第29卷 第5期 计 算 机 工 程 2003年4月
Vol.29 № 5 Computer Engineering April 2003
· 软件技术与数据库 · 文章编号：1000—3428(2003)05 —0078—04 文献标识码：A 中图分类号： TP311.5

净室软件工程在CMM中的应用技术研究

勉玉静，赵文耘，陈颂梅
（复旦大学计算机系软件工程实验室，上海 200433）

摘 要：净室是一种追求开发的零缺陷，以经济的方式生产高质量软件产品的新型软件技术。该文通过介绍CMM和净室软件工程技术，对净室技术在CMM的应用实践中遇到的问题作了分析，提出了基于CMM的净室裁剪的原则，从而给出了一种裁剪后的净室技术对各级CMM关键域的支持方案。

关键词：软件过程改进；能力成熟度模型；净室软件工程；净室裁剪

Research on Application of Cleanroom Software Engineering in CMM

MIAN Yujing , XHAO Wenyun , CHEN Songmei
(Software Engineering Lab., Dept. of Computer Science, Fudan Univ., Shanghai 200433)

【Abstract】Cleanroom software engineering(CSE) is a kind of new software technique. It pursues zero-defect in development in cost-effective ways. After introducing CMM and CSE techniques, this paper analyses the problems in the practice of CSE, gives some principles and approaches of CMM-based tailoring of CSE .

【Key words】Software process improvement; CMM; Cleanroom software engineering(CSE); Tailoring of CSE

1 概述

软件的质量决定了软件生产组织的生存和发展，因此，软件组织坚持不懈地寻找着提高软件质量的有效方法。从结构化编程到面向对象技术，新技术新方法层出不穷，但软件生产中仍常常出现无法按时、按预算、保质保量的支付软件产品的情况，即使按时完成，也往往留有很多隐患，需要高成本的维护工作。近10年来，随着软件过程工程的发展，人们逐渐意识到软件质量很大程度上取决于软件开发过程，通过过程改进可以持续有效地提高软件质量。

美国卡内基梅隆大学软件工程研究所提出的软件能力成熟度模型，成为软件组织进行软件过程改进以及评估和评价软件能力的基准。但在具体的过程改进实施中，需要有效的软件工程方法的支持。而净室软件工程正是为过程改进提供了具体实施方法，它能够及早发现并消除缺陷，显著提高软件的正确性、可靠性和可理解性，降低项目的成本，提高软件质量，延长软件的生命周期。

我们认为，完全可以将净室软件工程应用到CMM的实践中，从组织管理和技术工程实践两个方面改进软件过程，从而更加经济有效地提升软件质量。

1.1 CMM体系

CMM认为保障软件质量的根本途径是提升企业的软件生产能力，而这正取决于企业的软件过程能力。CMM为软件开发过程提供了一个框架，这个框架同软件的生命周期和开发技术均无关系。它将软件过程改进的进化步骤组织成阶梯型的5个成熟等级：初始级(1级)，可重复级(2级)，已定义级(3级)，已管理级(4级)和优化级(5级)。其中除初始级外，每一等级都包含一组关键过程区域，指明达到这个成熟度级别所必须着手解决的问题和必须满足的要求。每个区域通过实施相应的一组关键实践来达到所要求达到的过程目标，为过程不断改进奠定了循序渐进的基础。

CMM不仅为商业领域软件项目的招投标活动提供了评判依据，更重要的是，它为软件组织所不断进行的过程改进提供了由低到高、由浅入深的明确方向和目标。软件组织通过CMM给出的框架，对自身的软件过程作出评估，比照不足，规范自身的内部结构，并参照自身的结构来建立软件过程，以提高软件过程成熟度。但是，在CMM中，关键实践仅仅描述了应该“做什么”，并没有给出更没有规定“如何”去具体操作，操作的方法和步骤可以由也必须由软件组织自己去解决。因此，CMM只是对软件组织过程改进的指导，而非解决一切软件开发过程中的问题的法宝。在实施CMM的过程中，仍然需要有效的软件工程技术和方法，如“净室软件工程”方法的支持。

