

软件开发过程模型的发展

高 禹, 毕振波

(浙江海洋学院 计算机系, 浙江 舟山 316000)

摘 要:针对软件开发过程模型的发展问题进行研讨。通过分析几个典型的软件开发过程模型:瀑布模型、快速原型化模型、增量模型、螺旋模型、构件组装模型,分析每个模型的产生情况,研究软件开发过程模型整体的发展过程,寻找其中规律性的东西。讨论了促使软件开发过程模型发展的两个主要因素:一是需要新的软件开发过程模型来处理新情况和新问题;二是需要新的软件开发过程模型来提高开发的效率和质量。指出了软件开发过程模型可以按照三个方向去发展:一是通过对现有模型进行改进、扩充、综合去发展;二是遵循新的思维方式去发展;三是借助新技术和新工具去发展。

关键词:软件工程;软件开发过程模型;发展

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2008)07-0083-04

Development of Software Exploitation Process Model

GAO Yu, BI Zhen-bo

(Department of Computer, Zhejiang Ocean University, Zhoushan 316000, China)

Abstract: Aims at studying and discussing the problems of the development of the software exploitation process model. Analyzes several typical software exploitation process models, namely Waterfall Model, Rapid Prototyping Model, Incremental Model, Spiral Model and Component Integration Model, and origin of each model. Studies the overall course of software exploitation process model, and finds the relevant law. The two main factors are provided as follows so as to advance software exploitation process model. One is new things and problems should be done with and solved by new model, the other is software development efficiency and quality should be improved by new model. Finally, points out the three directions of software exploitation process model, one direction is the existed software exploitation process modes should be improved, extended and developed synthetically, the second is the existed models should be developed according to a new thinking, the last is the existed models should be developed through adopting new technology and tools.

Key words: software engineering; software exploitation process model; development

0 引言

从第一个软件开发过程模型——“瀑布模型”的产生到现在,人们陆续推出了许多软件开发过程模型^[1~4]。这些软件开发过程模型是如何产生和发展的?软件开发过程模型还会发展吗?软件开发过程模型如何发展?研究这些问题对于推动软件工程理论向前发展具有重要意义。下面对这些问题进行研讨。

1 对几个典型的软件开发过程模型的分析

下面分析几个典型的软件开发过程模型的产生情况,通过分析,既可以看到它们的内容又可以了解它们产生的原因。同时,也可以从整体上看到软件开发过

程模型发展的大致过程,在此基础上思考软件开发过程模型的产生和发展问题。

1.1 瀑布模型的产生情况

早期的软件开发活动带有明显的个体化特征,非常不规范,随意性很强,人们错误地认为软件就是程序,对程序之外的数据和相关的文档材料没有给予重视,对编写程序之外的软件开发活动(如需求分析、概要设计、详细设计、软件维护等等)没有给予重视,结果出现了软件危机。软件危机的典型表现有:开发成本急剧上升、开发进度一再拖延、软件难以维护甚至无法维护、软件质量无法保证、开发出的产品不能满足用户需要,等等。为了摆脱软件危机,人们开始研究软件开发方法,1968年提出“软件工程”的概念,主要思路是将人类从事各种工程项目积累起来的行之有效的原理和方法应用于软件的开发和维护活动中。在这种情况下,1970年瀑布模型被推出。

一个软件从开始计划到开发成功、交付使用,再到

收稿日期:2007-10-22

基金项目:浙江省教育科研项目(20050127)

作者简介:高 禹(1954-),男,吉林长春人,副教授,研究方向为软件工程、数值分析。

废弃不用,有一个完整的生命周期,称为软件的生命周期。瀑布模型按照软件的生命周期,将软件过程分为:问题定义、可行性研究、需求分析、概要设计、详细设计、编码、测试、维护等几个阶段。软件开发活动按顺序一个阶段接着一个阶段地进行,每个阶段完成一项特定任务,每个阶段的结果经审查合格后方能进入下一个阶段。瀑布模型严格地规定了每个阶段必须提交的文档,强迫开发人员采用规范的方法,要求每个阶段提交的产品必须经过专家的仔细验证。这样,软件质量得到了保证。由于各阶段提交了规范的文档,软件维护变得容易一些。瀑布模型的成功在很大程度上是由于它是文档驱动的模式。

瀑布模型的推出,是人们为了摆脱软件危机迈出的重要的一步。按照瀑布模型去进行软件开发活动,克服了开发中的个体随意性和不规范倾向,使软件开发有章可循,有效地遏制了日益蔓延的软件危机。

