

软件项目管理与精简并行过程

Simplified Parallel Process and Software Project Management

林 锐 (上海漫索计算机科技有限公司 201204)

董 军 (上海华东师范大学软件学院 200062)

摘要:本文简要回顾了软件工程的历史,概述了软件过程和能力成熟度模型的基本概念,介绍了目前的动向。在能力成熟度模型集成的基础上,面向软件开发的实际需要,给出了一种软件项目管理的策略。

关键词:软件项目管理 软件工程 软件过程 精简并行过程

1 引言

据介绍,1962年6月,美国飞向金星的第一个空间探测器水手I号,因其飞舱中的计算机程序之一的一个语句的语义出错,致使偏离航线无法成功。

其后,随着软件规模的扩大和对可靠性要求的增加,软件开发产生了越来越多的问题:程序质量低下,错误频出,进度无法保证,维护费用不断上升,可靠性得不到保障……。这些问题统称为“软件危机”。

在美国,20世纪70年代中期,失败的软件项目中,70%是由于管理不善引起的。90年代中期,2500亿美元用于175000个软件项目,其中31%的项目在完成前被取消,费用为810亿美元;53%的项目的费用是原估计费用的190%;只有10%的项目在预定的费用和进度下交付。后来,每年有2750亿美元用于200000个软件项目上,有16%的项目能按时并按预算完成。

然而,印度因为较多的软件企业通过能力成熟度模型(CMM)论证,2000~2001年度的软件产值为86亿美元,软件出口额达63亿美元。《财富》500强中有185家公司在印度采购软件产品,印度有1250家公司出口软件^[1]。中国软件企业长期缺乏一套国际标准的软件质量保证体系是十分关键的问题。

2 软件工程

软件^[3]有着如下特征:

- * 软件是被开发的,而不是象传统意义下那样制造的;
- * 软件无“磨损”;
- * 尽管工业是朝着基于组件装配方向发展的,但大多数软件还是客户定制的。

而软件本身有着双重作用:

- * 作为产品:生产、管理、获取、修改、显示或传播信息;
- * 作为分发产品的载体:操作系统、网络和软件工具与

环境。

然而,软件开发中始终困扰人们的问题是:

- * 为何要如此长的时间来完成软件?
- * 为何开发费用这么高?
- * 为何在把软件交给用户前不能发现所有错误?
- * 为何随着软件的开发在测量进展时不断遇到困难?

为了解决问题,1968年“软件工程”被正式提出,目的是用工程化的方法开发软件以图解决软件危机。软件工程尚无一致的定义。IEEE定义软件工程为:(1)应用系统化、纪律化和量化的方法进行软件开发、操作和维护;(2)对(1)中方法的研究。

软件工程必须回答下列问题:

- * 什么是要解决的问题?
- * 用于解决问题的实体(软件)的特征是什么?
- * 实体和解决方案如何实现?
- * 如何构造实体?
- * 用什么方法揭示实体设计和构造中的错误?
- * 在需要更正、适应和提高时,实体如何被长期支持?

而整个软件工程可分为三个阶段。

* 定义阶段:系统和信息工程,软件项目规划,需求分析。

* 开发阶段:软件设计、代码生成和软件测试。

* 支撑阶段:设计四种类型的改变:更正、适应、提高以及预防。本质上,预防性维护对计算机程序作改变,使之更容易被更正、适应、提高。

经过结构程序设计的讨论,20世纪70年代开始,人们认识到单纯寄希望于高级语言及其编译系统是不够的,还与方法论有关,从而有了“软件自动化”及“计算机辅助软件工程”思想。这里,集成化^[4]是核心的:

- * 松散的集成化:信息集成;
- * 紧耦合的集成:反映工具间的内在联系

此间,北美工业界强调软件技术。70 年代到 80 年代中期,希望通过提高自动化水平提高生产率。由于大型软件结构复杂,难以找到统一的风格和步骤;80 年代中后期,强调重用。但对语义理论过分忽略。

而西欧学术界一贯认为提高软件生产率的关键是提高软件的可靠性与可维护性。前者强调语义精确的规范语言,后者强调软件开发的逐步求精过程的语义一致性。这里,重要的是选取适当的软件体系结构,以与模块相对应的组件为基础。他们技术上强调模块可重用性和理论上强调语义可靠性这两种软件工程方向的合理结合。

在实际开发中,有不同的软件生命周期模型可遵循:

