# 基于 Linux 的安全操作系统开发过程与质量控制的研究*)

Study of Process and Quality Controlling Based on Researching Secure Linux OS

龚丽敏　刘　勇　沈　熙　谢俊元

(南京大学计算机软件工程中心　江苏南大苏富特软件股份有限公司)

(南京大学计算机科学与技术系　南京210093)

**Abstract** To find a new effective way to improve the quality and Productivity of software is always important in the area of software engineering. Presently, ISO9000 has been extensively applied in software area. CMM has also been known in China. In this paper, we introduce the above two standard, and then narrate how to use them flexibly to control the process and quality during the production of secure operating system based on Linux.

**Keywords** Process, Quality control, CMM, ISO9000, Linux

## 1. 引言

随着信息技术在政治、经济、文化等社会各领域的日益普及,特别是 Internet 技术的发展和应用,使信息共享和信息安全这对矛盾发展到了一个新的高度,信息安全问题日益突出。至今我国只能采用进口的 C 级信息产品。我国信息系统中使用的操作系统绝大多数是国外厂商生产的操作系统,而且很少具有安全防护功能。这就意味着我国绝大多数信息系统处于不设防状态,具有严重的信息安全隐患,极易受到外部、内部的安全攻击,面临着巨大的信息安全风险。这不仅会影响我们运用先进信息技术的效率和信心,进而影响我国的整体国际竞争力,也会影响到我国信息技术和产业的发展。因此,针对目前主流操作系统进行安全增强的研究和开发,提高信息系统的安全防护能力迫在眉睫,意义非常重大。这将有效缓解我国网络信息系统中的信息安全问题,促进信息化建设的健康发展。目前我国使用的主流操作系统包括 Windows、Unix 和 Linux 等几种。Linux 是源代码公开的共享软件。通过网络共享,吸引着大量的软件爱好者为之不断改进,这无疑有利于发现安全漏洞并及时修补。由于源代码是公开的,对其进行安全增强非常方便。系统结构安全缺陷、实现漏洞以及方便调试留置的后门等都容易发现并进行修改。因此,相对而言选择 Linux 进行系统增强可以获得较好的系统安全效果。

但是就目前国内的软件开发环境来看,基于 Linux 的安全操作系统的开发的经验几乎没有。本系统的开发会涉及大量核心源代码的修改以及各种上层应用接口的实现,其复杂度非同一般,因此如何来规划和组织开发过程,控制开发质量以保证成功完成预期目标,使得开发出来的操作系统在安全上能有质的飞跃,是在开发的整个过程中必须重视和探讨的问题。本文主要介绍了结合 CMM(软件能力成熟度模型)以及 ISO9000等标准来控制安全操作系统开发的过程。

## 2. CMM 和 ISO9000介绍

要控制软件质量,缩短开发周期和开发费用,提高生产效率,就必须参照一定的国际标准来制定开发中的方方面面,以提高产品在市场中的竞争力。ISO9000是当前最广泛应用的国际标准评估体系,它由五个部分组成:(1)质量术语标准;(2)质量保证标准;(3)质量管理标准;(4)质量管理和质量保证标准的选用和实施指南;(5)支持性技术标准。如图1所示。

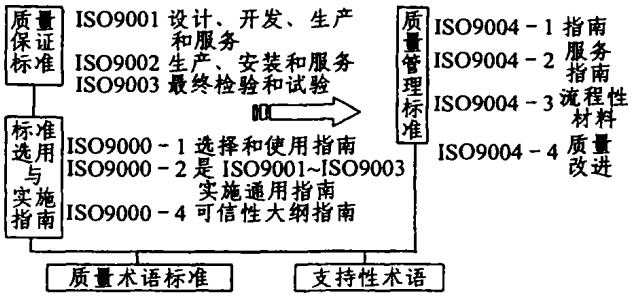


图1　ISO9000标准系列框架

CMM 是近年来脱颖而出的由美国 Carnegie Mellon 大学软件工程研究所(SEI)专门针对软件产品特点而定做的能力成熟评估模型。CMM 提供的一套过程控制和过程管理行之有效的方法已受到越来越多的软件开发者、经营者、软件消费者以及软件质量评估者的高度关注和欢迎。CMM 吸收了先进的管理思想,更强调过程控制,它代表了软件产品质量管理的发展方向。在软件评估、控制、诊断方面,CMM 模型是 ISO9000质量管理内容的必要补充。CMM 共分五个成熟级别,分别是初始级、可重复级、定义级、管理级,每个级别又包括2-7个关键过程域(KPA),52个目标(goals)和316个关键实践(KP)。各级结构图如图2所示。