1.2 快速原型化模型的产生情况

按照瀑布模型开发软件,虽然很有效,但灵活性不强,因为瀑布模型是按阶段顺序来操作的,必须在前一阶段的工作完成后才能进行下一阶段的工作。需求分析是一个重要的阶段,由于在开发早期用户的需求往往是模糊的,或由于某些原因用户的需求要发生变化,导致需求分析阶段的工作无法结束,造成下一阶段的概要设计工作无法进行。这时如果继续采用瀑布模型进行软件开发活动,显然不妥,因此为了解决这类软件开发问题,必须构建新的软件开发过程模型。在这种情况下,快速原型化模型被推出。

人们认识未知的事物,往往按照“实践、认识、再实践、再认识,逐步完善”的规律去做,经过反复多次的迭代式的实践和认识过程,达到基本了解事物情况的目的。快速原型化模型按照这个规律进行软件开发活动,首先快速建立一个能反映用户主要需求的原型系统,请用户在计算机上试用,通过试用,用户提出修改意见;开发人员按照用户意见快速地修改原型系统,然后再让用户试用;然后开发人员按照用户意见再去修改;如此反复多次,直到原型系统完全满足用户需求为止。

采用快速原型化模型进行开发活动,有效地解决了用户需求模糊不清和用户需求不断变化的问题。可以认为快速原型化模型是对瀑布模型的补充和完善,瀑布模型是用静止的观点来看待软件开发活动,将用户需求看成是固定不变的,这样实际上是将用户需求简单化了,这种理想状态实际很难找到。快速原型化模型是从变化的观点来看待软件开发活动,符合客观实际情况,所以许多后来推出的模型都吸收了快速原

型化模型的这种观点。

1.3 增量模型的产生情况

采用瀑布模型或采用快速原型化模型来开发软件时,是按照模型规定的开发过程,完成各开发环节的所有任务,得到一个完整的软件,将其提交给用户。面对软件规模越来越大、软件市场竞争越来越激烈、用户要求越来越高的形势,这样开发存在很多问题。当你将一个大的完整产品提交给用户后,用户要花费很多时间来学习这个新产品,短时间内很难适应这个新产品,给工作中应用该产品带来不便;这个产品完整提交后,用户再去评价和提出修改意见就没有意义了。这样,使开发风险加大,使开发时间增长,使用户满意度降低。为了解决这个问题,必须构建新的软件开发过程模型。在这种情况下,增量模型被推出。

人们解决大问题时,往往是将大问题分解为若干个小问题,每个小问题比较容易解决(其中有一个小问题是核心的关键问题),将这些小问题分别给予解决(对于核心的关键问题首先给予解决),那么大问题也就被解决了。一般来说,分解出的每个小问题具有相对独立性,即每个小问题与其它每个小问题联系不紧密,这样,既可以一个接着一个地顺序去解决每个小问题,也可以同时去解决多个小问题。增量模型按照这样的方法进行软件开发,将一个大的软件分解为一系列较小的“增量”,每个增量分别进行开发,通常开发的第一个增量是软件的核心部分,实现软件的基本需求。向用户一个增量接着一个增量地分批提交软件产品。采用增量模型,用户从拿到第一个增量时开始,就可以学习和熟悉软件,通过使用来评价软件及提出修改意见;开发人员根据用户对已经提交的增量的反馈,可以改进软件产品。这样,提交所有增量后,软件产品就达到比较完善的程度,也提高了用户满意度。

1.4 螺旋模型的产生情况

软件开发从始至终都存在着风险,项目规模越大、软件越复杂,开发该项目所冒的风险就越大。并且风险具有不确定性,可能发生也可能不发生,但是一旦风险变为现实,就会造成损失,甚至产生恶性后果。因此,如何识别风险、预测风险、驾驭风险,将风险可能造成的危害消除或减少,是软件开发中必须要考虑的问题。但是在螺旋模型之前所提出的各种软件开发过程模型,都没有强调“风险分析”。在这种情况下,螺旋模型被推出。

其实人们做任何事情之前,都要考虑风险。如果存在风险,那么一定要想办法去消除,否则成功希望渺茫。螺旋模型是在结合瀑布模型和快速原型化模型的基础上增加了“风险分析”的内容。螺旋模型在总体开

发框架上,带有瀑布模型的系统性、顺序性和“边开发,边评审”的特点。螺旋模型也是一种迭代模型,每一次迭代均可采用快速原型化模型方法,每一次迭代均作风险分析。螺旋模型由若干个螺旋周期组成,每一周期都包括需求定义、风险分析、工程实现和评审四个阶段,当项目按顺时针方向沿螺旋线移动时,每迭代一次,螺旋线就前进一个周期,软件开发又前进一个层次,系统又生成一个新版本(即构造一个新的原型,这个新原型是在风险被排除后得到的),当迭代过程进行到用户允许或可接受的范围时,迭代结束。

