

测算在软件开发中的实践^①

廖汗成 (江西财经大学 软件学院 江西 南昌 330013)

摘要: 软件测算是软件开发工程化管理的重要技术。但软件本身的复杂性、历史经验的缺乏、估算工具不完备性以及人为错误,导致软件项目的估算结果往往和实际情况相差甚远。为了改进测算结果,根据软件生命周期理论,结合 COCOMO II 模型,指出了软件开发模型中的关键测算点及相应可采用的测算方法,并且明确了软件测算人员应具备的素质要求及在典型的软件组织中的地位。最后通过实例,使用 USC 提供的一个免费软件测算工具对一个具体的软件项目进行测算,较好地控制开发过程。

关键词: 测算; 时点; 测算人员; 测算方法; 软件开发

Estimation in Software Development Practice

LIAO Han-Cheng

(School of Software, Jiangxi University of Finance & Economics, Nanchang 330013, China)

Abstract: Software estimation is an important prerequisite for project development management. The complexity of the software, the lack of historical experience, the incompleteness of estimating tools as well as human errors, often lead to software project estimation results far from the reality. In order to improve the estimation results, from a management point of view, the key estimation points in common software development model and the corresponding methods which can be used, are designated in light of the software life-cycle theory and COCOMO II model, and what basic essences estimator should have are proposed and their positions in a typical software organization are pointed out. Finally, an application example is given to illustrate how to apply these measuring points by using a free software tool provided by the USC.

Keywords: estimation effort; time-point; estimator; estimation method; software development

1 引言

软件项目开发测算是软件度量的基础。目前国际上已有许多软件项目估计的模型和方法。其中软件规模和进度的估算方法有: Allan Albrech 于 20 世纪 70 年代提出功能点分析(FPA: function points analysis)及其发展演化,包括: 国际功能点用户协会的 IFPUG 功能点法(IFPUG function points); 英国软件度量协会提出的 Mark II FPA 功能点法(Mark II function points); 荷兰功能点用户协会提出的 NESMA 功能点法, 以及软件度量共同协会提出的 COSMIC-FFP 方法等; 德尔菲法(Delphi technique)、特征点(feature point)、对象点(object point)、3-D 功能点(3-D function points)、Bang 度量(DeMar-

co's bang metric)、模糊逻辑(fuzzy logic)、标准构件法(standard component)等。模型包括: 由线性方程构成的 Far & Zagorski 模型(1965)和 NADC(Naval Air Development Centre)模型(1971)、使用生产率因子对分阶段软件开发的费用测算的 Kustanowitz 模型(1977)、基于诺顿/瑞利函数的 Putnam 模型(1977)和 Boehm 提出的不断演进的静态多变量 COCOMO 模型等等。常用的软件开发成本的估算方法主要有: 类比估算法、细分估算法、周期估算法、Parkinson 法、价格策略、自顶向下法、自底向上法、基于构件点的 SPECTRE 的方法、VSM (Vector Size Measure) 方法、WMC (Weighted Methods per Class) 方法等^[1-8]。M. Jorgensen 和 S.

① 收稿时间:2009-09-16;收到修改稿时间:2009-10-20

Grimstad (2008) 从心理因素的角度, 用经验数据证明, 过分乐观的软件估算及其高误差性, 在某种程度上, 是由于受到不相关或误导的信息影响。另外, Stephen H.Kan 对顾客满意项目度量要素进行了讨论^[9]; William A.Florac、Robert E.Park 和 Anita D.Carleton)对软件过程管理中的过程度量, 描述了过程管理和项目管理的关系^[10]。Barry W. Boehm 和 McCall 等人提出了软件质量度量的层次模型及度量方法。在关系到软件项目成功与否的众多与工程管理相关的因素中, 软件测算的结果直接影响了软件开发管理的决策。作为软件开发中的一个过程, 软件测算的应用散见于各种软件过程化管理得的标准体系中, 1991 年在美国 Carnegie Mellon 大学软件工程研究所(CMU/SEI)提出的 CMM/PSP/TSP, 英国的国家标准基础之上产生的 ISO 9000 质量标准体系及 2001 年 SEI 在 CMM 的基础上, 集成了软件以外其他知识领域的能力成熟度模型 CMMI。文献[11-17]从软件的架构和测算的技术角度, 对各种测算模型进行改进和应用。但因为软件本身的复杂性、历史经验的缺乏、估算工具不完备性以及一些人错误, 导致软件项目的估算结果往往和实际情况相差甚远。

