

# 基于 CMMI 的软件质量保证

袁安富, 伏萍

(南京信息工程大学 信息与控制学院, 江苏 南京 210044)

**摘要:**软件的质量保证是软件业最关注的一个问题。结合软件质量保证体系的国内外现状,以 CMMI 标准体系的优点与 5 级成熟度模型为依据,分析出过程是软件项目质量保证的制约因素以及基于 CMMI 的软件质量保证的具体内容。详细探讨了 CMMI 模型下项目计划开展与开发流程的问题。最后得出通过过程改进达到管理提高的目的,最终使得软件质量有所保证,结果减低企业的工程成本。该方法不仅适用于规模较大的组织和较大型的项目,而且也可以广泛应用于相当多的小型组织。

**关键词:**能力成熟度模型集成;软件;项目;质量保证

**中图分类号:** TP31

**文献标识码:** A

**文章编号:** 1673-629X(2012)01-0013-04

## Quality Assurance of Software Based on CMMI

YUAN An-fu, FU Ping

(College of Information and Control, Nanjing University of Information Science & Technology,  
Nanjing 210044, China)

**Abstract:** The quality assurance of software is the most concerned issue in software industry. Combined with the status at domestic and foreign software quality assurance system, based on the advantages and 5 maturity models of CMMI standard system, it is concluded that process is the factor of the quality assurance of software and the contents of the quality assurance of software based on CMMI standards. The project plans of quality assurance and development process, which based on CMMI standards have been discussed in the paper. From the analysis, it can be concluded that the introduction of CMMI will improve the ability to manage themselves, guarantee the quality of software and reduce the cost of the enterprise. The method not only applies for large organizations and projects, but also can be applied to a considerable number of small organizations.

**Key words:** CMMI; software; projects; quality assurance

## 0 引言

在日益竞争的软件市场中,软件企业面临着更多的挑战,比如:“认证”要求、市场宣传、投标资质、客户的要求等。软件企业必须首先使软件项目满足机构方针的要求,因此引入了 CMMI。CMMI 是 1997 年美国国防采购与技术办公室领导了一个由政府、企业和 SEI 的代表组成的团队开始开发一个 CMM 模型的集成框架<sup>[1]</sup>。2002 年 1 月 CMMI V1.1 正式发布,并立即被广泛采用。软件质量保证以预测、改进过程为主,保证问题被发现并解决。它是有系统、有计划的活动。而 CMMI 的引进解决软件项目过程改进难度增大问题,实现过程改进的最佳效益,使得产品能够达到客户

的要求、按时交付。从而在一定程度上大大保证了软件的质量,开发出具有多功能质量优越的产品,使得企业以最小的成本获得最优的效益。

## 1 质量保证体系的现状

多年来,国外很多的大公司在软件企业中纷纷实施 CMMI 准则,一方面是为了使大企业满足工程需求,另一方面是为了企业提升自身的管理水平。所有的这些准备工作都是为了使项目的质量有所保证,以最小的工程代价达到用户的要求。公司通过在企业中独立成立测试小组、SQA 小组的方式来进行监督。SQA 小组的职责包括监督开发,保证开发的项目与需求一致,在项目实施的过程中及时的发现错误并向上级汇报,对缺陷评审、追踪,确保项目组制定的计划、标准和规程适合项目组需要等。因此对 QA 的要求比较严格,国外的 QA 需要通过美国协会软件质量工程师相关的认证考试。

随着中国加入 WTO,我国软件企业陷入了危机。

收稿日期:2011-06-10;修回日期:2011-09-16

基金项目:江苏省高校自然科学基金(10KJB180004)

