

一种基于过程分解的工作量估算模型^{*}

汪琦 周福明 顾庆 陈道蓄

(南京大学软件新技术国家重点实验室 南京210093)

摘要 专家估算在工作量估算的实际运用中得到广泛的应用,但人们对其估算过程的透明性提出了更高的要求。针对专家估算在这方面的不足,本文提出了一种支持专家估算和类比估算的框架模型,借助 Delphi 方法,基于过程的分解来减小估算人员的主观影响。该框架简单实用,灵活性强,通过增加估算过程中客观因素的记录,使得估算过程具有较强的透明性和可重复性,适合软件组织实施和推广。

关键词 工作量估算,过程分解,Rational 统一过程,软件项目管理

A Software Project Effort Estimation Model Based on Process Decomposition

WANG Qi ZHOU Fu-Ming GU Qin CHEN Dao-Xu

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract The technique of expert estimation is widely used in the domain of effort estimation, while in most cases the transparency of the estimate process is unsatisfactory. In order to solve the problem of lacking transparency, an effort estimation framework based on process decomposition is put forward in this paper. This proposed framework supports expert judgments and analogy estimating. It uses Delphi method and process decomposition to reduce the subjectivity of the estimates, which makes the estimate process more transparent and more repeatable. The framework is concise, practicable, flexible and easy to popularize.

Keywords Effort estimation, Process decomposition, Rational unified process, Software project management

1 引言

近几年来,随着市场竞争的日趋激烈和信息技术的飞速发展,软件开发组织所开发项目的质量情况越来越受到关注。软件开发组织纷纷对软件过程的质量管理给予高度的重视,其中对软件开发项目的估算一直都是一个重要的课题,因为它可以用于项目的资源以及资源可使用的两种形式确定了项目的“边界”。专家学者们对估算模型做了许多研究,也提出了不少数学模型,但是在实际估算中,特别是工作量估算,最为流行的还是那些主观的、非结构化的估算方法,即所谓的专家法、类比法。

然而,运用主观的、非结构化的估算方法,估算效果往往取决于估算专家的观点,而估算过程又没有客观的依据,缺乏透明性。针对这一情况,业界往往采取一些措施来支持专家估算,比较流行的方法是让估算专家填写估算清单,召开估算组会议以减小估算过程中的主观影响,防止忽略某些重要的估算因素^[1]。

本文基于过程分解,为工作量估算中最为流行的专家估算法和类比估算法提供了一个模型框架,能够较好地解决估算过程透明性差的问题。本文将首先介绍传统的工作量估算方法,然后叙述基于过程分解进行估算的基本原理,在此基础上再详细说明基于过程分解的工作量估算模型,最后对模型的特性及效果进行讨论。

2 工作量估算方法

目前比较常用的工作量估算方法一般有三种:专家估算、类比估算和基于数学模型的估算。专家估算的思想就是由多位专家对工作量进行估算,取他们估算结果平均值作为估算

结果;类比估算可以说是专家估算的具体细化,通过与以前一个或多个项目比较来进行估算,类似的历史项目的工作量被用作对新项目的初始估算,然后根据二者之间的差异对估算进行调整;基于数学模型的估算往往提供一个估算方程,采用经验公式来预测软件项目的工作量。基于数学模型的估算中,模型一般用数学方式表达出估算所含的各种参数之间的关系,例如规模、工作量、进度、复杂度之间的关系,表达式中的控制变量一般都来自于已完成项目的分析数据^[2]。

对大量实际软件开发项目的调查显示,在软件开发的工作量估算中,使用专家估算法的占绝大多数^[3]。尽管专家学者提出了不少数学模型,但是经过测试,这些模型的估算效果很难令人满意,特别是对于中小型项目,往往即使在已经获得全部所需参数的情况下,估算结果与实际的工作量也完全不同,而且不同的估算模型得出的结果也大相径庭。所以很多学者认为,有丰富经验的专家估算要比基于数学模型估算的结果准确得多^[4]。

