黄红元　主编

上海证券交易所联合研究报告2013

证券信息前沿技术专集

上海人民出版社

黄红元　主编

上海证券交易所联合研究报告 2013

证券信息前沿技术专集

上海人民出版社

图书在版编目(CIP)数据

上海证券交易所联合研究报告.2013.证券信息前沿技术专集/黄红元主编.—上海:上海人民出版社,2014

ISBN 978-7-208-12658-9

Ⅰ.①上… Ⅱ.①黄… Ⅲ.①资本市场-研究报告-世界-2013 ②证券交易所-经济信息-研究报告-世界-2013 Ⅳ.①F831.51

中国版本图书馆 CIP 数据核字(2014)第 263880 号

责任编辑 吴书勇
封面装帧 王小阳

上海证券交易所联合研究报告 2013
——证券信息前沿技术专集
黄红元 主编
世纪出版集团
上海人民出版社出版
(200001 上海福建中路 193 号 www.ewen.co)
世纪出版集团发行中心发行 上海商务联西印刷有限公司印刷
开本 720×1050 1/16 印张 17 插页 4 字数 301,000
2014 年 12 月第 1 版 2014 年 12 月第 1 次印刷
ISBN 978-7-208-12658-9/F·2269
定价 45.00 元

序　言

信息技术的创新变革是推动市场发展的重要力量。始终保持对新兴技术的敏感性和关注度，加强先进技术在行业内的研究应用，对于促进市场创新发展，提高市场运行效率，提升行业整体竞争能力具有十分重要的意义。

上证联合研究计划是上海证券交易所通过承接国家科研课题并利用国内外学术资源联合开展的一系列科研活动，其中证券信息前沿技术专题已经开展了两期，积累了丰富经验，成效十分显著，相关研究成果不论是对交易所系统建设还是行业信息化工作都具有明显的促进作用。一是促进了交易所自身系统的完善和发展。上海证券交易所一贯重视对新兴信息技术的研究，并且注重研究成果与系统实际开发相结合，本次研究计划的开源集群与 RDMA 技术、下一代轻便高效交易系统等课题的研究成果对于交易系统未来建设工作有很好的启示、借鉴作用。二是促进了行业信息技术发展。本次研究计划的成果还包括服务行业技术需求的课题，比如行业云技术研究、快速订单柜台系统研究等，相关成果的推广应用对于服务市场发展，提升行业信息技术整体实力非常有益。

交易所是资本市场的核心参与者，不仅在市场运转过程中占有重要地位，更拥有人才、技术、资源等众多优势，因此，交易所要进一步发挥技术引领作用，协助科研成果在行业的推广应用，有效的促进行业技术创新、服务行业技术发展。同时，科研成果的推广应用要特别注重以下三个方面：

一是要符合行业总体规划的考量。证监会近日发布了《资本市场信息化建设总体规划(2014—2020)》，提出了行业应对互联网金融、新兴信息技术业态冲击背景下的信息化建设总体目标和具体任务。其中明确指出要“推广应用云计算、大数据、移动互联网等技术手段，探索多种基于云集中平台的应用实现”，这也是行业信息化发展的实际需求。因此，要根据行业信息化工作的总体规划，加大科研成果在行业公共基础设施建设工作中的投入应用，通过互联技术、云服务等方式，向行业提供规模化、集约化、专业化的信息技术服务，研究建设行业高可用云计算系统、高速行情网和行情云、测试云等。

二是要注重标准的研究制订工作。标准化工作对于科研成果的推广来说意义

重大。科研成果的载体不仅是某个技术系统，更应该通过行业标准明确行业同类系统的统一规范。比如交易系统，其延迟、边界在哪里，测试基准是什么，只有明确度量标准，才能够在国内或国际范围内做出比较。再如柜台交易系统，只有明确监察、资金等模块的具体接口标准，才能形成对相关产品的约束规范。只有建立在公开标准上的系统，才能真正做到松耦合、轻型化，并具有较高的灵活性和可扩展性，因此要加强研究成果通过标准载体落地的工作。