本文以软件开发模型中的生命周期理论为先导, 对软件企业在开发软件的实践过程中, 测算人员的选择、在组织中的地位和测算点的确定等实施问题进行讨论, 并用实例加以说明。

2 软件生命周期简介

一个软件产品或软件系统经历的孕育、诞生、成长、成熟、衰亡等阶段, 称为软件生命周期。整个软件生命周期划分为六个阶段, 每个阶段有明确的任务。

2.1 软件生命周期(SDLC)的六个阶段

(1) 问题的定义及规划

此阶段, 软件开发方与需求方共同讨论, 主要确定软件的开发目标及其可行性。

(2) 需求分析

需求分析阶段是整个软件开发项目成功的基础。由于需求是随着时间和软件开发进程的发展而不断变化和深入的, 因此必须制定需求变更计划来对软件需要实现的各个功能进行详细分析, 以保证项目的顺利进行。

(3) 软件设计

软件设计一般分为总体设计和详细设计。此阶段

主要根据需求分析的结果, 对整个软件系统进行设计。好的软件设计将为软件程序编写打下良好的基础。

(4) 程序编码

此阶段将软件设计的结果, 按照统一、标准的编写规范, 转换成计算机可运行的, 具有良好的可读性, 易维护性和运行效率的程序代码。

(5) 软件测试

白盒测试和黑盒测试是主要的两种测试方法。测试过程分单元测试(通常在编码阶段执行)、组装测试以及系统测试三个阶段进行。目的是发现软件存在的问题并加以纠正。

(6) 运行维护

软件维护是软件生命周期中持续时间最长的阶段。通过软件纠错性维护和改进性维护, 满足用户的各类应用需求, 延续软件的使用寿命。

2.2 软件生命周期(SDLC)理论的优点与不足

尽管传统的 SDLC 理论不适应一些新的软件开发方法(如: 极限编程和敏捷方法等), 但它仍然在软件开发实践中, 得到广泛的应用。软件生命周期理论的优点, 如表 1 所示。

表 1 软件生命周期理论的优点^[18]

优点	缺点
开发过程易于控制	开发时间长
易于监督管理大项目	开发成本高
开发步骤清晰明了	系统要预先定义
易于评估实现目标	缺乏灵活性
文档规范管理	难以估计开发成本项目开发时间
易于维护	
有开发和设计标准	

3 测算在软件开发中的实施

软件项目主要受到用户需求及变化、开发时间、资源(包括人员、硬件条件以及资金)和风险等四种因素影响。合理的软件项目测算是软件项目成功实施的基础, 一般认为, 满足如下条件的测算是一个成功的测算^[1,2]:

- a. 早期测算结果的误差不应超出最后实际支出的 $\pm 30\%$;
- b. 采用的测算方法易于使用, 能在软件生命周期的不同阶段, 对原有的测算结果快速进行改进完善,

达到更高的精确度,并且能用工具的进行文档标准化;

c. 项目的测算规则,能被所有项目开发人员所理解。

d. 测算过程能被软件开发组织认同和信服。

为了能够获得一个符合实际情况的软件项目测算,我们从实施管理的角度,对测算人员的选择、测算人员与软件开发人员的关系和涉及测算规则和方法的测算点的确定等进行讨论。

3.1 测算人员的选择

软件开发过程中,通常涉及到的人员有,软件项目经理、软件经理、软件分析人员和各类软件工程师。其中:

软件项目经理最关心的是如何在合同规定的费用和时间范围内,以较短的开发时间和费用,按合同的交付要求,完成软件的开发。项目经理容易从粗糙的需求分析结果,以过分乐观的估计,进行软件的测算工作,并用此测算为依据,进行谈判和制定项目开发计划。尽管项目经理对软件测算结果和开发计划,会随着项目的执行,进行调整,但其职责和乐观的心态,不能一定保证项目开发费用不增加和时间不延后。

作为软件开发中间管理层的软件经理,一方面要尽力严格按软件项目经理审定的项目开发计划,进行进度控制和人力、费用的调配,另一方面,要指导解决软件工程师所碰到的各类技术或协调问题。而且软件经理通常仅对软件项目的某一部分(一个或几个模块)负责,尽管对软件的全貌有一定的了解,但出于在组织中的位置和时间及费用的限制,常会偏向以项目经理的目的为标准的软件测算。

直接进行软件的开发工作的其他软件工程技术人,由于自信和工作热情,在尽快完成工作和尽量减少费用的要求下,也会倾向于乐观的软件测算。

由于上述人员与软件项目的结果有直接或间接利害关系,如果进行软件的测算工作,出于利益或其他感情的影响,很可能做出有缺陷的测算,并影响到后续的软件开发。

我们认为与软件开发结果没有利益冲突的专业测算人员,才能尽量减少各种因素的影响,并提供尽可能符合客观实际的软件项目情况。一个合格的测算人员应具有如下素质:

a. 有工程管理的教育背景、实际的软件开发经验和相关的领域知识,这有助于了解项目管理中的问题。

b. 熟练掌握各种软件测算方法、工具及其原理,这有助于选择合适的测算方法和工具,对具体的软件问题进行分析。

c. 具有良好的表达沟通和组织能力,这有助于清晰地展现其测算进程,并组织其他专家讨论。

d. 与软件配置管理相似,拥有一个形成和管理文档的方法或程序,有助于记录、说明测算工作内容,专家的审查和作为今后的基准。

3.2 测算人员与软件开发人员的关系

由于软件产品的特殊性,软件公司的组织结构从本质上来说,基本上是矩阵型组织机构。包括:市场开发部门、软件开发部门、软件系统实施与维护部门、财务、人事等其它职能部门。矩阵型组织把按职能划分的部门与按项目划分的部门结合起来,其主要优点:①项目目标清楚,责任明晰,客户直接与项目经理沟通,对客户的反应迅速。②在多职能部门环境中,有利于培养有能力的项目经理或项目管理人才。③可最大限度地利用组织资源,供所有的项目使用,由项目组与职能部门分担成本。④有良好的协调机制,可得到职能经理的更多支持和有益的建议。⑤信息流动良好,项目知识横跨各职能部门,技术信息贯穿整个组织。我们以典型的矩阵组织结构中,指出了软件测算部门的位置。

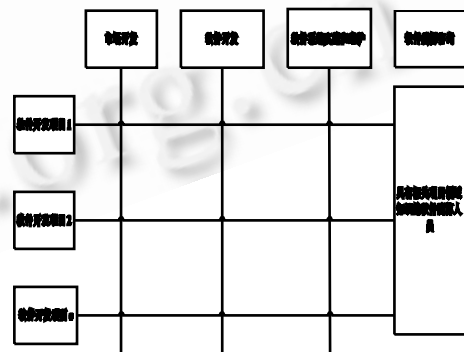


图1 软件测算人员在组织中的位置

从图1中可以看出,软件测算咨询部门与财务、人事等职能部门一样,无需对每个具体的软件开发项目负责。测算人员的主要任务是根据项目的具体情况,在软件生命周期的不同阶段,提供精度逐步提高的实际测算结果给软件项目组织的各级人员,对项目经理、软件经理的计划调整、进度控制和资源分配提供依据。软件测算工作始于程序量的测算工作,所以测算人员必须得到各类软件工程师的密切配合,参加项目开发的各种关键