作者简介:袁安富(1964-),男,江苏人,教授,硕士生导师,研究方向为系统分析与集成;伏萍(1984-),女,江苏人,硕士研究生,研究方向为软件质量保证。

为了能接到更多的国外项目,就必须与世界接轨,这里接轨的中介就是 CMMI 的评估认证。国外客户通过企业的 CMMI 的等级而决定是否把项目交付于该企业。近年来,我国许多软件企业组织实施了 CMM/CMMI 管理模式,例如:东软、托普、华为等,其中东软在 2004 年 12 月已通过了 CMMI5 级的评估,成为了我国第一家获得 CMMI5 级的企业组织。据 SEI 统计,截止 2006 年 11 月,中国共有 158 家企业组织通过了 CMMI 的评估认证<sup>[2]</sup>。但是由于国内软件公司多、规模小且分散的特点,质量保证的意识还没有健全等原因,只有一些能够通过 CMMI 认证的大公司能够独立成立测试小组、SQA 小组。而比较小的公司虽然通过了 CMM/CMMI 的认证,但还未能组织上成立质量保证组。即使有的大公司成立了 SQA 小组,但也只是形式上的,并没有真正发挥 QA 的作用。近年来一些刚刚成长起来的软件外包公司,只是把获得 CMMI 证书当作其终极目的,不是以过程改进为目的进行 CMMI 认证。

## 2 CMMI 标准的体系介绍

CMMI(Capability Maturity Model Integration),即能力成熟度模型集成<sup>[3]</sup>。目前为止它包括五级标准模式和 24 级过程域。与 CMM 相比,CMMI 是 CMM 的拓宽版本,在 CMM 基础上多了系统集成方面的改进,使其更加适用企业的过程改进实施。

### 2.1 CMMI 的优点

CMMI 准则,其目的是帮助软件组织改进其过程,在对软件产品或服务的开发、采购以及维护的能力中提供指导。其所依据的想法是:集中精力持续努力建立有效的软件工程过程的基础结构,不断进行管理的实践和过程的改进,就可以克服软件开发中的困难。其主要关注点就是成本效益、明确重点、过程集中和灵活性四个方面。CMMI 为改进一个组织的各种过程提供了一个单一的集成化框架,新的集成模型框架消除了各个模型的不一致性,减少了模型间的重复,增加透明度和理解,建立了一个自动的、可扩展的框架。因而能够从总体上改进组织的质量和效率。

研究调查显示影响软件项目成功的主要风险因素就是过程,过程严重制约着我国软件企业的国际竞争力的提升。可见软件过程是一个组织实现其软件能力改进的杠杆支点<sup>[4]</sup>,制约软件项目成功的主要问题不是技术而是管理问题。而 CMMI 正是推动过程改进,提高管理的工具和指南。CMMI 的本质是软件管理工程的一个部分。软件过程改善是当前软件管理工程的核心问题<sup>[5]</sup>,50 多年来计算机的发展使人们认识到要高效率、高质量和低成本地开发软件,必须改善软件生产过程。基于模型的过程改进是指用采用能力模型来

指导组织的过程改进,使之过程能力稳定的进行改善,该组织也能变得更加成熟。

### 2.2 CMMI 的成熟度模型

CMMI 成熟度共有 5 个级别<sup>[6]</sup>,除了初始级没有过程域外,其他四个级别已管理级、已定义级、定量管理级、优化级共计 24 个过程域。每个过程域含有特有的目标,每个目标又包含特有的实践。过程域、目标、实践三者之间是一层层的包含关系。目标和实践都有特定和通用的分类,也是一一对应的关系,即特定的目标包含特定的实践。CMMI 级别的高低与质量生产率成正比,与风险返工成反比<sup>[7]</sup>。

## 3 基于 CMMI 的软件质量保证

### 3.1 质量保证的具体内容

软件质量保证措施具体包括:完善详尽的项目计划、合理的开发流程、实时的测试、风险管理、进行度量等一系列活动<sup>[8]</sup>。这些活动中的评审以及各种文档对质量保证具有很重要的作用,进而对证明和表现项目的质量贡献很大。因此质量保证的有效证明方式就是所有过程文档化。

### 3.2 基于 CMMI 的项目计划

计划是完成工作不可缺少的步骤,针对完善详尽的项目计划,以往只是依靠项目负责人的经验和能力,过程不可预测和控制,软件过程是无序的,有时甚至是混乱的,对过程几乎没有定义,项目执行的成败在一定程度上取决于个人因数<sup>[9]</sup>。这仅仅只是 CMMI 的第一级(初始级)的要求,显见在 CMMI 初始级下制定的项目计划是没有质量保证的。下面把 CMMI 的第二级(已管理级)和第三级(已定义级)中的具体过程域一起合并 Excel 的格式的项目计划中,这样便于管理者查阅和修改,便于文档管理。