1.2 净室软件工程

1.2.1 基本概念

净室软件工程是一种应用数学与统计学理论以经济的方式生产高质量软件的工程技术，力图通过严格的工程化的软件过程达到开发中的零缺陷或接近零缺陷。“净室”一词源自半导体工业中硬件生产车间，通过严格、洁净的生产过程预防了缺陷的产生，而不是在事后再去排除故障。借用这个词，充分显示了净室技术“防患于未然”的主导思想。

净室软件工程为“如何”在软件过程改进中进行质量控制提供了具体手段。净室采用小组开发的组织形式和增量式的软件生命周期模型，依靠基于数学函数理论的形式化的开发过程和基于统计学原理的测试方法，及早发现并消除缺陷，减少在开发后期弥补和修正开发前期的缺陷而带来的巨大的支出，大大降低软件维护的成本，缩短软件的开发周期，提高软件生产率。

自20世纪80年代至今，净室软件工程在IBM、美国宇航局、美国国防部、爱立信、parx(法国第一家获得CMM4级

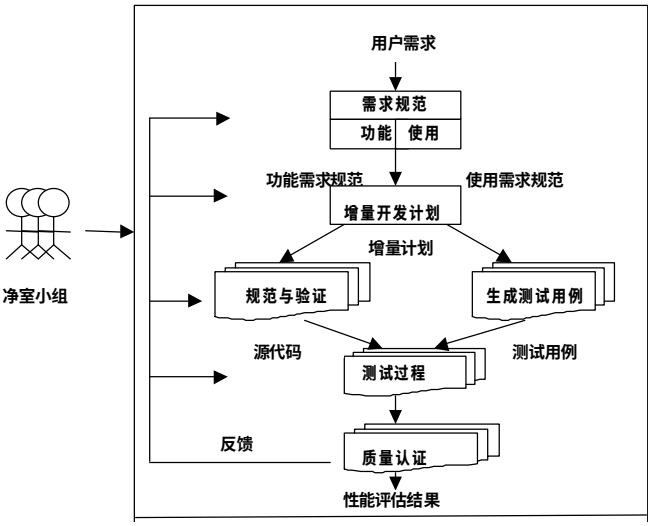
基金项目：国家“863”项目：基于Internet的以构件库为核心的软件开发平台（2001AA110241）
作者简介：勉玉静(1978~)，女，硕士生，研究方向：软件工程，软件过程，软件过程改进；赵文耘，教授；陈颂梅，硕士生
收稿日期：2002-03-28

认证的网站)等许多机构中得到实践，在提高生产率和降低缺陷率方面都取得了骄人的成果。1996年美国国防部软件数据和分析中心在其所作的软件方法比较分析中，报告净室具有真实的价值和质量优势。

1.2.2 过程模型

净室软件工程有其特定的开发过程模型，如图1所示。

表1简明列出了净室软件过程模型的特点和作用。



说明：粗体字表示过程。重叠部分表示为增量。箭头旁文字表示过程的中间产品。

图1 净室软件过程模型

表1 净室过程的优点

特点	小组开发的组织模式	统计控制下的增量开发	开发与测试的并行进行
所起作用	降低人员间的通信和协调减少对权威的依赖提高团队开发能力，小组评审尽早发现缺陷并显著降低成本	开发过程可预测开发进度可见使得开发在智能控制下易于适应需求的变化，促进持续的求精	开发阶段就完成预防缺陷和修正缺陷的工作测试过程完全从用户使用角度出发重视质量的度量与反馈

