ICS 35.080
CCS L00/09

团 体 标 准

T/ISC 0043—2024

软件代码自主率测评方法

Testing and evaluating methods for software source code self-development rate

发布稿

2024 年 1 月 29 日发布 2024 年 2 月 28 日实施

中 国 互 联 网 协 会 发布

目 次

前 言

本文件参照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》要求的格式进行编写。

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。

本标准由中国互联网协会归口。

本文件起草单位：中国信息通信研究院、北京酷德啄木鸟信息技术有限公司、北京风行网安科技有限公司、中国石油昆仑数智科技有限责任公司、中国移动通信集团设计院有限公司、中国电力科学研究院有限公司、中国民航信息网络股份有限公司、中建数字科技有限公司、南京众智维信息科技有限公司、安徽锋刃信息科技有限公司、北京德安信华科技有限公司、成都信息工程大学、四川赛闯检测股份有限公司、网宿科技股份有限公司、仁寿智仁智慧科技有限公司、四川仁恒智合科技有限公司、江苏大道云隐科技有限公司。

本文件主要起草人：蒋阿芳、马英轩、樊可欣、张合磊、赵平、郭治文、张志强、滕征岑、张嵩、李雪岩、周春山、程纯杰、周江山、缪思薇、周亮、左海峰、杨京煜、王宇、翟冬梅、段柯欣、贾大伟、梁小川、车洵、章帆、王晓宇、郝旭、冯丽、袁丽、方建康、周琼、黄莎琳、吕士表、杨志伟、刘丽娟、党杜均、邓恒、黄圣超。

软件代码自主率测评方法

1 范围

本文件规定了对软件和应用产品代码自主率的技术要求，描述了其测试方法。

本文件适用于软件开发企业开展软件和应用产品代码自主率的检测评价，也用于第三方检测机构开展软件产品的代码自主率检测。

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，标注日期的引用文件，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。

GB/T 25000.10—2016 系统与软件工程 系统与软件质量要求和评价（SquaRE）第10部分:系统与软件质量模型

GB/T 25000.51—2016 系统与软件工程 系统与软件质量要求和评价（SquaRE）第51部分:就绪可用软件产品（RUSP）的质量要求和测试细则。

GB/T 34943—2017 C/C++语言源代码漏洞测试规范

GB/T 34944—2017 Java语言源代码漏洞测试规范

GB/T 34946—2017 C#语言源代码漏洞测试规范

GJB/Z 141—2004 军用软件测试指南

QJ 3027A—2016 航天型号软件测试规范

3 术语和定义

GB/T 34943—2017、GB/T 34944—2017、GB/T 34946—2017、GJB/Z 141—2004、QJ 3027A—2016界定的以及下列术语和定义适用于本文件。

3.1 代码审计 source code audit

一种以发现程序错误，安全漏洞和违反程序规范为目标的源代码分析。

3.2 软件成分分析 software composition analysis

一种对软件的组成部分进行识别、分析和追踪的技术，用于分析了解应用程序中使用的开源组件和

依赖关系，以及如何自动化地使用这些组件和依赖关系。

3.3 源代码 source code

以适合输入到汇编程序、编译器或其他翻译器的形式表示的代码。
[ISO/IEC 2382-7:2000]

3.4 代码自主率 code autonomy rate(CAR)

代码自主率是指软件产品中自主开发的代码占总代码的比例，它反映了软件产品的自主创新能力和知识产权保护程度。一般来说，代码自主率越高，软件产品越具有竞争优势和市场价值。

3.5 自主代码行数 self-development code lines（SCL）

指去掉代码注释后的自主代码行数，即自主开发的有效代码行数。

3.6 总代码行数 total code lines（TCL）

指去掉注释后的代码总行数，即有效代码总行数。

3.7 开源源代码比例 open source code rate（OSR）

即已使用开源源代码的行数占总行数的比例。

3.8 开源组件比例 open source component rate（OCR）

即已使用开源代码组件/文件的数量占总文件的数量的比例。

4. 送检清单

为保证检测的有效性和效率，送检方需提供以下资料：

（1） 软件基本情况
（2） 软件著作权
（3） 软件功能清单，并对功能有简单介绍
（4） 软件安装手册，包含软件完整安装所需要的操作系统、数据库、中间软件等
（5） 软件使用手册
（6） 软件源代码，包含代码、配置文件等编译所需内容
（7） 软件所使用开源组件清单，包含组件名称、版本号以及许可协议类别

