

# 净室软件工程研究

常 郝

(安徽财经大学 信息工程学院, 安徽 蚌埠 233030)

**摘 要:**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。文中介绍了净室软件工程的理论基础,探讨了净室工程的关键技术,研究了净室参考模型及其不足。结果表明,净室软件工程对于成熟的软件开发组织和专业的技术人员,在资源充沛的情况下,其面向零缺陷的盒式结构开发方法和正确性验证能够有效地保证软件的质量。

**关键词:**净室软件工程;增量式开发;盒式规范与设计;净室参考模型

**中图分类号:**TP311.5

**文献标识码:**A

**文章编号:**1673-629X(2008)12-0127-03

## Research of Clean Room Software Engineering

CHANG Hao

(School of Information and Engineering, Anhui University of Finance and Economics, Bengbu 233030, China)

**Abstract:** Clean room is an engineering technology of developing high quality and reliability software based on mathematics and statistics. Introduced theory of clean room software engineering, and then researched some key technologies of clean room; at last, the clean room model and disadvantages were studied. The result proves that software quality can be ensured by correctness verification using box structure oriented zero defects in terms of well-rounded organization with professional technology and enough resources.

**Key words:** clean room software engineering; incremental development; box specification and design; clean room model

## 0 引 言

传统的软件工程方法包括分析、设计、编码、测试等步骤,随着软件规模的增大,软件开发过程内部的不一致性问题日渐突出,风险也越来越大。

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<sup>[1]</sup>。“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。

净室软件工程的目标之一就是实现软件开发的工程化,通过工程化的软件开发过程,有效地控制软件开发中的任何一个步骤,从而实现可预期的软件产出。净室软件工程的目标之二是改变以往在软件开发后期来改正软件失效的做法,通过增量开发、严格的开发规范及设计和基于统计学的可靠性测量来达到软件零失

效的目标。

经过 IBM, NASA 等机构在实践中使用净室技术开发的产品,显示出了卓越的质量水平及用户使用可靠性,使得净室方法得到了初步认可<sup>[2]</sup>。

## 1 净室理论基础

净室理论基础来自于数学。Mills 把计算机程序看作一个数学函数,同时 Mills 认为软件测试只是从无限的可能使用中抽取的样本<sup>[3]</sup>。这使得净室软件工程成为一个真正的工程学科。

### 1.1 函数理论

净室开发方法基于数学中的函数理论。函数定义了从定义域到值域的映射,定义域中的每个元素都可在值域中找到唯一的元素与之对应。程序的规范就是函数的规范,描述了程序的输入序列到输出空间的映射。一个定义明确的函数有如下特性:完备性、一致性和正确性,因此程序规范必须满足完备性、一致性和正确性,即每种可能的输入都必须定义,有且仅有一个输出与之对应;对于需求的正确性由领域专家判断,而对于给定设计及其规范的正确性通过基于函数理论的推理来验证。

收稿日期:2008-04-14

基金项目:安徽省自然科学基金资助项目(KJ2008B087)

作者简介:常 郝(1983-),男,安徽寿县人,讲师,硕士,主要研究方向为软件过程、生物特征识别技术、计算机网络与信息安全。

## 1.2 统计理论

净室测试方法基于统计学。当从经济上或技术上无法测试样本全体时,可以使用统计抽样的方法。如果统计结果没有达到质量目标,生产过程需做必要的调整。这种以统计学为基础的从产品度量到生产过程之间的反馈循环,得到了广泛的认可和应用。

在软件中,用于采样的全体是所有可能使用情况的集合。集合中的每个元素代表系统的一种可能运行情况。统计的目的是度量系统正确运行一个样本的能力。因为总体是无限的,完全测试是不可能的,所以必须利用统计学方法来对系统发生作一个有效的推理。测试过程不论如何扩展,在所有可能的输入序列中都只能算一个很小的集合,所有的测试活动只能是无限总体中的抽样。在净室软件工程中,统计测试可用于产品检测,也可用于过程检测。净室采用增量开发的迭代过程,这样可测量并提高运行的一致性。

## 2 净室工程的关键技术

净室技术是一种开发高质量和高可靠性软件的方法,由三大关键技术来刻画:统计过程控制下的增量开发,基于函数的规范、设计和验证,以及统计测试和认证<sup>[4]</sup>。

