ICS 03.080.99
CCS A 20

DB32

江　苏　省　地　方　标　准

DB 32/T XXXX—XXXX

小微企业质量管理提升规范 软件企业

Code for quality management improvement of the small and micro enterprise in software enterprise

（征求意见稿）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

江苏省市场监督管理局　　发 布

目　　次

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专利的责任。

本文件由江苏省市场监督管理局认证监督管理处提出并归口。

本文件起草单位：南京市雨花台区市场监督管理局、方圆标志认证集团江苏有限公司、江苏省产品质量监督检验研究院、江苏省质量和标准化研究院。

本文件主要起草人：顾正、马雪龙、姚卓、王卿、鲁杰、蒋林辉、赵铎荣。

小微企业质量管理提升规范 软件企业

1 范围

本文件规定了小微软件企业质量管理提升的总体原则，确立了需求分析过程管理、软件开发过程管理及软件测试过程管理的实质要求。

本文件适用于指导小微软件企业实施质量管理提升行动，小微软件企业自主开展质量管理提升工作时可参照执行。

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。

GB/T 19001 质量管理体系 要求

GB/T 8566 系统与软件工程 软件生命周期过程

GB/T 11457 信息技术 软件工程术语

T/SIA 002 软件企业评估标准

3 术语和定义

GB/T 8566、GB/T 11457、T/SIA 002界定的以及下列术语和定义适用于本文件。

3.1

小微企业 small and micro enterprise

小型企业、微型企业、家庭作坊式企业的统称。

注：符合国家统计局《统计上大中小微型企业划分办法》（国统字〔2017〕213号）规定的小型、微型企业划分，具有独立法人以及法人分支机构资格的社会经济组织。

[来源：国家统计局《统计上大中小微型企业划分办法》（国统字〔2017〕213号）]

3.2

质量管理 quality management

在质量方面指挥和控制组织的协调的活动。

[来源：GB/T 8566—2022/ISO/IEC/IEEE 12207:2017，3.1.42]

3.3

软件企业 software enterprise

在中国境内依法设立的从事软件产品（3.6）开发销售（营业）及相关服务的企业。

[来源：T/SIA 002—2019，3.1]

3.4

软件产品 software product

符合我国软件产品管理要求的软件成果，交付给客户的最终成果，在本文件中产品的含义也包含服务、系统。

[来源：GB/T 8566—2022/ISO/IEC/IEEE 12207:2017，3.1.54]

3.5

需求分析 requirement analysis

a）研究用户需要以得到系统、硬件或软件需求的定义的过程。

b）研究和重新定义系统、硬件或软件需求的过程。

[来源：GB/T 11457—2006，2.1363]

3.6

过程管理 process management

为开发一产品或执行服务要执行的工作的方向、控制和协调。

[来源：GB/T 11457—2006，2.1202]

3.7

需求 requirement

对需要及其约束和条件的表达。

[来源：GB/T 8566—2022/ISO/IEC/IEEE 12207:2017，3.1.44]

3.8

利益相关方 stakeholder

在系统或所属其特性中有权利、份额、声明或利益，以满足其需要及期望的个人或组织。

注：某些利益相关方可具有相互对立或系统对立的利益。

[来源：GB/T 8566—2022/ISO/IEC/IEEE 12207:2017，3.1.59]