三是要守住信息安全的基本底线。信息系统的安全稳定运行是资本市场健康发展的生命线。确保信息系统安全是促进行业信息化发展的重要基础，也是我们一贯强调的要求。要确保信息系统安全，就要不断提升信息系统研发技术和水平，增强自身技术竞争实力，避免核心系统关键技术受制于人。因此，要保持对前沿技术积极研究探索的态势，大胆尝试，勇于进取，不断增强信息系统自主可控能力。

望上海证券交易所再接再厉，祝愿前沿技术专题的联合研究工作越办越好！

张　野

中国证监会信息中心主任

2014年11月

目录

证券业关键业务系统代码质量保证检查方法与工具研究*

上海证券交易所—北京邮电大学联合课题组

第一节
研究背景与研究基础

证券业作为我国金融行业重要的组成部分，在信息化建设和信息安全保障方面对软件质量有更高的要求。证券业的关键业务系统既有与普通软件系统的共性，同时还包含一些证券行业特有的代码安全属性。本研究将从证券业的关键业务系统代码开发规范和检测方法两个方面开展研究。

一、 证券业软件质量保证现状

中国证监会根据证券业的网络安全现状和证券业务的安全要求，相继出台了《证券公司内部控制指引》、《证券期货业信息系统安全检查贯彻落实指引》、《证券期货业信息系统安全等级保护基本要求》、《证券经营机构营业部信息系统管理规范》、《网上证券委托暂行管理办法》等有关法规，这些法规从侧面体现了证券行业对 IT 系统治理方面的紧迫要求。2012 年 9 月 24 日，中国证券监督管理委员会令第 82 号公布《证券期货业信息安全保障管理办法》，自 2012 年 11 月 1 日起施行。该《办法》指出，证券期货业信息安全保障的责任主体应当执行国家信息安全相关法律、行政法规和行业相关技术管理规定、技术规则、技术指引和技术标准，开展信息安全工作，保护投资者交易安全和数据安全，并对本机构信息系统安全运行承担责任。要

* 课题主持：金大海；课题协调人：上海证券交易所王程程、赵文杰；课题研究员：张大林、董玉坤。

求核心机构应当对交易、行情、开户、结算、风控、通信等重要信息系统具有自主开发能力，拥有执行程序和源代码并安全可靠存放，在重要信息系统上线前对执行程序和源代码进行严格的审查和测试。核心机构和经营机构应当具有防范木马、病毒等恶意代码的能力，防止恶意代码对信息系统造成破坏，防止信息泄露或者被篡改。

众所周知，提高代码质量，除要提高逻辑合理性与加深对业务流程的理解外，代码本身也存在提高的空间，例如一些潜在的问题可以及早避免。类似于编码规范上的内容，然而如果全靠编码人员自行检查，无疑需要巨大的时间与人力成本，如果可以使用代码检查工具自动检查，将大大提高编码效率。另外，不同编程语言的语法拥有各自的灵活性，这种灵活性虽然可以提升代码编写效率，却相应增加了代码中存在隐患的可能性。在对代码质量进行保证的过程中，人工代码走查是其中不可或缺的一方面，若使用工具自动化完成部分标准化的代码走查，发现错误隐患，将是一个不错的补充手段。

二、 软件质量模型

软件系统的代码质量直接影响软件质量。

软件质量是软件产品具有满足规定的或隐含要求能力要求有关的特征与特征总和。软件产品质量需求一般要包括对于内部质量、外部质量和使用质量的评估准则，以满足开发者、维护者、需方以及最终用户的需要。

GB/T 16260.1—2006 定义了外部和内部质量的质量模型。它将软件质量属性划分为六个特性(功能性、可靠性、易用性、效率、维护性和可移植性)，并进一步细分为若干子特性，这些子特性可用内部或者外部度量进行测度，外部和内部质量的质量模型如图 1 所示。