（8） 诚信承诺书：

a) 所提交内容的真实性、完整性承诺；

b) 承诺拥有所送检软件的所有权利，不存在与第三方的产权纠纷。

5. 检测环境

检测环境应由检测方和送检方共同搭建。

送检方应按照所提供的安装手册完成系统检测运行环境部署以及代码编译环境部署工作，送检方须确保所部署系统的代码为检测系统的完整代码。

检测方负责提供硬件环境和软件操作系统环境，并应与送检方确认检测环境配置信息以及代码编译环境信息。

6. 前置条件

为保证检测的准确性，在检测之前需验证以下内容：

（1） 功能完整性：所安装的软件包含功能清单中所有功能；

（2） 所使用的操作系统、数据库、中间件等其他支撑软件或系统均与清单一致；

7. 检测内容和规则

7.1 检测内容

代码自主检测包含以下检测内容：

（1） 相关软件的自主可控检测；

（2） 开源组件依赖性的检测；

（3） 软件总代码行数检测；

（4） 软件的安全性检测；

7.2 相关软件的自主可控性检测

对软件运行所需要的相关软件进行分类：

（1） 操作系统；

（2） 数据库；

（3） Web 中间件等中间件；

（4） 其他必须的软件，如 FTP 等

检测规则如下：

（1） 如果送检方所送软件相关软件中，不含某类软件，则该类视为合格；

（2） 对于每一类分别评估是否具有充分可替代性（包含国产替代）或者是完整的开源软件，具备修改的能力和授权；

（3） 若某类不合格，则该类得分为 25%。

7.3 开源组件依赖性检测

通过单独使用或组合使用以下方式开展代码成分分析检测工作，获取检测数据：

工具检测（A）：使用成熟的商业成分分析工具对项目进行扫描，识别代码成分组成。

人工审核（B）：人工审核扫描结果，分析成分组成结果，判断组件缺陷危害程度，评估 License 风险。

与送检方确认最后的开源组件依赖清单。

7.4 代码总行数检测

使用工具对源代码部分的代码总行数进行检测。

代码总行数包括软件代码，包括配置文件。

注释内容和空行予以扣除。

原则上按照分行符计算代码行数。对于特殊代码，软件总行数小于代码命令数 10%，检测方可以酌情考虑是否按照代码结束符来计算代码行数。

7.5 代码安全性检测

对代码安全性进行综合检测，先对软件源代码进行静态代码安全审计，对运行环境进行系统扫描等安全检测，以及检测方认为必要的检测。

漏洞检测结果出来以后由检测方专家审核：严重漏洞：可以直接利用，实现重要数据泄露、代码泄露、恶意文件上传、软件被篡改或恶意操纵、软件无法正常运行等。

（1） 高危漏洞：通常安全检测工具判断的，真实存在的高危漏洞。

鉴于软件检测的特殊情况，对于不合理配置导致的漏洞，如果送检方可通过合理配置消除，则该漏洞可忽略。

代码安全性得分=(严重漏洞数*5%)+高危漏洞数/（代码总数/10000）*1%

7.6 代码自主率计算

代码自主率基础计算公式：

$$CAR1=(SCL/TCL)\times(1-\sum \alpha \times OSR-\beta \times OCR)$$

式中：

CAR——代码自主率

SCL——自主开发有效代码行数

TCL——有效代码总行数

OSR——开源代码行数/有效代码总行数

OCR——开源代码组件或文件数量占总文件数量的比例

α——开源代码协议权重；

β——开源组件协议权重。