

ICS 03.060

CCS A 11



中华人民共和国金融行业标准

JR/T XXXXX—XXXX

证券期货业开源软件治理能力建设与评估 指南

Guidelines on capacity building and evaluation for open source software governance
in the securities and futures industry

(征求意见稿)

XXXX—XX—XX 发布

XXXX—XX—XX 实施

中国证券监督管理委员会

发布

目 次

前言..... II

引言..... III

1 范围..... 1

2 规范性引用文件..... 1

3 术语和定义..... 1

4 治理能力模型..... 1

5 组织制度..... 2

 5.1 治理组织..... 2

 5.2 制度体系..... 3

6 治理流程..... 4

 6.1 开源软件生命周期管理..... 4

 6.2 风险管理..... 5

 6.3 二次开发..... 5

 6.4 社区贡献..... 5

 6.5 存量管理..... 6

7 工具平台..... 6

 7.1 架构模型..... 6

 7.2 通用能力..... 7

 7.3 研发流程集成能力..... 8

8 评估体系..... 8

 8.1 能力等级..... 8

 8.2 能力域..... 8

 8.3 评估指标..... 9

参考文献..... 18

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本文件由全国金融标准化技术委员会证券分技术委员会（SAC/TC 180/SC4）提出。

本标准由全国金融标准化技术委员会（SAC/TC 180）归口。

本标准主要起草单位机构：深圳证券交易所、上交所技术有限责任公司、中国期货市场监控中心、深圳证券通信有限公司、广发证券股份有限公司、南京证券股份有限公司、中国信息通信研究院、国家工业信息安全发展研究中心、华为技术有限公司。

本标准主要起草人：喻华丽、郭未、俞瑾、王佟、林林、颜飞、崔慧阳、刘政言、钟浪辉、刘翔宇、王海红、周子望、辛治运、李立峰、鹿群、潘智斌、江念南、张之浩、季常青、苏雁南、程贝、叶敏、郭雪、石硕磊、周峻松、王思檬、李峰风。

引 言

开源软件在证券期货业各领域得到广泛应用，在推动证券期货业行业机构科技创新和数字化转型方面发挥着积极作用，但同时也面临安全合规等方面的诸多挑战，对行业机构的开源软件治理能力提出更高要求。

本文件提出了证券期货业开源软件治理能力参考模型，围绕治理组织、制度体系、治理流程、工具平台及评估体系等方面，指导行业机构构建科学、有效的开源软件治理体系，旨在帮助机构提升开源软件安全应用水平和自主可控能力。

证券期货业开源软件治理能力建设与评估指南

1 范围

本文件规定了证券期货业机构开源软件治理能力建设与评估参考模型。

本文件适用于证券期货业机构开源软件统一管理工作，可作为评估机构自身开源软件治理能力建设水平的依据和准则。相关机构可参照开源软件治理能力模型要求，系统性地构建、持续维护并逐步完善开源软件治理能力体系。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

JR/T 0289—2024 金融业开源技术 术语

JR/T 0291—2024 金融业开源软件应用 评估规范

3 术语和定义

JR/T 0289—2024界定的以及下列术语和定义适用于本文件。

3.1

开源治理 open source governance

开源治理是指对开源软件、项目、社区以及相关的知识产权、合规性、安全性、质量和合规使用等方面的管理和控制。它涉及了从开源项目的选择、评估、使用、贡献、维护到退出整个生命周期的各个环节，旨在确保组织在使用和管理开源资源时能够遵循最佳实践，降低风险，并最大化开源带来的价值。

3.2

开源治理评估体系 open source governance evaluation system

开源治理评估体系是一个由多个指标构成的有机整体，用于全面评估一个组织在开源治理方面的能力和效果。这个体系旨在帮助组织了解自己在开源治理方面的现状，识别潜在的风险和改进点，从而推动开源生态的健康有序发展。

3.3

开源软件应用台账 open source technology application ledger

机构或组织为实现对内部所使用的各类开源软件的全生命周期管理，而建立的一套系统化、规范化的记录与追踪体系。

3.4

软件物料清单 software bill of materials (SBOM)

软件产品中所包含的全部开源组件及其相关元数据的结构化清单。

4 治理能力模型

开源软件治理能力模型共包括组织制度、治理流程和工具平台三个层次及其评估体系，开源软件治理能力模型如图1所示。

- a) 组织制度：明确机构开展开源软件治理时应具备的组织保障能力，包括治理组织、制度体系。
- b) 治理流程：明确机构开展开源软件治理时应关注的过程环节，包括开源软件生命周期管理、风险管理、二次开发、社区贡献、存量管理。
- c) 工具平台：明确机构开展开源软件治理时应建设的工具平台能力，包括通用能力、研发流程集成能力。
- d) 评估体系：明确组织架构、制度体系、生命周期管理、风险管理、生态参与、存量管理和工具平台等方面的相关评估能力项及评估细则。



图 1 开源软件治理能力模型

5 组织制度

5.1 治理组织

5.1.1 领导小组

领导小组宜由机构内技术治理委员会或类似组织组成，负责制定开源软件治理规划，建立考核和奖惩机制，统筹部署并推动机构开源软件治理工作发展。

5.1.2 管理团队

管理团队负责制定开源软件治理制度和流程，指导技术活动过程中的开源治理工作，包括制定治理计划、细化治理任务、监督治理进展、核查治理效果，定期向领导小组汇报，确保技术活动符合机构开源治理要求。

