

# 异地分布式敏捷软件开发的时间成本估算研究<sup>①</sup>

殷茗<sup>1</sup>, 成丽媛<sup>1</sup>, 邓国林<sup>1</sup>, 姜继娇<sup>2</sup>

<sup>1</sup>(西北工业大学 软件与微电子学院, 西安 710072)

<sup>2</sup>(西北工业大学 管理学院, 西安 710072)

通讯作者: 殷茗, E-mail: yming@nwpu.edu.cn

**摘要:** 为了解决异地分布式敏捷软件开发中的时间成本估算问题, 研究了影响异地分布式开发与效率的因素, 考虑了异地分布式敏捷软件开发的实际情境, 对 COCOMOII 模型的成本驱动因子进行了分析、筛选和新增, 在此基础上, 改进了 COCOMOII 模型, 使得改进 COCOMOII 模型能够合理地估算出异地分布式敏捷软件开发的时间和成本, 最后对改进 COCOMOII 模型的开发时间与成本估算方法进行了实现, 该改进 COCOMOII 模型可以较好地估算出异地分布式敏捷开发中的工作量和开发进度.

**关键词:** 异地分布式; 敏捷开发; 时间成本估算; COCOMOII 模型; 成本驱动因子

引用格式: 殷茗, 成丽媛, 邓国林, 姜继娇. 异地分布式敏捷软件开发的时间成本估算研究. 计算机系统应用, 2018, 27(6): 189-194. <http://www.c-s-a.org.cn/1003-3254/6383.html>

## Research on Time Cost Estimation of Geographically Distributed Software Development Project

YIN Ming<sup>1</sup>, CHENG Li-Yuan<sup>1</sup>, DENG Guo-Lin<sup>1</sup>, JIANG Ji-Jiao<sup>2</sup>

<sup>1</sup>(School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710072, China)

<sup>2</sup>(School of Management, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** In order to solve the problem of time cost estimation in the development of geographically distributed software projects, the factors that influence development time and efficiency of geographically distributed development has been studied, and the actual situation of distributed software development is considered synthetically. The cost driving factors of COCOMOII model are analyzed, screened, and newly increased. The COCOMOII model is improved on this basis, so that the improved COCOMOII model can reasonably estimate the time and cost of the distributed software development project. Finally, the development time and cost estimation of the improved COCOMOII model are realized. The improved COCOMOII model can better estimate the workload and the development progress of the distributed development.

**Key words:** geographically distributed; agile development; time cost estimate; COCOMOII model; cost drivers

异地分布式敏捷软件开发相对于传统的软件开发模式, 更具有灵活性, 能节约成本和缩短软件项目开发周期. 随着一些企业或公司采用异地分布式敏捷软件开发, 其所面临的挑战也迎面而至. 成本节约与周期的缩短, 需要考虑多方面的因素. 实施异地分布式敏捷开发需要考虑项目的工期和效率. 从项目管理的三个约束条件来说, 时间是最重要的, 它将决定工期、成本及

质量. 因此, 本文主要考虑影响时间与成本的因素, 研究异地分布式敏捷软件开发项目的时间成本估算方法.

本文的研究借鉴国外现有的研究成果, 进行异地分布式敏捷开发工期与效率影响因素的解析. 通过文献[1]研究得到 7 个因素, 包括远距离沟通、实际距离、站点间的相互依存关系、文化的适应性、体系结构的适当性、项目创新和项目周转, 这些因素能够很

① 基金项目: 国家自然科学基金(71201124); 教育部人文社会科学基金(16YJA630068); 航空科学基金(2016ZG53071); 西北工业大学中央高校基本科研业务费资助项目(G2017KY0101)

收稿时间: 2017-09-28; 修改时间: 2017-10-18; 采用时间: 2017-11-16; csa 在线出版时间: 2018-05-28

好地反映异地分布式敏捷开发的特点,考虑这些因素,使用一些合适的方法评估软件开发,从而制定出比较好的项目计划.本文使用COCOMOII模型进行项目的估算,按照COCOMOII模型,度量项目开发工作量和开发时间最重要的是成本驱动因子.针对异地分布式敏捷软件开发,本文参考了前人研究的成本驱动因子以及参考取值,对这些因子进行解析分析,考虑异地分布式敏捷软件开发的实际情况,对成本因子作了部分增加、增强或取消,改进了COCOMOII模型,使其能够合理估算异地分布式敏捷开发的成本与时间.