性会议，了解项目的目标和战略及已完成项目的检测标准，选择适当的测算模型和方法，进行测算工作。故此，软件测算可以通过外包的形式，请组织外部专业的软件测算咨询公司进行，也可由组织内设置测算部门负责。

3.3 测算点的确定

软件生命周期蕴含在几乎所有的软件开发模型中，我们根据软件生命周期的思想，结合 COMOCO II 模型，给出了软件开发管理和控制的重要时点-测算点及相应的测算方法，如图 2 所示。



图 2 软件生命周期与测算点

软件开发可行性分析阶段的任务不是具体解决问题，而是研究问题的范围，探索这个问题是否值得去解，是否有可行的解决办法。软件测算人员在软件系统技术分析员的协助下，可采用专家估算、类比方法、推理方法和价格策略等方法，独立地给出软件规模及开发时间的测算报告，提供给项目相关人员，对建议的系统进行仔细的技术、成本 / 效益分析。可行性研究的结果是应用部门负责人做出是否继续进行这项工程的决定的重要依据。

需求获取和需求分析阶段的任务仍然不是具体地解决问题，而是准确地确定“为了解决这个问题，目标系统必须做什么”，主要是确定目标系统必须具备哪些功能。测算人员根据需求规格说明书，通过与软件开发人员的沟通交流和已往项目的数据，采用专家估算、自顶(底)向下(上)的方法，选择算法模型(如

COMOCO II 和 Putnam 模型)，在前一次测算报告的基础上，独立地给出一个改进的测算报告，为软件设计人员提供一个涉及参照指标。

概要设计阶段的主要有两项任务，一项是问题的解决方案的选择；另一项是设计软件的结构。测算人员在此阶段，通过与设计与主要开发人员的合作，采用专家估算和适当的算法模型，根据经过客户认可的解决方案和设计人员设计的系统结构，进行开发工作量测算，供系统分析员制定实现系统的详细计划。

详细设计阶段的任务是设计出系统结构中每个模块程序的详细规格说明(包括输入、功能算法和输出设计)及所有必要的细节。测算人员此时，通过了解开发人员的技术水平，用定量模型，如: Judgmental combination, Size-based-based estimation models^[20]，给出系统开发的较为准确地工作量和开发时间估算，为项目管理人员调整项目计划与控制提供依据。

在实现阶段，程序员写出正确的容易理解、维护的程序模块，并且仔细测试编写出的每一个模块。测算人员在实际完成的工作量和时间的基础上，结合已往项目的经验数据，采用专家估算、类比和推理的方法，用改进的算法参数，对测算结果进行修正。

在系统测试、实施和移交阶段，开发人员通过集成测试和验收测试，使软件达到预定的要求。测算人员根据软件的质量，结合已往项目的经验数据，采用专家估算、类比和推理的方法，对维护阶段的工作量进行测算，为项目的评价和总预、决算提供依据。

维护阶段的关键任务是，使系统持久地满足用户的需要。虽然没有把维护阶段进一步划分成更小的阶段，但是实际上每一项维护活动实质上都是经历了一次压缩和简化了的软件定义和开发的全过程。因此，软件测算活动仍需迭代进行。

4 测算实例

利用 USC 提供的一个免费软件测算工具，我们对一个三人开发的软件，进行了度量。环境和测算结果，如下表 2-4 所示。

表 2 规模因子表

Scale Factors				
PREC	FLEX	RESL	TEAM	PMAT
2.48 H	2.03 H	5.65 L	2.19 H	3.12 H

表3 分模块测算结果表

Module Name	Module Size	EAF	NOM Effort DEV	ACT Effort DEV	PROD	Staff
input	1050	1.00	3.6	3.6	294.7	0.3
process	6000	1.00	20.4	20.4	294.7	1.9
store	2100	1.00	7.1	7.1	294.7	0.7
TOTAL SLOC	9150		OPTIMISTIC MOST LIKELY PESSIMISTIC	24.8 31.0 38.8	368.4 294.7 235.8	2.5 2.9 3.4

