术。

这三个要素中任何一个都不可忽视,否则很可能导致课题失败。而过去我们的注意力大部分集中在前两个要素上。引入了结构化程序设计、设计技术和自动测试工具等等之后,在技术领域曾取得重大的进展。但另一方面,却不大注意对课题进行妥善的管理。特别是对很大的课题,他们没有能力管理;而对小的课题却又不愿意在课题管理上花点精力。

在本章中,我们将用软件产业中软件开发课题的两个实例,来阐明忽视必要的课题管理而带来的危险性。在第一个课题中,缺乏行之有效的技术工具和管理技术;在第二个课题中,由于遵循了软件工程学的原理,因而取得了成功。将两个课题加以比较之后可以看出,虽然两个课题组都赞成使用软件工程学,但他们在管理方法上不同,以致于一个课题成功,而另一个失败。

首先,我们将讨论这两个课题,并探讨一下它们失败与成功的大致原因;然后用第一章中指出的软件开发的一般规范来对照这些课题;最后,我们要再次强调,不管软件课题的规模如何,也不管人们对它们是否熟悉,应用完善的软件工程学总是至关重要的。

2.2 失败的实例

2.2.1 IFMS-I 报表形成程序课题的历史*

IFMS-I 是用来建立财务计划模型的一个交互系统。它给非内行的用户提供一组对答式的交互命令。用户可利用这些命令方便地输入数据、随时处理各种改变了的数据并制作报表。IFMS-I 装配在 XDS-940 计算机中。这个系统是在七十年代早期开发的, 原先用 FORTRAN II 的 XDS 方言书写。

IFMS-I 报表形成程序课题 (GIRG) 的目标是在 IFMS-I 中增加一个报表形成程序子系统, 以增强其功能, 从而形成一个增强型系统。新的报表形成程序进一步提高了 IFMS-I 用户进行报表编辑和格式编排的能力。原来的系统已使用八年, 但随着硬件和软件的发展, 其第二代的局限性愈来愈突出。管理部门要求课题组继续在该系统上开发。管理部门最关心的是:

1. 尽快开发这一增强型系统。
2. 使增强型系统与现行的 IFMS-I 系统及已有的模型兼容, 以保证用户不经过新的训练或者稍许再训练一下就会使用这一系统。

在开始研究时, 系统和程序设计组曾经预计, GIRG 课题需要 8 人月的工作量, 因为当时已有一个完成了的课题 (IFMS-II), 只要仿照它并且提高一下, 就可将 IFMS-I 翻版至 PRIME 300 小型计算机上。由于 IFMS-I 报表形成程序仿照了 IFMS-II 报表形成程序, 所以该系统的要求也就明确地定下来了。

研究结果表明, 在过时的 XDS-900 计算机上开发软件, 对 GIRG 课题是不利的。有关 IFMS-I 的系统文献很少, XDS-940 的操作系统文献甚至更加缺乏。系统和程序设计小组中谁也没有参加过 XDS-940

* 尽管本章所述课题和软件产品是真实的, 但所用名字却是虚构的。