

基于 CMM 的软件过程改进实践

王艳慧

(建设部信息中心, 北京 100053)

摘要: CMM 是一种软件生产过程标准和软件企业成熟度评估标准, 该标准侧重于软件开发过程的管理及工程能力的提高与评估。基于 CMM 理论在软件过程改进中的应用, 针对软件开发过程中存在的问题, 采用 CMM2 级项目级实践, 实现 4 个关键过程域: 需求管理、软件项目计划、软件项目跟踪和监督、软件质量保证, 制定出符合项目需求和管理的规范与过程, 将 CMM 理论应用于软件开发过程中, 对项目各个环节的工作进行改进, 取得良好的过程改进效果, 并从中总结基于 CMM 模型进行过程改进的经验。

关键词: 软件能力成熟度模型; 软件过程改进; 软件质量保证; 软件项目跟踪和监督

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2008)05-0141-03

CMM - Based Software Process Improvement Practice

WANG Yan-hui

(Information Center, Ministry of Construction, Beijing 100053, China)

Abstract: CMM is a software production process standards and software enterprises maturity assessment standards. The standards focus on the software development process management and engineering capability increase and assessment. This paper is based on the theory CMM software process improvement in the application, aims at software development process problems that exist. Using CMM2 - project - class practice and achieve four key process areas: requirements management, software project planning, software project tracking and oversight, software quality assurance, formulated in accordance with the project management of the needs norms and processes CMM theory is applied to the software development process, all aspects of the project to improve, making good process improvement results. And it concluded experience based on the CMM process improvement.

Key words: software capability maturity model; software process improvement; software quality assurance; software project tracking and oversight

0 引言

能力成熟度模型 (Capability Maturity Model, CMM) 是由美国卡内基 - 梅隆大学软件工程研究所推出的评估软件能力与成熟度的一套标准^[1], 是目前国际上最流行也是最实用的一种软件生产过程标准和软件企业成熟度评估标准。该标准基于众多软件专家的实践经验, 侧重于软件开发过程的管理及工程能力的提高与评估^[2]。CMM 提供了一套国际标准的管理方法论, 是对自身软件过程进行评价和改进的行动指南。CMM 的用途主要有三方面: 软件过程改进 (Software Process Improvement, SPI), 软件过程评估 (Software

Process Assessment, SPA), 软件能力评鉴 (Software Capability Evaluation, SCE)。CMM 分为五个成熟度等级: 一级为初始级, 二级为可重复级, 三级为已定义级, 四级为已管理级, 五级为优化级。CMM 每一成熟度等级由若干个关键过程域 (Key Process Area, KPA) 构成, 每个成熟度等级由若干个关键过程域构成。

1 系统建设概况

为加强住房公积金的监督, 规范管理, 维护广大公积金所有者的合法权益, 建设部 2002 年组织建设“全国住房公积金监督管理信息系统” (以下简称住房公积金监管系统), 实现对全国各地住房公积金的适时监控和管理, 规范住房公积金管理制度的实施。以各设区城市公积金管理中心业务管理系统为基础, 以部、省住房公积金监管部门对城市公积金管理中心资金管理运作情况进行全面、及时、有效监管为目标, 建立部、省、市三级公积金监督管理信息系统网络。

收稿日期: 2007-08-26

基金项目: 建设部项目 (计投资[2003]174 号), 并获 2006 年《华夏建设科学技术奖》三等奖

作者简介: 王艳慧 (1974-), 女, 硕士, 工程师, 参与建设部组织的多项系统建设工作。

根据系统建设目标要求,住房公积金监管系统建设在 2002 年 3 月启动,根据系统建设初期阶段的目标,该项目应于 2002 年 6 月完成技术实施方案,2002 年 7 月完成需求调研分析,2002 年 9 月完成软件开发与测试,2002 年 10 月项目进行实施、培训,2002 年 10 月试运行,在 2002 年 10 月 1 日前完成住房公积金监管系统的安装并开始试运行。但截止 2003 年 12 月整个项目状态处于停滞状态,软件的开发和测试工作没有完成,软件功能还不能满足用户要求,由于用户需求不断变更,使开发人员忙于修改程序,在软件测试时,功能的实现错误很多,软件质量无法保障,软件无法交付使用,同时系统相关的开发标准和运行制度没有按期制定,部分省市由于资金经费无法落实、机构改革等原因,不能实现联网调试,项目处于无限延期状态,无法按时交付完成。

