

文章编号:1671-6833(2005)02-0106-04

一个软件复用成本模型对COTS、PLE 和TTM 的拓展

杨 林

(美国西弗吉尼亚大学计算机科学和电子工程系, 西弗吉尼亚州 摩根城 WV26505)

摘 要: 对当前存在的 4 种软件再利用模型进行分析, 并对现有模型的优缺点进行比较, 发现它们的共性及存在的问题是: 不能涵盖目前软件复用的最新领域. 以 Chmiel 的综合再利用模型为基础, 将其推广和扩展到 COTS (Commercial Off The Shelf) 和生产线工程 (PLE), 并把进入市场时间 (TTM) 的成本和利益整合进去. 基于这样的原理, 这个模型能涵盖到软件复用领域当前最新的行为, 包括 COTS、PLE and TTM.
关键词: 再利用(复用); 成品软件; 生产线工程; 进入市场时间
中图分类号: TP 311.56 **文献标识码:** A

0 引言

在过去的几年中, 存在着几种跟评估软件再利用成本有关的成本模型. 如 COCOMO、COCOMO II、COCOTS 和 Chmiel 模型^[1]. 然而却没有一种显著的和行之有效的模型能将技术和非技术因素一体化并使之及时投入市场发挥作用. 本文作者以 Chmiel 的综合再利用模型为基础, 将其推广和扩展到 COTS、PLE, 并把进入市场时间 (TTM) 的成本和利益整合进去.

1 模型结构

在建立并继承 Chmiel 模型基本结构的基础上, 带有 TTM 子模型的整体成本模型由 4 个工程周期和一个 TTM 子模型构成. 这 4 个工程周期和 TTM 模型分别为: 企业工程周期 (Corporate Engineering Cycle); 领导域工程周期 (Domain Engineering Cycle); 组件工程周期 (Component Engineering Cycle); 应用工程周期 (Application Engineering Cycle); TTM 子模型.

这个综合的成本模型可以通过为期数年的持续过程来包含 COTS 模型、PLE 模型和 TTM 模型.

2 组件工程周期

由 3 种方法使核心资产进入领域. 每种核心资产对应于一个组件工程周期^[2].

2.1 起始成本 (y = 开始年份 (SY))

2.1.1 挖掘遗留软件

成本来源于以下两个方面: ①估价成本, 要从公司现有的资源中选择并鉴定可再用组件; ②入库, 组件的成本要进入可再用库存. 因此总的起始成本就是

$$IC[COEC] = AC + II,$$

式中: $IC[COEC]$ 为组件工程周期起始成本; AC 为评估成本; II 为入库成本.

2.1.2 从零开始开发

成本来源于: ①开发成本, 开发可再用组件所耗费的成本; ②入库, 使组件进入可再用库存所耗费的成本. 因此总的起始成本为

$$IC[COEC] = ER + II,$$

式中: $IC[COEC]$ 为组件工程周期的起始成本; ER 为可再用开发成本.

如果 ER 不可得的话, 可以通过下述公式得到开发成本^[3]:

$$ER = \frac{RCWR \times E \times CFSP}{12}, E = a \times S^b,$$

式中: E 为单一使用的开发成本; $RCWR$ 为相关再利用率; a, b 为 COCOMO 参数, 跟工程的特性相关; $CFSP$ 为软件开发人员年成本; S 为组件大小, 以千行代码为单位.

2.1.3 COTS

估价成本为从市场上选择鉴定可再用组件所花费的成本; 购买成本为 COTS 组件的成本, 即在

收稿日期: 2005-01-20; 修订日期: 2005-03-08

作者简介: 杨林 (1975-), 男, 河南省获嘉县人, 西弗吉尼亚大学博士候选人, 主要从事软件工程领域软件复用经济模型的研究.

项目开始时,为得到可再用资产所耗费的成本;入库:使组件进入可再用库存所需要的成本.因此总的起始成本为

$$I[COEC] = AC + P + LI,$$

式中: $I[COEC]$ 为组件工程周期的最初成本; AC 为评估成本; P 为购买COTS的成本.

2.2 年成本 ($SY < y \leq SY + Y - 1$)

2.2.1 发掘遗留软件和从零开始开发

维护:维护成本来自于COCOMO维护成就因素.年变化交易(ACT)是每年开发成本而消耗的维护成本的比率.其典型变化于0.15左右^[3].因为每年的维护是单独使用从零开发成本的15%.