螺旋模型的推出,强化了人们的风险意识。通过使用原型来降低风险是一种行之有效的办法。螺旋模型集成了瀑布模型和快速原型化模型的优点,又有自身的特点,是一个实用性很强的软件开发过程模型。

1.5 构件组装模型的产生情况

面向对象技术出现之前所提出的各种软件开发过程模型,一般很少考虑“软件构件”的重复使用问题,即使编程时重复使用了一些库函数,量也不大,并且粒度小。因此,软件开发的任何一项工作基本是从头开始做,完整地做到尾。这样开发的缺点是成本高、时间长,当然出错的可能性也大。这里的“软件构件”一般指源代码,现在将需求规格说明、用户界面、软件体系结构等等也作为“软件构件”。人们考虑:如果在开发新软件时,能大量地重复使用已经开发过的软件中的内容,开发时间和成本不就降低了吗?又由于已经开发过的软件经过了严格的测试,重复使用这些内容在质量上当然是有保证的。面向对象技术的出现,为这个想法开辟了道路。在这种情况下,构件组装模型被推出。

重复使用的思想早已在许多领域广泛应用了,例如在工业生产中,重复使用各种零部件来组装生产新产品。在软件生产中,由于每个软件与其它软件都不同,在面向对象技术出现之前,重复使用难度比较大。面向对象技术将数据和操作该数据的算法封装在一起,做成一个个的“类”,将一个或多个相关“类”组合成一个“软件构件”,在某领域内使用过的所有“软件构件”被放到一个“软件构件库”中,这样为重复使用打下了基础,构件组装模型就是通过重复使用“软件构件库”中的软件构件来进行软件开发。使用构件组装模型开发软件时,根据被开发软件的目标和开发方案,选取软件构件库中的软件构件,组装成一个完整的软件版本。

构件组装模型的推出,使前人的劳动成果被有效地利用了起来。按此模型进行开发活动,可以节省时间、降低成本,软件质量也有保证。构件组装模型的出

现,使软件开发工作开始进入一个新时代。

1.6 几个软件开发过程模型产生情况小结

从以上分析几个典型的软件开发过程模型的产生情况可以看出:软件开发过程模型的出现,是人们为了消除软件危机、使软件开发活动有序化和规范化、高效率地得到高质量的软件产品而不断研究总结的结果,每一种新的软件开发过程模型的出现,都为当时软件开发遇到的某一类问题提供了解决方案,从而丰富了软件工程的内容,推动了软件工程理论向前发展。

2 促使软件开发过程模型发展的主要因素

现在已经有了这么多的软件开发过程模型,软件开发过程模型还会发展吗?答案是肯定的。通过上面的分析过程和深入的思考,可以得出促使软件开发过程模型发展的两个主要因素:

第一,客观世界的情况在变化,不断出现新的问题,需要用计算机处理。面对新情况和新问题,原有的软件开发过程模型无法胜任,因此需要推出新的软件开发过程模型来处理新情况和新问题。回顾软件开发过程模型的变化和发展的历史,许多软件开发过程模型是为了处理新情况和新问题而推出的。例如快速原型化模型是针对用户需求不完整和用户需求不断变化的情况而推出的。例如螺旋模型是针对风险控制问题而推出的。例如文献[5]所介绍的建立在面向 Agent 技术上的 Gaia 模型,是针对现有的软件开发过程模型在开发复杂分布软件系统时常常遇到困难而推出的。例如文献[6]所介绍的一种基于 Agent 的自适应软件过程模型,是针对软件过程所处的环境发生变化问题而推出的。

第二,人们希望软件开发的效率更高、质量更好、速度更快,因此人们不会满足现状,势必要研究并推出新的软件开发过程模型。例如构件组装模型的推出,就是人们不满足现状、遵循“重复使用”的思想所推出的软件开发过程模型。再如文献[7]所介绍轻载(敏捷)软件开发方法中的 XP 模型(极限编程),也是人们不满足现状,针对传统模型存在的问题,按照新的理念所推出的软件开发过程模型。

以上两个主要因素显然会长期存在,因此软件开发过程模型必然还要发展。

3 软件开发过程模型如何发展

既然还会有新的软件开发过程模型被推出,就是说软件开发过程模型还要发展,因此人们要思考软件开发过程模型如何发展这个问题。根据对软件开发过程模型有关情况的分析研究,软件开发过程模型可以

按以下三个方向去发展:

第一,可以通过对现有模型进行改进、扩充、综合去发展。

结合新问题的内容,针对现有模型存在的适用面窄、考虑问题欠周到等情况,可以通过改进和扩充某个软件开发过程模型的内容而得到一个新模型,或者通过综合运用几种软件开发过程模型的内容而得到一个新模型。如文献[8]介绍的一种新的软件开发过程模型,是在瀑布模型基础上进行改进和扩充的结果。再如增量模型,是综合运用瀑布模型和快速原型化模型的结果。再如文献[9]介绍的一种新的软件开发过程模型,是综合运用瀑布模型和构件组装模型的结果。再如文献[10]介绍的一种新的软件开发过程模型,是综合运用构件组装模型和并行过程模型的结果。

第二,软件开发过程模型可以遵循新的思维方式去发展。

现有的软件开发过程模型,每一个都体现出各自不同的思维方式,例如瀑布模型是所有采用线性思维方式模型的典型代表,快速原型化模型是所有采用反复循环迭代思维方式模型的典型代表。遵循新的思维方式去发展,就是说,新建立的软件开发过程模型应该是新的思维方式的体现,即按照新的想法去组织软件开发活动。例如 XP 模型(极限编程)就是按照新的思维方式去发展起来的。从 Agent 具有自主性、反应性、社会性等角度看,各种面向 Agent 的软件开发过程模型都是按照新的思维方式发展起来的。

第三,软件开发过程模型可以借助新技术和新工具去发展。

任何软件开发过程模型都是建立在一定的技术和工具基础之上,技术和工具的进步对软件开发过程模型的影响是巨大的,当新技术和新工具出现后,传统的开发方式势必要被改变,所以说新技术和新工具会推动软件开发过程模型更新发展。如构件组装模型、基于体系结构的软件开发过程模型^[11],就是在面向对象技术基础上发展起来的。再如 RUP^[12]模型,就是在 UML 这个开发工具基础上发展起来的。

4 结束语

软件开发过程模型的出现不是偶然的,它是软件开发活动到达一定程度后的必然结果。软件开发活动的多样性,决定了软件开发过程模型在形式上的多样性。每一个软件开发过程模型都有不同于其它模型的特点,这个特点体现了需要被解决的某一类问题所具有的特殊性。人的思维在发展,开发技术和工具在发展,推动着软件开发过程模型更新发展。在目前的基础上,研究软件开发过程模型的发展问题,对于提高软件开发的质量和效率具有重要的意义。

参考文献:

- [1] 陈松乔,任胜兵,王国军.现代软件工程[M].北京:清华大学出版社,2004:12-19.
- [2] 史济民.软件工程—原理、方法与应用[M].北京:高等教育出版社,2002:15-27.
- [3] 谷 烽,姜云飞,毛明志.软件过程模型回顾与分析[J].现代计算机,2005(5):28-30.
- [4] 张友生,李 雄.软件开发模型研究综述[J].计算机工程与应用,2006(3):109-115.
- [5] Wooldridge M J. The Gaia Methodology for Agent - Oriented Analysis and Design[J]. Autonomous Agents and Multi - Agent Systems,2000,3(3):285-312.
- [6] 赵欣培.一种基于 Agent 的自适应软件过程模型[J].软件学报,2004,15(3):348-358.
- [7] 徐 琛,杨宗源.轻载软件开发方法[J].计算机工程,2003,29(1):268-271.
- [8] 肖 汉,冯 娜,宋茂强.面向对象的软件重用瀑布模型[J].西华大学学报:自然科学版,2006,25(2):1-3.
- [9] 赵 鹏.一个三段式并行管道化软件过程模型[J].黑龙江大学自然科学学报,2006,23(2):245-249.
- [10] 王 珉,吴广茂,韩联庆.基于组件开发的并行过程模型研究[J].航空计算技术,2006,36(1):39-43.
- [11] 张友生.基于体系结构的软件开发模型[J].计算机工程与应用,2004(34):29-33.
- [12] Kruchten P. RUP 导论[M].麻志毅,译.北京:机械工业出版社,2004.

(上接第 82 页)

实的网络运行情况可能存在较大的偏差。而 NCTUns 简单易用,实验结果可信度更高,能方便地进行仿真实验。

参考文献:

- [1] 姜宁康,李毓麟.NS 网络仿真技术及其应用分析[J].小型微型计算机系统,2001,22(4):415-417.

- [2] 李向丽,李 磊,陈 静.网络实验仿真与网络技术实践[J].计算机技术与发展,2006,16(3):74-76.
- [3] Wang S Y, Chou C L, Huang C H, et al. The Design and Implementation of the NCTUns 1.0 Network Simulator[J]. Computer Networks,2003,42(2):175-197.
- [4] Wang S Y, Kung H T. A Simple Methodology for Constructing Extensible and High - Fidelity TCP/IP Network Simulators[C]//IEEE INFOCOM'99. New York, USA: [s. n.], 1999.