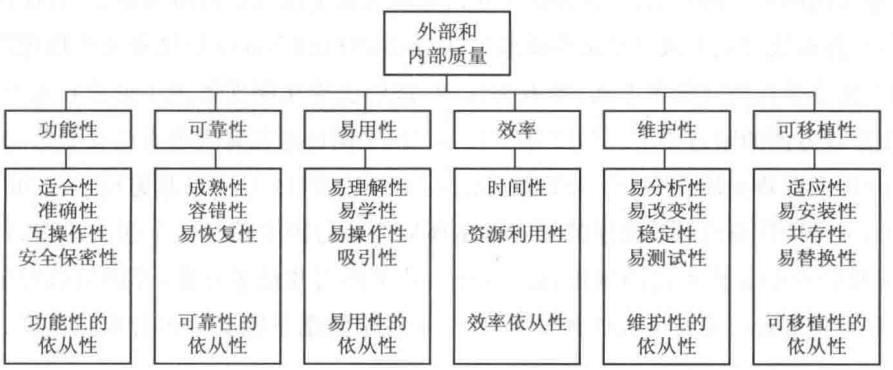

图 1　软件质量模型

GB/T 16260 的第 2 部分、第 3 部分和第 4 部分提出了与第 1 部分“质量模型”一起使用的一组软件质量度量(外部质量、内部质量和使用质量的度量)的建议。上述这些标准给出了测量、评估和评价软件产品质量的方法,旨在供开发者、需方和独立的评价者使用度量类型之间的关系如图 2 所示。

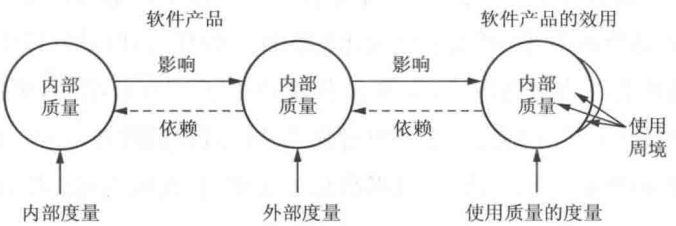

图 2 软件度量类型之间的关系

内部度量可用于开发阶段的非执行软件产品(例如标书、需求定义、设计规格说明或源代码等)。内部度量为用户提供了测量中间可交付项的质量的能力,从而可以预测最终产品的质量。这样就可以使用户尽可能在开发生存周期的早期察觉质量问题,并采取纠正措施。

外部度量可以通过测量该软件产品作为其一部分的系统行为来测量软件产品的质量。外部度量只能在生存周期过程中的测试阶段和任何运行阶段使用。在所属系统环境下运行该软件产品即可获得这样的度量。

使用质量的度量是测量产品在特定的使用环境下,满足特定用户达到特定目标所要求的有效性、生产率、安全性和满意度的程度。这只能在真实的系统环境下获得。

通过对上述软件质量模型分析可知,代码质量是影响软件质量的重要因素,尽早发现并纠正编码中存在隐患可以提高质量。

三、 质量保障方法

20 世纪 50 年代以来,各方陆续提出了许多软件质量保障方法,包括软件工程方法、软件质量国际标准体系、软件可靠性工程等几个方面。下面我们分别介绍这三种软件质量保障方法的主要思想及其应用背景。

(一) 软件工程方法

1. 工程化的设计与开发技术

软件工程的基本思想是摆脱传统的手工作坊式的编程模式,代之以工程化的

方法开发软件，最终实现软件工业化生产的目的。先进的设计与开发技术是软件质量保障的重要途径。20 世纪 70 年代人们就已提出了许多程序设计方法及结构化分析和结构化设计技术，以规范软件的分析和设计过程；同时，为了帮助人们更好地使用这些方法和技术，又将它们转换为相应的工具。由于软件工具间彼此独立，缺乏有效的通信机制，不能充分共享软件开发过程中产生的信息，于是，将相关软件工具有效地集成起来，形成了集成化的软件工程环境，以对软件生命周期内的管理、组织和开发活动提供连续、完整和有效的支持。20 世纪 90 年代，人们提出了软件过程和软件复用的概念，希望通过强调软件过程消除各部分工作间的冲突，实现软件过程的协调性和一致性，以提高总体效率，有效地实现总体目标。