表4 开发时间测算表

	OPTIMISTIC	MOST LIKELY	PESSIMISTIC
SCHEDULE	10.0	10.7	11.4

详细的时点测算结果如表5所示。

表5 PROJECT PHASE & ACTIVITY INFORMATION (Waterfall)

Life Cycle Phase	Life-Cycle - Effort	Life-Cycle - Schedule
Plans-And -Requirements	2.173 Person Months	1.933 Months
Product Design	2.173 Person Months	1.933 Months
Programming	18.892 Person Months	5.533 Months
Integration& Test	6.874 Person Months	2.472 Months

由于多种因素的影响,最终比测算的最可能时间,推迟了三个星期,让系统顺利地投入了运行。这表明,根据软件生命周期理论,对软件开发过程,实施分时点测算,能较好地控制软件开发过程,并对软件的及时交付及使用,具有促进作用。

5 结论

成功的软件系统开发意味着按时、按预算地交付符合用户和组织需求的软件系统。从国内外软件项目的开发经验来看,软件开发估算是软件开发管理控制的关键因素。因此,不断有新的或改进的软件估算模型和估算方法被提出。实际上,没有一种模型或方法明显优于其他模型或方法。在实践中,软件的复杂性要求软件测算人员了解不同估算模型及估算方法的优缺点及其适用环境,根据项目的具体情况,选择确定测算点,在不同阶段选用不同的方法,甚至是各种方法的结合使用,进行软件测算活动,这将一直是软件开发过程管理中的一个重要问题。

参考资料

1 Boehm B W. 软件工程经济学.北京:机械工业出版社,2004.

- 2 Londeix B.吴裕宪译.软件开发费用测算.北京:清华大学出版社,1991.
- 3 Boehm B. Software cost Estimation with COCOMOII. Prentice hall, 2002.
- 4 Jørgensen M, Grimstad S. How to Avoid Impact from Irrelevant and Misleading Information When Estimating Software Development Effort, IEEE Software 2008, (5/6):78-83.
- 5 Jørgensen M. A Review of Studies on Expert Estimation of Software Development Effort. Journal of Systems and Software, 2004,70(1-2):37-60.
- 6 Victor D, Daily K. Software Estimating at the task level -the Spectral approach, Project Control: Satisfying the Customer. London: Proceedings of ESCOM - SCOPE, 2001.
- 7 Thastings A S. A vector - based approach to software size measurement and effort estimation, IEEE Trans. Software Eng, 2001,27: 337-350.
- 8 GEORGES T. Measuring object oriented software with predictive object points. Proc. of ESCOM SCOPE 99 Conference, 1999.
- 9 Kan SH. Metrics and Models in Software Quality Engineering (2nd Edition), Addison-Wesley Professional, September, 2002.
- 10 Florac WA, Park RE, Carleton AD. Practical Software Measurement: Measuring for Process Management and Improvement. Software Engineering Institute Carnegie Mellon University, April 1997.
- 11 梁昌勇,王红星,刘业政.一种改进的软件项目成本估算方法.价值工程,2006,(8):105-107.
- 12 方海光,陈澎,余莉.基于范例库推理的软件成本估算模型研究.计算机工程,2006,32(19):191-192,220.
- 13 丁岳伟,马亦舟.贝叶斯校正算法在软件估算模型 COCOMO II 中的应用.计算机应用与软件,2007,24(1):151-153.
- 14 李明树,何梅,杨达,舒风笛,王青.软件成本估算方法及应用.软件原理,2007,18(4):775-795.
- 15 王璐,王宁.软件开发资金分配的一个非线性规划模型.工业技术经济,2007,26(9):80-83.
- 16 李虎雄,李虎俊. Web 系统软件开发项目的成本估算模型.计算机工程,2007,33(16):274-276.
- 17 冯楠,李敏强,寇纪淞,方德英.一种基于模糊决策树的软件成本估计模型.计算机工程与应用,2007,43(26):21-23.
- 18 http://en.wikipedia.org/wiki/Software_development_effort_estimation