- * 线性模型
- * 原形模型
- * 快速应用开发模型
- * 进化软件过程模型:增量模型,螺旋模型,并行开发模型
- * 基于组件的模型:合并了很多螺旋模型的特征,UML
- * 形式化方法模型:由于目前它比较花时间和昂贵,有足够背景的开发人员还很少,用户也不习惯。形式化方法尚未成为主流方法,但它为无缺陷软件提供了希望。

简言之,软件工程是集成了适合于计算机软件开发的过程、方法和工具的学科。三十余年来,软件工程逐步与计算机科学、计算机工程和信息系统一起成为并列的方向,形成相对独立的知识体系。

3 软件过程与能力成熟度模型

回顾软件工程的发展,它可划分为这么几个阶段^[5]:

- * 传统软件工程
- * 20 世纪 80 年代以来的对象工程
- * 80 年代中期开始的过程工程
- * 进入 90 年代之后出现构件工程

以探究软件开发过程中的方法、机制等内容为基础的思维方式,是原始的“过程观”。软件工程的模型的不同步骤(需求、设计等)把软件纳入工程化轨道,“软件过程”概念开始明朗,不过此时,管理、控制、人员等细节无法顾及。1984 年 10 月,第一届国际软件过程讨论会,正式提出“软件过程”:它是在软件生存周期中所实施的一系列活动的集合,且每个活动可由一些任务组成。

软件过程工程就是为建立软件过程所必须实施的一系列工程化的活动。它涉及与此有关的方法、工具和环境的研究。软件过程工程是以软件过程为中心的工程活动。软件工程所进行的是以软件开发为中心的工程活动,称为软件项目工程。它们共同构成了现代软件工程的框架。

CMM、SPICE(Software Process Improvement and Capability dEtermine,是国际标准化组织为组织软件过程评价标准的制定而成立的项目名称)和 Agile 是与软件过程相应的规范和原则。CMM/SPICE 经常不涉及经营目标,而 Agile 追求“刚刚好”(Just Enough)这样的效果。

CMM 是由美国卡内基-梅隆大学(Carnegie-Mellon)软件工程研究所(Software Engineering Institute, SEI)的 Watts Humphrey 于 80 年代末提出的,最初称为过程成熟度模型,后来改称为能力成熟度模型^[6]。

- * CMM 1.0 于 1991 年制定。
- * CMM 1.1 于 1993 发布,该版本应用最广泛。
- * CMM 2.0 草案于 1997 年制定(未被广泛应用)。
- * 到 2000 年,CMM 演化成为 CMMI,CMM 2.0 成为 CMMI 1.0 的主要组成部分。
- * CMMI-SE/SW 1.1(CMMI for System Engineering and Software Engineering)于 2002 年 1 月正式推出。

CMM 是用于衡量企业软件过程能力的事实上的标准,同时也是目前软件过程改进(Software Process Improvement, SPI)最好的参考标准。本质上,CMM 不但是一张国际市场的通行证,还是关系到软件企业能否从根本上走上正规化发展道路、改善管理、降低开发成本、提高产品质量和生产率、缩短开发周期、获得国际竞争力的关键。

值得注意的是:

(1) 成熟度等级不可以跳越,只能从低级一步一步地向高级前进。

(2) 有条件的话,允许在低等级时实施高等级中的一些关键活动。

(3) CMM 说明了“应该做什么”,是很充分的。CMM 对“如何做”也作了一些共性的建议,但这方面却是很不够的。所以企业仍然要花精力制定详细的软件工程规范,对 CMM 建议的内容进行适当的扩充、删减或者修改。

一个有趣的现象是,海外其它软件机构达到 CMM 2,3,4,5 等级的比例比美国的高。

4 热点与动向

国际产业界一般不提“软件产业”,代之以“软件服务业”:

- * 成套大众产品供应商(如微软);
- * 解决方案供应商(如 IBM);
- * 软件专业服务供应商(如普华永道)。

网络服务提供商体现了这种趋势。“软件就是服务”的观念日渐深入人心。

同时,软件运行环境已逐渐转变到开放和动态的网络

环境,而 Agent 的自主性和交互性能适应这样的改变,并带来软件在开发方法和使用方法等方面的一系列变化。

这里,对象和组件是核心内容。

面向对象方法中,面向对象程序设计是最早实践并且发展较为成熟。先看一下程序设计范式的演变^[7]。

最早出现也是最常用的程序设计范式是:决定想要的过程,使用能找到的最好算法。其核心是过程的设计,当然这些过程必须具有参数传递的功能。

随后,程序设计重点转到了数据组织上,程序设计范式是:决定想要的模块,将程序划分为不同的模块,而数据则隐藏在各模块中。这就是数据隐藏原则,它使得对数据的操作尽量局限在模块内部。