1.2.3 关键技术

在净室的规范、验证和测试的过程中，采用了以下几种关键技术：基于函数理论的盒式规范，正确性验证以及统计学理论为基础的软件测试方法(简称统计测试)。

表2简明列出这些技术在软件过程的优点与不足。

表2 净室过程中的关键技术分析

关键技术	优点	不足
盒式结构的规范方法	促进对需求的明确和理解，利于复用，规范的文档，便于验证	严格的形式化语言描述规范，难以掌握和使用，需要专有的CASE工具的支持，用户难以理解；初期投入大回报慢
基于数学理论的正确性验证	尽早发现并消除缺陷，提高质量显著降低成本	对一般软件来说，代价太昂贵；正确性证明难以寻找合适的预期函数规范
使用模型的建立和统计测试与认证	从用户使用角度出发，利于需求的理解和对开发人员的反馈，实现对质量和性能的量化	使用模型不易建立；需要统计学知识和CASE工具的支持；需要其他测试方法的补充

1.2.4 净室技术应用于软件企业的困难

从上文可以看到，净室软件过程和技术在减少和缺陷预防，提高软件质量方面有着突出的成效，但是净室自身的一些特点和不足也在一定程度上阻碍了它的推广，这一点在CMM3级以下的软件组织中尤为突出。

首先，净室软件工程有其特定的软件过程模型，在此基础上使用净室的规范、验证和测试技术才能更好地发挥其优势。CMM3级以下的组织，尚未建立自己明确定义的软件过程模型，此时直接套用对文档化、形式化要求较高的各种净室技术，不仅得不到预想的效果，更会带来众多方面不可预测的风险。不够成熟的软件组织永远无法从比其高级的方法中获益。

其次，净室软件工程有其一系列关键技术。这些技术大都与传统的开发技术不同。即使软件组织适宜引入净室技术，开发人员也必定要经过培训、尝试、反复等过程才能达到对净室技术的熟练掌握。任何一种新技术的引入，带来的可能是巨大的利益，同时也可能是更大的风险。对小型企业来说，不可能长期投入大量人力物力资源，更不愿影响正在进行中的软件开发工作，因此很可能在从净室中得到利益前就将其舍弃了。

第三，净室技术中程序正确性证明、统计测试等技术本身也被怀疑过于形式化和理论化。即使开发人员学过这些数学和统计学的知识，因为在传统的软件开发中很少使用，也已非常陌生。另一方面，这些技术的使用本身会提高软件的开发成本，从成本/效益分析的角度来看，并不适宜所有的软件开发。

2 基于CMM的净室裁剪

由于净室过程和技术的优点以及在软件企业中实施所遇到的困难，我们提出对净室进行基于CMM的裁剪。

2.1 基于CMM的裁剪的原则

针对实施CMM的软件企业，我们可以采用以下原则进行裁剪：

(1) 裁剪必须符合净室的基本原则

1) 设计原则：开发人员应该并且能够生产出在被测试前就已达到趋于零缺陷的软件产品来。

2) 测试原则：净室测试的目的不是寻找缺陷，而是度量软件产品的质量和性能，为软件过程的改进提供统计数据。

这是净室区别于传统软件工程方法的关键。

(2) 必须结合软件组织自身的能力成熟度现状。不同级别的组织自身软件能力不同，过程改进的主要目标也不相同。超越自身成熟度的裁剪，不能发挥净室的优势，却可能带来不可预测的风险；而低于自身成熟度的裁剪，达不到期望中提高成熟度的效果，反而浪费了资源。

(3) 必须结合所开发的软件的类型。应用程序易于实现但需求易变，军用、航天等所需的嵌入式系统需求稳定却对质量和性能有严格的要求，这两种软件的开发显然不能完全使用相同的过程和技术。

2.2 基于净室的裁剪方法

2.2.1 引入净室技术的3个阶段

从CMM各级的关键域可以看出，是否达到CMM3级是一个组织软件能力的分水岭。未达到3级的组织，过程改进的重点还在建立基本的项目管理过程，处于还不成熟的阶段。达到3级的组织已经确立了文档化、标准化的软件过程，软件过程能力已经比较成熟。而3级以上组织更重视对