CMM 与 ISO9000系列都是在国际上很有影响的质量评估体系,它们在降低软件开发风险、诊断与评价软件产品质量等诸多方面都做出了突出的贡献。然而,两者在研究范畴、评估的侧重面、论证的级别、质量管理应用的程度以及应用领域的范围等方面存在着差异:ISO9000-3侧重评价软件产品是否已达到了标准的各项指标,CMM 则基于软件的特点,基于软件改进必然性和长期性,强调软件开发的过程控制和预见性;ISO9000的标准涉及到从原料供应到产品销售的每一个环

*)本课题得到国家"八六三"高科技研究发展计划"基于 Linux 的安全操作系统"(863-306-ZD12－14-3)资助。龚丽敏　硕士研究生,主要从事安全操作系统、软件工程方面的研究。刘　勇,沈　熙　硕士研究生,主要从事安全操作系统方面的研究。谢俊元　教授,主要从事人工智能、先进操作系统方面的研究。

节，CMM 侧重软件开发和改进过程；ISO9000标准论证的只有两种结果，即通过和不通过，而 CMM 将软件成熟能力可以评价分为五个级别，通过论证，企业符合哪一等级的要求就将被评定为哪一级的企业；ISO9000仅论述了用户可接受的产品质量的最小集合，CMM 强调过程控制和过程管理，它是一把衡量软件开发过程的尺子，它更符合软件产品的开发特点。

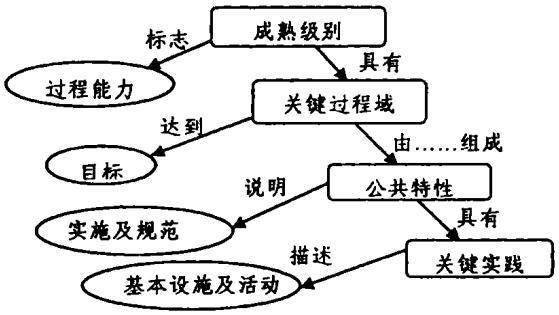


图2 CMM 的内部结构

以上论述表明，CMM 和 ISO9000各有所长，我们要基于 Linux 进行安全操作系统的开发，在国内还是一个相对陌生的领域，所以要特别注意产品开发的持续性和高效率性以及产品最终的质量问题，这就必然要借助广泛应用的各种产品开发和质量控制的国际标准，同时根据产品的特性做适当的裁减，使之发挥最大的作用。在持续性方面我们可以充分利用 CMM 的特点，使得复杂无序的操作系统的开发的每一个阶段都逐渐地成熟和完善，尽可能地减少无用功和不必要的开销，同时保证过程能持续进行。而 ISO9000是目前被广泛应用的质量评价标准，涵盖范围广，它可以补充实现 CMM 模型中不够细致的部分，提供整个生产、安装和服务的质量保证模式。二者紧密结合，从而能尽可能地保证开发顺利进行，达到预期目标。

## 3. 安全操作系统开发中的过程与质量控制

### 3.1 组织管理规范化

组织管理规范化是 CMM 的核心。基于 Linux 的安全操作系统的开发规模大、层次多、内部分工细而且涉及到庞大数量的核心源代码，所以需要强调分工后的高度统一和管理的高度规范化。

在 CMM 模型的第3级中专门就组织过程焦点、组织过程定义等关键过程域进行了描述。从而明确了组织工作的目标、责任、工作序列以及对组织工作的评价标准等。CMM 要求确定一个这样的组织结构，该结构要符合组织的战略目标，要和经营环境达成一致，并且该组织要保证关键过程域是相互独立的活动实体。CMM 中特别强调了要保证软件质量保证组(SQA)和系统测试的独立性。它要求 SQA 是独立的，在 SQA 中成员的工作不会受开发进度和已消耗成本高低的影响。它要求为 SQA 中的成员提供一个较大的工作自由度，使他们成为高级管理人员的耳目，从而保证高级管理人员得到客观的、真实的信息。CMM 要求系统测试的工作要独立于软件开发者的测试工作，他们不应受到开发者、维护者的决策影响。

因此，根据本安全操作系统研制开发的特殊性，首先在开发小组通过一段时间的阅读分析 Linux 源代码来进行可行性分析之后，我们确定由一位在质量控制方面训练有素的人来担当 SQA 的工作，该工作将贯穿整个开发过程：根据预定目标来研究、确定相应的开发安全操作系统的国际国内参考标准；监督制定需求分析；维护整个开发过程不偏离开发计划，保证开发能持续进行；维护和审核整个过程中产生的各类文档；制定统一的代码设计规则和统一完善的文档格式来记录安全操作系统设计中对核心源代码和数据结构所作的任何修改；要求代码设计尽可能地模块化，从而能容易实现系统的升级；记录设计中出现的错误，从而避免在以后的升级开发或其他类似开发中再犯；制定测试计划日程、方式等。

