

COCOMO 估算模型改进研究

唐林燕

(广东轻工职业技术学院, 广东 广州 510055)

摘要: 针对代码行估算软件规模的不足, 改进了软件规模的估算方法, 并据此改进了 COCOMO 工作量估算公式。应用实例证明, 使用改进后的估算公式, 估算准确性高于原 COCOMO 工作量估算公式。

关键词: COCOMO, LOC, 软件规模, 估算

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2006)12-0058-03

Improvement Research of COCOMO Estimation Model

TANG Lin-yan

(Guangdong Light Industry Technical College, Guangzhou 510055, China)

Abstract: Aiming at the deficiency of software sizing by using LOC (line of code), this paper improves the way of LOC. Based on that, this paper also improves the effort formula of COCOMO. According to actual validation, the outcome of estimating cost by this improved formula is more accurate than the old one.

Key words: COCOMO, LOC, Software size, Estimate

1 引言

什么是软件的规模这个问题至今还没有做出明确的定义, 导致规模的度量单位也没有统一。比较现有的软件规模估算方法, 没有哪一种是十全十美的。COCOMO 模型是当今世界上应用最广泛的软件成本估算模型^[1]。考虑到 COCOMO 模型的成熟性、应用广泛性, 及其输出产物的强大上, 我们又不偏向其规模估算方法——LOC 估算方法。虽然 LOC 估算方法存在很多的缺陷, 我们可以用一些方法来改进它, 提高它的估算准确性。

2 软件规模估算改进研究

通过对软件规模度量的总结, 软件规模及其度量有以下一些特征:

(1) 软件规模是个模糊的概念, 目前缺乏明确的定义。

(2) 软件规模的大小只有近似值, 并没有精确值^[2]。

(3) 软件规模的度量必须服务于工作量和成本的度量目的。

(4) 软件规模的度量独立于物理实现和开发技术。

(5) 软件规模度量单位的选择可以根据不同应用环境而不同, 其尺度的大小也可以不同。

为了改变传统 LOC 规模估算的单一结构, 我们需要结合软件规模度量的这些特征, 从软件的属性集中选择适合规模估算的属性, 来弥补传统 LOC 规模估算的不足。

软件属性可分内部属性和外部属性两大类^[2]。软件内部属性, 是指能够仅仅根据软件本身来进行度量的属性, 独立于其行为, 即“完全取决于自身”, 无需执行系统就可以对其进行度量, 如规模、复杂性和软件模块间的依赖关系; 软件外部属性, 是指只有考虑到软件如何与其环境相关的部分关联才能进行度量的属性, 如软件的可靠性、可维护性、可使用性和效率等。内部属性通常在生命周期早期就能度量到, 而外部属性只有在软件产品完成或几乎完成时才能度量到。由此, 我们需要在软件的内部属性中寻找合适的属性。

软件有三个非常重要的内部属性: 规模、软件复杂性和技术含量。软件规模若以代码行长度、软件复杂性和技术含量这三个属性来度量软件的规模, 就能改变传统 LOC 规模估算只针对代码行长度这个单一片面的估算结构。代码行长度描述了软件的长度规模, 复杂性描述了软件的知识辐射面, 技术含量描述了软件的技术难度辐射深度。

收稿日期: 2005-12-12

基金项目: 广东省自然科学基金项目(04009863)

软件开发活动是人的高智力的解决项目问题的活动。对软件开发人员来说,开发技术是不得不考虑的问题。笔者认为,技术要分为两类。一类是软件自身所包含的技术含量。另一类是开发者的技术素质。软件的技术含量是软件非常重要的一个内在属性。给定一个有一定技术含量的软件项目,开发者的技术素质低于该技术含量时,该项目对他而言就是一个难对付的项目,他在同等功能的代码行上所花费的工作量就要多些;而当开发者的技术素质高于该技术含量时,该项目对他而言就是一个相对简单的项目,他在同等功能的代码行上所花费的工作量就要少些。从规模度量必须服务于工作量度量和成本度量这点来说,软件技术含量理应当包含在规模度量范围之内。