软件质量和可靠性的度量，以量化的数据支持高层的决策，处于成熟度很高的阶段。相应的，我们认为对净室的渐进引入也可以分为3个阶段：

初级阶段：尚在为达到CMM2级努力的组织，几乎没有真正确定的软件过程，其过程能力不可预测，产品性能只能根据相关人员的个人工作能力预测。此时首先要引入净室小组开发的组织模式和质量控制下的增量式生命周期模型，将开发与测试分离，建立起包括需求、计划、规范、验证和测试的基本项目过程。结合自身能力，引入形式化程度较低的黑盒规范与验证方法，实现对软件项目的跟踪和监督；借助质量控制下的增量式开发和功能测试来完成质量保证。

中级阶段：准备通过3级认证的组织，已建立基本的项目管理过程，为一个有纪律的管理过程提供了可重复以前成功实践的项目环境。过程改进的主要目标是将软件过程标准化、文档化。此时可在引入净室过程模型的基础上，加入更多必须的管理规范，明确定义自身的软件过程。同时引入比较形式化的净室规范和验证技术，进一步降低开发阶段的缺陷率，提高软件生产率。并根据需要进行有限的统计测试。

高级阶段：以CMM4、5级为目标的组织，拥有完整的软件过程，对过程的度量和缺陷预防给予更多的关注。此时引入净室统计测试技术，能够很好地实现对质量和性能的量化，为高层的决策提供可靠的数据依据。很可能还需要将统计测试和其他的测试方法相结合。比如统计测试对使用概率很低的功能的测试就可能不足，而这种功能出错后导致的后果可能是非常严重的。此时必须进行极端测试。缺陷预防方面，此时也有能力完整地引入严格形式化的盒式规范和验证技术，以科学的理论保证软件开发阶段的趋于零缺陷。

这3个阶段中，引入的技术的形式化程度由低向高，这就需要对净室技术本身做相应的裁剪，采取形式化程度不同的实现方式。

2.2.2 针对净室技术形式化程度的裁剪

(1) 对盒式规范技术的裁剪

黑盒规范对系统的外部可见行为做一个完整的定义，隐藏了软件设计和实现的所有细节，适用于软件开发的任何粒度中。规范的描述形式可以不同。形式化程度较低的比如自然语言，虽然不够严密，但是易于使用，用户也易于理解。半形式化的规范语言(如Z语言)部分克服了自然语言的二义性，同时加深对需求的理解，得到更为准确、严谨的规范。而严格的函数表达方法，对规范的描述更加精确，易于验证和追踪，但不易使用和理解，文档数量大，对管理要求高，属于非常严格的形式化规范方法。

状态盒规范是对系统内部数据的描述，它的实现形式依赖于黑盒规范。自然语言描述的盒式规范甚至可以省去状态盒，直接对黑盒细化完成明盒。明盒规范是对黑盒与状态盒逐步求精的实现,最终形式便是源代码。既可以是结构化的，也可以是面向对象的，不受开发方法和语言的限制。

(2) 对盒式规范验证技术的裁剪

验证过程基于非执行的测试方法寻找并消除开发阶段的缺陷。因盒式规范的形式化不同，验证方法也有相应变化。

检查方法简单易行，但是不够严格，基于潜在错误清单的审查方法有规范的步骤，对审查的过程、目标、参与人员、结果记录都有明确的要求，能够使错误在软件过程的早期就被发现，规范的文档又便于对验证过程的跟踪。是一种经济有效的错误检测方法。

而基于函数理论的正确性证明，要求在盒式规范过程中，建立明确的预期函数，这就要求盒式规范本身的形式化程度较高，此外要求评审人员有相应的数学知识和专用CASE工具的支持。但是对于那些事关重大或者通过成本效益分析认为迫切需要证明的软件，这种证明仍是必要的。

(3) 对统计测试技术的裁剪

统计测试中使用模型的建立对需求和规范的要求很高，在形式化要求不高的情况下，同样是从用户使用角度出发，可以使用一般的功能测试(即黑盒测试)的方法来评估软件的质量。它同样无须对软件内部结构的了解，比如可以对没有提供源码的构件的功能进行测试；测试本身并不基于对内部的了解，减少了测试人员与开发人员犯同样错误的概率。规范和验证阶段采用的技术都不严格时，更需测试过程来保证产品发布前的低缺陷，以减少产品的维护费用。