5.1.3 研发团队

研发团队根据信息系统建设需要开展开源软件技术调研和评估，提出开源软件引入申请并经过相关团队审批后执行。

5.1.4 运维团队

运维团队负责开源软件的安装部署、运行监控、变更维护等工作，确保使用开源软件的信息系统安全稳定运行。

5.1.5 安全团队

安全团队负责开源软件的安全审查，收集开源软件安全信息，评估安全风险及影响，确定安全修复方案，及时通知相关团队消除安全风险，并在此过程中给予指导。

5.1.6 合规团队

合规团队确保开源软件使用符合开源软件相关协议、法律规定，识别并确认开源软件的使用范围，及时通知相关团队消除开源软件使用中涉及的合规风险，并在此过程中给予指导。

5.2 制度体系

5.2.1 开源软件生命周期管理制度

5.2.1.1 引入管理制度

机构宜建立开源软件引入管理制度，由使用方提出引入申请，管理团队组织相关团队审核评估，审批通过后方可正式引入开源软件。开源软件引入后由管理团队统一记录和管理。

开源软件选型规范可参考 JR/T 0291—2024 中的引入评估内容对开源软件进行综合评估。

5.2.1.2 使用管理制度

机构宜建立开源软件使用管理制度，由引入方制定开源软件相应的研发、部署、运维等使用规范，由管理团队组织相关团队审核通过后发布。管理团队建立开源软件使用台账，保证开源软件使用有迹可循。

5.2.1.3 持续评估制度

机构宜建立开源软件持续评估制度，由管理团队牵头评估开源软件在流程规范、技术规范、安全合规等方面的符合性，记录问题并及时通知相关团队处理。

5.2.1.4 退出管理制度

机构宜制定开源软件退出管理制度，由管理团队组织相关团队评估确认拟退出开源软件或开源软件版本，制定退出方案。管理团队统一记录和管理退出的开源软件或开源软件版本。

5.2.2 风险管理制度

机构宜制定开源软件风险管理制度，由安全团队负责开源软件安全漏洞、供应链安全等风险管理，合规团队负责许可证合规、知识产权等风险管理，并做好相关风险的识别、记录、评估并及时处置。

5.2.3 二次开发管理制度

机构宜建立开源软件二次开发管理制度，明确二次开发规范、代码管理、版本控制、知识产权归属等要求，确保二次开发过程安全、合规。

5.2.4 社区贡献管理制度

机构宜建立开源软件社区贡献管理制度，规范员工参与开源社区贡献的行为，明确贡献内容审核、知识产权声明、合规性审查等要求。

5.2.5 运营管理制度

机构宜建立开源软件运营管理制度，将治理要求内嵌至工程实践，实现对开源资产持续、自动化、可审计的风险控制，并促进组织知识体系和人才能力的同步演进。

6 治理流程

6.1 开源软件生命周期管理

6.1.1 引入管理

遵循开源软件引入管理制度，开源软件依据相关要求检查、评估并做记录，审核通过后方可引入，主要包括以下事项。

- a) 开源需求确认，明确业务场景需求及引入开源软件的必要性，包括功能需求、性能需求等。
- b) 开源软件选型，依据开源软件选型规范选择开源软件，输出有优先级的备选清单和选型结论。
- c) 技术专家审查：
 - 技术审查：宜由管理团队组织研发、运维等相关团队开展技术评审，评估开源软件功能及性能符合性，判断技术成熟度、先进性、自主可控性和可持续性。
 - 安全审查：由安全团队负责安全评审，识别安全、供应链等风险。
 - 合规审查：由合规团队负责合规评审，识别许可证合规、知识产权等风险。
- d) 开源软件入库，由管理团队将评审后的开源软件的代码、制品拉取到机构内部，供后续使用和管理。
- e) 开源软件信息维护，由管理团队维护开源软件的信息，包括开源软件名称、版本、开源许可证、下载源、类别、引入方、引入时间等信息。

6.1.2 使用管理

遵循开源软件使用管理制度，建立开源软件使用台账，保证开源软件使用有迹可循，主要包括以下事项。

- a) 依据开源软件使用情况，建立台账，实时记录开源软件的使用版本、使用方、系统名、联系人等，保证开源软件的使用情况可追溯。
- b) 使用已引入开源软件时，需从制品仓库获取相关介质，不宜自行下载。
- c) 针对数据库、缓存、消息队列等重要的开源软件宜编制开源软件在信息系统设计、开发、测试、集成等研发方面的技术规范。
- d) 针对重要的开源软件宜编制开源软件在安装部署、运行监控、变更维护等运维方面的技术规范。对于涉及核心业务的开源软件应提供高可用、容灾解决方案及应急预案。

6.1.3 持续评估

遵循开源软件持续评估制度，定期开展开源软件在制度流程、技术规范、安全合规等方面的符合性评估，记录并通知相关团队处理发现的问题，主要包括以下事项。

- a) 定期审查开源软件制度流程遵从情况，包括引入、退出、二次开发、社区贡献等环节。
- b) 定期审查开源软件技术规范落实情况，包括研发、运维等技术规范。
- c) 定期审查开源软件安全合规问题整改情况，包括安全漏洞、许可证合规等。
- d) 定期跟踪开源许可证情况，及时对变更的开源许可证做出评估，在变更为证券期货行业机构不能接受的开源许可证时，考虑对此开源软件重新进行选型，使用同等功能的其他开源软件替换。