4 质量管理提升原则

4.1 问题导向原则

以问题为导向，聚焦小微软件企业质量管理中存在的痛点、难点问题，建立符合行业特征和企业需求的质量管理体系。

4.2 主体原则

确立小微软件企业在质量管理提升中的主体地位，强化主体责任，发挥主体作用。

4.3 培育原则

构建小微软件企业分层分级培育机制，完善质量管理提升服务体系，引导企业质量管理能力提升。

4.4 信息保密原则

在质量管理提升过程中获取的企业所有资料（包括图片、影像、录音等），未经上级部门批准不得向任何对象泄露。

5 需求分析过程管理

5.1 控制要点

5.1.1 需求分析应实现利益相关方需求向技术解决方案的转化，将期望的能力从面向用户或利益相关方的视图，转化为满足用户运行需要的技术视图。

5.1.2 需求分析过程管理应遵循下列控制要点：

5.1.2.1 需求收集

——应识别所有相关方的需求，将非形式的需求表述（如，用例、用户故事、应用场景）转换为结构化需求条目；

——应绘制项目原型图，采用图、线、框的形式表示，形成完整需求集；

5.1.2.2 需求分析

——应评估需求的技术可行性，识别需求集潜在问题；

——应建立需求优先级矩阵，对需求优先级进行排序，评价维度应包含业务价值、实现难度和风险等级；

——应形成需求分析报告，包含可行性分析、冲突解决方案和成本效益评估。

5.1.2.3 需求描述

——应确认所有需求已被充分获取和表达，充分响应利益相关方的期望；

——应开展需求的动态行为分析和静态数据分析，分析结果应形成记录；

——经确认的需求应形成软件需求说明书，包含功能性需求（输入处理、输出控制、数据管理）、质量属性（可靠性、响应时间）和接口协议（API 调用规范、数据交换格式）。

5.1.2.4 需求验证

——应组织跨部门评审验证，查验利益相关方需求、软件需求完整集、软件需求分析记录、软件需求说明书的完整性和可追溯性；

——应实施基线化管理，建立变更控制程序。未经基线化的需求不得进入下一阶段开发工作，基线化后的需求不得擅自更改，任何变更应经过评审。

5.2 实施指南

5.2.1 需求分析阶段应执行以下流程：

5.2.1.1 原始需求

——记录用户或利益相关方非形式的需求表述（如，用例、用户故事、应用场景），可不设置验收指标以及责任人；

——应标注需求采集日期，记录需求来源（客户、法规、内部改进）；

——应对需求进行初始分类，分类维度应包含业务需求、用户需求、系统需求。

5.2.1.2 系统需求

——应将非形式的需求表述（如，用例、用户故事、应用场景）转换为结构化需求条目；

——应形成明确的可验证条款与验收指标，可不设置责任人；

——应建立需求-测试用例映射表。

5.2.1.3 任务分解

——应对系统需求进行拆分，形成的独立功能点；

——功能点应与利益相关方的需求一致，功能点间交互关系应清晰；

——应设置明确的可验收指标以及责任人，作为最小化需求的跟踪。

5.2.2 需求分析过程输出应包括但不限于：

——所有相关方的需求和期望收集完整；

——采用软件工程的语言结构将原始需求转换为结构化需求条目；

——评估需求的技术可行性，建立需求优先级矩阵；

——软件需求（功能需求、性能需求、过程需求、非功能性需求和接口需求）和设计约束明确、可实现；

——明确软件产品及需求评估的关键性能指标；

——形成文档记录，包含利益相关方需求、软件需求完整集、软件需求分析记录、软件需求说明书，文档记录完整、可追溯。

5.3 检查改进

5.3.1 设置指标实施监控，包含需求评审缺陷密度、需求变更率和测试用例覆盖率。

5.3.2 建立需求跟踪矩阵：

——应记录每个需求的相关属性，明确每个需求的关键信息；

——典型属性应包括但不限于唯一标识、需求表述、需求收录缘由、所有者、需求来源、优先级别、版本、当前状态和状态日期；

——适宜时，可增加其他关联属性（如，稳定性、复杂性、性能指标和验收标准）。

5.3.3 实施持续改进措施：

——应开展需求实现偏差分析，设置偏差纠正措施；

——应评估需求管理工具效能，优化响应流程；

——应定期更新需求优先级评估模型，基于实际情况调整权重因子。

6 软件开发过程管理

6.1 控制要点

6.1.1 全生命周期质量管理体系

——应制定覆盖需求分析、系统设计、编码实现、测试验证的全流程质量管控方案；

——应建立质量追溯机制，各阶段质量要素可验证、可追溯。

6.1.2 项目计划管理

——应编制软件开发计划，包含功能点分解、进度节点、资源配置；

——应输出经评审的软件设计说明书，明确系统架构和接口规范；

——应实施版本基线管理，确保开发过程可追溯。

6.1.3 业务机构协同

——业务部门应全程参与原型设计、功能验证等关键项目节点；

——应建立业务需求与技术实现的联合评审机制；

——应定期开展架构合理性评估，形成架构优化建议。

6.1.4 数据质量控制

——应制定数据完整性校验规则和异常数据处理机制；

——应实施人工抽样和自动化脚本相结合的双重比对机制；

——应建立数据质量监控体系，定期开展数据质量分析。

6.2 实施指南

6.2.1 需求分析阶段应执行以下流程：

6.2.1.1 概要设计

——开发团队应依据经批准的软件需求说明书进行功能模块分解，采用结构化分析方法完成系统架构设计，输出概要设计说明书；

——概要设计说明书应包括模块化功能组织结构图、系统全局数据结构定义、模块间消息接口规范、分层数据流图、状态转换图和模块耦合度分析；

——应设计跨部门评审机制，组织评审会议，参与人员应包括项目经理、系统架构师、质量保证工程师、测试人员和运维支持人员；

——评审会议应输出评审会议纪要、问题跟踪表、架构合理性评估和风险项记录。

6.2.1.2 详细设计

——应对概要设计说明书细化分解，输出详细设计说明书；
——详细设计说明书应包含接口参数校验规则、数据结构存储方案、错误码定义规范；
——应实施双人互审机制，确保设计逻辑与需求规格双向追溯，接口定义与概要设计保持一致。