但是,专家估算法也存在着一些明显的缺陷,其中受指责最多的一点是:估算过程缺乏必要的透明性。所谓估算过程的透明性,是指在估算过程中留下客观的估算依据,外部可以通过这些信息了解估算过程的具体细节,在估算结束后有足够的信息可以重现该估算过程。针对缺乏透明性的不足,许多支持专家估算的方法应运而生,例如填写估算清单、专家组讨论、工作分解结构等,但始终没能很好地解决专家估算透明性差的问题。

3 基于过程分解的工作量估算

一个项目的工作量并不是由软件项目本身造成的,而是由对应的活动所造成的^[4]。实际开发中,开发过程的组织结构

^{*} 本文得到国家863项目支持,项目编号:2001AA113090。汪琦 硕士研究生,研究方向:软件估算,软件过程管理。周福明 硕士研究生,研究方向: workflow 系统,软件过程管理。顾庆 博士,副教授,研究方向:分布式计算。陈道蓄 博士生导师,研究方向:分布式计算与并行处理。

往往对一个软件产品的工作量大小起决定性作用,因此本文提出了一个基于过程分解的工作量估算模型。

3.1 软件项目的两种分解模式

该估算模型包含了在实施专家估算之前对软件项目的分解。一般来说,对软件项目的分解有两种模式:一种是按所需交付的产品分解,如组件、接口、界面等;另一种是按构建项目所需的各个活动分解。

3.2 基于过程的分解模式

在一个软件开发组织内部,所有的开发项目应当采取统一的分解模式,这样可以降低估算本身的工作量,而且有利于估算过程的重现和比较。然而由于项目的异构性,从产品的角度几乎不可能得到一个统一的分解模式。而基于过程使这种统一的分解模式成为可能,因为软件项目中总会有相同的活动需要执行;因此,开发过程被分解成一组标准化的过程组件,例如工作流和活动。于是对项目过程中所有过程组件工作量估算的总和就是整个项目工作量的估算结果。这种标准化的分解模式只有在公司的开发过程发生改变时才需要进行修改。

一个标准化的分解模式需要支持在项目的不同阶段进行估算,因此这种分解需要提供不同等级的详细程度。在项目的开始,由于缺乏对细节的了解,估算是非常粗糙的;但随着项目开发进程的深入,可以获得的细节越来越多,必然要求项目后期进行的估算比前期准确,这就需要过程分解得更详细。因此基于过程的分解能够与基于产品的一样,提供不同详细程度的分解。此外,基于过程还具有以下优势:

(1)在基于过程的分解模式中,过程模型定义了过程进行中需要执行的各种活动,一旦分解完成,过程所有层次的细节都被包含在分解结果当中。分解结构在项目过程中不会改变,变化的只是结构细节的详细程度,因此具有稳定性。一个稳定的分解模式使得开发计划具有更高的可信度,也有利于各估算项目之间的比较。而在基于产品的分解中,当分解完成时却无法得到所有关于产品的细节,于是分解随着项目的进展而改变,缺乏稳定性;当增添新的产品需求时,将不可避免地导致分解结构的变化。

(2)对项目规模进行准确的估算是相当困难的,特别是在项目早期,规模估算的失误更是工作量估算差错的主要来源。然而基于过程分解的估算,与基于数学模型的估算不同,规模估算不再是工作量估算的前提条件,对工作量估算结果的影响也大大减小。