- e) 登记发现的问题，通知相关团队处理。

6.1.4 退出管理

遵循开源软件退出管理制度，开源软件依据相关要求评估，评审通过后方可退出。开源软件退出时，需明确整体方案。主要包括以下事项。

- a) 开源软件的退出必要性评估。
- b) 开源软件退出的技术方案、实施计划制定并组织评审。
- c) 开源软件退出执行：
 - 升级：使用该开源软件的新版本替换旧版本，新版本引入参考“引入管理”流程。
 - 更换：使用同类开源软件替换现有开源软件，新技术引入参考“引入管理”流程。
 - 弃用：如果该开源软件无业务需求时，进行退出操作。

6.2 风险管理

遵循开源软件风险管理制度，开展风险识别、风险记录、风险评估、风险处置，主要包括以下事项。

- a) 风险识别，重点识别以下风险：
 - 安全漏洞风险：识别开源软件漏洞、后门等安全风险。
 - 供应链风险：防范上游提供方停止维护、供应中断等风险，以及源码或文档中承载政治主张、恶意代码注入等风险。
 - 许可证合规风险：识别开源软件许可证的兼容性、使用限制、衍生软件分发等合规风险。
 - 知识产权风险：分析开源软件涉及的专利、著作权、注册商标使用情况，针对商用场景评估知识产权风险。
- b) 风险记录，记录识别到的开源软件风险。
- c) 风险评估，确认风险等级，分析风险影响范围，评估风险的可能性和后果，输出风险优先级处置列表。
- d) 风险处置，依据风险优先级处置列表，确定风险处置方案，通知相关团队采取相应措施修复或规避风险，确保在面临风险时能够及时妥善解决。

6.3 二次开发

遵循开源软件二次开发管理制度，按照机构的软件研发过程管理规范，修改、优化、扩展或定制开源软件，主要包括以下事项。

- a) 在设计阶段，确认开源软件现有情况，包括功能、接口兼容性、接口版本、开源许可证要求等，并确认采用独立分支策略或跟踪社区版本策略。
- b) 在开发阶段，明确开发分支负责人，标记并记录分支与主版本差异部分。采用跟踪社区版本策略时要定期与社区版本归并。如相关代码需要贡献给社区，在编码时宜遵循开源社区的编码规范和要求，按照开源许可证相关规定保留原版权信息并添加开发者的版权信息。
- c) 在测试阶段，对软件的相关功能进行必要的测试，如配置测试、单元测试、持续集成测试等。
- d) 在发布阶段，可采用灰度发布的方式，先在非核心业务上小范围使用，保证整体系统的稳定。

6.4 社区贡献

遵循开源软件社区贡献管理制度，在对外贡献代码或发起开源项目时，确保如下事项。

- a) 对待开源的软件进行软件成分合规性审查，确保使用的开源软件和开源代码不存在开源许可证兼容性与合规性问题。
- b) 待开源的软件或代码不含有第三方商业组件和代码。

- c) 待开源的软件或代码不含有涉及内部专利技术的代码。
- d) 待开源的软件或代码不包含我国出口管制相关技术。
- e) 待开源的软件或代码遵从对应社区的代码提交规范。
- f) 待开源的软件或代码宜设置统一提交人/者。
- g) 待开源的软件或代码宜发布在国内代码托管平台上。
- h) 待开源的软件或代码根据开源传播、开源运营和开源许可证合规性等要求,选择合适的开源许可证,且附有完整的版权声明信息。
- i) 待开源的软件或代码,宜制定贡献者许可协议(CLA)或开发者原创证书(DCO)供贡献者签署。
- j) 待开源的软件或代码确认不会对机构的服务以及内部系统造成安全风险。

6.5 存量管理

针对开源软件的存量情况开展梳理、记录与分析,主要包括以下事项。

- a) 制定和更新存量开源软件的管理策略与计划。
- b) 识别开源软件,记录开源软件的版本号、许可证、官网地址、可信下载源、制品仓库地址等,形成开源软件清单。
- c) 识别开源软件使用方,记录使用方的系统名、联系人等,形成开源软件使用清单。

7 工具平台

7.1 架构模型

工具平台包含通用能力与研发集成能力两个能力要素,工具平台架构模型如图 2 所示。通用能力包括开源软件资产视图、开源软件信息管理、应用系统信息管理、安全管理、合规管理、开放接口。研发流程集成能力包括引入管控、使用管控、持续集成管控、准出管控、退出管控。