项目计划主要包括以下几个部分:过程定义、项目估算、角色定义以及通讯录、项目监控、项目进度、评审计划、质量目标、交付件清单、培训计划、环境与工具、重用、相关方承诺。

其中过程定义主要包括过程名以及文档引用的章节<sup>[10]</sup>。交付件清单是指把具体版本的设计规格、需求文档、数据库接口设计说明书、过程文档、软件修改单、测试文档、回归问题列表等文档打成包交付给相关负责人。相关方承诺是随着计划书通知到相关方,并得到相关方的确认,确认方式可以是邮件、会议纪要、签字件。下面详细说明项目计划中的其他内容。

#### 3.2.1 项目估算

项目估算主要包括对开发、测试、资料三方面的初次估算。表 1 表示了开发初次估算;表 2 描述了测试初次估算。

资料的初次估算与测试的相类似,只是估计规模以页为单位,采用的生产率为每人每天产出多少页。通过以上的详细描述,可以初次粗略的估算出具体部门对整个项目的具体模块的消耗。此处项目管理者必须掌握好人员的分配问题。

表 1 开发初次估算

估算类型	模块名称	开发语言	估计模块	采用的生产率(模块/人天)	可用资源(人)	估计工作量(人天)
开发初次估算						
总计						

表 2 测试初次估算

估算类型	模块名称	估计规模(测试用例的个数)	采用的生产率(用例个数/人天)	可用资源(人)	估计工作量(人天)
测试初次估算					
总计					

3.2.2 角色定义以及通讯录

角色定义以及通讯录的文档化是为了明确具体个人对具体模块的责任。当测试时对具体模块的具体错误可以找到具体的相关负责人。通讯录的记录是为了在负责人随时随地不在工作状态时可以联系到具体负责人,及时快捷地解决问题。

3.2.3 项目监控

项目监控是项目负责人定期地做项目进展报告,将各项监控的结果记录在项目进展报告里,以使项目主管部门及项目组成员及时地了解项目的真实进展状况<sup>[11]</sup>。此处主要采用的是项目报告机制和项目会议机制。项目会议机制是定期定人的召开会议一起讨论项目过程中的进度、技术、问题、变更等问题。项目报告机制如表 3 所列。

表 3 项目报告机制

序号	报告类型	准备人	频率	报告机制
1	工作日报	项目组每个成员	每日一报	开发测试人员报告给组长,抄送 PM,组长报告给 PM,抄送主管,依次逐级上报。
2	项目进度跟踪表	具体模块组长	每日一报	组长对模块的工作进度检查及工作任务安排,报告给 PM,抄送模块组成员。
3	项目周报	PM (项目经理)	每周一报	公司领导,抄送项目组成员
4	里程碑报告	PM	阶段结束	公司领导,抄送项目组成员
5	验收问题的统计分析报告	PM	每天一次	公司领导
6	项目结项总结报告	PM	项目结束	公司领导
7	各轮测试完成后的本轮测试报告	PM	每轮归档	公司领导
8	例外报告	PM	待定	公司领导

表 3 描述了项目报告机制的具体类型、准备人、报告的频率和报告的机制。项目组中的模块负责人不仅要写每日工作日报,还要写自己负责模块的进度表。

每日工作日报中要清晰地记录个人每天的工作内容,记录越清晰越好,时间单位可以精确到分。比如:需求分析累计工时 1 小时,代码编写累计工时 1.5 小时,评审累计工时 1 小时,测试用例编写累计工时 2 小时,搭建环境(配置环境)累计工时 1.5 小时,培训学习累计工时 1 小时,总计工时 8 小时。如果企业是第三方外包公司的话,PM 必须要把项目周报、里程碑报告、验收问题的统计分析报告、各轮测试完成后的本轮测试报告、例外报告发给合作方的 PM。因为合作方有权知晓项目的具体进度,以便合作方及早校验开发的项目是否与具体需求相符合。

3.2.4 项目进度

项目进度是指制定项目中具体阶段的开始和结束日期。这里的具体阶段细分为系统规格、需求分析阶段、编写测试计划阶段、编码阶段、单元测试阶段、联调测试阶段、系统测试阶段。其中系统测试需要的时间是最长的。

