

试论软件过程模型及其重要性*

余 金 山

(华侨大学计算机科学系,泉州 362011)

摘要 讨论软件过程模型的基本概念、实质含义和研究范畴,阐述软件过程及其模型的重要性。

关键词 软件过程,软件过程模型,软件生命周期,软件开发,软件管理

分类号 TP 311.5

软件过程模型(Software Process Model)这一基本概念的建立,被认为是软件工程的一次重要突破^[1],对缓和软件危机起了重要的积极作用。软件过程模型亦称软件加工模型或软件生命周期模型(Software Life Cycle Model)。软件过程模型不但体现客观世界中工程问题的一般性的求解过程,而且体现软件开发本身所具有的许多特殊性。多年来,软件开发者在实践中成功地应用它解决软件开发中存在的许多严重问题。随着经验的不断积累和软件工程其它领域的迅速发展,软件过程模型也在不断发展和完善。本文论述软件过程模型的基本概念、实质含义、研究范畴及其重要性。

1 软件过程模型的实质含义及重要性分析

1.1 基本概念

(1) 软件(ISO/IEEE),是与计算机系统操作有关的程序、过程、规则以及任何有关资料的完整集合。(2) 软件生命周期(IEEE),是从设计软件产品开始到产品不能再使用时为止的时间周期。典型地包括了需求分析、设计、实现、测试、安装和验收、运行和维护等阶段,有时还包括退役阶段。(3) 软件生命周期模型,是在软件生命周期中按照预定的模式或步骤,使用某些特定的规则和方法,配以适当的软件工具,进行软件产品的开发、作用和维护^[2]。(4) 软件过程和软件过程模型,软件过程,是指软件开发人员的一组活动、方法和实践的集合。软件过程模型,是软件过程的一种宏观的控制结构,它充分且明确地表示了一组特定过程的共同属性。

1.2 软件过程和软件生命周期

软件开发,是把客观世界中用户的要求变换成计算机世界中的软件产品,而软件产品投入运行后又不断进行维护。维护的实质上,把新的用户需求或原产品中不符合要求的地方

* 本文 1993-08-01 收到

再变换成软件的一种活动(附图)。因此,软件生命周期(SLC)的定义是明白的,它是一种静态的观点,软件过程(SP)则是一种动态的观点,是软件生命周期的一种“过程”。从软件工程知识体系^[3]的观点分析,而软件过程包括软件开发、控制、管理、操作与维护四种不同的活动,每种活动,由抽象

技术、表现技术、软件方法、软件工具、评价、通讯等六个方面组成。因此,软件过程可看作是一个控制和指导软件开发的软件。显然,深入分析和研究 SP 本身是从根本上解决软件生产率和软件质量的关键。

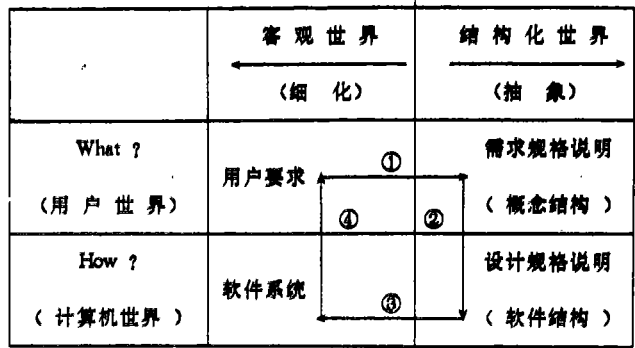
1.3 软件过程模型和软件生命周期模型

软件过程模型(SPM)和软件生命周期模型(SLCM)的含义是相同的,一般均不作严格区分。SPM 或 SLCM 是软件过程研究的一个方面,是一种用来表现软件整个生命周期的模型。

