

关于小型软件公司开发模型的应用研究

张志辉

(三峡大学 电气信息学院, 湖北 宜昌 443002)

摘要:对常见软件开发模型进行分析,同时结合小型软件公司软件开发目前的实际状况,提出针对软件开发模型在软件开发过程中如何应用的见解.

关键词:软件开发模型; 小型软件公司; 软件开发过程

中图分类号: TP311.5 **文献标识码:** A **文章编号:** 1672-948X(2008)02-0077-03

Application Study of Software Development Model in Small Software Company

Zhang Zhihui

(College of Electrical Engineering & Information Science, China Three Gorges Univ., Yichang 443002, China)

Abstract Several common software development models are analyzed. Meanwhile, combining the actual situation of software development at present in small software company, some thoughts about how to apply the models in the software development process are proposed.

Key words software development model; small software company; software development process

从20世纪60年代软件危机引发软件工程革命以来,关于软件开发标准化、规范化已经作了很多研究,因此出现了多种软件开发模型,如瀑布模型、原型模型、螺旋模型、演绎模型等等.由于各种开发模型的哲学基础、形式概念、本质特征均有所不同,造成大多数小型软件公司面对复杂的现实如何正确使用开发模型时,感觉到力不从心.造成该现象原因有很多种,如开发人员对模型的实质领悟、公司的管理机制、人员的素质、企业的能力等等.主要就小型软件公司在应用软件开发过程中如何使用开发模型提出几点看法,以帮助小型软件公司在应用软件开发模型时跳出复杂的模型概念,追究软件过程的本质.

1 开发模型综述

开发模型是软件开发的指导思想和全局框架,是软件开发的全部过程、活动和任务的结构框架,它能清晰、直观地表达软件开发全过程,明确规定了要完成的主要活动和任务,用来作为软件项目开发的基

础^[1].在一般情况下,企业软件开发与软件模型之间并不存在对应关系,项目经理在带领IT团队进行独立开发、联合开发过程中,瀑布式、原型式、螺旋式3种模式及其衍生模型都可以使用,只是它们之间的区别主要在对项目风险的控制能力、及时收集用户的反馈信息以及及早对软件进行测试能力上的不同^[2].目前已提出或使用的开发模型有很多,典型的开发模型有:瀑布模型(waterfall model);螺旋模型(spiral model);快速原型模型(prototype model).

(1)瀑布模型.瀑布模型将软件生命周期的各项活动规定为依固定顺序联接的若干阶段工作,形如瀑布流水,最终得到软件产品.

优点:a.强调开发的阶段性;b.强调早期计划及需求调查;c.强调产品测试.

缺点:a.依赖于早期进行的唯一一次需求调查,不能适应需求的变化;b.由于是单一流程,开发中的经验教训不能反馈应用于本产品的过程;c.风险往往迟至后期的开发阶段才显露,因而失去及早纠正的机会.

收稿日期:2007-09-25

通讯作者:张志辉(1976-),男,硕士研究生,主要研究方向为计算机应用技术.

(2) 快速原型模型. 快速原型模型是指在进行了基本需求分析之后, 快速开发出产品的原型, 然后基于这个原型, 同客户沟通、交流, 更好地了解客户需求, 不断修改这个原型, 到了双方认可的程度, 再做详细分析、设计和编程, 最终开发出令客户满意的产品.

一般步骤如下: 先定义软件的总体目标, 根据已知的需求来规划出可实现的区域. 然后是“快速设计”, 集中于系统的总体框架、基本功能和直观的输入方式和输出格式等. 有了原型, 使客户对系统实现哪些具体功能、功能实现到什么程度有更好的理解. 开发者可以边开发边评估, 不断细化软件的需求, 逐步调整原型使其满足客户的要求. 这形成一个迭代的过程. 即使开始建立的原型过于简单或性能很差, 难以使用, 但为下一次建立适用的模型积累了经验, 而浪费的成本、时间有限.

(3) 螺旋模型. 螺旋模型是瀑布模型与原型模型相结合, 并加入两者所忽略的风险分析所建立的一种软件开发模型. 该模型于 1998 年由美国 TRW 公司 (Boehm B W) 提出. 软件项目风险的大小作为指引软件过程的一个重要因素, 引入这一概念有可能使得软件开发被看作一种元模型, 因为它能包容任何一个开发过程模型^[3].

