

ICS 35.240.99  
UNSPSC 43.23.27  
CCS L 77



# 团 体 标 准

T/UNP 258—2024

## 内嵌软件语音降噪系统技术规范

Technical specification of embedded software voice noise reduction system

2024 - 11 - 14 发布

2024 - 11 - 14 实施

中国联合国采购促进会 发布



## 目 次

前言 .....	III
引言 .....	IV
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语、定义和缩略语 .....	1
3.1 术语和定义 .....	1
3.2 缩略语 .....	2
4 基本要求 .....	2
4.1 软件开发 .....	2
4.2 软件文档编制 .....	2
4.3 软件质量保证 .....	2
4.4 兼容性 .....	3
4.5 可靠性 .....	3
4.6 互通性 .....	3
4.7 扩展性 .....	3
4.8 稳定性 .....	3
4.9 易操作性 .....	3
5 系统架构 .....	4
6 功能要求 .....	4
6.1 实时语音降噪 .....	4
6.2 语音活动检测 .....	4
6.3 回声消除 .....	5
6.4 噪声抑制 .....	5
7 性能要求 .....	5
8 安全要求 .....	6
8.1 硬件安全 .....	6
8.2 软件安全 .....	6
8.3 信息安全 .....	6
9 运维要求 .....	6
9.1 日常维护 .....	6
9.2 性能监控 .....	7
9.3 故障处理 .....	8
9.4 应急预案制定 .....	8
9.5 问题跟踪机制 .....	8
9.6 升级管理 .....	9
10 评价改进 .....	9

参考文献..... 10

全国团体标准信息平台

## 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。

本文件由武汉铃声科技有限公司提出。

本文件由中国联合国采购促进会归口。

本文件起草单位：武汉铃声科技有限公司、湖北睿唐空间智能科技有限公司、武汉吉客威睿数字科技有限公司、湖北国威时代信息技术有限公司、武汉锐佳讯科技有限公司。

本文件主要起草人：郑伟莲、卢踮跹、余汉汉、李嘉茵、胡彬。

## 引 言

为助力中国企业参与国际贸易,推动企业高质量发展,中国联合国采购促进会依托联合国采购体系,制定服务于国际贸易的系列标准,这些标准在国际贸易过程中发挥了越来越重要的作用,对促进贸易效率提升,减少交易成本和不确定性,确保产品质量与安全,增强消费者信心具有重要的意义。

联合国标准产品与服务分类代码(UNSPSC, United Nations Standard Products and Services Code)是联合国制定的标准,用于高效、准确地对产品和服务进行分类。在全球国际化采购中发挥着至关重要的作用,它为采购商和供应商提供了一个共同的语言和平台,促进了全球贸易的高效、有序发展。

围绕UNSPSC进行相关产品、技术和服务团体标准的制定,对助力企业融入国际采购,提升国际竞争力具有十分重要的作用和意义。

本文件采用UNSPSC分类代码由6位组成,对应原分类中的大类、中类和小类并用小数点分割。

本文件UNSPSC代码为“43.23.27”,由3段组成。其中:第1段“43”为大类,表示“信息技术广播和电信”,第2段为中类,“23”表示“软件”,第3段为小类,“27”表示“网络应用软件”。