综上所述，本文提出了根据CMM的不同级别，分3个阶段，由浅入深地引入净室过程及技术的裁剪方案，这种方案既符合不同成熟度软件组织的能力现状，又能满足过程改进需求的目标。表3显示了引入的净室技术与其所支持的CMM关键域的对应关系，以此作为对上文裁剪方案的总结。

表3 净室过程和技术对CMM关键域的支持

CMM级别	CMM关键域	需要引入的净室过程和技术
2可重复级	需求管理	需求规范
	项目规划	增量式开发，小组开发模式
	跟踪和监督	自然语言描述的黑盒规范质量控制下的增量
	分包合同管理	无专门技术
	质量保证	小组评审制度，功能测试
	软件配置管理	无专门技术
3已定义级	组织过程焦点	净室过程的引入
	组织过程定义	结合净室过程，定义自己的软件过程
	培训大纲	无专门技术
	集成软件管理	无专门技术
	软件产品过程	半形式化语言描述的黑盒，状态盒
	组际协调	小组开发模式中的组际开发模式
	同行评审	基于潜在错误清单的审查
4已管理级	定量过程管理	建立使用模型、统计测试
	软件质量管理	建立使用模型、统计测试，其他测试方法的协助
5优化级	缺陷预防	严格形式化的盒式规范、明盒的正确性证明
	技术变更管理	无专门技术
	过程变更管理	无专门技术

从表3可以看到，CMM18个关键域中，只有以下几个关键域没有净室技术提供直接的支持：外包管理，配置管理，培训大纲，集成管理和变更管理，它们大都属于管理的范畴，与采用的具体技术无关。而净室的过程和技术与它们是一致的，并没有任何冲突。

CMM与净室技术都不是万能的。CMM提出的是完整的软件开发和管理的过程，而净室更多的是技术方面的支持。两者相互一致并相互补充。将二者合理地结合，能够获得更

高的软件质量、更低的开发成本，更高的生产效率和更长的软件生命周期。

3 结束语

爱立信公司在手机操作系统OS32的开发中引入了经过裁剪的净室技术。73人以个人或组际(team of teams)的方式工作了33个月，通过15个增量过程开发了约33万行代码。以通信业广泛使用的SDL语言完成盒式规范；使用了小组评审技术，但没有进行正确性证明；建立了使用模型，但只进行了很有限的统计测试。这个项目取得了令人满意的结果：集成和测试的时间减少了，缺陷率比预期低了50%，而生产率则提高了70%以上。这足以说明，通过合理裁剪的净室技术是能够非常有效地预防缺陷，提高软件质量和生产率的。

本文针对净室技术在CMM中的应用，提出了基于CMM的净室裁剪的原则和一些方法，给出了一种净室对CMM各级关键域的技术支持方案。我们认为通过适当裁剪将净室技术应用于CMM的实施当中，有以下优点：

(1) 引入净室技术，能够及早发现并消除缺陷，降低成本，提高软件质量和生产率。

(2) 净室的引入为CMM的关键实践提供了具体有效的技术，易于基于CMM的软件过程改进的具体实施。

(3) 根据软件组织的成熟度现状渐进地引入净室技术，提高了净室技术的可操作性。

(4) 对成功的软件开发而言，管理和技术缺一不可。净室技术在CMM的实施中的应用,实现了科学的技术与先进的管理的结合。

在今后的研究中，还需要对净室裁剪作更深入的探讨，寻找更为合理有效、易于实践的裁剪方法，制定更为成熟的净室对CMM的支持方案，为改进软件过程，提高软件质量寻求更好的方法和技术。