2 系统过程改进实践

为了能尽快完成系统建设任务,并保证按时交付高质量的软件产品,开始研究如何能有效控制软件开发进度,满足用户需求,提高软件质量,改善软件可靠性和减少开发系统成本等问题。经过多方面的研究论证,在《住房公积金监管系统》项目中应用 CMM 理论指导软件开发过程改进活动。根据项目的规模和实际情况,改进活动以 CMM2 级为目标。CMM2 级的 KPA 关注的是项目级的实践,按照标准软件过程执行项目时就会涉及到这些实践。这些关键过程域的实现定义了执行具体活动时的流程和使用的方法。根据项目实际情况进行裁减,确定要实现 4 个关键过程域:需求管理、软件项目计划、软件项目跟踪和监督、软件质量保证,制定出符合项目需求和管理的规范与过程,将 CMM 理论应用于软件开发过程中,对项目各个环节的工作进行改进,最终实现项目成功完成并运行。

2.1 需求管理过程改进

根据 CMM2 的需求管理关键过程域要求,需求管理过程改进重点关注需求跟踪技术和需求变更管理,项目遵照规定的变更流程进行需求变更,并通过需求变更的影响分析,考虑风险、成本和进度等多方面因素,以确定需求变更的实施策略。

(1)需求跟踪的主要目的在于获得需求目前的实现状态,确保用户所有的需求都得到满足。实践中采用了比较常用的跟踪矩阵的方法(见表 1),它提供了从需求到有效验证软件是否实现了所有需求,以及是否对全部的功能进行了测试。这样的跟踪信息还可以在其它方面起到作用,如在需求发生变更时,可以据此判断这一变更的影响范围。而对需求发生变更时,可

以据此判断这一变更的影响范围。

表 1 跟踪矩阵

需求编号	需求概述	详细设计	实现	单元测试	整体测试	验收测试
3.3	报表上报模块设计	3.3.1	JTJK-BRSB	3.3.12	3.3.13	
4.1	报表录入模块	4.1.1	JTJK-BBLR	4.1.12	4.1.13	

(2)需求变更是软件开发过程中最重要环节,既然需求变更不可避免,如何将需求变更带来的风险有效化解就成为至关重要的问题。需求变更管理通过定义一个规范的流程来处理需求变更,力图使需求变动影响到工作产品尽可能的少,保证在需求发生变化的情况下项目仍能取得成功。通过“需求变更记录表”(见表 2)进行记录,并通过“变更跟踪累计表(见表 3)”累计需求变更对工作时间的总影响,以便对项目进行管理。

表 2 需求变更记录单

项目名称	公积金监管系统	日期	2004.9.16
请求号	29	需求编号	SR009
需求变更	对资料上报,要求可全部一次点击按钮上报或按城市进行上报,在后台进行数据检验,要区分已上报和未上报的资料		
影响分析	前台要修改上报功能,后台要对资料进行判断分析		
变更工作量	2 人天	进度影响	没有

表 3 变更跟踪累计表

请求号	请求日期	变更概述	工作量(人天)	状态	关闭日期
SR009	2004.9.16	对资料上报,要求可全部一次点击按钮上报或按城市进行上报,在后台进行数据检验,要区分已上报和未上报的资料	2	关闭	2004.9.20
SR010	2004.9.20	上报情况查询中要求增加手工录入的统计情况	1	关闭	2004.9.25
...
工作量总计					

2.2 项目计划管理过程改进

项目计划的重点是根据用户需求和实现目标,对项目的工作量和规模等方面做出估计,并根据估计结果制定出合理的进度计划,软件项目计划管理在软件开发过程中处于十分重要的地位,它体现了对用户需求的理解,是开展项目活动的基础,项目的管理者按照计划确定的内容对项目进行管理,计划的合理性将直接关系到项目的成败。在项目计划管理过程改进中,主要加强风险的管理。通过风险管理活动可以将意外事件对项目的不良影响减至最小。通过对项目的各个方面可能存在的风险进行识别和分析,确定避免或减轻风险的策略及措施以达到降低风险,保证项目顺利进行。

住房公积金监管系统风险管理计划见表4。

表4 住房公积金监管系统风险管理计划

编号	风险项	风险出现的后果	风险出现概率	降低风险以及风险出现的解决办法	风险解决情况
1	过多的需求变更	项目延期	20%	从客户那里获得对最初需求规格文件;说明客户相信需求变更会影响进度;定义一个规程来处理需求变更;协商按实际工作量付费	
2	采用新技术风险	项目失败	20%	考虑新技术适应性;将学习曲线的时间包括在进度内;验证该技术的可行性	
3	业务知识误解	项目延期	20%	主动学习相关业务知识;模拟业务流程并建立原型;积极与用户交流沟通	
	