3.2.5 评审计划

评审贯穿与全部项目流程,是不可缺少的,每个阶段都必须评审。评审分为外评和内评。通过评审交流这样可以更加及时的发现和纠正在项目监管方面的问题,提高产品质量和管理水平,并及时化解风险。评审方式有评审通知、预评审、会议确认、返工、跟踪等。评审计划就是对评审项安排具体时间,评审专家进行交流沟通。此处将评审计划列为一个表格统一放在项目计划 Excel 表格中管理。

3.2.6 质量目标与培训计划

质量目标是对项目缺陷密度的控制,但是缺陷必须控制在一定范围里<sup>[12]</sup>。在质量目标里规定了项目级质量目标的上下限和目标。度量方式主要用缺陷密度来描述,可以是每页缺陷个数、所有测试用例中未达标的用例个数、模块中缺陷的个数、行代码中缺陷个数等。

质量活动需要特定的过程技术的培训。在培训计划里必须要标明培训范围和对象、组织培训的责任人、培训讲师、培训的目标。培训的对象是人,需比较在培训前后,个人和团队在技术上是否有所提高。从长远教育上来看,培训过的人员独立解决问题的能力是否得以提高。

3.2.7 环境工具与重用

环境具体指开发、测试、运行环境。工具指的是开发工具。重用包括重用资产使用清单和重用资产提交清单。在重用资产提交清单中可以列出计划在版本开发结束后可以提交的可供重用的资产。重用的内容包括但不限于:(1)源代码;(2)模块设计;(3)架构设计;(4)设计模式;(5)开发或测试工具。

### 3.3 基于 CMMI 的开发流程

产品的质量在很大程度上取决于用以开发和维护该产品过程的质量,所以下面讨论合理的开发流程,从过程上加以改进。以往的开发流程中,很少注重评审的重要性,但是评审贯穿于 CMMI 的 5 个标准,评审也贯穿于整个项目的全过程,是质量保证的主要直接手段。基于 CMMI 标准的开发流程如图 1 所示,图中可以清晰地看出从需求规格开始直到单元测试都必须评审。

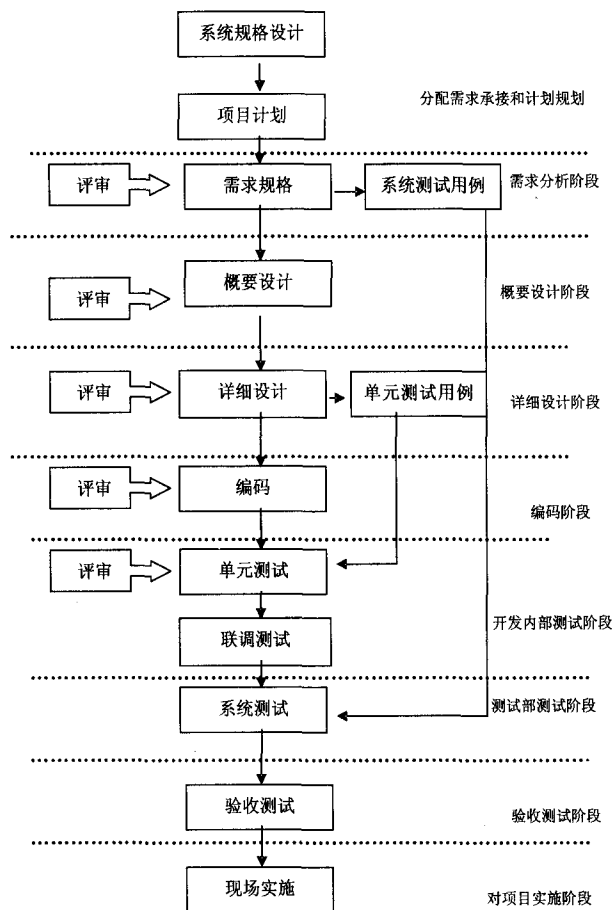


图 1 基于 CMMI 标准的开发流程

加入评审的需求规格书的形成如图 2 所示,图中评审需求虽然只显示了一次,可是这里的评审是一个循环的过程,只有通过不断的评审所有管理者对需求文档意见达成一致时才可以形成需求规格书。当然在此期间还需要用户的参与。