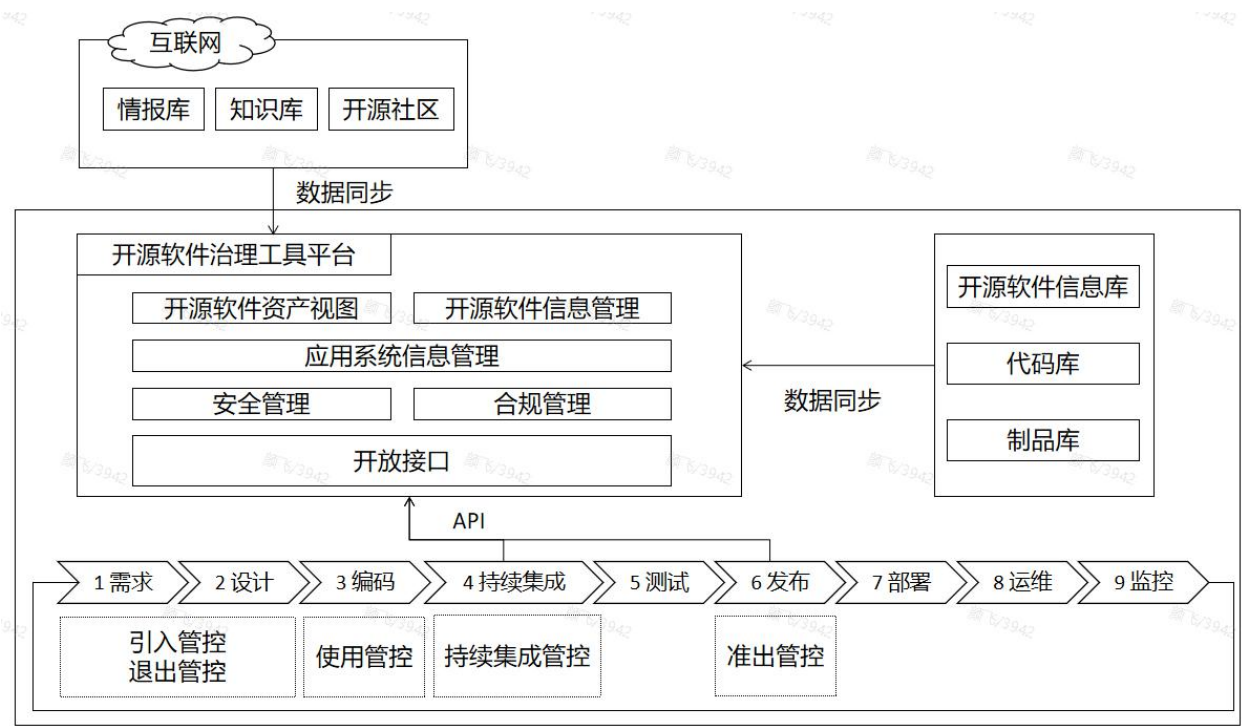


图 2 工具平台架构模型

7.2 通用能力

7.2.1 开源软件资产视图

开源软件资产视图模块宜包括：

- a) 提供不同层级组织的资产分布视图，多维度展示机构、部门、项目各使用者相关信息，可快速了解掌握相关软件资产信息，辅助决策。视图宜包含：
 - 开源软件使用情况总览与排行情况。
 - 新版本和漏洞提醒。
 - 机构、部门、项目级风险评估与统计。
 - 待办事项。
 - 项目状态跟踪。
 - 风险与缺陷流程跟踪。
- b) 提供开源软件在机构内部的分布情况，便于快速清理和升级问题开源软件。

7.2.2 开源软件信息管理

开源软件信息管理模块宜包含以下方面。

- a) 开源软件信息展示，包括名称、版本、描述、标签、引用情况、安全风险、合规风险、下载源、引入方、引入时间。风险详细信息包括历史风险记录及修复建议。
- b) 开源软件信息维护，包括描述、标签等信息维护。
- c) 开源软件引入、退出流程管理。

7.2.3 应用系统信息管理

应用系统信息管理模块宜包含以下方面。

- a) 应用系统信息展示，包括名称、版本、描述、开源物料清单、安全风险情况、合规风险情况、系统负责人及部门。
- b) 应用系统成分分析，支持主动上传指定格式的应用包来扫描分析应用系统并展示分析结果。

7.2.4 安全管理

安全管理模块宜包含以下方面。

- a) 与外部情报平台对接，新漏洞发布时对应用系统开展自检，对相关部门和人员发布漏洞预警。
- b) 漏洞查找定位，通过 CVE，CNNVD 等平台发布的漏洞编号，实现应用系统安全自查。

7.2.5 合规管理

合规管理模块宜包含以下方面。

- a) 开源软件的许可证信息展示与评估，支持多许可证检测及兼容性检测、无许可证告警、许可证类型变更告警。
- b) 开源许可证自动发布管理，支持开源许可证声明与文档自动生成。
- c) 开源软件的知识产权信息展示与评估，支持知识产权风险告警。