2.3 软件项目跟踪和监督过程改进

软件项目跟踪和监督的目的是建立软件项目实际进展状态的可视性,使管理者能在软件项目性能偏离开发计划时采取有效措施^[3]。在这个阶段中,由项目经理负责依据开发计划对项目实施跟踪与监督工作,并在项目的执行过程中要求项目中的各级负责人查阅和分析组织软件过程数据库和文件库,使用组织级的经验对项目进行监控。当确定未实现拟订的目标时,采取纠正措施。对跟踪资料分析的结果在项目的状态报告中记录,说明项目在这一阶段的管理活动和软件工程状态,指出存在的问题和解决方案。

2.4 软件质量保证过程改进

软件质量保证过程主要通过监控软件的开发过程来保证产品的质量,软件质量保证涉及审查和核实软件产品及其活动,以便验证它们与项目采用的过程与标准的一致性^[4]。保证生产出的软件和软件开发过程符合相应的标准与规程,保证软件产品、软件过程中存在的不符合问题得到处理。在实施软件质量保证过程中的一个明显的效果就是促进了规范化过程的实施,保证了组织制定的软件过程得到项目人员的有效执行,制定相应的质量保证计划、流程审计报告、软件工作产品复审报告、质量保证计划进度表、SQA问题跟踪解决进度表。

3 系统过程改进效果

项目通过一年的软件过程改进活动,2005年4月该项目顺利通过验收,按时交付《全国住房公积金监督管理信息系统》2.0版,系统进入正式运行阶段,截止目前为止,已联通15省(区)、直辖市及110多个设区城市,包括河北省、河南省、湖北省、湖南省、吉林省、福建省、山东省、重庆市、上海市、天津市、甘肃省、新疆、新疆兵团、云南省等。建设部和省(区)住房公积金监

管部门可通过系统对城市住房公积金使用情况进行远程查询、动态预警、报表接收等监管功能。

应用CMM2级理论在住房公积金监管系统的过程改进中取得成功,在项目的需求管理、项目计划、质量管理、进度控制等方面取得良好的改进效果,实践证明,采取CMM2级进行一系列的软件过程改进后,软件开发效率不断提高,时间控制严格,品质有保证,管理更有序,从中获得以下改进效果和收获:

(1)建立良好的需求管理控制机制,对需求变更进行记录和控制,需求管理水平得到很大的提高。通过规范和完善的需求管理规则,为需求分析的工作质量、需求管理可视程序的提高打下了很好的基础。正确、有效和完整的需求规格说明书、需求管理和需求变更的控制,是进行下一步软件开发工作的基础。从整个软件开发生命周期中看,完善的需求管理提高了开发效率,缩短了开发时间和成本,提高了产品质量,增加了用户满意度。

(2)建立良好的项目计划体系,制定有效的进度计划,并对软件风险进行管理。跟踪与监控的规范的流程,随时掌握软件项目的实际开发过程,使软件项目能按计划进行。对项目计划的建立、项目资料的收集、项目跟踪和监控进行了完善。软件开发水平得到很大提高,在实施的进程中,意外事故、超额工作量大大降低,实际的开发过程越来越靠近计划和预测结果,从而降低了风险,提高了发展和改进的可预测性,确保软件开发工作按计划进行。

(3)在质量管理方面,通过评审和审核软件产品的活动,验证它们是否与应用的标准和规范一致,从而获得可见性。软件开发架构实行模块化,采用统一定义的标准接口,使得开发和测试更容易。提高检验率,降低错误发生率,在采用CMM2级过程改进以前,隐藏在产品中的错误较多,例如:在需求定义中的错误只有40%被检验出来,过程改进后,需求定义中的错误检验率提高到80%,从而大大减低了因为修改和改进所花费的工作量。

4 系统过程改进经验

在建设部《住房公积金监管系统》中采用CMM模型来进行软件改进过程的应用证明,采纳CMM的方法是一种提高软件企业过程能力的有效办法。在CMM框架的指导下,可以大大提高软件开发能力和软件产品的质量,降低软件开发失败风险,降低错误发生率,缩短整体开发时间,减少开发成本,从而保证用户能得到满意的软件产品。在应用CMM理论的过程

(下转第171页)

状图等。历史信息查询模块增加判断查询 IP 来源区域功能,当管理员输入要查询的 IP 历史信息时,先经过 IP 地址来源区域判断,确定 IP 来源区域后,再到相应区域的 LINUX 服务器数据库中取相应历史记录信息。