(3)基于过程分解的估算直接支持项目过程管理,一个具体到活动层次的估算过程使得项目经理可以在活动层次上将实际结果与估算值进行比较,从而具备更好的项目控制能力。

3.3 RUP 过程分解

在实施专家估算之前对软件项目过程的分解,可以使用任何基于过程的软件开发模型作为分解的基本模型,本文中的一个实例将使用 Rational 统一过程(RUP, Rational Unified Process)。RUP 是一套基于统一建模语言的软件过程框架,各个软件组织可根据自身的实际情况,以及项目规模对 RUP 进行裁剪和修改,以制定出合乎需要的软件过程。如图1中所示^[5],横坐标代表 RUP 的软件生命周期随着时间分为四个依次进行的阶段:先启(Inception)、精化(Elaboration)、构建(Construction)、产品化(Transition);纵向坐标上的工作流(workflows)从技术角度描述 RUP 的静态组成部分,可进一步细化到活动(activities)、工件(artifacts)、角色(workers)。RUP 体现了软件开发过程的一种标准分解结构,分为工作流和活动两个层次。一个软件项目包含9个核心的工作流:业务

建模、需求分析、分析设计、实施、测试、部署、配置与变更管理、项目管理和环境。每个工作流又由八至十五个活动组成。

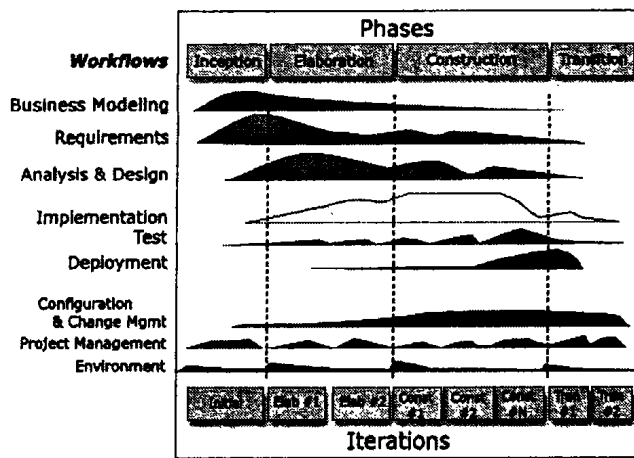


图1 RUP 的过程框架

3.4 一个过程分解实例

本文提出的估算模型中,RUP 过程分解的结果是一张表格,包含了组成软件开发过程的工作流和活动信息。表格1以业务建模工作流为例展示了 RUP 中的过程分解情况,并为业务建模工作流及其相关活动提供了一组参数作为专家估算的参考。业务建模工作流包含了表格中的14个活动,其中角色作为相关信息提供给估算人员,另外如果需要,还可以附上各活动生成的工件作为参考。

表1中工作流参数一栏和活动参数一栏中的参数分别用来支持在工作流层次上的估算和活动层次上的估算。这些参数刻画了工作流和活动工作量大小的一些相关信息,为估算专家的主观判断提供了客观依据。为了满足专家估算的需要,这些参数的大小必须比过程组件本身更容易被估算,因此它们一般是一个容易度量的数值,比过程组件本身容易量化。

本文提出的基于过程分解的估算模型具有很强的灵活性,在项目生命周期的任意阶段,都可以根据项目的进展情况进行估算。在开始阶段,项目的详细信息还无法获取,因此只能在工作流层面进行估算;随着开发项目的进展,得到项目在活动层面上的详细信息,便可以在活动层面上进行更精确的估算。而且,在一次估算过程中,可以对项目中信息掌握度高的部分进行活动层次上的估算,对信息掌握度低的部分进行工作流层次上的估算。

以表1中为例,如果是在工作流层次上的估算,业务建模工作流本身作为一个过程组件成为估算的对象;而在活动层次上的估算,业务建模工作流的每一个活动就作为一个过程组件成为估算的对象。显然在活动层次上的估算可以得到更精确的估算结果,但需要的关于项目的信息也更详细。对业务建模工作流而言,即使在项目刚开始也可以在活动层次上进行估算,因此工作流参数与活动参数很接近,但对于其他一些工作流,如实现工作流和测试工作流,它们的工作流参数和活动参数则有着明显的区别,在项目早期由于无法获得活动层次上的详细信息,因此只能进行工作流层次上的估算。