2. 工程化的软件测试技术

软件测试是软件质量保证工作的一个重要环节。软件测试出现在软件生命周期每一阶段中，无论是静态测试还是动态测试，要求检验每一个阶段的成果是否符合质量要求和达到定义的目标，尽早发现错误并加以修正。如果不在早期阶段进行测试，错误不断扩散、积累常常会导致最后成品测试的巨大困难、开发周期的延长、开发成本的剧增等等。对比国外可以看到，国外软件开发机构会把 40%的工作花在测试上，测试费用则会占到软件开发总费用的 30%到 50%，对于一些要求高可靠性、高安全性的软件，测试费用则相当于整个软件项目开发费用的 3 至 5 倍。因此，软件测试在软件生存期中占有非常突出的位置，是保证软件质量的重要手段。

（二）软件质量国际标准体系

工程化是提高软件质量的途径之一，而工程化必须借助管理来实现。作为管理的一项重要内容，加强质量管理、建立质量体系必须走国际化的道路，这样才能与国际接轨，并在激烈的国际竞争中占有一席之地。国际上的软件质量标准体系可分为两类：一类强调产品质量，如 ISO 9000 系列；另一类则注重过程质量，如 CMM。ISO 9000 系列国际质量标准认为：质量形成于生产全过程；必须使影响产品质量的全部因素在生产全过程中始终处于受控状态；应使企业具有持续提供符合要求的产品的能力；质量管理必须坚持质量改进；质量管理的核心是预防而不是补救。CMM 将成熟度分为初始、可重复、已定义、已管理和优化等 5 个等级。

（三）软件可靠性工程

软件可靠性是指在特定的环境下，在给定的输入条件或指定的时间段内软件正确运行的概率，可用作软件开发或测试阶段评估软件质量的基本量度。软件可

靠性工程用于对软件系统的质量进行评价、预测和管理。内容包括可靠性预测、可靠性评估、观察、度量、选择模型、建模和决策等。由于提供给用户的所有功能都要经过测试，活动只关心软件生命周期的测试阶段。

四、 软件测试技术

上述3种软质量保证方法分别从不同维度提供软件质量保证策略。软件工程方法侧重于开发角度，软件质量标准体系侧重于软件开发管理视角，而软件可靠性工程则侧重于软件质量建模和预测。

随着人们对于软件质量的重视程度越来越高，软件测试在软件开发中的地位也越来越重要。软件测试是程序的一种执行过程，目的是尽可能发现并改正被测试软件中的错误，提高软件的可靠性。它是软件生命周期中一项很重要且非常复杂的工作，对软件可靠性保证具有极其重要的意义。在目前形式化方法和程序正确性证明技术还无望成为实用性方法的情况下，软件测试在将来相当一段时间内仍然是软件可靠性保证的有效方法。

（一）软件测试策略

按软件测试方法可划分为：

（1）黑盒测试：指基于需求和功能性的测试，已知产品所应具有的功能，通过测试来检测每个功能是否都能正常使用。在测试时，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，测试者在程序接口进行测试。它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息，并且保持外部信息（如数据库或文件）的完整性。

（2）白盒测试：指基于一个应用代码的内部逻辑知识，即基于覆盖全部代码、分支、路径、条件的测试。它是知道产品的内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常执行；按照程序内部的结构测试程序，检验程序中的每条通路是否都有按预定要求正确工作。

按程序是否执行可划分为：

（1）静态测试：在不执行程序的情况下对代码进行分析，通过计算程序抽象语义的方法来预判未来所有可能的运行情况。静态测试的基本特征是在对软件进行分析、检查和测试时，被测试的程序没有被实际运行。它可以对各种文档进行测

试，是软件开发中十分有效的质量控制方法之一。

（2）动态测试：指通过运行软件来检验软件的动态行为和运行结果的正确性。动态测试的特性使得它只存在于软件生存期的编码阶段之后。动态测试包括两个基本要素：一是被测程序；二是用于运行软件的数据，称为测试数据。

（二）面向缺陷的静态测试技术

无论是动态测试技术还是静态测试技术都很难完全检测软件安全漏洞和系统编码质量。现有测试方法，可以检测出大部分的软件缺陷，但软件中有些缺陷是难以测试的，例如发生概率极小的事件所产生的缺陷或者难以再现的内存问题等等，这些问题必须通过专门的测试技术解决。