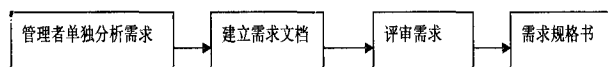


图 2 需求规格书的形成

在软件设计的全阶段通过不断的评审给出最终的概要设计说明书、详细设计说明书等。在评审计划中,会列出各个评审项的评审时间。在实施基于 CMMI 的开发流程图时可以参考评审计划。

### 3.4 引用 CMMI 准则的结果

软件质量保证措施中的测试、风险管理、进行度量

等一系列活动都合并到完善详尽的项目计划中。基于 CMMI 的第二级(已管理级)和第三级(已定义级)的项目计划书指导企业从一个杂乱、不成熟的状况向成熟、规范的过程演进,使生成率比改进前有所提高。通过对项目开发、测试、资料三方面的初次估算,在给定资源、预算的范围内,安全地完成符合质量目标的客户要求。

基于 CMMI 的开发流程比以往的流程图多了多次评审。通过一次次的评审才能够在定量的范围内预测控制过程,实现及时纠正错误,能够交流沟通引进先进的新思想、新技术促使过程持续不断改进。

不管是项目计划还是开发流程都是在 CMMI 标准的模式下设计的。企业必须严格按照标准流程设计计划;按标准流程图走开发流程,才能真正的通过过程改进达到管理提高的目的,最终使得软件质量有所保证。

## 4 结束语

CMMI 标准模型的实践只是告知要做的事,但并未说明如何做,文中作者创新点:详细说明了在 CMMI 标准模型下的项目计划以及开发过程,并将计划中的各个活动都制成表格放在 Excel 中统一管理,监控软件过程,从而保证软件的质量。这里的基于 CMMI 标准的项目计划和开发流程不仅只适用于规模较大的组织和较大型的项目,而且也可以广泛应用于相当多的小型组织。

### 参考文献:

- [1] Siponen M, Willison R. Information security management standards: problems and solutions[J]. Information & Management, 2009, 46(5): 267-270.
- [2] 黄 劲, 辛小霞, 黄 镔, 等. 能力成熟度模型 CMMI 与 CMM 的比较和研究[J]. 现代计算机, 2008(9): 27-30.
- [3] Mathiassen L, Scrensen C. The capability maturity model and CASE[J]. Information Systems Journal, 2008, 6(3): 195-208.
- [4] Florac W A, Park R E, Carleton A D. Practical software measurement measuring for process management and improvement[M]. Pittsburg: SEI, 1997: 180-181.
- [5] 张仲雷. 基于 CMMI 的软件项目质量管理框架[J]. 中小企业管理与科技(下旬刊), 2009(9): 278-279.
- [6] 管东升, 吕小刚, 赵云丰. 基于能力成熟度的信息系统监理过程改进研究[J]. 计算机技术与发展, 2011, 21(1): 137-139.
- [7] 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.
- [8] 聂 南, 邓璐娟, 夏启明. 面向不同行业软件的质量评测模型及实践[J]. 计算机科学, 2011, 38(2): 156-159.

(下转第 20 页)

-AGC 拦截目标程序对堆内存管理服务程序的调用,需要拦截的函数如表 1 所示。

表 1 堆内存服务程序列表

CRT 函数	Window API 函数
malloc/free	HeapAlloc/HeapFree
realloc/free	GlobalAlloc/GlobalFree
calloc/free	LocalAlloc/LocalFree

以上函数的实现代码都在动态链接库中,由所有 Windows 应用程序共享。当应用程序中出现对以上函数的调用,连接器会将函数所在的动态链接库导入该应用程序空间。当应用程序调用一个来自其它模块的函数,由编译器发出的调用指令不会将控制直接转移到该模块中的函数,而是利用一条 JMP 指令跳转到真实的函数入口点。函数入口点的地址保存在 PE 文件的 .idata 段中。

RT-AGC 修改目标应用程序 .idata 段中的函数地址,将需要被拦截的服务函数地址指向托管函数,在托管函数中执行信息的收集工作之后,再调用实际的服务函数,以获得已分配的堆内存信息。