# 内嵌软件语音降噪系统技术规范

## 1 范围

本文件规定了内嵌软件语音降噪系统的基本要求、系统架构、功能要求、性能要求、安全要求、运维要求及评价改进。

本文件适用于内嵌软件语音降噪系统的设计和应用。

## 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 8566 系统与软件工程 软件生存周期过程

GB/T 8567 计算机软件文档编制规范

GB/T 28171 嵌入式软件可靠性测试方法

GB/T 28172 嵌入式软件质量保证要求

## 3 术语、定义和缩略语

下列术语和定义适用于本文件。

### 3.1 术语和定义

#### 3.1.1

**内嵌软件语音降噪系统** embedded software voice noise reduction system

安装在特定设备内部，旨在对语音信号进行处理，降低背景噪声，提升语音质量的软件系统。

#### 3.1.2

**回声消除** echo cancellation

运用特定算法和技术消除扬声器和麦克风之间因物理路径引起的回声干扰的过程。

#### 3.1.3

**噪声抑制** noise suppression

采用频域与时域相结合的算法等手段，降低或消除不需要的噪声成分，保持语音清晰度的功能。

#### 3.1.4

**自适应滤波器** adaptive filter

能根据输入信号和期望输出信号自动调整其参数的滤波器。

#### 3.1.5

**非线性处理** non-linear processing

在回声消除或其他语音处理环节中，针对非线性回声或回声与语音信号重叠等复杂情况所采用的处理方式。

#### 3.1.6

**频域处理** frequency-domain processing

通过对语音信号进行频谱分析，识别噪声在不同频率上的分布情况并加以抑制的噪声抑制处理方式。

#### 3.1.7

**时域处理** time-domain processing

针对噪声在时间上的特性，如突发、持续等情况进行处理的噪声抑制处理方式。

#### 3.1.8

**全量备份** full backup

对整个系统数据或指定数据集合进行完整复制的备份方式。

### 3.1.9

**增量备份 incremental backup**

只备份自上次备份（全量备份或增量备份）以来发生变化的数据的备份方式。

### 3.1.10

**数字签名 digital signature**

用于验证软件来源和完整性的技术手段。

### 3.1.11

**哈希函数 hash function**

将任意长度的数据映射为固定长度哈希值的函数。

## 3.2 缩略语

下列缩略语适用于本文件。

ARM: 高级精简指令集机器 (Advanced RISC Machine)

DSP: 数字信号处理器 (Digital Signal Processor)

FPGA: 现场可编程门阵列 (Field Programmable Gate Array)

SNR: 信噪比 (Signal-to-Noise Ratio)