总之，组织管理的规范化和 SQA 的一切工作，目的就是维护整个开发过程中过程的质量和最终形成的产品的质量。

### 3.2 重视团队精神

CMM 将管理方法和技术手段相结合，强调团队精神，强调分工后的协作和过程的相互制约。进行基于 Linux 的安全操作系统的开发，设计模块将涉及到的有登录、文件系统、进程、共享内存、网络管理以及审计等，各模块之间相互协作才能实现一个操作系统完善的功能，所以团队精神显得非常重要。

CMM 的模型是透明的，它将复杂的系统分解成相互独立的模块，每个成员都知道自己的工作范围、工作标准以及工作的发展目标，每个成员在规定的范围内发挥自己的创造力并为整个安全操作系统的开发做出自己的贡献。在前一节所描述的规范管理的前提下，本操作系统的开发不会零乱无章，而是依靠团队精神以评价模型为依据，依照既定的目标前进。

在需求管理的活动标准中规定需求管理由安全操作系统开发组负责。开发组由负责人和技术人员组成，他们需要在硬件软件环境以及输入输出方面确定需求的内容、建立文档，并对需求定义的质量负责。在系统正式实施过程之前，该需求定义还要经过整个安全操作系统开发组的评审和修改，他们共同发挥作用，保证了需求定义具有较高的质量。在系统开发的过程中，各模块之间需要设计相应的接口程序来相互协作，如审计模块就需要设计接口程序来记录文件系统和进程中涉及到的系统调用的使用情况，而不管是审计或是文件系统中安全功能的实现都必须使用到内存管理的问题，这也需要内存模块的负责人提供相应的分配回收程序接口。在这整个过程中，团队之间相互交流和协作的重要性也不言而喻了。测试工作也依靠团队的力量。在项目开发初期，开发小组成员就要参加计划测试过程。小组的测试工作不是一次性的短期行为，而应每隔一段时间周期性地持久地进行模块测试和整体测试，如有条件更要送与专家评审，本安全操作系统就由国家863项目组以及上海 Intel 公司多次评审，这有利于促进本安全操作系统功能上更加完善，性能上更加稳定。

重视团队精神，将会在一定程度上缩短开发周期，减少不必要的重复劳动和开销。

### 3.3 实际开发中产生的代码以及文档的控制

以上两点都是从宏观上来控制开发过程和质量，基本上采用 CMM 的相关思想。在关于实际开发中产生的代码和文档的规范管理上，CMM 也阐述了相应的思想。如 SPF(Software Process Framework)就是关于软件开发过程定义和管理得很好的参考文件。但是，CMM 关键实践本身并未形成完整的过程和过程文档，它仅仅提供了一个符合 CMM 的过程文档框架，设计软件过程文档不是一件轻松的事情，即使对一个经验丰富的开发人员或管理人员，如何将软件过程清晰、准确、条理清楚地表达出来也是一件很费脑筋的事情，即便是参考 SPF 也是如此。

鉴于如此，我们参考 ISO9000的相关标准，结合 Linux 核

心代码开发的特点，来控制代码和文档的制定。对代码规范化的控制主要包括：(1)由于 Linux 源代码的复杂性，在定义变量的时候尽可能地用局部变量，同时在对全局变量和全局数据结构命名时，应加上适当的前缀，一旦命名与原有核心程序中的变量命名有重复，后果不堪设想；(2)使用子程序，即可以把循环和判断从子程序中拖出来，使其成为一个独立的子程序，以降低原有子程序的复杂性，尽量将新增程序或代码与原有的核心代码有明显的分离，若不得不嵌入其中，则必须有清晰统一的解释说明；(3)隐含顺序，例如在核心代码开发中经常涉及到对某一结构的操作集，为了降低阅读和审核的复杂性，应该把相关的代码以常规思维顺序集中在同一子程序中，相关的结构定义集中在同一头文件中等；(4)在首次开发安全操作系统时，可能有一些功能代码并未起作用，但是它可以作为一个功能程序族而存在，我们可以开发一些相关的接口，以便日后的移植或升级；(5)由于 Linux 已根据 POSIX 标准制定了完善的错误处理系统，我们可以利用它原有的错误处理机制来控制系统的运行。

我们参照 ISO9000-3中制定的相关软件产品开发文件编制指南，在安全操作系统开发的不同阶段制定相应的技术和帮助文档，文档要求格式统一、描述详细易懂、参考性强。对产生的文档要进行统一的管理和实时的更新，使之与当前开发进度和实际开发中采取的方案保持一致性，为以后的审核、测试提供必要的参考。