参考文献

1 Mills H D, Dyer M, Linger R. Cleanroom Software Engineering. IEEE Software, 1987-09

2 Linger R. Cleanroom Process Model. IEEE Software, 1994-03

3 贲可荣, 张志祥, 张秀山等译. 净室软件工程：技术与过程. 北京: 电子工业出版社，2001-06

4 Linger R, Trammell C. Cleanroom Software Engineering Reference Model1.0, CMU/SEI-96-TR-022,1996

5 杨一平. 软件能力成熟度模型CMM方法及其应用. 北京: 人民邮电 出版社，2001-04

☆☆

（上接第29页）

么内容能够用于获得数据，或反之可以保护敏感数据。此处仅考察在与数据无关的框架中基于视图的查询计算的问题。这样，基本上在一定条件下解决了何时视图包含足够计算查询的信息问题。

3 基于实化聚集视图的一种应用模型

在一些应用情形中，如OLAP、决策支持系统DSS和数据仓库中，往往不需要每次查询均需要准确地回答。快速查询回答首先适用于聚集查询，它可以快速提供最重要数据的回答。近似查询方法主要有两类：在线联机查询处理和预计算提纲方法。第一种方法由用户控制回答的情况与改进，第二种方法需在查询之前建立或存储查询提纲，接到查询时回答提纲。第二种方法需要对提纲进行更新维护，这些工作都基于预处理方法，通过图1可以了解其工作原理：

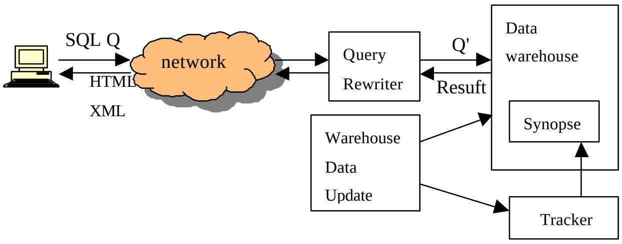


图1 查询工作原理

用户发出的查询Q，通过网络期望从数据仓库中获得查询结果，此计算模式在数据仓库与用户网络之间增加了一个中间件，该中间件负责将查询Q通过重写算法产生查询重写Q', Q'首先查询数据仓库中的提纲，重写查询根据提纲可以快速查询到所需信息，通过数据仓库返回查询结果到中间件，最后通过网络以用户需要的方式提供查询结果。其中，提纲中的信息内容由跟踪器对其进行更新，提纲的内容由数据仓库中根据查询概率产生，数据仓库与跟踪器的数据由数据仓库的数据更新机制实现。近似查询处理方法与聚集查询计算相结合获得的这种计算框架，可以广泛应用于具有统计信息特征的数据的高效处理。

4 总结

本文主要围绕在统计数据库、数据仓库和联机分析处理OLAP 中的建立在聚集数据上的一类特殊实化视图——实化聚集视图为核心，对基于实化聚集视图的查询计算问题及相关研究的进展进行了分析和总结。基于实化聚集视图的研究，虽取得了一些有理论价值和意义的结果，但这只是一些初步的工作与结论。本文在分析上述内容的基础上，提出了具有广泛应用前景的计算方案，该方案主要依据所获得的数据仓库的实化聚集视图的查询计算理论同较新的面向应用的近似查询处理方法相结合，提出了一种基于实化聚集视图的近似查询处理计算框架，其主要意义：可将其应用于具有统计信息特征的数据处理，有广阔的应用前景和社会效益。

参考文献

1 Cohen S, Nutt W, Serebrenik A. Rewriting Aggregate Queries Using Views. In Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Philadelphia, Pennsylvania，ACM Press，1999-05-31~06-02:155-166

2 Rafanelli M, Bezenchek A, Tininini L. The aggregate Data Problem: A System for Their Definition and Management. ACM Sigmod Record， 1996,25 (4):8-13

3 Grumbach S, Tininini L. On the Content of Materialized Aggregate Views. In Proceedings of the Nineteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database System，2000

4 Chaudhuri S, Das G, Narasayya V. A Robust， Optimization-based Approach for Approximate Answering of Aggregate Queries. ACM SIGMOD 2001

5 Sristava D, Dar S, Jagadish H, Levy A. Answering Queries with Aggregation Using Views. In Proc. of Intl. Conf. on Very Large Data Bases, 1996:318-329