## 4 基本要求

### 4.1 软件开发

4.1.1 软件开发应遵循 GB/T 8566 规定, 涵盖软件生命周期的各个阶段, 包括需求分析、设计、编码、测试、维护等。

4.1.2 需求分析阶段应深入调研用户需求和应用场景, 明确系统功能、性能和界面要求, 形成详细的需求文档。

4.1.3 设计阶段应采用先进的架构模式和设计理念, 保证系统的可扩展性、灵活性和稳定性。应进行模块化设计, 降低模块间的耦合度, 提高代码的复用性。

4.1.4 编码过程应遵循规范的编程风格和代码结构, 并添加注释。

4.1.5 测试阶段应进行全面的单元测试、集成测试和系统测试。制定详细的测试计划和用例, 及时发现并修复软件缺陷。

4.1.6 维护阶段应建立有效的维护机制, 及时响应并处理用户反馈的问题。应对软件进行定期更新和升级, 适应不断变化的需求和环境。

### 4.2 软件文档编制

4.2.1 软件文档编制应符合 GB/T 8567 的规定, 编制全面、准确的软件文档。包括需求文档、设计文档、测试文档、用户手册等。

4.2.2 需求文档应详细描述系统的功能需求、性能需求、数据需求以及用户界面需求等, 为开发团队提供明确的目标和方向。

4.2.3 设计文档包括系统架构设计、模块设计、数据库设计等, 阐述系统的内部结构、模块之间的交互关系以及数据存储方式。

4.2.4 测试文档记录测试用例、测试结果、问题反馈和解决情况等。

4.2.5 用户手册应以通俗易懂的语言向用户介绍系统的安装、使用方法、操作流程和注意事项。

### 4.3 软件质量保证

4.3.1 软件质量保证应符合 GB/T 28172 的规定, 建立完善的软件质量保证体系。制定质量计划, 明确质量目标和质量控制流程。

4.3.2 在开发过程中应进行代码审查和质量检测, 采用静态分析工具和代码审查流程, 检查代码的规范性、安全性和可靠性。

4.3.3 应制定详细的测试策略和计划, 进行全面测试, 包括功能测试、性能测试、兼容性测试和安全

性测试等。

4.3.4 应建立质量反馈机制，收集用户和测试人员的反馈意见，及时分析和处理问题，持续改进软件质量。

#### 4.4 兼容性

4.4.1 系统应兼容多种嵌入式平台，如 ARM、DSP、FPGA 等。针对不同平台的硬件特性和指令集架构，进行优化和适配。

4.4.2 在 ARM 平台上，应利用其低功耗和高性能的特点，优化系统的资源利用率。

4.4.3 在 DSP 平台上，应发挥其数字信号处理能力，提升语音信号处理的效率和精度，实现与系统其他部分的无缝对接。

4.4.4 再 FPGA 平台上，通过硬件编程实现高速并行处理，满足实时性要求高的应用场景。

#### 4.5 可靠性

4.5.1 应采用成熟的软硬件技术和硬件设备，构建稳定的系统架构。系统应具备备份、维护保障、容错和系统恢复能力。

4.5.2 应建立完善的备份机制，定期对关键数据进行备份，并确保备份数据的安全性和可恢复性。备份方式包括全量备份和增量备份，根据数据的重要性和变化频率选择合适的备份策略。