### 2.1 增量式开发过程

统计质量控制下的增量式开发是软件项目建立和保持管理控制的净室途径。增量式开发有助于早期的和连续的质量评估、用户反馈并方便开发进度的过程改进,避免了在开发周期中期部件集成后风险的继承,而且增量式开发允许在开发周期整修过程中根据需求变化进行系统协调。

增量式开发的技术基础是引用透明性特征。在软件开发的前后,这种特征要求规范及其实现定义同样的数学函数。当拥有了这种特征时,设计就能显示出与其规范的一致性。大的软件系统由各个部分组成。系统各个部分组成的方式对项目的成功有重要的影响。增量式自顶向下的开发途径表现为软件系统的已开发和已测试部分作为功能累积子集的序列。在最早增量中开发了一个小系统,然后把功能添加到每一个后续增量中直到系统完成。这种软件系统增长方式有利于客户、管理者,同样有利于技术人员。

已在净室中实践的增量式开发为统计过程控制提供了基础。每一个净室增量都是过程的一个完整周期,包含规范、开发和新的用户函数的验证,以及到目前为止所有已完成的测试。作为统计过程控制的典型,把过程的每一次的性能度量与性能目标相比较,以决定过程是否一直在控制之下。

净室软件小组通常使用在测试中的开发性能度量作为过程控制的标准。通常使用的度量包括每千行代码的错误数、失效的间隔时间(MTTF)、可靠性及可信性。其它过程控制方法或许依赖于所管理的事务,而不是产品的质量。进度一致性、预算一致性、整体计划的一致性,都是按增量的实际性能与目标性能相比较而言。净室增量度量依据的标准描述了过程控制的具体级别。如果标准不符合,开发小组能从增量中检测执行数据,确定问题所在,必要时调整项目计划,修改软件开发过程,避免此类问题的再次发生。例如,如果增量的测试提示过程失去控制(如质量标准不符合),开发者们应停止测试,返回设计阶段;如果过程是在控制之下,下一步增量工作才能继续。

### 2.2 盒式规范与设计

盒子结构是在规范和设计中对现实系统的外在基本属性的功能描述。净室软件工程中描绘了三种盒子:黑盒、状态盒、明盒。这些盒子不仅展示了外部行为,而且还提高了内部可见度。黑盒确定了一个系统或系统组件的外部行为。状态盒指定了完成外部行为所需的状态数据。明盒则进一步把状态盒具体化,它确定了完成状态盒行为的过程设计。它可使用已有的黑盒或引入新的黑盒,这些黑盒将在以后细化,每步细化是根据前一步进行验证的。这样盒子结构将系统开发的行为、数据和过程三个方面的规范分离开,但又把它们连成一个细化和验证的内聚过程。盒子结构是基于对象的,并支持软件工程的关键原则:信息隐藏和实现分离。

### 2.3 净室软件认证和测试

净室软件测试和认证方法是基于模型的统计测试在软件上的一种应用。统计测试时,需要建立软件运行时的使用模型,测试用例由该使用模型随机产生。然后,按照数学和统计学模型对结果进行分析,获取软件的质量度量,并判断测试的充分性。传统的结构化测试方法是净室统计使用测试方法的一种补充,因此,不必放弃该方法。不过,大量实践表明,基于使用模型的测试更经济有效,并且能获得实用软件的高可靠性。

软件系统基于使用模型的统计测试提供了软件产品和过程质量的度量标准,它将用于软件的整个生命期的管理和决策。由于使用模型是基于规范而不是基于代码的,因此,源于模型构筑的洞察可用于产生在工程的早期阶段避免出现问题的有价值的管理决策。

就统计测试而言,软件测试被看作是一个统计学方法的问题。先产生软件所有可能使用的一个子集,并以这个子集所表现的性能作为依据来考虑整体使用性能。换句话说,就是通过样本来描述总体。作为一