同样,对于软件的复杂性,它也应该是规模度量要考虑的因素之一。比如,同样是数据库操作,含有三个数据表的数据库与含有一百个数据表的数据库,后者要复杂、“庞大”得多,即它的规模大,开发人员对付它所需的工作量也将增大。而如果后者是异地的不同类的数据库组成的数据仓库,则不仅复杂性高,其技术含量也高。

考虑软件技术含量的另一个主要原因是,高水平的程序员往往想办法用短小精悍的代码去解决项目问题,而水平一般的程序员往往用冗长的代码。对于相同的软件模块,如果只用代码行长度来估算软件规模,对高水平的程序员极为不合理,而且估算出来的工作量也将失真。

至于为什么选用软件技术含量而不是用软件人员的开发技术来度量软件的规模,是因为软件规模的度量要独立于具体的实现方法和开发技术,而仅仅描述软件本身。

综上所述,在估算软件规模大小时,可以将它看作是代码行长度、技术含量和软件复杂度这三个属性变量的函数,即:

$$\text{Size} = F(C, T, L)$$

其中 Size 为软件规模; C 为软件复杂度; T 为软件技术含量; L 为软件的源代码长度,单位与 COCOMO 规模度量单位一致。

为计算简便起见,可设软件规模大小为这三者的累乘积,即把复杂性和技术含量当作代码行长度的权值计算。

软件复杂性的度量准则及权值等级可参考 COCOMO 的复杂性成本驱动因子,即综合考虑控制操作、计算操作、设备相关操作、数据管理操作和用

户界面管理操作要求。

技术含量需要参考当前社会软件行业的技术发展水平,以确定适当的权值。例如,对于一个软件系统,在一定的时间范围内,如果对一个平均很低水平的人员,如未接受任何计算机知识教育,但接受短暂的培训后就能胜任去完成,则可设其技术含量为很低;如果对一个平均很高水平的人员,如具有丰富经验的专家,但还需深入研究后才能胜任,可设其技术含量为很高;如果对一个有着四年计算机本科教育背景的平均中等水平的人员就能胜任时,可设其技术含量为标称。其它级别类推。

3 COCOMO 模型改进研究

前面对基于代码行估算软件规模进行了改进研究,并提出了一个新的估算软件规模的方法,那么 COCOMO 工作量估算公式就得做一下调整。

新的 COCOMO 工作量估算公式与原 COCOMO 工作量估算公式形式一样,核心方程均为:

$$PM = A \times \text{Size}^E \times \prod_{i=1}^n EM_i$$

这两个估算公式的不同点是:

(1) 新估算公式的“Size”不仅只是原始的源代码千行数,而是源代码千行数再乘上软件复杂度和技术含量权值。

(2) 新估算公式的工作量乘数 EM_i 去除了软件复杂性因子,其它均不变。软件复杂性被移到软件规模计算中去,当作软件规模的“驱动因子”。

4 COCOMO 改进模型的验证

现在我们就把它应用在 COCOMO 实践当中,去检验其是否有改进,是否对提高软件成本估算的准确性有所帮助。

对 Boehm 用来分析软件项目成本及其开发属性的 63 个已完成项目数据组成的数据库^[4]中复杂度不为标称(1.0)的中等开发模式下原 COCOMO 81 估算公式与新估算公式的估算值和估算相对误差量级 MRE 值的比较。改进效果显著,如表 1 所示。

两者的估算准确度比较如表 2 所示。

从表 1 可以看出,新 COCOMO 公式的计算结果普遍比原 COCOMO 公式计算结果更接近于实际值。并且可以得出,对于平均相对误差量级 MMRE,原 COCOMO 公式计算结果为 20.7,而新 COCOMO 公式计算结果为 19.8,准确情况要好于原 COCOMO 公式。

表 1 新估算公式与原估算公式估算结果比较

项目序号	实际值	原估计值	新估计值	原 MRE	新 MRE
1	2040	2218	2063	8.7	1.1
2	1600	1770	1709	10.6	6.8
3	243	245	237	0.8	2.5
4	240	212	205	11.7	14.6
6	43	30	30	30.2	30.2
8	1075	669	918	37.8	14.6
9	423	397	418	6.1	1.2
10	321	214	225	33.3	29.9
:					