4.5.3 应提供维护工具和接口，方便技术人员进行系统维护和故障排查。维护工具应具备系统监控、性能分析、故障诊断等功能。

4.5.4 应具备容错能力，系统能自动检测和处理硬件故障、软件错误和网络异常等情况。采用冗余设计和错误检测机制，确保系统在出现故障时仍能保持部分功能的正常运行。

4.5.5 应具备系统恢复能力和机制，在故障发生后能迅速恢复到正常运行状态。制定详细的系统恢复流程和应急预案，减少故障对用户业务的影响。

#### 4.6 互通性

4.6.1 平台安全防护设备应开放相应的服务端口，实现系统各平台间的数据通信和信息共享。

4.6.2 在数据传输过程中，应进行数据加密和校验，保障数据的安全性和完整性。采用安全传输协议，如 SSL/TLS 等，对敏感数据进行加密处理。

4.6.3 应建立数据缓存机制，优化数据传输的性能。

#### 4.7 扩展性

4.7.1 系统应采用模块化设计，将各个功能模块划分为相对独立的单元。每个模块具有清晰的接口和明确的功能定义。

4.7.2 设计时宜考虑系统未来的发展需求，预留一定的扩展空间和接口，能方便集成新的技术和功能。

#### 4.8 稳定性

4.8.1 应运用冗余部署和负载均衡等技术手段，冗余部署包括硬件冗余和软件冗余，如服务器冗余、数据存储冗余等。

4.8.2 能根据系统的负载情况，动态分配用户请求到不同的服务器或处理节点上。

4.8.3 应对系统进行实时监控，及时发现并处理潜在的性能问题。建立性能监控指标体系，包括 CPU 使用率、内存使用率、网络带宽利用率等指标。

4.8.4 当发现性能问题时，应能及时采取相应的措施进行优化和调整，如调整系统参数、优化算法流程、增加硬件资源等。

#### 4.9 易操作性

4.9.1 应提供清晰、简洁、友好的人机交互界面。界面设计遵循直观性原则，操作流程简单明了，功能布局合理。

4.9.2 操作控制方式应简便、灵活，用户可以通过鼠标、键盘、触摸屏等多种方式进行操作。同时，系统应提供实时的操作提示和反馈信息，帮助用户正确完成操作。

4.9.3 对于系统的管理和维护，应提供便捷的管理工具和可视化的监控界面。管理员可以通过管理工

具进行系统配置、用户管理、权限设置等操作，通过监控界面实时了解系统的运行状态和性能指标。

## 5 系统架构

内嵌软件语音降噪系统架构见图1。

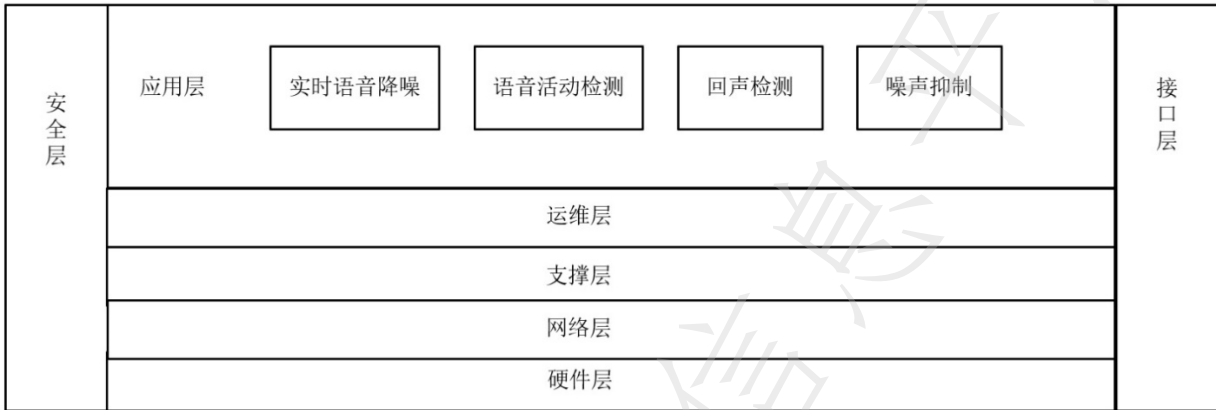


图1 内嵌软件语音降噪系统架构

## 6 功能要求

### 6.1 实时语音降噪

系统能实时捕获并处理语音信号，动态降低背景噪声，具体包括：

- a) 算法优化：
  - 1) 基于先进的信号处理技术和机器学习算法，系统能自动识别并适应不同类型的噪声环境，如街道噪声、交通噪声、办公室噪声等。通过对大量噪声样本的学习和分析，建立噪声模型库，实时匹配当前环境噪声，选择最优的降噪算法；
  - 2) 具备自学习能力，能根据用户的反馈和使用习惯不断优化降噪效果。系统能记录用户对降噪效果的评价和调整操作，通过数据分析和模型更新，逐步改进算法参数。
- b) 低延迟处理：
  - 1) 采用高效的信号处理算法和硬件加速技术，在捕获到语音信号的同时能立即进行降噪处理；
  - 2) 能自动优化算法流程和数据处理架构，减少数据传输和处理环节的延迟，保证语音信号的实时性。
- c) 多通道支持：
  - 1) 支持多通道输入/输出，如立体声或多麦克风阵列。多麦克风阵列技术能利用多个麦克风采集到的不同空间位置的信号，通过算法对信号进行分析和处理，实现对特定方向噪声的有效抑制和语音信号的增强；
  - 2) 对于立体声输入/输出，系统能提供更加丰富和逼真的音频体验，满足音乐播放、视频观看等多种应用场景的需求。

### 6.2 语音活动检测

系统能检测语音活动的开始和结束，在无声时段关闭降噪算法，具体包括：

- a) 高灵敏度与低误报率：
  - 1) 能准确捕捉语音活动的细微变化，包括轻声说话、短暂的停顿和语音的起始与结束。采用先进的语音信号分析技术和算法，降低环境噪声对检测的影响；
  - 2) 能通过多轮验证和滤波处理，过滤非语音信号，提高检测准确性。
- b) 自适应调整：

- 1) 能根据当前的噪声环境和用户的使用习惯进行自适应调整。实时监测环境噪声的变化，自动调整语音活动检测的阈值和参数，适应不同的环境条件；
  - 2) 能学习用户说话习惯，如语速、语调、停顿时间等，优化检测算法，提供更加精准的语音活动检测服务。
- c) 多场景应用：支持多种应用场景下的语音活动检测功能，如电话会议、在线教育、远程医疗等。