3 RT-AGC 的运行测试

为了测试 RT-AGC 的运行效果,编写了测试目标应用程序 TestTargetAppl;TestTargetAppl 使用 C 语言编写,每隔固定时间间隔申请堆内存但从不释放。活动指针分布于全局数据区、栈区和堆空间。在以下两种情况下运行 TestTargetAppl 并查看其内存使用情况:

- 1. TestTargetAppl 在没有 RT-AGC 干预的情况下独立运行:内存使用量随时间呈线性增长,其增长率与每次申请的堆内存大小相关,说明 TestTargetAppl 发生了严重的内存泄漏;
- 2. 在 RT-AGC 干预的情况下,TestTargetAppl 的内存使用量在经过一段时间的线性增长后,逐渐趋于稳定。

以上实验说明 RT-AGC 虽然不能完全清除内存泄漏情况,但是可以有效地抑制内存泄漏。这一点在需要长期、大量的申请堆内存的应用程序上表现的尤为明显。

4 结束语

文中描述了一种在 Windows 下应用程序垃圾内存

的回收器的设计与实现。回收器采用 Mark-Sweep 算法,其中关键信息的获取通过分析 PE 文件结构和调用 API 函数实现。最后通过实验证明了其有效性。

但是文中只是从较高层对 Windows 应用程序的垃圾回收器的设计与实现进行描述,尚有一些细节问题需要考虑:文中实现的 RT-AGC 无法有效的作用于已加壳的应用程序;另外,对于应用程序中由其它模块分配的堆内存,RT-AGC 无法进行拦截和回收。在这两方面进行改进将是下一步工作。

参考文献:

[1] Jones R, Lins R. Garbage Collection: Algorithms for Automatic Dynamic Memory Management [M]. [s. l.]: John Wiley & Sons, 1996.

[2] Boehm H J, Weiser M. Garbage collection in an uncooperative environment[J]. Software Practice and Experience, 1988, 18(9): 807-820.

[3] Wentworth E P. Pitfalls of conservative garbage collection[J]. Software Practice and Experience, 1990, 20(7): 719-727.

[4] Willard B, Frieder O. Autonomous garbage collection: resolving memory leaks in long-running server applications[J]. Computer Communications, 2000, 23(10): 887-900.

[5] Aho A V, Sethi R, Ullman J D, et al. 编译原理[M]. 李建中, 译. 北京:机械工业出版社, 2003.

[6] Horowitz E, Sahni S, Anderson-Freed S, et al. 数据结构(C语言版)[M]. 李建中, 译. 北京:机械工业出版社, 2006.

[7] Michael G. Easy Detection of Memory Leaks[EB/OL]. 2005-08-04 [2011-04-26]. <http://www.codeproject.com/KB/cpp/MemoryHooks.aspx>.

[8] 杰夫瑞, 克里斯托夫. Windows 核心编程[M]. 葛子昂, 周靖, 廖敏, 译. 北京:清华大学出版社, 2008.

[9] Robert K. Three Ways to Inject Your Code into Another Process[EB/OL]. 2003-08-21 [2011-04-20]. <http://www.codeproject.com/threads/winspy.asp>.

[10] 王 峥, 姜渊胜. 远程线程注入技术在监控系统中的应用[J]. 计算机技术与发展, 2010, 20(3): 207-210.

[11] Pietrek M. An In-Depth Look into the Win32 Portable Executable File Format[EB/OL]. 2002-02 [2011-05-10]. <http://msdn.microsoft.com/en-us/magazine/cc301805.aspx>.

[12] Randy K. Managing Heap Memory[EB/OL]. 1993-04-03 [2011-05-03]. <http://msdn.microsoft.com/en-us/library/ms810603.aspx>.

(上接第 16 页)

[9] 王艳慧. 基于 CMM 的软件过程改进实践[J]. 计算机技术与发展, 2008, 18(5): 141-143.

[10] 龚 波. 能力成熟度模型集成及其应用[M]. 北京:中国水利水电出版社, 2003.

[11] 李红梅. 软件项目跟踪与监控[J]. 科技论坛, 2005(17): 17-17.

[12] 邓治文, 商惠华, 戴汇川. 基于需求分析的软件质量目标策划方法[J]. 微计算机信息, 2010, 26(18): 187-188.