## 1 传统COCOMOII模型的开发时间和成本估算

早期的COCOMO模型<sup>[2]</sup>由Boehm提出,主要用于软件的成本和开发时间估算.COCOMO模型能够很好地分析项目的可行性并有效管理软件开发过程<sup>[3]</sup>,为软件开发提供重要的估算依据,并被广泛地接受和使用<sup>[4]</sup>.然而由于软件复杂度的提升,COCOMO模型对现在的软件开发的成本和时间估算已不再适用,Boehm对原模型进行了改进,提出了COCOMOII模型<sup>[5]</sup>来估算工作量、规模、开发进度和人员安排,有利于合理地制定软件开发计划,对软件开发起到至关重要的作用.

COCOMOII有粗略估算的早期设计模型、基于对象点的应用组合模型和根据软件体系结构估算的后体系结构模型<sup>[6]</sup>.本文采用的是后体系结构模型,它要求了解软件体系结构功能,并使用规模度量因子和成本驱动因子来进行估算,具有估算效率高、基本不受人员主观因素影响、能够综合考虑成本驱动因子的特点.但是对于特定领域的时间和成本估算的因素考虑不足,模型计算时需要输入规模的估算值,并且受一些成本因子的影响,导致估算不准.

在COCOMOII中,最重要的估算就是对工作量和开发进度的估算,其中开发进度即开发时间,工作量一般用PM表示,是指一个人在一个月内进行软件开发的时间数,单位是人月.根据项目的不同情况,工作量可以有标称进度NS或者调整进度AS,同时标称进度的工作量估算不含开发进度SCED,SCED反映了软件面临的进度压力.开发时间根据工作量的大小来计算,一般单位是月.

标称工作量的计算公式为:

$$PM_{NS} = A \times Size e^E \times \prod_{i=1}^n EM_i \quad (1)$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad (2)$$

式中, $PM_{NS}$ —人月/PM; $A$ —常数,取2.94; $Size$ —估算的规模/KSLOC; $EM_i$ —工作量因子; $E$ —规模指数; $B$ —常数,取0.91; $SF_j$ —比例因子.

对于开发成本的计算,通常是由工作量与单位劳动时间成本相乘得到.

指数 $E$ 是比例因子作用的体现, $E>0$ ,表示软件项目在总体规模上表现为非经济性,一般是由于人员交流及大型系统集成的成本的影响;若 $E<0$ ,表示软件项目在总体规模上表现为经济性,这样工作量不会因为软件项目规模的增加而发生变化;若 $E=0$ ,表示软件项目在总体规模上表现为经济性和非经济性的平衡,适用于小型项目的成本估算.指数 $E$ 中的每个比例因子SF从低到高都有一个权重,这5个比例因子分别是先例性(PREC)、团队凝聚力(FLEX)、体系结构/风险化解(RESL)、开发灵活性(TEAM)、过程成熟度(PMAT).Boehm整理这5个比例因子并给出了其参考值.

$EM$ 是工作量因子,在COCOMOII模型中有17个成本驱动因子,用来表示对项目开发工作量的影响程度.若 $EM>1.0$ ,表示该驱动因子将会产生较多的工作量,若 $EM<1.0$ ,表示该驱动因子将会减少工作量.Boehm<sup>[3]</sup>提出的成本驱动因子包括产品属性(REPL软件可靠性、DATA数据规模、CPLX产品复杂度、RUSE可复用开发、DOCU文档要求),平台属性(TIME执行时间约束、STOR主存储约束、PVOL平台易变性),人员属性(ACAP分析员能力、PCAP程序员能力、PCON人员连续性、APEX应用经验、PEXP平台经验、LETX语言和工具经验),项目属性(TOOL软件工具的使用、SITE多点开发、SCED要求的开发进度).Sharma<sup>[7]</sup>通过对大量的项目研究,得到了成本驱动因子取值.