表 2 新旧估算公式估算准确度比较

估算准确度	原 COCOMO 公式	新 COCOMO 公式
PRED(0.10)	36%	42%
PRED(0.25)	68%	70%
PRED(0.30)	72%	76%

从表 2 可以看出,对于新 COCOMO 公式,有 42% 的项目其估算偏差在实际值的 10% 以内,而原 COCOMO 公式只有 36%。对于估算偏差在 25% 和 30% 以内的项目比例,新 COCOMO 公式计算结果均比原 COCOMO 公式的高。由此可以验证,将软件复杂性当作软件规模的“驱动因子”,并把它用于软件规模的计算中而不是把它当作成本驱动因子,估算的准确度高于原 COCOMO 估算公式的计算结果。

5 结束语

深入分析了软件规模的特征,改进了传统 LOC 规模估算方法,认为软件规模应该由软件的三个非

常重要的属性:代码行长度、技术含量和软件复杂性来度量,并且软件复杂性不适合当作开发“环境变量”——成本驱动因子去“事后调整”工作量,而应当当作软件所特有的内在属性、当作软件规模的“驱动因子”——长度权值估算软件规模,然后再去估算软件工作量。经过 COCOMO 数据库的验证,这个新工作量估算公式计算的结果更接近于实际值。

未来的工作,就是通过更多的详细历史项目数据以及专家法确定技术含量的权值,以及通过回归技术调整参数值并用贝叶斯方法校验,完善新 COCOMO 工作量估算公式。这里笔者先提出这种改进思想,以供大家评定与参考,更深一步的工作等待大家共同研究和探讨。

参考文献:

[1] Barry W Boehm. 李师贤,等译. 软件成本估算 COCOMO II 模型方法. 北京:机械工业出版社,2005

[2] Norman E Fenton,Shari Lawrence Pfleeger. 杨海燕,等译. 软件度量. 北京:机械工业出版社,2004

[3] J P Lewis. Limits to software estimation. ACM, Software Engineering Notes, 2001, 26(4): 54

[4] Barry W Boehm. 李师贤,等译. 软件工程经济学. 北京:机械工业出版社,2004

作者简介:

唐林燕 女,(1962-),教授。研究方向为软件工程与计算机应用技术。

(上接第 57 页)

(3) 能够安全的保护日志记录或者其它敏感信息。即使非法用户侵入个别日志服务器,并且看到了部分日志记录,但是它不一定能够获得日志记录的全文。要想达到此效果,必须侵入更多台日志服务器。

(4) 只要有少数几个日志服务器,就能取得日志记录的完整信息。

必须指出,本模型采用冗余的方法,提供分布式的日志服务,增加了系统组建的成本。本系统并不能提供绝对的保护。即当几乎所有日志服务器上的日志被破坏的话,那么系统无法恢复原来的日志记录。

参考文献:

[1] 白银姬,谢维信,喻建平等. 基于主动秘密共享的安全容忍入侵方案. 兰州交通大学学报(自然科学版),2004,

23(1): 87-89

[2] 张宇. 一种基于门限 RSA 的容侵系统结构. 中南民族大学学报, 2004, 23(4): 68~71

[3] 黄文. 基于日志分析策略的分布式网络入侵预警系统模型. 湘潭大学自然科学学报, 2004, 26(4): 39~42

[4] Richard O Duda, Peter E Hart, 李宏东等译. 模式分类. 北京:机械工业出版社,2003

[5] 高温丽. 日志系统的检测系统的应用研究. 河北建筑工程学院学报, 2004, 22(2): 103~106

[6] 李伟生. 基于贝叶斯网络的态势评估. 系统工程与电子技术, 2003, 25(4): 480~484

[7] 宫秀军. 主动贝叶斯网络分类器. 计算机研究与发展, 2002, 39(5): 574~579

作者简介:

黄光球 男,(1964-),教授,博士。研究方向为电子商务与网络安全。