6.2.1.3 软件编码

——应制定软件编码规范，明确命名规则、注释标准、安全编码要求，建立编码规范检查清单；
——应实施代码版本基线管理，开展静态代码扫描、编码规范符合性检查和交叉代码走查；
——软件开发人员不得擅自变更接口参数、全局变量、数据库结构，应提出变更需求，经评审后进行修改；
——变更实施后应执行影响范围分析、回归测试用例更新、关联文档修订。

6.2.2 软件开发过程输出应包括但不限于：

——制定各阶段节点的开发计划；
——输出设计说明书，包含概要设计与详细设计；
——组织设计方案评审会议，输出评审会议纪要、问题跟踪表、架构合理性评估和风险项记录；
——完善开发计划和设计说明；
——确认所有相关方在需求和设计实现方面达成一致；
——文档记录归档保管；
——实施软件开发工作。

6.3 检查改进

6.3.1 软件质量保证计划

——软件质量保证计划应包括技术评审、软件测试、质量保证、缺陷跟踪以及其他可能影响软件产品质量的技术要点。
——应实施代码质量量化管理，量化指标应包含设计文档评审缺陷密度、代码重复率和单元测试通过率。
——适宜时，建立代码审查缺陷跟踪机制和分类统计机制。

6.3.2 项目配置工具

——应建立配置管理体系，对目录结构进行系统性策划：
——应建立配置目录结构，设定目录访问和存取权限；
——每个配置项应标识作者、时间、版本号、状态等信息；
——应定期编制配置状态报告，应包含基线变更统计、版本发布记录、备份完整性验证；
——应建立配置管理审计机制，定期检验不符合项关闭情况。

7 软件测试过程管理

7.1 控制要点

7.1.1 代码质量管理

——应执行静态代码分析，检查范围应包含逻辑结构、属性定义、命名规范和代码布局；
——应实施代码重构机制，消除冗余代码，统一编码风格。

7.1.2 测试体系

——应建立分层测试策略，包含单元测试、集成测试和系统测试；
——适宜时，可采用组合测试方法。

7.1.3 缺陷管理

——应实施缺陷全流程跟踪，包含缺陷发现、定位、修复及验证；
——应建立自动化测试框架，提升阶段集成测试效能。

7.1.4 文档标准化

——应制定测试文档规范，包含语法规范、语义表述和逻辑结构的要求。

7.2 实施指南

7.2.1 单元测试

——测试用例应覆盖全部逻辑路径及边界条件；
——应建立回归测试机制，维护阶段变更后须执行再验证；
——宜采用单元测试框架实施自动化测试；
——应输出单元测试报告，纳入配置管理。

7.2.2 集成测试

——测试对象应聚焦模块接口协议及消息交互机制；
——集成测试应在单元测试通过后开展；
——应编制接口测试矩阵，明确输入/输出参数验证规则；
——应实施与单元测试的环境隔离策略。

7.2.3 系统测试

——应依据软件需求说明书逐项验证功能实现；
——应执行非功能性测试，包含性能、安全和兼容性测试；
——缺陷修复后应执行全量回归测试；
——应分析缺陷根本原因，制定预防措施。

7.3 检查改进

软件测试过程的检查改进应符合以下要求：
——应检查源代码的逻辑、属性、命名标准和代码布局；
——应验证代码的编译、链接、集成以及构建；
——应使用开发工具自带的编译功能或专门的程序对软件源代码进行检查；
——应关注源代码是否无矛盾或遗漏、无定义变量、变量无赋值、死循环的情况；
——应开展软件质量评估工作，包括但不限于软件质量特性的内容，即使用性、可测试性、正确性、维护性、可靠性、移植性、效率、重用性、完整性、互操作性和适应性；
——应组建专业测试团队，使用自动化软件测试工具，完成全方位的软件质量验证；
——应遵守质量保证计划中设定的关键节点开展检查工作。