再往后,出现了抽象数据类型概念,程序设计范式变为:决定想要的类型,为各类型提供一个完整的操作集合。其核心是用户定义类型,隐藏性和灵活性更好了。

数据抽象由下列两方面支持:为某一类型定义一操作集合;对该类型的对象的存取限于该操作集合。由此便导出了类的概念。类概念本身是在程序设计领域中由类型概念抽象得出的,体现了人们的一种抽象思维过程。抽象过程的主要手段是分析。

将具有一定集成度并可以重复使用的软件组成单元(包括文档)称为软组件。

然而,已有工程化方法并不能解决人们遇到的各种问题,形式化方法始终被关注着和研究着。

形式化方法用严格的数学方法说明、设计和验证基于计算机的系统,不明确、不完整、不一致能被容易地发现和更正,但不是通过特别的评审,而是通过数学分析的应用。IBM 的净室软件工程^[8]设计方法提出了开发零缺陷或接近零缺陷的成功方法,是一种以合理的成本开发高质量软件的基于理论、面向工程的方法。其动机是提高软件可靠性和效费比,它结合了基于工程的技术,以及基于对象的系统论证、正确性验证和统计质量论证等技术,使软件开发在性能方面有本质改进,在可靠性和生产率两方面形成竞争优势,是有坚实理论基础的软件工程技术。

5 一种策略

企业做一次比较完整的 CMM 2-3 级咨询和评估大约要花费 50~90 万元。企业要组建软件工程过程组(SEPG)专门从事 CMM 研究与推广工作,这笔费用并不比咨询费低。如果企业再购买一些昂贵的软件工程工具(例如 Rational 的产品),那么总成本会更高。

即使企业舍得花钱,也不意味着就能够容易地提高软件过程能力。曾经有一段时间,经常争论“CMM 好不好”、“值

不值得推广 CMM”等话题。我们关注的焦点则是“企业如何以比较低的代价真正提高软件过程能力”。我们的研究对象是某特大型企业。

近几年来,该企业平均每年有好几十个研发项目,研发经费达数亿元。公司约有 1500 名研发人员,半数以上是软件开发人员。由于公司的研发管理能力不够强,特别是软件过程能力比较薄弱,大量以软件为主的项目的开发过程比较混乱,导致新产品的质量严重,进度不断地被拖延,直接经济损失近亿元。

在和很多同行专家交流时我们发现,该企业面临的软件工程和项目管理问题在很大程度上代表了国内 IT 业界面临的共性问题。

项目管理是个比较成熟的学科,美国的项目管理协会(PMI)制定了“项目管理知识体系”(相当于项目的标准),全世界以 PMI 为参考标准的项目管理工具软件也非常多。然而软件项目管理有其自身的特点,传统工业的项目管理方法很难套用到软件项目上。业界实践证明,软件项目管理最好的参考标准是 CMM/CMMI。

基于这样的背景,一套比较通用的“CMMI 3 级软件过程改进解决方案”得以提出^[9],被命名为“精简并行过程”(Simplified Parallel Process, SPP),其规范模型如图 1 所示,它的特色是:

* 一是采用 CMM/CMMI 为参考标准,专注于软件项目管理领域。该工具融入了 CMM/CMMI 的过程域管理方法,更加贴切于真实的软件开发过程。

* 二是将“项目规划、项目监控、质量管理、配置管理、需求管理和电子社区”全部集成于 Web 环境。人们不必多次购买分立的管理软件。使用 Future 工具将大大提高管理人员和开发人员的工作效率。

CMM/CMMI 论述的重点是“做什么”,而精简并行过程论述的兼顾“做什么”和“怎么做”。

目前市面上最流行的项目管理工具是 Microsoft Project,该软件在项目规划方面的功能很强,但是没有提供质量管理、配置管理、需求管理等非常重要的“CMM 过程域”功能。虽然 Microsoft 也拥有非常流行的配置管理软件 Visual SourceSafe,但是上述两个软件都是独立运行的,没有集成在 Web 环境,相互操作十分不便。

国际上著名的项目管理系统有 Rational Products, PMoffice、Primavera、Oracle Project 等。尽管这些系统的功能十分强大,但是由于功能繁多,导致使用复杂并且价格昂贵。