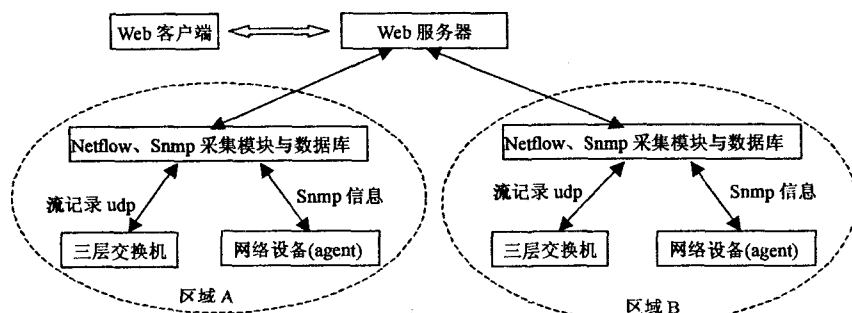


图2 分布式系统逻辑结构图

4 结束语

设计了一种基于 NETFLOW 与 SNMP 的网络监控系统。该监控系统要求不高,配置较为简单,只要园区网核心交换机支持 NETFLOW 功能,各接入交换机支持 SNMP 功能,即可实施,目前较大的园区网都满足这两项要求。

利用所设计的监控系统,能很好地对管辖范围内

的网络数据流进行有效监控,给网管人员提供丰富的决策依据。通过实时监控,网管人员可以找到病毒、蠕虫和网络攻击等网络异常行为的源头,并采取有效措施进行阻断,以阻断异常流量的影响进一步扩大。本监控系统功能还不太完善,更多功能有待进一步研究与开发。

参考文献:

- [1] Cisco NetFlow Services and Applications[EB/OL]. Cisco White Paper. 1999. <http://www.cisco.com>.
- [2] Stallings W. SNMP 网络管理[M]. 北京:中国电力出版社,2001.
- [3] 熊齐邦,黄明哲. 基于 NetFlow 和异步服务的网络流量监测系统[J]. 计算机工程,2006(13):144-146.
- [4] 何海涛,罗笑南,郭清顺. Netflow 在边界网流量测量中的应用研究[J]. 计算机工程与应用,2004(11):11-14.
- [5] Netflow Services Solutions Guide[EB/OL]. 2005. <http://www.cisco.com/en/US/products/sw/netmgtsw/ps1964/products-implementation-design-guide09186a00800d6a11.html>.

(上接第143页)

改进活动中总结以下经验:

(1)在应用 CMM 理论进行软件过程改进中,要根据软件企业的发展情况及其系统开发环境,从中找出制约软件企业开发效率和能力的关键因素,提出可行的标准和实践的步骤。根据软件企业的实力确定过程改进的各个阶段目标,不断总结、改进、提高,找出一条适合本软件企业的 CMM 实施路线。

(2)针对软件企业自身的特点,对 CMM 过程进行适当裁剪,对于 CMM 的模型与标准,不能生搬硬套,而应将其作为参考^[5]。必须结合项目的情况和企业自身的特点、要求与现实条件,制订软件过程 and 选择实行改进的部分。在引进、消化、吸收的基础上需要自主创新,让 CMM 更实用化,形成自己的管理模式。

(3)充分认识改进过程本身就是一个规范的过程,需要循序渐进、逐步改进,因为软件过程成熟度的升级本身就是一个有生命周期的过程,而且全面引进 CMM 所涉及的范围非常广,要求人力、财力与设备资源的投入跟得上。在最初实施 CMM 时,软件企业可以试行某些关键过程域的一部分关键实践活动。并逐步规划出软件过程建立与改进的短、中、长期目标,分

清轻重缓急,逐步取得经验。

(4)实施 CMM 对软件企业的发展有着重要的作用,CMM 过程本身就是对软件企业发展历程的一个完整而准确的描述。软件企业通过实施 CMM,可以更好地规范软件生产和管理流程,使企业组织规范化,提高软件企业的能力成熟度,改进软件的开发、维护过程,按时、按预算为用户提供高质量的软件,提高产品和企业的竞争力。

参考文献:

- [1] 卡耐基梅隆大学软件工程研究所. 能力成熟度模型(CMM)软件过程改进指南[M]. 刘孟仁,等译. 北京:电子工业出版社,2001.
- [2] Persse J R. CMM 实施指南[M]. 王世锦,蔡愉祖译. 北京:机械工业出版社,2003.
- [3] Caputo K. CMM 实施与软件过程改进[M]. 于宏光,王家锋,等译. 北京:机械工业出版社,2003.
- [4] 郑人杰. 基于软件能力成熟度模型(CMM)的软件过程改进[M]. 北京:清华大学出版社,2003.
- [5] Raynus J. CMM 软件过程改进指南[M]. 邱仲潘,等译. 北京:电子工业出版社,2002.