软件过程模型,主要研究软件开发中各种宏观结构(概念结构、软件结构、程序代码)的构造以及它们之间的转换过程。不同过程模型,有着共同的目标(把用户要求变换成软件)和共同的基本步骤,如图 1 所示。软件过程模型不同于软件方法,其主要作用:确定软件开发或演化所涉及的各个阶段的顺序;建立阶段之间的过渡准则,包括标志当前阶段完成的准则和进入下一阶段的选择准则。因此对一个软件项目,过程模型至少应回答两个问题:(1)下一步做什么;(2)该项工作应持续多长时间。然而,软件方法和重点在于如何指导每一阶段的工作,如何进行和表示各个阶段的产品,例如功能分割、确定数据流和控制流、使用结构图或状态转换图等等。软件过程模型是软件方法和软件工具研究与应用的基础,但软件过程模型必须有一定的方法与工具的支持才能充分发挥该模型的优点和作用。当然,它们之间的研究和发展也是相互促进的。下面介绍两个软件过程模型,进一步说明 SPM 的实质含义和作用。

瀑布模型^[4],是软件产业界中应用最广并已成为一种传统和标准的经典性模型。结构化方法是与其相应的软件方法。该模型的特点和使用规则是:(1)把 SLC 划分成若干严格定义的顺利阶段;(2)只有在前阶段的阶段性产品完成后才能开始下一阶段的工作;(3)每阶段都要配合生成相应的具体文档,它们类似于其它工程中的图纸、资料,是管理和下一步工作的依据,极为重要;(4)每个阶段的结束都要经过严格的定量或定性复审和确认。

喷泉模型(Fountain Model),是随着当今大有风靡全球之势的面向对象(Object-Oriented)的软件方法的研究和进展而提出的。该模型的若干主要特点:(1)虽仍然沿用瀑布模型中的“阶段”概念,但强调了阶段之间的大量重叠和反复,即阶段之间已无明显的界限;(2)簇/类/模块的开发可与系统的开发并行且互相影响;(3)要求分析和规格说明仍然是系统开发的基础;(4)模型不但适用于整个系统的开发而且更适用于簇/类/模块的开发;(5)显式地引入了聚集(Aggregation)和概括(Generalization)阶段;(6)模型置根于客观世界不断向上“喷射”,外形上象个喷泉。



附图 软件开发的宏观结构及转换

从以上的分析可以看到,软件过程模型给出了看待软件的一种全局性的系统观点.其重要作用和意义可归纳为如下九个方面.(1)强调了“软件系统存在于一个较长的时间周期内”这一事实,从而提醒人们要注意软件生命周期的各个不同阶段,而不仅仅是“编码”.(2)为研究软件产品的投入与产出以及提高软件产品的效益,提供一种框架和一定的方法.例如,对软件产品效益曲线和所谓的“长寿技术”的研究.(3)可以根据软件开发中不同阶段的特点和任务,研究和应用相应的技术、方法与工具.(4)使得人们对软件及其开发能有一个系统性和深刻的了解.(5)为工程化研制软件提供一个框架,使人们能按一定的模式有条不紊地、规范化地进行软件生产的管理和开发.(6)提供一种把软件生产这一高度复杂的过程进行分解和抽象的有效途径.(7)使各类人员,对工作的基本步骤、目标以及在不同时期的具体任务和活动十分明确.(8)使得软件开发的组织机构更具结构化和合理化.(9)软件过程模型既然是一种模型,那么它应具有模型的一般要求和作用:(a)真实性.能确切反映和符合系统的客观实际;(b)简洁性.在真实性的基础上应该简单明了、抽象,并可以进行实验或/和的逻辑处理;(c)适应性.随着模型应用条件的变化,具有一定的适应能力;(d)在实际系统建立前,可凭借模型来有效地求系统设计所需的参数,并据此确定各种制约条件;(e)可用来预测各个替代方案的性能、费用和效益,以利于各种方案的比较;(f)能获得表达系统基本性质的形式.利用这些形式进行分析,可最后确定系统的性质和设计;(g)通过对模型的分析 and 计算,为研究系统有关的技术和经济指标,提供必要的信息.