其基本的做法是在“瀑布模型”的每一个开发阶段之前, 引入非常严格的风险识别、风险分析和风险控制. 直到采取了消除风险的措施之后, 才开始计划下一阶段的开发工作. 否则, 项目就很可能被取消.

(4) 基本模型的衍生模型. 迭代模型. 迭代模型主要针对事先不能完全定义需求的软件开发. 用户可以给出待开发系统的核心需求, 并且当看到核心需求实现后, 能够有效地提出反馈, 以支持系统的最终设计和实现. 软件开发人员根据用户的需求, 首先开发核心系统. 当该核心系统投入运行后, 用户试用之, 完成他们的工作, 并提出精化系统、增强系统能力的需求. 软件开发人员根据用户的反馈, 实施开发的迭代过程. 第一迭代过程均由需求、设计、编码、测试、集成等阶段组成, 为整个系统增加一个可定义的、可管理的子集可看作是重复执行的多个瀑布模型. RAD 模型. RAD (rapid application development) 模型, 即快速应用开发模型. 由于其模型构图形似字母“V”, 故也称 V 模型, 是属于线性顺序一类的软件开发模型. 它通过使用基于构件的开发方法来缩短产品开发的周期, 提高开发的速度. RAD 模型实现的前提是能做好需求分析, 并且项目范围明确, 这一点正好和原型模型相反.

RAD 模型一般适合信息系统应用软件的开发,

而不适合高性能、技术风险高或不易模块化的系统开发. 如果一个系统难以被适当地模块化, 那么就很难建立 RAD 所需的构件; 如果系统具有高性能的指标, 且该指标必须通过调整接口使其适应系统构件才能达到, 使用 RAD 方法可能会导致整个项目失败.

2 开发模型在软件开发中的应用

从横向角度来看, 软件开发过程包括开发过程中服务的管理过程, 一般包括项目管理、质量管理与配置管理 3 方面. 从纵向角度看, 一个软件的开发过程按照软件工程关于软件生存周期的观点分为问题定义、可行性研究、需求分析、总体设计、详细设计、编码和单元测试、综合测试、软件维护 8 个阶段. 对于小型软件公司人力和财力都有限, 将软件项目的开发过程分为软件需求、软件开发和软件产品 3 个阶段涵盖以上 8 个阶段.

(1) 软件需求阶段

用户需求的模糊性和动态性及用户与开发商存在专业的差别、心理因素造成开发商与用户甚至开发商与开发商之间的沟通, 不可避免地存在鸿沟和误解, 需求会发生变动. 如何保证变化能够得到控制, 并且最后的产品能够满足用户的需求, 这是两个关键的问题.

瀑布模型强调阶段和阶段的子目标, 因而其强调需求规格说明书的绝对重要性, 以此来确定需求. 对于需求简单、目标明确的单一部门级应用软件或大型应用软件的部分简单模块均可采用此种方式, 例如简单的办公软件应用或 ERP 模块中简单、单一的物流、财务、库存软件开发可采用此种模型. 但瀑布模型强调开发的阶段性, 强调早期计划及需求调查, 强调产品测试, 但其带来的结果往往是系统交付客户之后才发现原先对于需求的理解是错误的, 系统设计中的问题要到测试阶段才能被发现; 对于项目风险的控制能力较弱; 软件项目延期完成或开发费用超出预算, 项目开发进度往往会被意外发生的问题所打乱, 需要进行返工或其他一些额外的开发周期, 造成项目延期或费用超支. 另外, 在瀑布模型中, 需求和设计中的问题是无法在项目开发的前期被检测出来的, 只有当第一次系统集成时, 这些设计缺陷才会在测试中暴露出来, 从而导致一系列的返工: 重新设计、编码、测试, 进而导致项目的延期和开发成本的上升.