### 6.3 回声消除

系统能消除扬声器和麦克风之间的物理路径引起的回声干扰，具体包括：

- a) 自适应滤波器：
  - 1) 采用自适应滤波器技术，实时调整滤波器参数，适应不同的声学环境和扬声器/麦克风配置。通过对回声路径的实时跟踪和分析，不断优化滤波器参数，消除回声干扰；
  - 2) 当声学环境发生变化时，如用户移动设备位置、房间布局改变等，系统能迅速感知并自动调整滤波器参数，保持良好的回声消除效果。
- b) 双讲检测：
  - 1) 具备双讲检测功能，能判断当前是单人讲话还是双人同时讲话。在双人同时讲话时，相应地调整回声消除算法，平衡双方语音信号的处理；
  - 2) 利用语音信号的特征分析和实时监测技术，准确区分不同说话人的语音信号，实现智能的回声消除处理。
- c) 非线性处理：
  - 1) 支持非线性回声消除算法，处理复杂的回声情况，如非线性回声或回声与语音信号重叠的情况。采用先进的信号处理技术和模型，准确识别和分析非线性回声的特征，进行针对性的消除处理；
  - 2) 对于回声与语音信号重叠的情况，通过特殊的算法和技术手段，实现语音信号和回声的有效分离和处理。

### 6.4 噪声抑制

系统能降低或消除噪声成分，保持语音清晰度，具体包括：

- a) 频域与时域处理：
  - 1) 支持频域与时域相结合的噪声抑制算法，同时处理不同频率和时间段的噪声信号。在频域上，通过频谱分析识别噪声的频率成分，采用相应的滤波器进行抑制；
  - 2) 在时域上，对噪声的时间特性进行分析，如噪声的突发、持续等情况，采取时域滤波、自适应阈值处理等方法进行处理。
- b) 动态范围调整：
  - 1) 具备动态范围调整功能，自动调整降噪算法的增益和阈值，适应不同音量级别的语音信号。当语音信号音量较大时，适当降低降噪算法的增益；
  - 2) 当语音信号音量较小时，提高降噪算法的灵敏度，确保有效抑制噪声。通过实时监测语音信号的幅度和功率，动态调整参数。
- c) 用户自定义设置：
  - 1) 提供用户自定义设置功能，用户可根据自己的喜好调整降噪强度、滤波器等参数。例如，用户可以在不同的环境下选择不同的降噪强度级别；
  - 2) 提供可视化的设置界面和参数调节工具，方便用户进行操作和调整，提高用户的参与度和满意度。

## 7 性能要求

系统性能符合以下要求：

- a) 降噪效果：应将 SNR 提升至少 5 dB，对于不同类型的噪声环境，如稳定的白噪声环境、复杂多变的现实生活噪声（包含多种频率和幅度变化的声音，如街道交通、市场嘈杂声等）以及特定频段突出的噪声（如工厂机械噪声等），均应保证 SNR 的稳定提升；

- b) 残留噪声水平：在降噪后的音频中，残留噪声不应影响语音的可懂度和清晰度。可通过测量残留噪声的能量或主观听感来评估；
- c) 实时性：系统整体延迟（从语音采集到输出）应小于 100 ms，对于实时性要求高的应用场景，如在线游戏语音、实时视频会议中的交互环节等，应将延迟控制在 50 ms~80 ms；
- d) 功耗：在正常工作状态下，系统功耗应低于设备总功耗的 10%。

## 8 安全要求

### 8.1 硬件安全

#### 8.1.1 物理安全

物理安全应符合以下要求：

- a) 具备数据的物理保护机制；
- b) 具备在受到暴力移除或拆卸时的防护预警机制。

#### 8.1.2 硬件接口安全

硬件接口安全符合以下要求：

- a) 对于使用无线和有线外围接口的设备，宜通过指示灯或显示屏等方式，显示数据传输的监控状态；
- b) 对于具有调试功能的接口，应在出厂时设置为默认关闭。

### 8.2 软件安全

- 8.2.1 软件应包含供应商或者开发者的数字签名信息和软件属性信息（如版本名称、版本信息和描述等）。
- 8.2.2 软件应对于认证信息提供安全性措施。
- 8.2.3 软件应提供认证失败处理机制。
- 8.2.4 软件向系统申请权限时应遵循最小化原则以及合理的申请方式。