标准开发进度(项目开发时间)估算公式为:

$$TDEV_{NS} = C \times (PM_{NS})^F \quad (3)$$

$$\begin{aligned} F &= D + 0.2 \times 0.01 \times \sum_{j=1}^5 SF_j \\ &= D + 0.2 \times (E - B) \end{aligned} \quad (4)$$

式中:  $TDEV_{NS}$ —标称开发进度;  $C$ —常数, 一般取 3.67;  $F$ —指数;  $D$ —工作量因子。

在 COCOMOII 中针对不同的项目, 有两种方法计算软件规模的大小, 最后都换算为源代码行来计算。这两种方法分别是直接估算软件的源代码数和估算软件未调整功能点、然后转化为源代码行。

基于 COCOMOII 模型的开发时间估算与成本估算如果有一些项目的历史数据, 可以根据历史数据来调整驱动因子的值和常数  $A$ 、 $B$ 、 $C$ 、 $D$  的值。具体项目开发时间估算流程如下:

步骤 1. 首先检查是否有历史数据, 若有, 根据历史数据调整因子和指数, 否则采用标准值。

步骤 2. 输入功能点, 通过确定的未调整功能点数, 将其转换为源代码行数。

步骤 3. 估算工作量和成本, 使用 COCOMOII 模型的估算公式 (1), 计算工作量并计算成本。

步骤 4. 估算开发时间, 使用 COCOMOII 模型的估算式 (3), 计算项目的开发时间, 结束。

## 2 改进 COCOMOII 模型的开发时间与成本估算的原则和驱动因子

### 2.1 改进 COCOMOII 模型的改进原则

由于 COCOMOII 模型的 17 个驱动因子没有综合考虑影响异地分布式敏捷软件开发的全部因素, 本文对其做了调整, 通过合理规划成本驱动因子来完成成本和工期的估算, 为异地分布式敏捷软件开发的成本和工期做出一个度量性的探究。

为了适应异地分布式敏捷软件开发环境, 需要适当地调整成本估算因子。对 COCOMOII 模型的相关成本驱动因子的调整, 可以通过项目统计或者专家评估, 依据一定的经验来进行<sup>[8]</sup>。调整应遵循以下 3 个原则<sup>[9]</sup>: 全局优先性原则、独立性原则、简单性原则。根据异地分布式敏捷软件开发的实际情况, 本文对模型的成本驱动因子进行分析、筛选和新增。

本文保留比例因子, 因为这 5 个因子是普遍软件都适用的规模经济型因子。确定成本驱动因子主要是为了体现项目的特点及项目的组织差异性<sup>[10]</sup>。因此本文对 COCOMOII 的改进主要是研究成本驱动因子然后进行工期估算。对成本驱动因子的改进主要体现在增强驱动因子的作用和新增驱动因子上。

### 2.2 改进 COCOMOII 模型的驱动因子

异地分布式敏捷开发是将敏捷开发过程应用于异

地分布式开发中, 加强在本地团队中的项目管理和协调, 敏捷过程的原则是个体与互动高于流程与工具, 可用软件高于详尽文档。根据异地分布式敏捷开发的这些特征, 对影响其时间与效率的因素进行解析研究, 改进了 COCOMOII 模型, 使得改进后的 COCOMOII 模型能够更好的估算异地分布式敏捷项目的开发与成本。

#### 1) 增强的驱动因子

根据筛选因子应遵循的原则, 确定需要增强的驱动因子如下:

数据规模 (DATA), 表示大量的测试数据对工作量的影响。增强原因是考虑到软件测试时间的约束, 由于开发时间短, 需要测试的工作量相对较大对项目的工作量有重要影响。其取值如表 1 所示。

表 1 DATA 驱动因子取值

因子级别	低	标称	高	很高
DATA 描述	$D/P < 10$	$10 \leq D/P < 100$	$100 \leq D/P < 1000$	$D/P > 1000$
DATA 取值	0.90	1.00	1.20	1.31

分析员能力 (ACAP), 表示设计和分析项目的水平。敏捷过程注重人的因素, 设计员需要了解体系结构, 创建容易分布在团队的体系结构导致在设计阶段使用更多的时间。其取值如表 2 所示。

表 2 ACAP 驱动因子取值

因子级别	很低	低	标称	高	很高
ACAP 设计分析水 描述	设计分析水 平相当低	设计分析 水平低	设计分析水 平一般	设计分析水 平较高	设计分析水 平相当高
ACAP 取值	1.44	1.18	1.00	0.84	0.70

人员连续性 (PCON) 体现了敏捷软件开发的灵活性特征, 表示每年项目组人员的灵活变动情况。异地分布式敏捷情境下需要多次频繁选择合格、可用的工程师, 人员流动性大。其取值如表 3 所示。

表 3 PCON 驱动因子取值

因子级别	极低	很低	低	标称	高
PCON 描述 (%)	$\geq 78$	46	24	12	6
PCON 取值	1.70	1.29	1.12	1.00	0.90