7.2.6 开放接口

平台提供开放 API，与机构内部已有工具深度集成，构建一体化的流程驱动平台，例如开源组件管理和制品库的集成、代码扫描工具等。

7.3 研发流程集成能力

7.3.1 引入管控

通过引入管控，确保只有通过开源风险综合评估的开源软件才被允许在研发流程中使用，未通过评估的组件应阻止被下载。

7.3.2 使用管控

开发人员根据平台知识库给出的信息，对所需要使用的组件进行安全评估，决定是否能使用该软件，评测过程宜标准化，评测文档系统存档。

7.3.3 持续集成管控

在应用构建过程中，开源治理平台自动分析项目中使用的组件情况，有违反安全策略的，可以阻断项目构建过程。

7.3.4 准出管控

在制品交付到上线之前，制品成分开源风险扫描结果数据作为交付准出的决策维度之一，在确认制品无风险之后，再通过部署平台发布到线上环境。

7.3.5 退出管控

开源软件在退出之前评估即将退出的版本对业务的影响面，全面评估退出后对业务的影响，制定合适的退出方案并执行，争取做到不影响业务稳定性和连续性。

8 评估体系

8.1 能力等级

指南将机构开源软件治理能力按照从低到高的顺序，共分为3级，分别为“探索级”、“提升级”、“成熟级”。每个成熟度级别标明机构开源软件治理能力所达到的水平。

- a) 探索级：机构对开源软件的治理处于初步和自发状态，尚未建立系统化的管理机制。开源软件的引入、使用和退出多依赖个人或团队的经验，缺乏统一的政策、流程和工具支持。风险管理被动零散，通常只在问题暴露后才临时应对，整体治理能力较弱，存在较高的安全、合规和运营风险。
- b) 提升级：机构已建立起系统化、制度化的开源软件治理框架，制定了明确的管理制度、流程和职责分工。实现了开源软件从引入、使用、持续评估到退出的全生命周期管理，能够主动识别和管理安全、合规等风险，并建设统一的工具平台支持治理活动。治理工作有序开展，但仍存在部分环节依赖人工、集成度不高等局限性。
- c) 成熟级：机构的开源软件治理已完全融入技术战略和业务流程，形成高度自动化、智能化和价值驱动的治理体系。通过平台化、数据化的手段实现风险的预测性识别与闭环管理，治理活动深度集成至 DevOps 流程。机构积极参与并影响开源生态，具备量化评估和价值回报体系，展现出卓越的治理效能和行业引领能力。