面向缺陷的软件静态测试技术是 2000 年由美国提出并得到应用。与传统的软件静态测试方法不同，该技术首先定义软件的缺陷模式，然后通过静态分析的方法对程序进行抽象计算，利用计算结果从程序中推导出与所定义模式相匹配的程序元素，称为 IP（Inspective Point，检查点），最后通过人工或自动方式确认 IP 是否为真正的缺陷。

缺陷模式必须满足以下几个条件：

（1）该模式下的缺陷是符合实际的。也就是说，该模式下所定义的缺陷在工程中是实际存在的。

（2）基于该模式的缺陷数目是可以容忍的。通常来讲，缺陷个数与系统规模呈线性关系。

（3）该模式下的缺陷是可以测试的。应存在一个算法可以检查出这个缺陷。

与其他测试技术相比，基于缺陷模式的软件静态测试技术应具有以下特点：

（1）针对性强。如果某种模式的缺陷是经常发生的，并且在被测代码中是存在的，则面向缺陷模式的测试可以检测出此类缺陷。

（2）能发现其他测试技术难以发现的故障，这个故障往往是小概率、不明显的。

（3）工具自动化程度高，测试执行效率高。

（4）缺陷定位准确：对测试所发现的缺陷可以准确定位。

（5）具有较强的扩展性：针对新的缺陷，能够迅速扩展检测工具。

（6）易学、易使用。

测试工具都是存在漏报的，基于缺陷的测试工具也一样。漏报（False Negative），是指程序中实际存在某类问题，但分析工具没有发现该问题。程序中实

际存在多少问题常常是未知的。这里所谓的漏报，通常是指由手工置入若干个特定错误或者在和同类工具进行比较时发现的漏报。

设 P 是待测程序，将缺陷模式 M 分成类 $M=\{M_1, M_2, \cdots, M_n\}$，每类分成 $Mi=\{Mi_1, Mi_2, \cdots, Mi_L\}$，从 P 中计算出和 M 相匹配的检查点的集合 $IP=\{IP_1, IP_2, \cdots, IP_m\}$，可以定义如下技术指标：

(1) 漏报率(ER)：设 P 是程序，M 是缺陷模式，A 是算法，$IP(M, A, P)$是 IP 总的数目，考虑到测试算法实现过程中的不同假设，会导致 $IP(M, A, P)$不同。漏报率定义为：

$$\text{ErrorRatio}=\frac{|\text{IP}(M, A, P)-\text{IP}(M, P)|}{|\text{IP}(M, P)|}$$

在理论上，在 M 和 P 给给定之后，$IP(M, P)$是确定的，但是在实践中很难得到 $IP(M, P)$。假设不同的测试工具算法为 $A_1, A_2, \cdots, A_n$，则通常用 $\bigcup_{i=1}^{n}\text{IP}(M, A_i, P)$ 表示 $IP(M, P)$。

对于每个 IP 通常需要人工去判定该 IP 是否真的缺陷，考虑程序的逻辑复杂性以及测试代价等因素，IP 经确认后分为 3 种情况，用 $IP_Y(P, A, M)$、$IP_N(P, A, M)$、$IP_U(P, A, M)$分别表示 IP 确认为缺陷的数目，确认为非缺陷的数目，以及不能确定是否缺陷的数目。显然，$IP_Y(M, A, P)+IP_N(M, A, P)+IP_U(M, A, P)=IP(M, A, P)$。

(2) 准确率(CR)：

$$\text{CorrectRatio}=\frac{\text{IP}_Y(M, A, P)+\text{IP}_U(M, A, P)}{\text{IP}(M, A, P)}$$

(3) 误报率(DR)：

$$\text{DistortRatio}=\frac{\text{IP}_N(M, A, P)}{\text{IP}(M, A, P)}$$

(4) 缺陷检测率(DDR)：

$$\text{DefectDetectingRatio}=\frac{\text{IP}_Y(M, A, P)}{\text{IP}(M, A, P)}$$

(5) 自动缺陷检测率(ADR)：

用 $IP_{AY}(P, A, M)$表示不需人工确认，工具可以自动缺陷的检测个数。

(6) 计算复杂性：理论上，基于缺陷的软件测试技术可以 100%的检测所定义的缺陷模式。但由于缺陷的检测可以模型化程序的遍历问题，对于大型程序，全部遍历虽然可以提高精度，但需要花费大量的时间，因此，该技术存在一个性价比的问题。