多站点开发 (SITE), 由于实现多站点开发, 将会导致项目组成员地理位置分布问题和交流问题。增强原因是由于实际距离、远距离沟通造成的项目成本上升。其取值如表 4 所示。



表4 SITE 驱动因子取值

因子级别	很低	低	标称	高	很高	超高
SITE 描述	困难, 仅电话	电话或传真	小范围	大范围, 电子沟通	大范围电子沟通, 偶尔视频交流	交互式多媒体交流
SITE 取值	1.12	1.07	1.00	0.92	0.85	0.78

## 2) 保留的驱动因子

部分软件开发项目的成本驱动因子, 要根据软件开发项目的实际情况来定, 所以要保留其驱动因子, 保留的驱动因子研究如下:

产品复杂性 (CPLX), 表示由于产品本身实现的复杂程度对工作量的影响. 在原模型中明确说明了该因子是不受国情或者环境影响的, 因此可以保留取标称值.

软件可靠性 (REPL), 表示可靠性对项目的影响, 主要由程序员本身的水平决定, 主要针对商业性质的软件, 一般商业软件主要面向普通用户, 可靠性要求不高, 因此作保留取标称值.

可复用性 (RUSE), 表示在现在或将来去构造复用的组件所需要的工作量, 由于分布式敏捷软件, 人员流动性大, 软件代码库的设立比较难, 一般也只能满足一个项目的使用, 保留为标称即可.

平台经验 (PEXP)、应用经验 (APEX) 以及语言与工具 (LTEX) 对于开发人员来说, 可能都要具备才能胜任软件开发的工作, 将其合并为经验因子 (EXP), 取 3 个的乘积. 其取值如表 5 所示.

表5 EXP 驱动因子的值

驱动因子	很低	低	标称	高	很高
EXP	1.74	1.31	1.00	0.73	0.58

程序员 (PCAP), 表示编程水平高低直接影响项目的工作量. 这个因子在 COCOMOII 有详细的介绍, 对于分布式敏捷软件开发也适用, 因此作保留取标称值.

软件工具使用 (TOOL), 表示在软件开发生命周期中对管理工具的使用熟练程度. 软件管理工具的应用熟练度差异较大, 敏捷开发不需考虑软件生命周期的全过程, 因此按实际估算取标称值即可.

平台易变性 (PVOL), 表示不同的开发项目需要不同的软硬件平台. 根据实际情况确定其具体值.

要求的开发进度 (SCED), 传统方法对开发进度的度量一般取决于实际经验, 较少采用软件工具, 异地分布式敏捷开发过程涉及多次快速迭代, 对开发进度要

求更高, 因此保留作为驱动因子.

## 3) 取消的驱动因子

由于原模型中的部分成本驱动因子不太符合异地分布式敏捷软件开发的情况, 考虑这些成本驱动因子的作用有限, 故取消一些成本驱动因子. 取消的驱动因子研究如下:

匹配生命周期的文档编制 (DOCU), 表示在软件开发项目执行过程中相关文档齐全. 由于分布式敏捷软件开发中, 团队成员相互熟悉工作内容, 充分了解项目进度, 要完成的文档极少, 故取消.

执行时间的约束 (TIME), 表示软件在系统上的执行时间大小, 目前由于处理器系统和计算机性能的提高, 处理速度明显提高, 因此对执行时间的约束可取消.

主存储时间约束 (STOR), 表示使用主存空间大小对项目的影响, 由于实际情况对主存空间大小的影响很小, 因此主存储时间约束可取消.

## 4) 考虑异地分布式敏捷开发工期与效率因素后新增的成本驱动因子

考虑影响异地分布式敏捷开发工期与效率的因素, 作为新增的成本驱动因子, 并根据影响的大小, 使用 PERT 方法评估具体值. 考虑的影响范围在 0 到 2 之间, 使用 PERT 评估出最佳值 ( $a$ )、可能值 ( $m$ ) 和悲观值 ( $b$ ) 进行加权求平均值来计算. PERT 评估计算公式为:

$$EV = \frac{(a + 4m + b)}{6} \quad (5)$$

团队协作性 (TC): 异地分布式团队成员之间相互协作, 考虑团队成员的实际距离和团队之间合作的关系, 主要是交流和文化适应度影响. 其取值如表 6 所示.