2 从 SPM 的发展和应用看软件过程模型的重要性

软件过程模型的发展,在本质上体现了人们对软件开发这一认识的提高和飞跃.一些专家认为,软件开发可划分为四个时代.(1)第一代(70年代以前).这个时期尚无软件生命周期的概念,也无特定的软件方法论和技术.(2)第二代(70年代到80年代初).这个时期的特点是在软件生命周期模型下的结构化方法.(3)第三代(80年代中期——?).这个时期以新的观点重新看待软件开发,提出许多不同的生命周期模型,并从不同角度采用新的技术和方法.其中,具有代表性的是快速原型模型和面向对象的方法.(4)第四代(21世纪,展望).以面向并行对象模型和知识信息处理软件为特点.以下我们以几个典型的 SPM 为例,简要地说明软件过程模型的发展及其对软件开发实践的影响.

2.1 无模型时期.

这个时期尚无软件生命周期的思想,没有明确的或根本没有需求分析、设计、测试、维护的概念,也无编写文档要求的习惯.软件开发的任务主要是编写代码,因而也有人称这样的软件开发是基于编码-修正(Code-and-fix)模型的,其基于这种模型开发出来的软件存在着普遍问题:(1)无法满足用户需求且应用范围十分有限(几乎是专人专用的);(2)软件既结构可言,又无文档资料,因而测试、修改、维护均十分困难,代价异常的高昂;(3)软件质量及生产率极低,是导致软件危机的主要原因之一.下面列出它们的4个典型实例^[4].(1)在两个大型的指挥控制系统中,由于不能满足用户要求,有67%~95%的软件在交付使用后不得不重新编写.(2)Univac-United 航空公司机票预定系统耗资5600万美元;美国某先生的后勤系统耗资近 2.2×10^4 万美元,由于无法实现预期目标而被迫取消.(3)据设在英国的IBM公司于1966年10月统计,他们花费30亿美元,只生产出500万条指令.(4)IBM OS/360系统的困境,其负责人

Brooks 生动地把它描述为“在泥潭中作垂死挣扎的巨兽”。

2.2 Waterfall Model 的提出及应用

瀑布模型的特点和作用,前面已作叙述。若从软件工程经济学的观点分析,使用瀑布模型还有另外两条支持理由:(1)为了得到成功的软件产品,无论如何必须在某一阶段实现所有的子目标;(2)对应实现的子目标的顺序凡作任何调整,都会导致较差的软件产品,而瀑布模型却给出这些子目标的正确顺序,且是“每一时刻针对一项子目标”的。值得强调的是,列举实例已不足于显示瀑布模型的重要性,然而“瀑布模型在软件开发中至今仍占据领导地位.它已发展成为一种标准的、为人们广泛接受的经典性模型”的这一事实,充分地、有力地说明了瀑布模型的成功之处及其对软件开发实践的重要作用。

2.3 快速原型模型(Rapid Prototyping Model)

快速原型是为克服瀑布模型的缺点,在 80 年代初提出的一种著名的软件过程新模型。经过一段时间的研究、发展和实践,人们逐渐认识到瀑布模型也存在许多不足之处。突出的问题是:它无法适应用户需求与软件开发过程具有高度的动态性这一客观事实,因而不但导致了高昂的修改代价和维护代价,而且拖延了开发进度。实际的统计分析指出:软件的需求变化平均为 25%(IBM);由需求变化引起的生产率变化范围为 1.78(COCOMO);某些大型项目因需求变化造成的生产率损失因子为 4。需求的易变性,导致软件开发失败的典型例子有:(1)某雷达应用软件^[3]因需求变化不得不重新开发的软件达 318%(交付 33 kB 指令,但废弃的指令达 105 kB),成本递增 3 倍;(2)法国某气象卫星软件^[4],由于规格说明不完善,从而导致在某次运行中错误地把 141 个高空气象气球中的 73 个摧毁。