五、 研究目标及拟采用的研究路线

本研究基于证券业软件质量保证现状，同时参考软件质量模型对现有软件保障方法进行了分析梳理，结合现有研究基础和证券业软件质量保证的实际需求，采用基于缺陷模式的软件静态测试技术作为软件质量保证的技术支撑。然而，缺陷模式更主要来源于具体行业的软件质量保证需求，因此梳理并形成一个完善的证券业软件开发规范势在必行，然后针对开发规范，我们设计并实现一款自动化测试工具来支撑测试规范。

本研究所采用的研究方法，包括文献调研法、访谈法和系统开发方法。

（一）证券业的关键业务系统代码开发规范

在本研究中，一个程序缺陷是指在程序代码中，对某一程序规范或安全属性的违反，即在执行质量保证检查时候，违反我们制定的证券业软件安全开发规范就会报告一个缺陷。由此可知，我们的证券业软件安全开发规范总结的越科学全面，我们的代码质量保证检查越可靠。

本研究的证券业软件安全开发规范涉及软件质量保证的各个方面，主要包括：(1)可用性、稳定性，即程序能够平稳有效运行，不会出现异常崩溃。(2)可靠性、安全性，即程序可靠、安全。(3)规范、可读性，即程序书写风格、命名规则等要符合规范。(4)效率与性能，指软件系统的整体效率或者某个模块(或者子模块、函数)的本身效率。

（二）关键业务系统代码质量保证检查工具

为了保证证券业的关键业务系统代码质量，依据证券业软件安全开发规范而进行的代码审查工作必不可少，人工审查费时费力，同时正确性还没有严格保证，因此就进一步催生了自动化检查工具的研发需求。

由上文证券业软件安全开发规范可知，违反证券业软件安全开发规范则形成一个软件缺陷，因此证券业软件安全开发规范与证券业软件缺陷模式是一一对应的。同时，我们发现证券业软件安全开发规范是可扩展的，规范的数量可以动态变化，一条规范的删除与增加不影响其他规范。基于上述特点，本研究提出了使用面向缺陷的静态测试技术来解决现有问题。

本研究中的缺陷模式指的是程序中的缺陷所呈现出的语法或语义特征。缺陷模式是对程序属性的一种描述，如果违反该属性则造成一个缺陷。例如，申请的资

源在使用完后必须释放，否则造成资源泄漏缺陷；数组下标的使用必须在其数组声明大小范围以内，否则会造成数组越界缺陷；指针在解引用之前必须确保其指向非空，否则会造成空指针引用缺陷。

面向缺陷的静态测试技术关键是对缺陷模式进行定义和检测，能处理的缺陷模式种类越多则分析检测能力越强。

第二节
证券业软件开发规范研究

在代码质量保证检查方法方面，本研究首先对已有的证券业代码质量保证经验和方法进行总结抽象，基于证券行业已有的软件开发标准，同时借鉴国内外相关领域的开发标准，最终归纳出证券业软件开发规范。

一、 现有软件开发规范分析

本研究分别调研了 4 个领域的现有软件开发规范。

（一）证券业相关开发规范

我们首先调研了证券业相关开发规范。根据《上海证券交易所技术开发部编码规范》、《上海证券交易所技术开发部 C 语言开发提示(Tips)》、《上海证券交易所高可用包编码规范》、《上海证券交易所软件开发 QA 规则》，本研究共整理出 66 条带有证券业特点的代码规则。

我们通过对这 66 条证券业特点的代码规则整理和分析后发现，现有的证券业软件开发规范 C 语言部分共有 8 部分，分别从以下这些方面对 C 语言编程作出规定：

(1) 可读性和可维护性：封装和信息隐藏、空格、注释、命名规范、标准命名、变量命名、大小写、Type 和常量 Constant、平台命名规则，共 9 个分类。

(2) 程序组织：程序文件、标准库、头文件、标准的文件名后缀，共 4 个分类。

(3) 文件组织：文件开端、程序设计、Include 指令、Define 和 Typedef、外部数据声明和定义、函数的顺序，共 7 个分类。

(4) 函数组织：函数开端、函数参数、外部变量声明、内部变量声明、段落显示、

开始和返回语句、函数调用，共6个分类。

（5）数据类型、操作符和表达式：变量、常量、变量定义和声明、类型转换和强制类型转换、指针、指针转换、操作符格式、赋值操作和表达式、条件表达式、优先级和求值顺序，共10个分类。