表6 TC 驱动因子的值

因子级别	很低	低	标称	高	很高
TC 描述	比较难以互动	难以互动	可以合作互动	主要合作互动	需要较高的合作互动
TC 平均值	1.45	1.21	1.00	0.9	0.76
因子等级	很低	低	标称	高	很高

项目周转 (PT): 异地分布式敏捷软件开发有更高的周转迭代, 相对于传统集中式开发, 需要更多地运用知识迁移保证交流. 其取值如表 7 所示.

表 7 PT 驱动因子的值

因子级别	很低	低	标称	高	很高
PT 描述 (%/年)	50	25	20	10	5
PT 平均值	0.86	0.94	1.00	1.21	1.30

项目负责人经验 (PME): 项目负责人是否具有管理分布式敏捷软件开发的经验, 他/她对项目交流、协作和控制团队活动、并制定开发计划具有重要的影响力. 其取值如表 8 所示.

表 8 PME 驱动因子值

因子级别	很低	低	标称	高	很高
PME 描述 (月)	<3	6	12	36	72
PME 平均值	1.32	1.12	1.00	0.89	0.78

### 3 改进 COCOMOII 模型的开发时间与成本估算方法

由改进模型的驱动因子, 可以确定对异地分布式敏捷软件开发项目进行估算的有效驱动因子为 10 个, 由于异地分布式开发中, 过程成熟度很低, 取值为 7.8, 可得到调整后的工作量估算公式为:

$$PM = A \times Size^E \times CPLX \times PEPL \times RUSE \times EXP \times PCAP \times TOOL \times PVOL \times SCED \times DATA \times ACAP \times PCON \times SITE \times TC \times PT \times PME \quad (6)$$

$$= A \times Size^E \times EXP \times PVOL \times SCED \times DATA \times ACAP \times PCON \times SITE \times TC \times PT \times PME$$

$$E = B + 0.01 \times (PREC + FLEX + RESL + TEAM) + 0.078 \quad (7)$$

开发时间的计算公式如下:

$$TDEV_{NS} = C \times (PM_{NS})^F \quad (8)$$

$$F = D + 0.2 \times 0.01 \times (PREC + FLEX + RESL + TEAM) + 0.0156 \quad (9)$$

$$= D + 0.2 \times (E - B)$$

改进 COCOMOII 模型未给出 A、B、C、D 具体的值, 只给出了基于异地分布式敏捷软件开发的计算公式和驱动因子的评估值. 判断模型的有效性, 可根据大量的异地分布式敏捷软件开发项目历史数据来验证, 对该模型中相应参数的校正也要根据具体的相关历史数据才能完全确定. 只要坚持统计具体历史数据, 不断的进行估算, 随着历史数据的不断积累, 模型将越来越有效, 估算也越来越准确.

### 4 改进 COCOMOII 模型的开发时间与成本估算的实现

根据改进 COCOMOII 模型的研究, 设计了基于 COCOMOII 的项目开发与成本估算方法. 在实际项目“蓝剑计划联盟项目”中进行了实现验证. 其实现流程如图 1 所示, 估算开始, 判断是否有历史数据, 若是, 则根据历史数据调整因子和指数, 若否, 根据改进模型中增强的驱动因子、保留的驱动因子和增加的驱动因子进行参数设置, 界面如图 2 所示. 然后进行规模估算, 可以采用直接估算软件的源代码数或先估算软

件未调整功能点再转换为源代码数. 根据公式 (6) 计算工作量, 根据公式 (8) 计算开发时间. 最后显示结果, 如图 3 所示. 项目负责人根据项目估算结果, 制定项目计划.