结合有效的标准，对代码以及文档进行规范化的管理，使得整个开发过程有精确的文字记录，可以有效地保证开发不偏离预期轨道，也有利于从中总结经验，为以后做其他类似的软件开发过程提供参考价值，提高企业软件开发过程的整体水平。

**结束语** 本文介绍了目前国际上软件产品开发中被广泛采用的过程和质量控制评估标准 CMM 和 ISO9000系列，并以此为基础，结合实际开发经验，介绍了基于 Linux 的安全操作系统开发中如何控制过程、提高开发质量的一些方法。采用国际开发标准、重视开发过程和产品质量是我国软件业走向国际市场必须重视的一个环节，而我们开发本安全操作系统的实际经验表明，结合要进行的开发过程的特点，采用合适的标准并进行相应的裁减，使之应用到开发过程中的方方面面，会使得软件产品的开发事半功倍。希望藉此能引发国内软件业关于过程和质量控制的研究。

## 参 考 文 献


1 何新贵，王纬，王方德，等编著. 软件能力成熟度模型. 北京：清华大学出版社，2000. 11

2 杨一平，等. 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. 4

3 America Department of Defense. TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA. CSC-STD-001-83，15 Aug 83

3 郑人杰，等. 实用软件工程. 北京：清华大学出版社，1997

4 MIL-STD-498. Software Development and Documentation

5 Paulk M C，Curtis B，Chrissis M B，Weber C. Capability Maturity Model，Version 1. 1. IEEE software，July 1993. 18～27

6 CMU-SEI-93-RT-24，Capability Model for Software，Version 1. 1. Feb. 1993

7 林汉川主编. ISO9000在质量管理中的应用. 广州：广东人民出版社，1999. 1

8 林汉川主编. ISO9000在服务、流程性材料、软件中的应用. 广州：广东人民出版社，1999. 1


（上接第17页）

上 $S$-不变量的条件，就可除去那些不属于原网的可达标识的标识(这一点将另文讨论)；至于语言(即变迁可发生序列)的综合，正如定理3. 3所指出，通过语言的同步合成运算就可以实现。

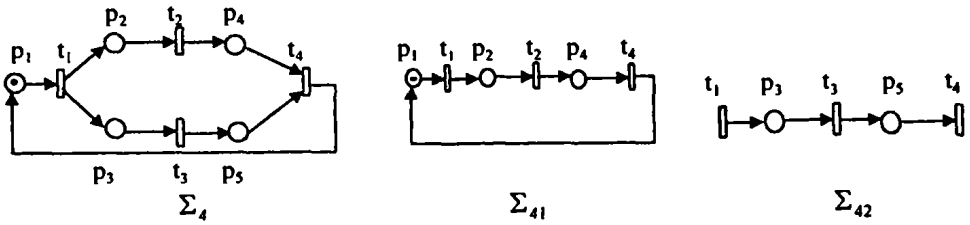


图5 一个 Petri 网 $\Sigma_4$ 和它的分解子网 $\Sigma_{41}$ 和 $\Sigma_{42}$

**结束语** 文[8]分析了系统合成(同步合成和共享合成)过程中合成网同子网之间满足动态不变性。文[6]和[7]提出的网的“和分解”和“并分解”，均是从保持系统的结构性质方面考虑系统的分解。本文提出的基于库所指标的 Petri 网分解技术，这种方法使分解得到的子系统保持原系统的行为和状态。同以往的工作相比，本文可以说是文[7]的逆过程；同文[5]和[6]相比，本文则是着重从保持系统的动态不变性(行为不变性和状态不变性)方面考虑分解网系统。

## 参 考 文 献


1 Hyunglee K，et al. Generalized Petri Net Reduction Method. IEEE Transaction on Systems，Man and Cybernetics，1987，SMC-17(2)

2 Suzuki I. A Method for Stepwise Refinement and Abstraction of Petri Nets. J. of Computer and System Sciences，1983，27：51～76

3 Murata T. Petri Nets：Properties，Analysis and Applications. Proceedings of The IEEE，1989，77(4)

4 Jiang Changjun，Wu Zhehui. Net Operations. Computer Science and Technology，1992，7(4)：333～344

5 王培良，等. P/T 系统的和分解. 计算机科学，1999年(增刊)

6 王培良，等. Petri 网的并分解. 控制理论与应用，2001，18(1)：116～118

7 蒋昌俊. Petri 网的动态不变性. 中国科学(A 辑)，1997，27(6)：567～573

8 Garg V K，Ragunath M T. Concurrent regular expressions and their relationship to Petri nets. Theoretical Computer Science，1992，96：258～304