8.2 能力域

8.2.1 组织架构

评估机构开源软件治理的组织架构和职责分工情况。

8.2.2 制度体系

评估机构开源软件治理的制度体系建设情况，包括治理规划、管理制度、运营制度。

8.2.3 开源软件生命周期管理

评估机构开源软件全生命周期的管理能力，覆盖开源软件引入、使用、持续评估、退出阶段。

8.2.4 风险管理

评估机构在开源软件风险管理方面的能力。

8.2.5 生态参与管理

评估机构参与开源软件生态的能力，包括二次开发、社区贡献。

8.2.6 存量管理

评估机构存量开源软件梳理、记录与分析的情况。

8.2.7 工具平台

评估机构的工具平台对开源软件的管理水平，包括通用能力、研发集成能力。

8.3 评估指标

评估指标细则内容见表1。

表 1 评估指标细则

能力域	能力项	能力子项	能力等级	评估细则
组织架构	岗位设置	设立开源软件治理领导小组，明确其领域岗位及相关职责。	探索级	明确开展开源软件治理工作，但未明确开源软件治理领导小组岗位及职责。
			提升级	a) 明确领导小组岗位，发布领导小组成员名单。 b) 明确领导小组职责，制定开源软件治理战略，建立组织、制度、流程等工作机制，能够有效推动机构开源软件治理工作。
			成熟级	领导小组参与制定机构技术路线和战略决策，驱动开源软件治理工作，融入开源治理生态，评估开源投入和回报。
		设立开源软件管理团队，明确研发、运维、安全、合规领域岗位职责。	探索级	开源软件分散管理，尚未明确岗位及职责分工。
			提升级	a) 设立开源软件管理团队，统筹开源软件治理工作。 b) 建立跨职能的治理组织，集中管理开源软件，人员能力和数量配置合理，有清晰的职责分工，责任到人，能够有序开展开源软件治理相关工作。
			成熟级	开源软件管理团队成为机构技术战略的贡献者，有年度专项工作目标和规划。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
制度体系	治理规划	结合自身的技术基础、业务需求及行业发展趋势，精准分析当前在开源软件应用中存在的优势、短板与潜在风险，制定开源软件治理规划。	探索级	初步形成机构级开源软件治理战略，开源治理发展规划不明确。
			提升级	a) 制定了机构级开源软件治理战略，明确了治理目标、原则和重点领域。 b) 有明确的开源软件治理规划，包含实现目标的关键举措或者重点任务，组织保障和资源投入、考核与奖惩等责任落实情况。
			成熟级	动态评估战略的正确性、适应性及执行的有效性，持续优化开源软件治理规划的目标和任务。
		以开源软件治理规划为指引，从开源软件应用、流程机制、技术支撑、风险管控等维度实施开源软件治理工作方案。	探索级	对开源软件的引入和使用有初步的工作方案。
			提升级	依据开源软件治理规划，制定不同场景下针对性开源软件治理工作方案。
			成熟级	a) 有明确的开源软件治理工作方案实施主体。 b) 有清晰明确的开源软件治理工作方案执行流程。 c) 对开源软件治理工作方案执行效果进行评估、考核与奖惩，并有相关文档记录。
	管理制度	建立开源软件的生命周期、风险、二次开发和社区贡献方面的管理制度，包括相关流程。	探索级	a) 形成配套管理规章制度，但尚未规范开源软件全生命周期中各环节管理举措。 b) 对安全漏洞、许可证合规等风险的管理是被动和临时性的，通常只在问题暴露后才着手处理，未建立系统的风险识别、记录、评估和处置机制。 c) 二次开发未遵循特定规范，社区贡献也多属于员工个人行为，未建立组织层面的审核流程与支持体系。
			提升级	a) 制定了系统的开源软件生命周期管理制度，对全生命周期管理提出明确规定。 b) 制定成文的开源软件风险管理制度，明确了安全团队和合规团队的职责分工，能够系统性地识别安全漏洞、供应链风险、许可证合规等，并建立风险记录，推进风险评估和处置。 c) 建立了二次开发管理制度，明确了开发规范、代码管理和版本控制等要求。 d) 建立了社区贡献管理制度，规范员工参与开源社区贡献的行为。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
制度体系	管理制度	建立开源软件的生命周期、风险、二次开发和社区贡献方面的管理制度，包括相关流程。	成熟级	a) 开源软件引入评估与机构技术战略融合，并深度集成到DevOps流程，建立高度自动化且智能化的持续评估机制，退出机制充分考虑对业务连续性的影响，具有完善的影响分析和无缝迁移方案。 b) 开源软件风险管理深度融入机构整体风险管理和技术战略，建立量化的风险度量体系，风险管理活动高度自动化并集成到DevOps流程，能通过工具链实现风险的预测性识别与自动阻断，实现安全管控左移。 c) 机构积极培养内部核心贡献者并在关键开源社区寻求领导角色，以影响项目方向和塑造行业生态。 d) 机构有策略地将内部项目主动开源，以引领生态。
				a) 员工对自身在开源治理中的角色和职责认知模糊，岗位说明书中未包含明确的开源治理职责。 b) 知识零散分布在个人手中，没有统一的管理平台，没有成文的培训材料，缺少正式的培训计划，培训内容不成体系。 c) 通过口头传达、临时邮件等方式解决突发问题，出现重大事件后进行临时性宣贯。
	运营制度	构建开源治理运营制度，体系化地传递知识、明确职责，并激发参与，确保从开发者到管理者都能在开源软件治理工作中安全合规、步调一致。	探索级	a) 岗位说明书中明确开源治理相关的职责，为关键岗位定义清晰的任职资格和技能要求。 b) 建立统一的开源软件知识库，系统性地沉淀开源政策、流程、指南、最佳实践和常见问题解答，按角色分层分类，方便精准获取。 c) 制定年度开源培训计划，开发标准化的培训课程并记录完成情况。
				a) 设立激励机制，鼓励员工分享经验，鼓励核心人员参与行业开源社区。 b) 知识库演进为智能化、场景化的赋能平台，能主动向员工推送其所需的知识。 c) 设立可量化的开源治理运营指标体系。
			成熟级	a) 没有正式的引入管理制度，依赖个人或团队自发选择和使用开源软件。 b) 选型时未开展系统性的安全、合规和技术评估，仅关注功能实现。 c) 引入的开源软件没有统一的登记和管理，使用情况不可见。
开源软件生命周期管理	引入管理	依据引入管理流程要求检查、评估并做记录，审核通过后方可引入。	探索级	a) 没有正式的引入管理制度，依赖个人或团队自发选择和使用开源软件。 b) 选型时未开展系统性的安全、合规和技术评估，仅关注功能实现。 c) 引入的开源软件没有统一的登记和管理，使用情况不可见。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
开源软件生命周期管理	引入管理	依据引入管理流程要求检查、评估并做记录，审核通过后方可引入。	提升级	a) 制定了正式的引入管理制度，规范了申请、审核、批准的引入流程及要求，明确了参与角色和职责分工。 b) 系统化开展选型评估，依据选型规范，在初步筛选中评估许可证、产品认可度、产品活跃度、安全性等维度，在终选阶段评估性能效率、可维护性、可扩展性等更深层次的技术指标。 c) 组织跨团队的选型评审，管理团队能组织研发、安全、合规团队进行技术、安全、合规三方面的审查，并留存评审记。 d) 审核通过的开源软件统一拉取到内部仓库，并由管理团队维护包含名称、版本、许可证等核心信息的台账。
			成熟级	a) 引入评估与机构技术战略和业务目标深度融合，选型时不仅考虑当前项目，更评估其对技术生态的长期价值。 b) 具备自动化与智能化能力，将安全、合规扫描工具深度集成到 CI/CD 流水线，实现关键风险的自动识别与阻断。
	使用管理	依据开源软件使用管理流程要求，通过开源软件使用台账，保证开源软件使用有迹可循。	探索级	a) 未建立统一的开源软件使用台账，使用情况不可见、不可查。 b) 开发者可自行从互联网下载开源软件组件，来源无法保证，存在安全与版本不一致风险。 c) 没有制定研发、部署、运维等方面的技术规范，使用方式依赖个人经验。
			提升级	a) 制定了正式的开源软件使用管理制度，规范了台账、下载、技术规范的使用流程和要求，明确了各环节参与角色和职责分工。 b) 建立并维护统一的开源软件使用台账，及时更新，记录使用版本、使用方、系统名、联系人等关键信息，保证使用情况可追溯。 c) 要求使用已引入的开源软件时，必须从统一的内部制品仓库获取，禁止自行下载，确保组件来源可信、版本一致。 d) 针对重要的开源软件，制定研发技术规范 and 运维技术规范，并由管理团队组织审核后发布。
			成熟级	a) 使用管理与研发和运维过程深度集成，确保开源软件使用台账实时、准确。 b) 技术规范内嵌到工具链和模板中，并能根据运行数据持续优化技术规范。 c) 所有涉及核心业务的关键开源软件，均具备经过演练的高可用、容灾解决方案和应急预案，确保业务连续性。 d) 能够量化分析统一使用管理和技术规范带来的价值，如部署效率提升、故障率下降等。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