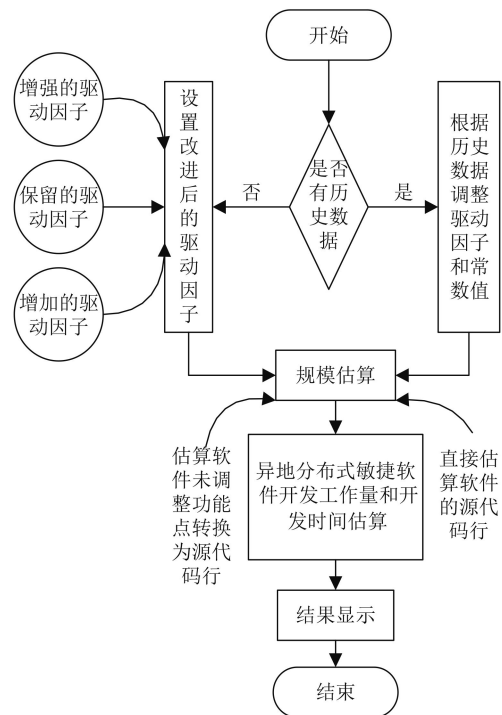


图 1 改进 COCOMOII 模型的开发时间估算的实现流程



图2 改进COCOMOII模型估算的参数设置

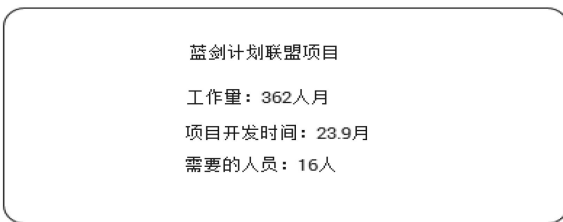


图3 输出项目的开发时间

改进模型的开发时间估算的核心代码如下图4所示。

```

1 public static const double INDEX_B = 0.91; //常数 B 可以根据实际情况调整
2 public static const double INDEX_A = 2.94; //常数 A 可以根据实际情况调整
3 public static const double INDEX_D = 3.67; //常数 D 可以根据实际情况调整
4 public static const double INDEX_C = 0.28; //常数 C 可以根据实际情况调整
5 //计算指数 E 的值
6 double index_E = INDEX_B + 0.01*sumSF (paramSF_Value);
7 //计算项目工作量
8 personMonth = INDEX_A * (Math.Pow(dataSize, index_E))*multiplyEM(paramEM);
9 //计算指数 F 的值
10 double index_F = INDEX_D + 0.2 * (index_E - INDEX_B);
11 //计算项目开发时间
12 double projectTDEV = INDEX_C * (Math.Pow(personMonth, index_F));
13 //计算需要的人员
14 int personNumber = Convert.ToInt16(personMonth / projectTDEV);
    
```

图4 改进COCOMOII模型的开发时间估算的核心代码

## 5 结论

本文主要解析了影响异地分布式敏捷软件开发的因素,研究了COCOMOII模型估算项目开发进度与成本估算的方法,针对传统的COCOMOII模型中开发与成本估算没有完全考虑异地分布式敏捷软件开发的重要因素的问题,综合考虑影响异地分布式开发的影响因子,根据异地分布式敏捷软件开发的实际情况,对模型的成本驱动因子进行分析,筛选和新增,并改进COCOMOII模型,使得改进后的COCOMOII模型能够合理地进行异地分布式敏捷软件开发的开发时间和成本估算。

## 参考文献

- 1 Carmel E. Global Software Teams: Collaborating Across Borders and Time Zones. Upper Saddle River: Prentice Hall PTR, 1999: 80-81.
- 2 Boehm BW. Software engineering economics. In: Broy M, Denert E, eds. Pioneers and Their Contributions to Software Engineering. Berlin, Heidelberg: Springer, 2001.
- 3 Huang XS, Ho D, Ren J, et al. Improving the COCOMO model using a neuro-fuzzy approach. Applied Soft Computing, 2007, 7(1): 29-40. [doi: 10.1016/j.asoc.2005.06.007]
- 4 In HP, Baik J, Kim S, et al. A quality-based cost estimation model for the product line life cycle. Communications of the ACM, 2006, 49(12): 85-88. [doi: 10.1145/1183236.1183273]
- 5 Boehm BW, Abts C, Brown AW, et al. Software Cost Estimation with COCOMO II. Upper Saddle River: Prentice Hall PTR, 2001: 1-4.
- 6 Yoon MY. An improved calibration method for the COCOMO II post-architecture model. Journal of the Korea Industrial Information Systems Research, 2000, 5(2): 47-55.
- 7 Sharma TN. Analysis of software cost estimation using COCOMO II. International Journal of Scientific & Engineering Research, 2011, 2(6): 1-5.
- 8 Menzies T, Port D, Chen ZH, et al. Validation methods for calibrating software effort models. Proceedings of the 27th International Conference on Software Engineering. Saint Louis, MO, USA. 2005. 587-595.
- 9 冯静, 朱小冬, 刘彦斌. 应用IFPUG功能点方法的军用软件规模估算. 计算机工程, 2006, 32(10): 50-52. [doi: 10.3969/j.issn.1007-130X.2006.10.018]
- 10 Sachan RK, Nigam A, Singh A, et al. Optimizing basic COCOMO model using simplified genetic algorithm. Procedia Computer Science, 2016, 89: 492-498. [doi: 10.1016/j.procs.2016.06.107]