### 8.3 信息安全

#### 8.3.1 数据保护

数据保护安全符合以下要求：

- a) 保护用户的音频数据，不应被未经授权的人员访问或使用；
- b) 在用户未授权的情况下，不应保存用户音频。

#### 8.3.2 数据传输

数据传输安全符合以下要求：

- a) 应采用超文本传输安全协议等安全传输协议进行数据传输，对敏感数据（如个人身份信息、账户信息等）进行加密处理；
- b) 各执行主体之间进行数据传输时，应保证数据传输的完整性和机密性；
- c) 检测到数据完整性遭受破坏时，应采取措施恢复或重新获取数据。

## 9 运维要求

### 9.1 日常维护

#### 9.1.1 数据备份

##### 9.1.1.1 应根据数据重要性和变更频率确定备份周期：

- a) 核心语音数据和关键系统配置信息：每日全量备份，每小时增量备份；
- b) 稳定系统参数和基础架构信息：每周全量备份。

##### 9.1.1.2 备份数据宜采用磁盘阵列、磁带库等冗余存储介质。

## 9.1.2 数据恢复

9.1.2.1 应建立针对不同场景的恢复流程手册，具体场景包括但不限于：

- a) 硬件故障；
- b) 软件错误；
- c) 人为误操作；
- d) 自然灾害。

9.1.2.2 应进行恢复操作预演和测试，通过数据哈希值比对、数据记录数核对等校验方法保证数据一致性和完整性。

## 9.1.3 日志管理机制

9.1.3.1 应分类存储和统一管理不同类型日志，包括但不限于：

- a) 操作系统；
- b) 应用程序；
- c) 网络设备；
- d) 安全设备。

9.1.3.2 关键系统日志保存期限不少于1年，定期迁移旧日志至长期存储介质归档。

## 9.1.4 日志检查与更新

9.1.4.1 安排专人每日用分析工具和脚本检查日志，提取关键信息并分析，包括但不限于：

- a) 错误码；
- b) 异常事件；
- c) 访问控制违规。

9.1.4.2 根据分析结果制定更新计划（软件补丁安装、系统参数调整、安全策略优化等），于非业务高峰时段操作，提前通知用户，做好测试和验证。

## 9.2 性能监控

### 9.2.1 性能阈值

9.2.1.1 应针对CPU使用率、内存使用率、磁盘I/O速度、网络带宽利用率、系统响应时间、语音编解码帧率、降噪算法处理时长、回声消除效果评估指标等关键性能指标设阈值。