因此年总成本是

$$AC[COEC](y) = MN(y) + OC(y);$$
$$MN(y) = \frac{0.15 \times E \times CESP}{12}; OC(y) = \frac{L \times LESF}{n \times 12},$$

式中: $AC[COEC](y)$ 为组件工程周期的年成本; $MN(y)$ 为年维修成本; $OC(y)$ 为年操作成本; L 为库管理人员数量; n 为可再利用率库存总数量; $LESF$ 为软件开发人员年成本

2.2.2 COTS

通过使用COTS组件,公司就可以节省维护的功力和成本.Hari指出,内部开发软件的软件维护成本是产品生命周期的总成本的55%,这包括弥补错误及完成增强相的成本^[4].NASA软件开发管理手册指出,维护成本可达70%.因此,总的年成本是

$$AC[COEC](y) = AL + OC(y);$$
$$OC(y) = \frac{L \times LESF}{n \times 12},$$

式中: $AC[COEC](y)$ 为组件工程周期的年成本; AL 为年许可费用; n 为可再利用率库存总数量

2.3 起始利润 ($y = SY$)

2.3.1 从零开发

对于这种核心资产来说,没有起始利润.

2.3.2 发掘遗留软件和COTS

这两项的起始利润来自于节省的开发成本.这些利润是通过开发现有组件或购买的组件而在组件工程周期中节省的劳动成本中得到的.因此,组件工程周期的起始利润为

$$IB[COEC] = \frac{ER \times CESF}{12},$$

式中: $IB[COEC]$ 为组件工程周期的最初利润.

2.4 年利润 ($SY < y < SY + Y$)

年利润是指投资周期第一年之后的年份里得到的利润.只有当应用工程周期使用核心资产时,

核心资产的价值才能被实现.影响这些核心资产能够带来多少利润的主要因素有:①相对黑盒子(RBP):卖给应用工程师黑盒子的价格.这项研究使用的是Chmiel提供的默认值0.4.COTS的使用方法也是如此.②相对白盒子(RWP):卖给应用工程师白盒子的价格.这项研究使用的是Chmiel提供的默认值0.15.

2.4.1 从零开始开发和挖掘遗留软件

$$AB[COEC](y) = FreqB(y) \times RBP \times E + FreqW(y) \times RWP,$$

式中: $AB[COEC]$ 为组件工程周期的年利润; $FreqB(y)$ 为组件黑盒子的年再利用频率; $FreqW(y)$ 为组件白盒子的年再利用频率; RBP 为相关黑盒子; RWP 为相关白盒子.

2.4.2 COTS

$$AB[COEC](y) = FreqB(y) \times RBP \times E,$$

式中: $AB[COEC]$ 为组件工程周期的年利润.

3 领域工程周期

在领域工程周期中,领域工程师定义了领域特征、一般属性和体系结构、集中数据,分析了领域内组件的可再用性,定义可用性指导方针;分析再使用结构.

3.1 成本 ($SY < y < SY + Y$)

领域分析成本是分析可再用构造和体系结构从组件工程周期逐级继承的成本.领域工程周期的年成本是从组件工程周期中逐级继承得到的.

3.2 利润 ($SY < y < SY + Y$)

领域工程周期的利润是从组件工程周期中逐级得到的.

4 应用工程周期

在应用工程周期中,应用工程师要为再使用的吸收、制作、粘合代码和系统挥发性付出额外的成本.

4.1 起始成本 ($y = SY$)

4.1.1 从零开始开发和挖掘遗留软件

为投入应用,应先获得黑盒子资产的成本、白盒子资产的成本,EDSI成本、粘合代码成本.从以下的估计适用量化可以计算等价递送源码指令Equivalent delivered source instructions(EDSI)^[3].

$$I[PRJ] = \frac{RBP \times FTSP}{12} \times \sum_{j=1}^m E_j + \frac{RWP \times FTSP}{12} \times \sum_{k=1}^n E_k + \frac{FTST}{12} \times a \times$$

$$\left(\sum_{i=1}^{m+n} (EDSI_i + GC_i)\right)^b,$$

式中: $IQ[PRJ]$ 为工程的最初成本; m 为发掘遗留软件或者从零开发的组件在一个工程中黑盒子的使用数; n 为发掘遗留软件或者从零开发的组件在一个工程中白盒子的使用数量; GC 为粘合码的大小.