以活动“评估目标组织”为例,为了支持专家对该活动工作量的估算,引进了“涉众数量”作为参数。涉众是指会受到系统结果重大影响的个人,这个值在项目早期就可以度量,而且要比从项目角度评估目标组织的工作量容易得多。一个只有4个涉众的项目和一个有24个涉众的项目在评估目标组织上的工作量是有非常大的区别的。而对这个区别度的把握则

由估算专家们根据经验来判断,参数只是提供一个估算的客观依据,与估算结果并不存在函数上的关系,这是本模型中参

数与基于模型估算法中参数的根本不同之处。

表1 RUP 业务建模工作流的标准化分解

workflow	workflow参数	活动	活动参数	角色
业务建模 workflow	涉众数量 目标组织规模 业务建模规模 业务用例包的规模	评估目标组织	涉众数量,目标组织规模	业务流程分析员
		设定和调整目标	涉众数量,目标组织规模	
		制定业务规则	业务用例包的规模	
		获取常用业务词汇	业务用例包的规模	
		查找业务主角和用例	业务主角数量,业务用例包的规模	
		建立业务用例模型	业务用例包的规模	
	业务角色数量 业务实体数量	定义业务构架	业务主角数量,业务用例包的规模	业务设计员
		详细说明业务用例	业务用例包的规模	
		查找业务角色和实体	业务角色数量,业务实体数量	
		详细说明业务角色	业务角色数量	
	业务用例包的规模	详细说明业务实体	业务实体数量	业务模型复审员
		定义自动化需求	业务用例包的规模	
	复审业务用例模型	业务用例包的规模		
	复审业务对象模型	涉众数量,业务用例包的规模		

4 基于过程分解的估算模型

4.1 概述

本文提出的工作量估算模型建立在一个标准化的过程分解基础之上,分解的结果就是专家估算的对象。根据对估算项目细节掌握情况的不同,可以从两种详细程度对项目进行分解: workflow层和活动级。

为了适应软件组织估算水平的不同,根据标准化程度高低的不同,该模型分为基础模型和高级模型。高级模型是基础模型的扩展,具备更多的客观依据和更好的透明性,当然对软件组织估算水平的要求也越高。

该模型本质是基于专家估算法的。由于专家估算法缺乏一个稳定的框架,而且无法提供可重复、可理解的估算过程,这种方法常常被认为是主观的,不透明的^[6]。然而,考虑到专家估算法的易实现性和广泛应用,如果能够给予透明性好、易于理解的过程框架的支持,那么专家法在实际运用中将更受欢迎。本文正是为支持专家估算法提供了一种框架模型,改进估算过程的透明性,并引入类比估算的一些特征,使得原本主观的估算过程具备更多的客观因素,更适应软件组织的需要。

4.2 两种模型

对软件组织而言,该估算模型根据标准化程度要求的不同有两种选择:基础模型和高级模型。高级模型建立在基础模型之上,加入了类比估算的特征。如图2所示,虚线部分就是两种模型的区别所在。

4.2.1 基础模型 基础模型不包括图2中虚线框的部分,整个流程借用传统的 Delphi 方法^[2],根据适当的估算层次对项目进行标准化的过程分解得到相应的过程组件,然后根据引入的参数结构对相关参数进行量化。以量化结果为依据,每一位估算专家独立地对每个组件都给出最小值、最可能值和最大值三种工作量估算结果。这三种结果在专家们汇总他们各组件的估算结果时将会被加权统计,并经过计算讨论得出每个组件的最终估算值以及标准方差;如果方差超出预定的域值,专家们将各自重新对该组件进行估算,再加权统计,直至方差减小到可接受范围之内。最后,再把对所有过程组件的估算结果进行整合,就得到对整个项目的工作量估算以及方差。