开源软件生命周期管理	持续评估	依据开源软件持续评估流程要求，定期开展开源软件在制度流程、技术规范、安全合规等方面的符合性评估，记录并通知相关团队处理发现的问题。	探索级	a) 未建立持续评估管理制度，评估活动是被动和临时性的。 b) 在出现安全事件、合规问题或业务故障时，才对特定开源软件进行事后审查。没有定期的审查计划，对许可证变更、社区停止维护等长期风险没有监控。 c) 记录与跟踪不足，发现的问题没有正式登记和跟踪流程，依赖口头或临时沟通。
			提升级	a) 制定了正式的开源软件持续评估管理制度，规范了持续评估相关流程和要求，明确了各环节参与角色和职责分工。 b) 定期组织审查，开展系统性的符合性评估，包括检查引入、退出、二次开发等环节是否按流程规定执行，检查研发、运维等技术规范是否被遵循，跟踪已知安全合规的整改情况。 c) 建立对开源软件许可证变更的监控机制，一旦发生变更能及时评估，并在必要时启动替换方案。 d) 建立统一的问题登记册，记录评估发现的问题，并通知相关团队处理与跟踪闭环。
			成熟级	a) 通过集成工具链自动监控流程符合性、安全合规风险、社区健康度与许可证变更等，实现预测性风险识别。 b) 建立关键开源软件依赖关系地图和风险图谱，能系统性分析单一组件风险对全局业务的影响，并制定应急预案。 c) 建立量化的评估指标看板，包括漏洞平均修复时间、规范遵从率，并利用评估数据反向驱动制度和规范的持续优化。
	退出管理	依据开源软件退出管理流程要求，提出退出申请，评估和方案评审通过后按要求退出，并制定退出方案和计划。	探索级	a) 缺少退出管理制度，开源软件因系统下线、出现无法解决的严重漏洞等原因的被动退出。 b) 退出前缺少必要性评估，也无技术方案和评审，引发业务中断或遗留风险。 c) 退出的软件没有统一记录，导致后续无法追溯，产生法律合规风险。
			提升级	a) 制定了正式的开源软件退出管理制度，规范了退出的全过程流程和要求，明确了各环节参与角色和职责分工。 b) 在退出前由管理团队组织相关团队进行必要性评估，制定详细的技术方案（升级、更换、弃用）和实施计划，并组织评审。 c) 新版本或替代品的引入严格遵循引入管理流程，确保平滑过渡。 d) 管理团队统一记录和管理所有已退出的开源软件及其版本，形成退出台账，保证有迹可循。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
开源软件生命周期管理	退出管理	依据开源软件退出管理流程要求，提出退出申请，评估和方案评审通过后按要求退出，并制定退出方案和计划。	成熟级	a) 退出管理与技术债务管理、IT 架构演进战略相结合，能主动规划技术栈的迭代。 b) 建立组件依赖关系地图，在评估退出方案时，能自动化、可视化地分析该变更对所有业务系统和流程的潜在影响。 c) 对于核心业务系统使用的关键开源软件，提前制定和演练退出应急预案，确保业务连续性。 d) 将退出过程中的迁移方案、工具、脚本沉淀为可复用的知识资产和工具包，显著降低未来同类迁移的成本与风险。
风险管理	风险管理流程	依据开源软件风险管理流程要求，开展风险识别、风险记录、风险评估、风险处置。	探索级	a) 未建立系统的开源软件风险管理制度，管理活动是被动和临时性的，通常仅在出现安全事件或合规问题时才采取行动。 b) 风险识别是被动和个案驱动的，依赖个人经验，未系统化覆盖安全合规领域。 c) 风险记录不完整，未建立规范的风险评估流程来确定等级和优先级。 d) 风险处置通常是在问题发生后才进行，未建立前瞻性的规避和修复计划。
			提升级	a) 制定了成文的开源软件风险管理制度，规定了风险管理流程和要求，明确了安全团队与合规团队的职责分工。规范了退出必须经过申请、评估、方案评审、批准和记录的全过程流程和要求，明确了各环节参与角色和职责分工。 b) 能够主动并系统性地识别开源软件在安全漏洞、供应链风险、许可证合规和知识产权等方面的风险。 c) 建立统一的风险登记册，记录已识别的风险。引入流程进行风险评估，确认风险等级并输出优先级处置列表。 d) 依据优先级列表，有流程地通知开发、运维等相关团队采取措施修复或规避风险，包括通过建立许可证白名单管理合规风险，或为关键组件准备备用机制。
			成熟级	a) 风险管理制度深度融入机构整体风险管理和技术战略，成为业务连续性和战略决策的组成部分。 b) 风险管理活动高度自动化并集成到DevOps流程，实现风险的预测性识别与自动阻断。 c) 建立量化的风险度量体系，能够评估风险对业务的影响。 d) 风险处置不再局限于内部，而是通过主动参与社区、培养内部核心贡献者，从源头降低和规避风险，并具备影响技术路线的话语权。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
生态参与 管理	二次开发	遵循开源软件二次开发管理制度，按照机构研发过程规范，修改、优化、扩展或定制开源软件。	探索级	a) 没有正式的二次开发管理制度，开发过程高度依赖项目团队或个人的自觉性。 b) 未系统考虑开源软件的许可证限制、接口兼容性、与社区的关系等。 a) 未明确分支策略。 b) 未保留开源软件版权信息、未明确二次开发代码的知识产权归属等存在合规风险。
			提升级	a) 制定了正式的开源软件二次开发管理制度，规定了开发规范、代码管理、版本控制和知识产权归属，建立了向社区贡献代码的内部审核与批准流程。 b) 将二次开发融入机构自身的标准软件研发生命周期，并在设计、开发、测试、发布等关键阶段设置控制点，遵循开源社区规范，避免合规风险。 c) 确立“上游优先”原则，积极将修复和改进贡献回社区，从源头减少维护成本。 d) 开源软件二次开发可以解决行业共性问题，或者提升开源软件整体应用水平。
			成熟级	a) 二次开发流程深度集成到 DevOps 流程，自动化工具确保版权信息规范、代码风格符合要求，并自动完成与开源社区的基线比对。 b) 机构在关键社区中拥有维护者，能影响社区方向以利于自身技术战略，能够评估不同分支策略的长期成本和收益，并基于产品路线图做出最优决策，与开源社区的同步和归并高效、低冲突。 c) 能够量化分析二次开发活动的投入产出，包括社区贡献带来的品牌提升、技术风险降低和人才成长价值。
	社区贡献	遵循开源社区贡献管理制度，在对外贡献代码或发起开源项目时，确保合规与安全。同时，对开源社区有一定贡献，全面反映组织在开源生态中的参与深度与价值输出能力。	探索级	a) 没有正式的社区贡献管理制度，员工贡献属于个人行为，机构不干预也不知晓。 b) 贡献前未进行任何合规、安全或知识产权审查，无意中泄露商业秘密、第三方代码或受管制技术。 c) 贡献过程不遵循社区规范，也未设置统一的机构身份，削弱了贡献的影响力和专业性。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
生态参与 管理	社区贡献	遵循开源社区贡献管理制度，在对外贡献代码或发起开源项目时，确保合规与安全。同时，对开源社区有一定贡献，全面反映组织在开源生态中的参与深度与价值输出能力。	提升级	a) 制定了正式的社区贡献管理制度，明确贡献前的审核流程、知识产权声明和合规性审查要求。 b) 建立强制性的贡献前审查流程，确保待贡献的代码/软件满足安全合规要求。 c) 机构的主要贡献者加入合法合规的开源组织，要求贡献时遵循社区代码规范，并设置统一的机构提交者身份，以提升品牌形象。
			成熟级	a) 社区贡献与机构技术品牌塑造、行业影响力和人才战略深度融合，贡献是有计划、有目标的，旨在影响关键开源项目。 b) 贡献审查流程高度集成到开发者工具链中，实现快速、高效的“一键式”合规检查与审批，提升开发者体验。 c) 主动发起并主导有影响力的开源项目，建立了完善的引入贡献者许可协议（CLA）或开发者原创证书（DCO）签署与管理体系，并拥有专业的开源运营团队来维护社区。 d) 机构的主要贡献者成为社区的维护者或核心决策者，对开源决策发挥影响力。
存量管理	存量开源 软件管理	根据开源软件使用情况，建立开源软件应用台账，并定期更新。	探索级	a) 没有制定存量开源软件的管理策略与计划。 b) 没有建立统一的开源软件清单和使用清单。 c) 依赖人工记忆或零散记录，无法系统性地识别和记录开源软件。
			提升级	a) 制定了存量开源软件的管理策略与年度盘点/更新计划，明确了管理目标和方法。 b) 系统性地识别并建立了开源软件清单和使用清单，制定了存量开源软件的管理策略与计划并定期开展全面排查。 c) 开源软件清单和使用清单由管理团队统一维护和管理，保证了数据的权威性和一致性。
			成熟级	a) 开源软件清单和使用清单的识别与更新高度自动化，与DevOps、制品仓库等系统深度集成，实现实时或准实时的数据同步。 b) 在基础清单之上建立开源软件全景视图，能深度分析与洞察，包括依赖关系、风险热力图，可视化全局风险。 c) 存量数据直接支撑战略决策，可用于技术债务评估、架构演进规划、漏洞应急响应影响分析等。