（6）语句和控制流：顺序语句、选择控制语句、迭代控制语句，共3个分类。

（7）可移植性和性能：可移植性的建议、性能方面的建议，共2个分类。

（8）其他：类型使用、特殊宏定义、代码检查，共3个分类。

规范的Java语言部分共有以下6部分，分别从如下方面对Java语言编程作出规定：

（1）Java程序组织：类结构、类组织、代码格式，共3个分类。

（2）Java命名标准：一般标准、文件名、类名、方法名、变量命名、参数名和局部变量名、数组、静态变量名（即类变量名）、常量名称、异常名称、接口名称，共11个分类。

（3）Java程序设计风格：类声明、抽象方法、单入口/单出口、访问实例变量、可见度、方法大小和一页规则、异常、括号和嵌套方式、变量（一般用法）、实例变量、类变量、常量、局部常量、初始化、原始数据类型的包装类，共15个分类。

（4）最佳编程实例：条件语句、异常、循环、Switch语句、特定关键字和构造函数的说明，共5个分类。

（5）Java代码中的注释：注释、代码单元、方法、变量、空格，共5个分类。

（6）Javadoc使用标准：概要、首句、Javadoc标记、类描述、Javadoc中的HTML标记、方法描述，共6个分类。

从以上内容可以看出，规范分别对C语言和Java语言开发中需要遵守的原则进行了规定和建议。这些内容从开发人员编写代码的视角出发，比较全面的从语法层面进行规定，但是此规范也存在一些问题，主要表现在以下几个方面：

（1）对代码运行时故障问题和安全问题涉及较少，而这些问题恰恰是引起证券业软件系统缺陷的高风险问题。

（2）对涉及软件运行效率的相关规范涉及较少，对于证券业关键业务系统而言，软件运行效率也尤为关键，很多运行时错误往往是由系统低效造成。

（3）语法规定虽然比较全面，但是往往是一些软件开发所面临共同问题，而对高可信软件行业特有的编码规范体现不足。

（二）银行业软件安全开发规范

银行软件开发部门从代码安全的角度对软件开发做了详细规定，我们通过调

研，得到银行业相关部门对代码编写安全问题的分类如下：

输入处理。这个主题下给出了验证程序输入的方法、进行输入校验的策略、实现这些策略的方法。

Web 应用程序。这个主题覆盖了创建安全 Web 应用最重要的相关内容。

外部连接。这个主题主要针对数据库和文件操作会带来的安全隐患。

开发技术。这个主题下的内容涉及软件开发中经常用到的，与系统安全风险相关的各种技术。

安全特征。这个主题下的内容包括密码、认证、网络、口令等软件系统会涉及的各类安全特征。

语言特征。这个主题分别针对 C/C++和 Java 语言开发时各自的安全问题。

通过分析发现，该规范将代码质量问题分成三个层面：第一是语言特征方面，这是所有软件开发的共性问题；第二是具体安全问题，这个具有领域特征，具有很好的适用性；第三是特定运行环境和编程环境问题，比如数据库缺陷、Web 应用程序缺陷、输入处理缺陷等。

（三）航天嵌入式软件 C 语言开发标准

航天嵌入式软件 C 语言开发标准是国家军用标准在航天型号软件的开发中对 C 语言的安全规范（以下简称“国军标”）。

国军标主要从以下 15 个方面进行规定：

声明定义类，共 31 条规则。

版面书写类，共 12 条规则。

分支控制类，共 8 条规则。

指针使用类，共 5 条规则。

跳转控制类，共 3 条规则。

运算处理类，共 22 条规则。

过程调用类，共 13 条规则。

语句使用类，共 11 条规则。

调用返回类，共 5 条规则。

程序注释类，共 3 条规则。

循环控制类，共 5 条规则。

类型转换类，共 4 条规则。

初始化类，共 4 条规则。