基础模型通过对参数的量化对组件进行估算,提高了估算过程的可理解性,并且对于结果出现较大分歧的情况(即方差超出预设域值)提供了一个可重复的估算过程。最后的估算结果不仅仅是一个值,而是一个最可能值和相应的标准方差,这体现了估算活动本身的不确定性。整个流程中,根据不同的估算要求可以使用两种估算详细程度: workflow层和活动层。

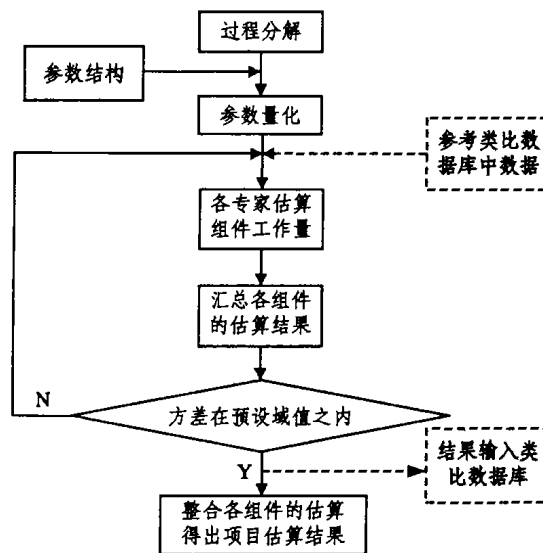


图2 基于过程分解的工作量估算模型

以此模型为框架的专家估算法与以往方法相比,实现简单、运用灵活,估算框架具有一定的规范,结果也更容易被理解,并且估算过程可以重复,具备更好的透明性,能够有效避免传统专家估算法过分依靠主观判断的缺陷。

4.2.2 高级模型 尽管和基础模型采用相同的过程分解和参数量化措施,高级模型的可重复性和可理解性得到进一步改善。这是因为在高级模型中加入了类比估算的特征——历史项目数据库,专家们通过查询数据库,利用相似项目的经验数据对当前项目的过程组件进行估算。

如图所示,该模型的流程与基础模型基本一样,在基于标准化的过程分解之后,专家们根据量化参数对各个过程组件的工作量进行估算。不同的是,高级模型中建立了一个历史项目数据库,其中保存了以往估算项目的相关数据,包括整个项

目估算的工作量和实际工作量,还有项目各个组件的参数值、估算工作量和实际工作量,为专家们在估算时提供了相似项目的数据以供参考,有效提高了估算的准确性,并使得估算过程更加透明。这种与历史项目的类比不仅是项目级别上的类比,更包括过程组件级别上的类比。因此,高级模型同样可以适用于 workflow 层和活动层两种分解详细程度。无论分解后的过程组件是 workflow 还是活动,量化的参数值都体现了该组件的属性特征,专家们可以通过在数据库中搜索相近属性的参数值来搜索相似的历史项目,再利用经验通过新旧项目的比较,决定新的估算项目中过程组件的工作量与历史项目相比是大于、小于还是等于。

与基础模型相比,运用高级模型的估算具有以下几点优势:

(1)类比的引入进一步降低了专家们主观判断对估算结果的影响,使估算过程更加透明,可重复性更好;

(2)专家们利用过去类似项目的估算结果,以此作为共同的估算起点,这将有利于减小最后估算结果的标准方差;

(3)在新的估算活动中可以吸取以往项目的经验教训,有利于估算人员逐步提升估算技能,改善软件组织的估算水平。

一个能够为项目类比提供足够历史数据的数据库是使用高级模型的先决条件。建立这样一个数据库需要数月甚至数年的时间,而在此之前,软件组织可以先使用基础模型,对于每个项目,把工作量估算结果和项目完成的实际情况全部输入数据库。当记录在数据库里的历史数据累积到一定数量之后,就可以扩展到高级模型来改善估算过程的准确性和透明性。