个出发点,这种类比的前提是:不可能对软件的所有可能应用都进行测试。软件使用的过程被认为是一个随机过程。一个 Markov 过程就是一个具有 Markov 性质的随机过程,其中,序列中的下一个事件只依赖于当前而与过去无关。Markov 理论已经用于软件使用模型的分析 and 开发之中,相关的数学方法也已被运用到模型优化之中<sup>[5]</sup>。软件的使用模型可用有穷状态、离散参数的 Markov 链表示。Markov 链的标准分析结果将有助于分析长期运行使用的情况。给定一个使用模型的约束系统,通过数学方法可以得到满足一定目标条件的最优化模型。形式化思想在净室软件认证中的应用,为当前的实践和技术进步提供了坚实的理论基础。

### 3 净室参考模型

软件能力成熟度模型注重于软件生产的管理和组织,定义了五个软件成熟度级别,而净室软件工程则侧重于技术和实践,从工程的角度将软件生产过程化,从而实现对软件生产进行可预期的控制和管理。二者互相支持,从管理和技术两方面保证软件工程的成功实施<sup>[6~8]</sup>。净室参考模型(CRM)定义了完整的净室软件生产过程,它由四个相互独立又相互影响的过程组成:

(1)管理过程,包括项目规划、项目管理、性能改进、工程变化;

(2)规范过程,包括需求分析、功能规范、使用规范、结构规范、增量开发计划;

(3)开发过程,包括软件再工程、增量设计、正确性验证;

(4)认证过程,包括建模测试计划、统计测试和认证。

#### 3.1 管理过程

在项目规划过程中,依据项目要求裁剪净室过程,制定和维护软件开发计划。在项目管理过程中,管理和控制增量开发和认证,定义净室质量目标。在性能改进过程中,从开发计划、过程控制等方面来评估项目性能,并提交改进办法。在工程变化过程中,为工程实施中可能出现的变化制定相应的策略和记录,并评价其影响大小。

#### 3.2 规范过程

在需求分析过程中,清晰完整地提出目标系统所需完成的功能。在功能规范过程中,基于需求定义软件功能的外部规范。在使用规范过程中,定义软件用户对软件的使用方式及使用环境。在结构规范过程中,明确软件的系统结构,并分析所用系统结构怎样在本次软件计划中体现其优越性。在增量计划过程中,

建立软件的增量式开发和认证计划,以组件的方式逐步实现用户功能,对软件质量实现分步控制,最终形成目标系统。

#### 3.3 开发过程

在软件再工程过程中,实现软件的重用,避免同一功能重复开发从而减少软件失效的概率。在增量设计过程中,为特定的软件功能分配一系列增量,并根据整个项目结构和进度来安排这些增量的开发。在正确性验证过程中,初步找出在增量开发中出现的软件失效并加以改正。

#### 3.4 认证过程

在测试过程中,建立模型用于软件的测试和认证,并使用模型来产生测试用例。在统计测试和认证过程中,将测试纳入软件开发的全过程,并认证软件的正确性,评价其性能。

净室参考模型应用于实践的开发流程如图 1 所示。

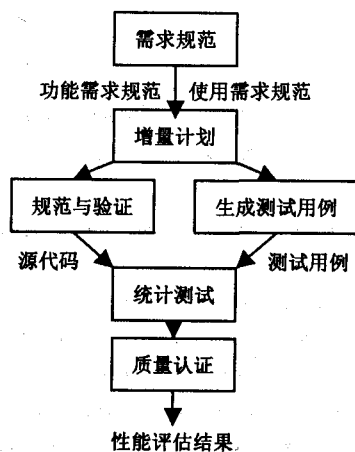


图 1 净室参考模型

### 4 净室软件工程的不足

净室软件工程在减少和缺陷预防,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<sup>[7]</sup>。

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。对于一些不成熟的软件开发组织,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。

其次,净室软件工程有其一系列关键技术。即使软件组织适宜引入净室技术,开发人员也需要经过培训、尝试、反复等过程才能达到对净室技术的熟练掌握

(下转第 133 页)