快速原型的思想是:强调用户合作,以尽量小的代价和尽快的速度建立目标系统的一个原型,通过原型的实际运行、评价和完善,揭示和适应需求的模糊性与易变性,并对系统的设计方案进行审定。因此,快速原型模型的最大优点,在于能快速且廉价地提供一种连续的且非常有效的反馈机制,供动态地求精以及完善需求描述和系统设计。同时,它能使用户的验证提前,及早排错;能使许多原来必须在维护阶段进行的工作提早进行,从而提高软件生产率和软件质量。然而,快速原型模型也不完善,仍有不足之处。从实践结果分析,下面是一些有说服力的事实和经验:(1)与瀑布模型相比较,在开发小型的商用软件中,两者的平均生产率(以 DSI/MH 计)相同,在性能大致相当的前提下,快速原型的产物可减少约 40%的 DSI 并节省 40%的人时;瀑布模型项目在排错和完整性方面困难较少;(2)快速原型模型不适用于某些领域,如大多数的实时分布系统、指挥控制系统和大型的集成化信息系统;(3)原型并非万能的,规格说明仍十分重要;(4)快速原型模型在中、大型系统中的应用,尚有待实践与研究。

2.4 面向对象方法与喷泉模型

喷泉模型(FM)是由 B. Henderson 等人于 1990 年提出的一种新模型,其目的是用于支持面向对象的软件开发。面向对象方法是一种全新的软件方法,它一反传统的面向功能的软件开发风范,用一种截然不同的世界观和方法论来看待问题和解决问题。即把问题领域看成是由一些相关的对象所组成,并用计算机中的对象对客观对象进行模拟,从而得到问题的解空间。需要指出的是,FM 作为一种新模型,尚有待进一步研究,其作用也有待实践的检验。

2.5 其它模型

其它一些有较大影响的 SPM,是螺旋模型(Spiral Model)、渐增式(Incremental)模型、重

用(Reuse)模型和自动变换(Automatic Transform)模型.螺旋模型与渐增式模型,已被成功地应用于若干大型的复杂系统的开发,如成本高达1亿美元的某阵地防卫系统(渐增式模型)、具有1300万行源令的TRW-SPS系统(螺旋模型).软件重用和自动变换,被认为是提高软件生产率和软件质量的根本途径.尽管目前在技术上仍困难重重,但在某些小型项目中,4GL或应用生成器的应用,减少了60%的DSI和60%的人时.日本的一些实践经验也指出,软件重用可提高生产率25%~60%.

3 结论

对软件过程及其模型的研究,是对软件开发的本质及其原理的研究.软件过程及其模型是指导思想,也是软件开发实践的全局性框架.有关概念的提出和应用,大幅度地提高了软件生产率和软件质量,它的重要作用远超过数据模型对数据库的影响.更具现实意义的是,目前我国软件产业的发展正处于一个关键的时期.软件人员在软件开发中应充分认识到软件过程模型的重要性;项目负责人应根据具体项目、具体环境以及其它约束条件,选择合适的过程模型并正确地加以运用.不同的过程模型的确定将直接影响到软件开发的各个方面,包括开发速度,开发成本,项目的预期、控制、配置和管理,产品的质量,等等.

参 考 文 献

- 1 张维业,扬芙清.软件生命周期模型与快速原型.计算机科学,1989,16(1),27~32
- 2 贾耀良.快速成型方法及其支持工具,软件产业,1989,1~8
- 3 余金山.软件工程学科建设的若干问题,计算机科学,1991,18(4),69~74
- 4 Boehm B W. Software engineering economics Englewood Cliffs, NJ, Prentice-Hall, Inc., 1981. 50~150

On the Software Process Model and Its Importance

Yu Jinshan

(Dept of Computer Science, Huaqiao Univ., 362011, Quanzhou)

Abstract The Author deals with the software process model. Its basic concept is briefed; its substantial implications and scope of study are discussed; and its importance is expounded.

Keywords software process model, software life cycle, software development, software management