4.3 模型效果分析

上文提出的包含基础模型和高级模型的一套估算框架,运用基于过程分解项目的思想,在项目开发的不同阶段可以对项目进行两种详细程度不同的分解,然后自底向上,通过对分解后各过程组件工作量的估算来实现对整个项目工作量的估算,能够有效减小估算本身的工作量,同时增加项目计划的稳定性,使得估算过程具有更好的透明性。基于过程分解使得软件规模的估算不再是工作量估算的先决条件,一方面减轻了估算人员的负担,另一方面有效降低了规模估算所造成的误差对工作量估算影响。而且,基于过程的分解支持项目过程管理,提高了项目经理对项目的控制能力。

在经典的 Delphi 方法中,专家们以相同的假设条件进行估算,直至他们各自提交结果,这期间他们的估算过程完全是不透明的,协调人只能根据各结果进行总结,对于结果差异过大的组件只能请各专家重新估算。因为缺乏必要的依据,重新估算的专家们并不知道问题出在哪里,所以多次重复估算后,结果未必能令人满意。

而应用本模型的专家估算,由于量化参数的引入,专家们的估算过程留下了依据,估算过程的透明性得到明显改善。以上文对业务建模工作流的估算为例,在汇总“定义业务构架”活动的工作量时,各专家的结果有据可依,讨论的重点放在业

务主角数量和业务用例包规模与该活动工作量大小的对应关系上,因此如果需要对该活动重新估算,专家们也知道估算的差异所在,有利于改进自己的估算过程。若是应用高级模型,情况得到进一步改善,因为在估算“定义业务构架”活动的工作量时,除了量化的参数外还引入了历史数据,即以往类似项目中“定义业务框架”活动的工作量大小和业务主角数量及业务用例包规模的数量对应关系。因为有相同历史数据作为专家们估算的起点,估算过程具备更好的透明性,方差超出预定域值的概率也大大减小。

这种基于标准过程分解的自底向上的估算方法在工作量估算领域还是很少见的。通常的分解估算技术没有一个标准的分解过程,也没有一个关于分解后的估算对象如何被估算的方法框架。功能点分析(FPA, Function Point Analysis)的方法尽管采用了自底向上分解的方法^[5],但仅局限于规模估算领域,并不涉及工作量估算。

总结与展望 本文在 Delphi 估算方法的基础上,引入参数量化、类比估算等机制,基于过程分解提出了一种支持专家估算法的工作量估算模型。使用该模型的估算过程具有较好的透明性和可重复性,同时易于理解。由于本估算框架的基础模型并不以经验数据为基础,因此可以适用于任何规模的软件组织。不过,简单、灵活的特点使得该框架更加适合中小规模的软件组织的工作量估算。目前,这一估算模型框架已经被应用于一项正在实施的软件研发项目中。该项目是一个基于 Web B/S 结构的面向 CMM 的企业级软件过程支持系统,其目标是为了辅助软件企业在 CMM 的框架下实施软件过程的管理,目前,第二版本的设计和开发工作已接近尾声。项目开发中包含一个可以独立使用的软件估算工具,支持本文提出的估算框架中的基础模型,在实际应用中取得了很好的效果。不过对于整个框架的估算效果的评估,还需要进行更深入的研究和实验。

参考文献

- 1 Jørgensen M. A review of studies on expert estimation of software development effort. *Journal of Systems and Software*, 2004, accepted for publication
- 2 Boehm B. *Software Engineering Economics*, Englewood Cliffs, 1981
- 3 Wrigley C D, Dexter A S. Software development estimation models: A review and critique. In: *Proc. of the ASAC Conf.* University of Toronto, 1987. 125~138
- 4 Hull M, et al. Software development processes-an assessment. *Information and Software Technology*, 2002, 44(1): 1~12
- 5 Kruchten P. *The Rational Unified Process*, Reading/Mass., 1998
- 6 Hughes R T. Expert judgment as an estimating method. *Information and Software Technology*, 1996, 38(1): 67~75
- 7 Boehm R. Function Point FAQ. at <http://ourworld.compuserve.com/homepages/softcomp/fpfaq.htm>