由于上述原因,上述软件难以在国内广泛应用。社会越发达,企业和大学科研机构对用户需求的反应就越快,所以“短平快”的中小型项目的增长率就比大型项目的高。大企业

也趋向于把大型项目分解成若干中小型项目以求降低风险。

我们认为国内 IT 企业和大学科研机构迫切需要的是低成本的、能够有效管理中小型项目的方法学和工具。我们的方案就是根据这个需求创作的,两者构成了完整的软件过程改进解决方案,对提高中国 IT 企业和大学科研机构的管理水平很有价值。

6 小结

要想提高企业的软件过程能力,本质上是靠规范化的企

业管理。软件开发是如此的灵活,如果没有规范来指导与制约,就容易因无序而导致混乱。但是规范如果不切实际或者太严密了,就容易畸变成为死板的教条,会扼杀开发人员生机勃勃的创造力。软件过程规范应当力求简单实用。

诚然,我们不能老是犯从一个极端到另一极端的错误。技术与管理一样,也经常成为失败的原因^[10]。对很多企业而言,技术开发过程的规范化比项目管理过程的规范化尤为重要与迫切。

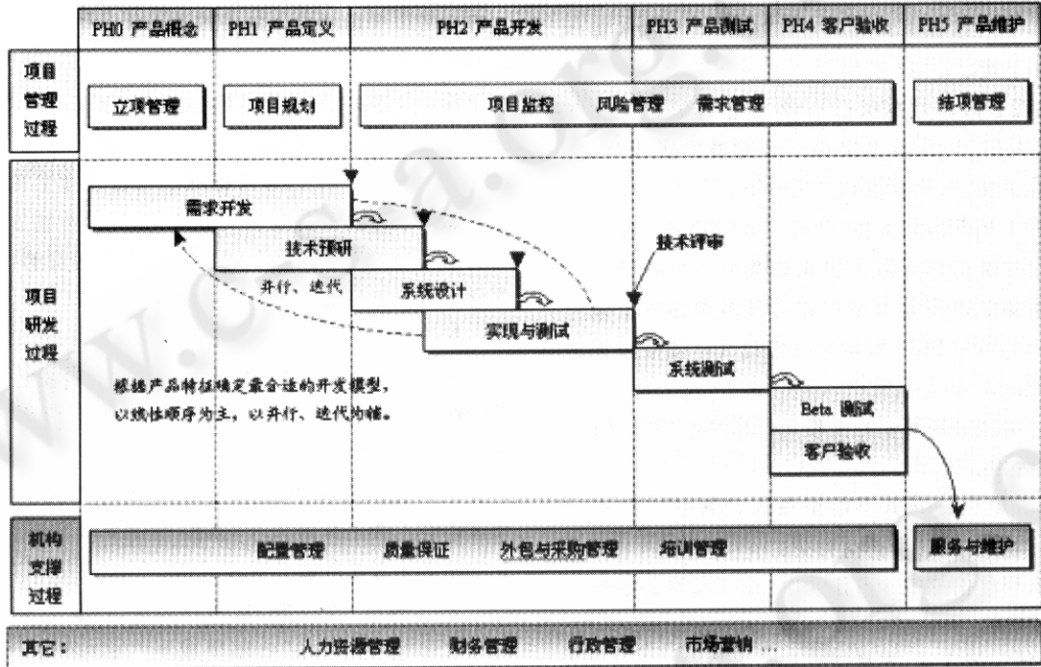


图 1 精简并行过程模型

参考文献

- 1 联合国计划开发署, 2001 年人类发展报告(中文版), 中国财政经济出版社, 2001: 37.
- 2 科技日报, 2002.8.22: 8.
- 3 Roger S. Pressman. Software Engineering, A practitioner's Approach(Fifth Edition). Tsinghua University Press, McGraw-Hill Companies, Inc.: 4-47.
- 4 唐稚松, 时序逻辑程序设计与软件工程, 科学出版社, 1999: 16-19.
- 5 朱三元、钱乐秋、宿为民, 软件工程技术概论, 科学出版社, 2002: 101-108.
- 6 Carnegie Mellon Software Engineering Institute.
- 7 Stroustrup B. What is Object-oriented Programming? IEEE Software, 1988, 5(3): 10-20.
- 8 Stacy J. Prowell etc. 贲可荣等译, 净室软件工程技术过程, 电子工业出版社, 2001.
- 9 林锐、王慧文、董军, CMMI3 级软件过程改进方法与规范, 电子工业出版社, 2003.1.
- 10 [美]罗伯特·格拉斯、陈河南等译, 软件开发的滑铁卢——重大失控项目的经验与教训, 电子工业出版社, 2002.2.