9.2.1.2 宜综合考虑系统设计容量、历史性能数据、业务服务水平协议确定阈值。

### 9.2.2 报警机制

9.2.2.1 应具备多级报警功能，根据问题严重程度通过短信、电子邮件、即时通讯工具、系统管理界面弹窗等通知不同级别运维人员和业务负责人。

9.2.2.2 报警信息包含性能指标名称、当前值、阈值、超出阈值持续时间、可能影响业务范围，建立报警升级机制。

### 9.2.3 负载测试

9.2.3.1 应定期开展全面负载测试，涵盖正常业务高峰、异常业务高峰（突发大量用户使用语音功能）、高并发数据传输、复杂语音环境（多种噪声源同时存在）等场景。

9.2.3.2 使用专业负载测试工具模拟真实用户行为和数据流量，对各组件和功能进行压力测试。

### 9.2.4 性能评估

9.2.4.1 应关注响应时间、吞吐量、资源利用率等基本指标和语音清晰度、流畅度、高负载下降噪效果等用户体验指标。

9.2.4.2 应生成详细性能评估报告，包括性能瓶颈定位、根本原因分析、对业务潜在影响、优化建议，对比历史数据分析性能变化趋势。

### 9.3 故障处理

#### 9.3.1 故障信息收集

9.3.1.1 故障发生时启动监控系统，收集时间、涉及设备和模块、受影响用户范围、故障表现形式（语音中断、噪声过大、系统死机等）信息。

9.3.1.2 根据基本信息初步判断故障类型和影响范围。

#### 9.3.2 硬件故障排查

应使用硬件检测工具（诊断卡、网络测试仪、音频分析仪等）检查硬件物理连接、工作温度、电源供应、组件运行状态。

#### 9.3.3 软件故障排查

9.3.3.1 应通过查看系统日志、应用程序日志、核心转储文件等分析软件运行轨迹和错误信息确定故障点。

9.3.3.2 宜采用分层排查和分段排查结合方法，逐步缩小故障范围直至定位原因。

#### 9.3.4 故障修复与验证

9.3.4.1 硬件故障宜依据保修和备件情况更换或维修硬件；软件故障宜根据类型采取补丁修复、配置文件修改、程序重启等措施。

9.3.4.2 修复后全面测试和验证系统，应记录故障处理全过程（排查步骤、工具方法、原因、修复措施、结果）。

### 9.4 应急预案制定

#### 9.4.1 应急处理流程

应针对不同故障类型制定应急处理流程和责任分工，包括针对数据中心火灾、地震等自然灾害的紧急恢复计划。

#### 9.4.2 联系人信息

应详细列出应急处理涉及的联系人信息（内部运维团队成员、硬件和软件供应商技术支持人员、上级领导、业务部门负责人等）及其职责和联系方式。

#### 9.4.3 预案演练

应定期演练和测试应急预案，模拟真实故障场景，根据演练结果修改和完善预案。

### 9.5 问题跟踪机制

#### 9.5.1 问题跟踪系统

9.5.1.1 应建立全面问题跟踪系统，为每个问题分配唯一标识号，记录发现时间、报告人、描述、严重程度、影响范围、处理状态等信息。

9.5.1.2 支持多维度查询和统计，方便分类分析和趋势预测。

#### 9.5.2 问题处理进度

应实时更新处理进度（已采取措施、结果、下一个步骤、预计完成时间），复杂问题建立升级机制。

#### 9.5.3 问题复盘与反馈

问题解决后应复盘总结根本原因，将经验教训反馈到设计、开发、运维等环节。

## 9.6 升级管理

### 9.6.1 版本控制

9.6.1.1 应建立版本控制体系，版本号遵循统一命名规则（含主版本号、次版本号、修订号、内部版本号或构建号）。

9.6.1.2 版本控制系统应记录发布时间、发布说明（新增功能、改进功能、漏洞修复列表等）、适用环境、依赖版本、升级步骤和注意事项，并保存代码、配置文件、文档等。

### 9.6.2 升级测试

9.6.2.1 测试环境搭建：升级前应搭建模拟真实生产环境的测试环境（硬件配置、软件环境、网络环境、数据量、用户负载等）。

9.6.2.2 应进行多维度测试，包括但不限于：

- a) 功能测试：逐一测试系统功能，确保升级后功能完整正确，关注新旧功能受影响情况；
- b) 性能测试：检查不同负载下性能表现，对比升级前后指标；
- c) 兼容性测试：验证与硬件设备、操作系统、其他软件应用程序兼容性；
- d) 安全测试：全面评估升级后系统安全，检查是否引入新安全漏洞（用户认证、数据传输、存储安全等方面）。

9.6.2.3 应详细记录测试结果，反馈问题给开发团队修复至测试通过。

### 9.6.3 升级通知

9.6.3.1 升级计划确定后，应提前通过系统公告、电子邮件、短信等多渠道向受影响用户和相关部门发通知。

9.6.3.2 通知内容应包括升级时间、时长、内容、目的、对用户影响（业务中断时间、功能变化等）、用户准备事项（保存数据、退出系统等）。

9.6.3.3 升级过程中应保持沟通渠道畅通，向用户通报进展情况（已完成阶段、是否有问题、预计剩余时间）。

9.6.3.4 出现意外情况时，应及时向用户解释原因并提供新预计完成时间，升级完成后收集反馈评估效果并改进。

## 10 评价改进

依据第6章~9章规定的要求，定期开展内嵌软件语音降噪系统的功能、性能、安全和运维方面的评价，审查不合格项，并有针对性地采取纠偏措施并持续改进。

### 参 考 文 献

- [1] GY/T 322.1—2019 网络音频应用的开放式控制架构 第1部分：框架
  - [2] GB/T 35319 物联网系统接口要求
  - [3] GB/T 37025 信息安全技术物联网数据传输安全技术要求
-