4.1.2 COTS

成本包含为应用而获取 COTS 的成本、适应努力、粘合代码努力. 因此, 起始成本为

$$IQ[PRJ] = \frac{RBP \times CFSP}{12} \times \sum_{i=1}^n E_i + \frac{AESP}{12} \times \frac{73.2}{45.5} \times \sum_{i=1}^n GE_i,$$

式中: $IQ[PRJ]$ 为工程的最初成本; n 为工程中的 COTS 组件数量; GC 为粘合码大小; GE 为粘合码成本.

4.2 年成本 ($SY <_y < SY + Y$)

表 1 为 COTS 的努力分布.

表 1 COTS 努力分布
Tab. 1 COTS distribution

COTS 开发阶段	努力百分比 /%
评估	7.8
制作	27.7
粘合码	45.5
挥发性	19.0

$$VC = \frac{19}{45.5} \times 6E,$$

式中: VC 为挥发性成本; GE 为粘合码成本.

$$AQ[PRJ](y) = \left[\sum_{i=1}^n VC_i\right] \times \frac{AESP}{Y \times 12},$$

式中: $AQ[PRJ]$ 为工程的年成本; n 为组件数量; Y 为工程的年限.

4.3 起始利润 ($y = SY$)

起始利润是因使用核心资产而节省的开发成本得到的.

$$IB[PRJ] = \frac{AESP}{12} \times a \left(\sum_{i=1}^n S_i\right)^b,$$

式中: $IB[AEC]$ 为应用工程周期的最初利润; S 为组件大小; n 为组件数量.

4.4 年利润 ($SY <_y < SY + Y$)

年利润是由质量提升得到的. 可再用开发的组件其质量要比单独使用从零开始开发的组件好, 因为它的设计更加谨慎, 测试更加彻底, 文件更加精确^[1]. 因为 ACT 是 0.15, 可再用资产的年变化交易 (ACT') 是 0.09, 而因软件可再用的操作成本微分是开发成本的 9%^[1]. 因此质量提升为

$$(ACT - ACT') \times E;$$

$$AB[PRJ] = \frac{AESP}{12} \times (ACT - ACT') \times$$

$$a \left(\sum_{i=1}^n S_i\right)^b, SY <_y < SY + Y;$$

$$AB[PRJ] = \frac{AESP}{12} \times (ACT - ACT') \times$$

$$a \left(\sum_{i=1}^n S_i\right)^b + R, y = SY + Y;$$

式中: $AB[PRJ]$ 为工程年利润; n 为组件数量; R 为提高的收入, 可以得到更多的销售收入 R 源自 TTM 子模型.

4.5 AEC 中的成本和利润总和 ($SY <_y < SY + Y$)

应用工程中的成本和利润从在周期中进行的所有项目中得到.

$$C[AEC](y) = \sum_{i=1}^n C[PRJ]_i(y);$$

$$B[AEC](y) = \sum_{i=1}^n B[PRJ]_i(y),$$

式中: $C[AEC](y)$ 为应用工程周期的年成本; $B[AEC](y)$ 为应用工程周期的年利润; $C[PRJ]_i(y)$ 为工程年成本; $B[PRJ]_i(y)$ 为工程年利润; n 为 AEC 的工程数量.

5 TTM 子模型

5.1 缩短的 TTM 带来的利润

从应用周期之外得到利润, 在项目完成和产品投放到市场之后得到回报.

缩短的 TTM(STTM): 这项利润是这个项目整个生命周期节省出来的时间; 收入提高 (R); 更多的市场份额和更多的销售收入.

5.2 量化 TTM 利润

5.2.1 缩短的 TTM(STTM)

通过使用现有的核心资产而不是从零开始开发, 可以缩短应用工程周期. 这个计算建立在从中间 COCOMO 得到的等式基础上, 这个方程式计算了开发的劳动成本和时间^[3].

$$TDEV = 2.5 \times MM^f,$$

式中: $TDEV$ 为成本和开发时间; MM 为月人工. 利用这个公式可以计算出一个月的时间里用多长时间开发. 因此缩短的 TTM 可以用以下方式导出.

$$STTM = T_1 - T_2,$$

$$T_1 = 2.5 \times \left[a \times \left(\sum_{i=1}^n S_i\right)^b\right]^c,$$

$$T_2 = 2.5 \times \left[a \times \left(\sum_{i=1}^n EDSI_i + \sum_{i=1}^n GC_i\right)^b\right]^c,$$

式中: T_1 为从零开始开发的时间; T_2 为使用复用组件所消耗的额外时间; S 为组件的大小; $EDSI$ 为等

价递送源码指令; GC 为粘合码; n 为组件数目.