快速原型模型是一种用较小的代价和较快的速度生成一个可对用户的需求和目标系统进行审定的、具体的、可执行的模型. 它提供一种连续的、有效的信

息反馈机制来不断地、动态地求精和完善需求描述和系统设计,使用户通过亲身的经验对系统的功能和性能作出全面客观的验证和判断,拥有较好的工具、软件可重用技术和环境是充分发挥原型模型优越性和效率的关键。

(2) 软件开发阶段

在软件开发阶段须建立一个合理有效的软件项目计划,对软件项目的规模、资源要求和风险等要有一个合理的估算。对所有的项目活动和所有参加方及利益关系人的责任要明确并且文档化,以保证有效地将计划传达给项目各个参加方。在计划实施过程中,项目负责人必须跟踪控制计划,必要时对计划进行调整修改和维护。项目后期的修改往往导致成本增加很大,因而项目负责人必须保证计划的合理有效地设计和执行及修改计划的必要。从软件系统设计角度,设计人员必须保证设计与需求的一致,防止导致“错误扩大”效应。

在此阶段最常用的是螺旋模型,基本做法是在“瀑布模型”的每一个开发阶段之前,引入非常严格的风险识别、风险分析和风险控制。直到采取了消除风险的措施之后,才开始计划下一阶段的开发工作。其强调了严格的全过程风险管理和各开发阶段的质量,并提供机会以检讨项目是否有价值继续下去,但其因为引入非常严格的风险识别、风险分析和风险控制,对风险管理的技能水平提出了很高要求,并需要大量的人才、资金和时间投入。对于普通企业采用此种方式独立开发或联合开发软件则有些得不偿失,估计只有像微软那样财力、物力、人力齐备的软件公司才会考虑采用此种模型。

迭代模型也经常用在软件项目开发阶段,它允许研发人员变更、优化系统需求,并通过向业务部门演示迭代所产生的部分系统功能从而可以尽早地收集业务部门对于系统的反馈,及时改正对业务部门需求的理解偏差,从而保证开发出来的系统真正解决他们的问题。其次,可逐步集成元素,在传统的项目开发中,由于要求一下子集成系统中所有的模块,集成阶段往往要占到整个项目很大比例的工作量(最高可达40%),这一阶段的工作经常不确定并且非常棘手。在迭代式模型中,集成可以说是连续不断的,每一次迭代都会集成一些新的系统功能,但要集成的元素都比过去少得多,所以工作量和难度都是比较低的。由于在每次迭代中总是不断地纠正错误和产生一个可运行的系统,并对开发项目进行系统进行测试,因此可

以及时发现缺陷并改正,尽早发现并处理性能上的瓶颈,不断提高软件开发的产品质量。

(3) 软件产品阶段

软件产品阶段包含了中间产品、提交给用户的产品和开发商所提供的关于软件产品的相关服务,这是软件推向用户的阶段。

随着面向对象技术的发展,软件开发的技术发展方向已经从“提升被开发系统的执行效率”转变为“提升开发效率”^[4],面向对象(OO)技术降低了解决方法域与问题域的差别,提供了良好的复用机制,能够更加有效提高软件开发效率,完全顺应了软件开发技术的发展方向。目前采用UML以及RUP(Rational Unified Process)指导软件的开发已经在国内外取得了广泛的应用。UML是一个通用的可视化建模语言,可以对复杂的系统建立可视化系统模型。在软件需求分析、设计、产品测试阶段,基于UML的面向对象需求分析克服了传统的分析、设计、测试在问题领域上受时效的限制和对系统功能无法把握其精确程度等缺点;同时解决了分析、设计的层次复杂性的问题,使开发者和用户能够更好的关注需求^[5]。

3 结 语

对小型软件公司在软件开发过程如何使用开发模型发表了一些看法,参考了软件过程理论和软件过程模型及CMM(能力成熟度模型)理论和UML模型,依据关于规划小型软件公司软件开发工作流程的想法,提出软件开发模型在小型软件公司开发过程中如何实际应用的观点。

参考文献:

- [1] 齐治昌. 软件工程[M]. 北京:高等教育出版社,2004.
- [2] 谢 芳. 现代软件工程在现代软件开发中的合理应用[J]. 计算机与现代化,2005(4):120-122.
- [3] 张海藩. 软件工程导论[M]. 4版. 北京:清华大学出版社,2003.
- [4] 邱 剑. 面向对象软件工程方法分析[J]. 计算机技术与发展,2002,12(1):32-35.
- [5] 戎 玫. 基于软件体系结构和UML的图书管理系统设计与实现[J]. 计算机科学,2005,32(6):224-227.

[责任编辑 张 莉]