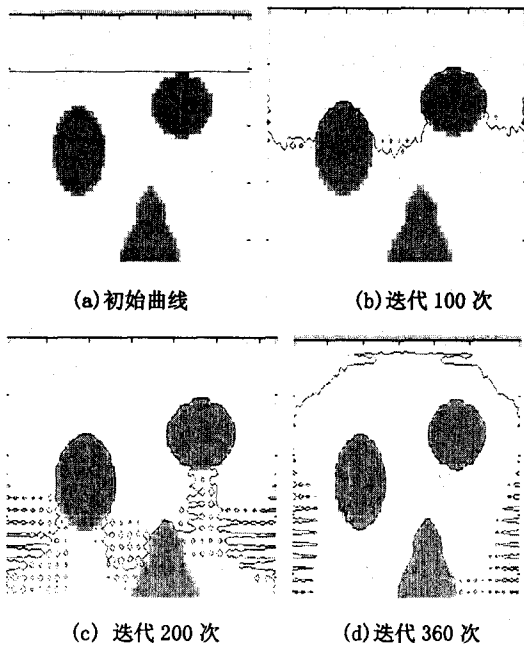


图 3 初始轮廓为一水平线

## 参考文献:

- [1] Han X, Xu C, Prince J. A topology preserving level set method for geometric deformable models[J]. IEEE Trans. Patt. Anal. Mach. Intell., 2003, 25: 755-768.
- [2] Caselles V, Catta F, Coll T, et al. F. Dibos, A geometric model

- for active contours in image processing[J]. Numer. Math., 1993, 66: 1-31.
- [3] Malladi R, Sethian J A, Vemuri B C. Shape modeling with front propagation: a level set approach[J]. IEEE Trans. Patt. Anal. Mach. Intell., 1995, 17: 158-175.
- [4] Osher S, Sethian J A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations[J]. Journal of Computational Physics, 1988, 79(1): 12-49.
- [5] 冈萨雷斯. 数字图像处理[M]. 第 2 版. 北京: 电子工业出版社, 2003.
- [6] 李俊. 基于曲线演化的图像分割方法及应用[D]. 上海: 上海交通大学, 2001.
- [7] Alvarez L, Lions P-L, Morel J-M. Image selective smoothing and edge detection by nonlinear diffusion II[J]. SIAM Journal on Numerical Analysis, 1992, 29(3): 845-866.
- [8] Peng D, Merriman B, Osher S, et al. A PDE-based fast local level set method[J]. J. Comp. Phys., 1999, 155: 410-438.
- [9] Gomes J, Faugeras O. Reconciling distance functions and Level Sets[J]. J. Visual Commun. and Imag. Representation, 2000, 11: 209-223.
- [10] Arnold V I. Geometrical Methods in the Theory of Ordinary Differential Equations [M]. New York: Springer-Verlag, 1983.

(上接第 129 页)

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。

最后,净室技术中程序正确性证明、统计测试等技术本身极具形式化和理论化,即使开发人员学过这些数学和统计学的知识,但是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。

## 5 结束语

净室软件工程是一种应用数学与统计学以经济的方式生产高质量软件的工程技术,它提出一种强调正确性的数学验证和软件可靠性的认证的软件工程模型,其目标是极低的故障率,面向零缺陷的盒式结构开发方法和正确性验证将有效保证软件的质量,这是使

用传统的欠形式化软件工程方法难以做到的。

## 参考文献:

- [1] 冯建湘,楚涤修. 基于体系结构的构件化软件净室设计方法[J]. 武汉大学学报:工学版, 2004, 37(3): 123-126.
- [2] Prowell Stacy J, Trammell Carmen J. Clean Room Software Engineering Technology and Process[M]. Reading, MA: Addison Wesley, 1999.
- [3] Mills H, Dyer M, Linger R. Clean Room Software Engineering[J]. IEEE Software, 1987, 4(5): 19-25.
- [4] 张志斌,高峰,唐朝京. 净室软件工程中的关键技术研究[J]. 计算机应用研究, 2003(2): 17-20.
- [5] Whittaker J A, Thomason M G. A Markov Chain Model for Statistical Software Testing[J]. IEEE Transactions on Software Engineering, 1994, 20(10): 812-824.
- [6] 熊伟,贲可荣. 净室技术与软件能力成熟度模型的融合[J]. 武汉大学学报:自然科学版, 1999, 45(5): 691-694.
- [7] 勉玉静,赵文耘,陈颂梅. 净室软件工程在 CMM 中的应用技术研究[J]. 计算机工程, 2003, 29(5): 78-81.
- [8] 杨涛,王铮,谭歆,等. 从设计角度对净室技术的分析[J]. 重庆大学学报, 2004, 27(1): 88-91.