5.2.2 收入增长(R)

这项利润来源于因缩短的TTM而得到的更多的市场份额和销售收入.

$$R = BC \times DC \times (STTM/12),$$

式中: R 为收入提升进; $STTM$ 为缩短的TTM,由于再利用组件的使用而节省的时间; BC 为利润系数; DC 为开发成本利润系数是企业为了确定产品价格以在市场上获得利润的一个系数.

6 总体工程周期

总体工程周期使最高层次的周期,要在其它3个工程周期的全部成本和利润以及TTM子模型的基础上做出最终的决定.

6.1 起始成本($y = SY$)

在总体工程周期中的起始成本来自于建立再用结构、员工培训等,还有逐级来自于领域和应用工程周期的成本.

$$IC[CEC](SY) = IS + C[DEC](SY) + C[AEC](SY),$$

式中: $IC[CEC]$ 为公司工程周期的最初成本; IS 为基础建设的成本; $C[DEC](SY)$ 为DEC的年成; $C[AEC](SY)$ 为AEC的年成本;DEC为域工程周期;AEC为应用工程周期.

6.2 年成本($SY < y < SY + Y$)

$$AC[CEC](y) = C[DEC](y) + C[AEC](y),$$

式中: $AC[CEC](y)$ 为公司年成本; $C[DEC](y)$ 为域工程周期成本; $C[AEC](y)$ 为应用工程周期成本.

6.3 利润($SY < y < SY + Y$)

利润逐级来自于领域和应用工程周期.

$$B[CEC](y) = B[DEC](y) + B[AEC](y),$$

式中: $B[CEC](y)$ 为CEC年利润; $B[DEC](y)$

为DEC年利润; $B[AEC](y)$ 为AEC年利润.

7 结论

目前软件复用经济模型中有一些是不成熟的,还处在试验阶段,还有一些具有片面性.比如,COCOTS模型只能在最初阶段起作用,而对长期的再利用工程却不起作用.当今软件再利用实践中最关键的问题是,以软件经济为基础,没有一种模型能概括定义并开发必要的细节以支持有效的COTS软件使用过程模型和PLE模型.本文作者分析了当前存在的几种模型并总结出它们的共性和存在的问题,通过给Chmiel增加一些新的特性,建议将其应用于COTS再利用体系和PLE模型中.此外,对缩短的TTM的益处进行量化的尝试也在本文得到了阐述,并总结出一个TTM子模型以解决这些关键问题,它可以针对长期的工程并能和缩短的TTM模型利润一体化.最终,在公司工程周期和TTM子模型利润的基础上,可以计算出NPV值和PQI值,它是公司的经营管理层做出最后决策的依据.

参考文献:

[1] SENTA Fowler Chmiel .An Integrated Cost Model for Software Reuse[R] .Li menrick ;Proceedings from ICSE 2000 Conference ,2000.

[2] USC COCOMO Company .USC COCOMO reference Manual [M] .California ;University of Southern California , 1994.2~3.

[3] BARRY W Boehm , Software Engineering Economics[M] . Englewood Cliffs ;Prentice Hall , 1981.71~73.

[4] HARRIS K .Using an Economic Model to Tune Reuse Strategies[R] .New York ;Proceedings of the fifth annual software reuse workshop ,1992.1~2.

Generalization of an Integrated Cost Model and Extensions to COTS , PLE , and TTM

YANG Lin

(Lane Department of Computer Science and Electrical Engineering , West Virginia University , Morgantown WV 26505, USA)

Abstract : This paper aims at investigating , analyzing the existing software reuse models and identifying their common properties and problems , comparing the existing models for their advantages and disadvantages . Based on Chmiel 's comprehensive reuse model , this paper generalizes and extends it to integrate use of Commercial Off The Shelf (COTS) , Product Line Engineering (PLE) , and integrates costs and benefits of Time To Market (TTM) . Based on this rationale , this model can cover the most recent software reuse activities such as COTS , PLE , and TTM .

Key words : reuse ;COTS ;PLE ;TTM