表 1 评估指标细则（续）

能力域	能力项	能力子项	能力等级	评估细则
工具平台	工具平台能力	工具平台包含通用能力与研发流程集成能力。通用能力包括开源软件资产视图、开源软件信息管理、应用系统信息管理、安全管理、合规管理、开放接口，研发流程集成能力包括引入管控、使用管控、持续集成管控、准出管控、退出管控。	探索级	a) 没有统一的开源软件治理平台，使用零散单点工具。 b) 开源软件资产、安全、合规等数据无法关联和整合，依赖人工收集和整理报表。 c) 工具未与机构的标准研发流程集成，管控全靠人工自觉，无法阻断高风险行为。 d) 无法提供全局的资产视图和风险视图，决策没有数据支持。
			提升级	a) 建设了统一的开源软件治理工具平台，资产视图、信息管理、安全管理、合规管理等核心模块全流程贯通。 b) 工具平台关联和整合开源软件资产、安全、合规等数据，可自动化生成相关报表。 c) 工具平台与机构标准的研发流程集成，在引入、使用、持续集成、准出等阶段及时发现和阻断开源软件风险。 d) 工具平台能够建立应用系统与开源组件的关联关系，形成初步的软件物料清单和资产视图，跨域数据初步形成关联。
			成熟级	a) 工具平台通过开放接口与机构所有研发工具链深度集成，形成一体化的、数据自动流转的开源软件治理体系。 b) 资产视图演进为实时、动态的供应链地图，能可视化分析影响范围。安全管理具备预测性风险情报，能智能识别潜在威胁。合规管理能进行复杂的、跨组件的许可证兼容性自动化推理。 c) 工具平台提供丰富的度量看板和数据分析能力，用于衡量治理效能、优化流程并指导战略决策。 d) 工具平台能与外部开源社区、情报源进行数据交换，能自动化管理机构对开源社区的贡献。

参 考 文 献

- [1] 《中国人民银行办公厅 中央网络安全和信息化委员会办公室秘书局 工业和信息化部办公厅 中国银行保险监督管理委员会办公厅 中国证券监督管理委员会办公厅关于规范金融业开源技术应用与发展的意见》（银办发〔2021〕146号）. 2021-09-28
- [2] JR/T 0290—2024 金融业开源软件应用 管理指南
- [3] GB/T 43698—2024 网络安全技术 软件供应链安全要求
-