ICS 03.120.10
CCS A 00

`

中华人民共和国国家标准

GB/T

汽车软件质量与缺陷管理规范

Automotive software quality and defect management specification

（征求意见稿）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

XXXX - XX - XX 发布　　XXXX - XX - XX 实施

国家市场监督管理总局
国家标准化管理委员会　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。

本文件由全国产品缺陷与安全管理标准化技术委员会（SAC/TC463）提出并归口。

本文件起草单位：国家市场监督管理总局缺陷产品召回技术中心、上海市嘉定区国际汽车质量标准化协会、奇瑞汽车股份有限公司、吉利汽车集团有限公司、深圳引望智能技术有限公司、上海蔚来汽车有限公司、同济大学、广州小鹏汽车科技有限公司、比亚迪汽车工业有限公司、广州汽车集团股份有限公司、重庆长安汽车股份有限公司、北京理想汽车有限公司、浙江清华长三角研究院、清华大学、上海质界科技有限公司、德赛西威汽车电子股份有限公司、博世（中国）投资有限公司、科大讯飞股份有限公司、博泰车联网科技（上海）股份有限公司、东软集团股份有限公司、上海机动车检测认证技术研究中心有限公司。

本文件主要起草人：肖凌云、徐华、王海、李艳、李文昭、苏雯、周敏彪、贺兴、胡文浩、余荣杰、萧晗、谭玉函、席明、赵宗芳、李江坤、杨龙龙、张恒、王福华、陈亮、张伟伟、熊璐、王剑、黄晋、李秉奎、王波、惠阳阳、尹路、李小锋、苍学俊、董红磊、曲现国、余艳丽、周凡华、陈振威、蒋治文、许乃平、周小红、郭悦、赵俭平、冯占军、沈兴华、许晓春。

引　　言

《中华人民共和国国民经济和社会发展第十五个五年（2026－2030年）规划纲要》指出，我国将加快智能网联新能源汽车等战略性新兴产业发展，随着自动驾驶、智能座舱、车用人工智能、车载操作系统等关键创新技术在汽车领域不断广泛深入应用，汽车质量安全内涵和外延都发生了深刻改变，智能网联新能源汽车的可靠性与安全性将高度依赖汽车软件质量水平。相较于汽车硬件，汽车软件在设计、开发、部署、升级和迭代等全生命周期环节具有显著差异，软件运行过程的不直观性、不透明性，以及人工智能系统的不可解释性，均对传统汽车质量管理模式构成巨大挑战。

在软件定义汽车背景下，汽车的功能、性能和用户体验等质量和安全问题越来越受到软件的影响。由于整车系统架构的复杂性和使用工况的特殊性，软件问题在整车系统中可能引发高风险等级的安全后果，其影响范围广、隐蔽性强，需作为重点质量控制和安全验证对象。本文件在软件质量管理基础上，集成汽车功能安全、预期功能安全、信息安全、人工智能安全、数据安全等特定要求，引入风险评估与缺陷处置，形成覆盖需求、设计、实现、验证与运维全生命周期的车规级软件质量与缺陷管理规范。

本文件采用过程方法，融合PDCA循环与基于风险的管理思维。通过策划（Plan）软件质量计划、实施（Do）过程质量保证、检查（Check）节点评审与风险评估、处置（Act）缺陷等关键活动，实现软件质量安全的过程控制，通过持续的风险评估，实现缺陷识别前置和预防控制，旨在全面提升软件质量与组织的质量保障能力，确保汽车软件质量安全目标达成。本文件体现了全生命周期软件质量和缺陷管理，主要包括质量策划、过程质量保证、关键节点评审、软件风险评估、软件缺陷管理共5个部分，相互之间关系模型如图1所示：

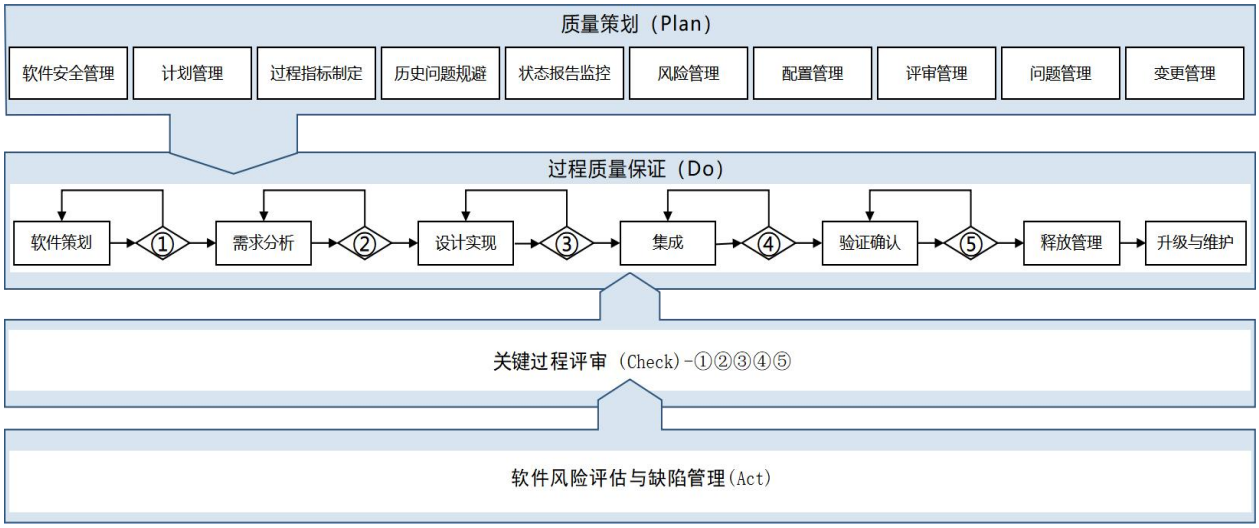

图1 全生命周期软件质量和缺陷管理

本文件适用于从事汽车软件全生命周期活动的相关组织，包括全新开发、增量开发及现有软件的升级维护，旨在满足以下需求：

——确保软件质量满足基本要求，降低由软件问题导致的安全风险；

——建立、实施、保持和改进组织的软件质量与缺陷管理体系；

——提供组织之间共享软件风险分析与安全管理的最佳实践；

——加强组织与相关监管部门与机构之间的协调与合作。

汽车软件质量与缺陷管理规范

1 范围

本文件描述了汽车软件质量与缺陷管理的总体实施框架与方法，对软件产品全生命周期内的软件策划、过程质量保证、关键过程评审、软件风险评估及软件缺陷管理等活动给出了具体的要求与指南。

本文件适用于汽车产品生产者、软件及相关组件供应商、系统供应商以及供应链上下游中其它为车辆提供软件产品或服务的组织。

本文件适用于汽车软件及与汽车进行直接交互的软件，包含车辆嵌入式软件、云端软件、车用人工智能软件、车载操作系统、基于机器学习的软件组件和移动端实时控制软件等。

本文件适用于汽车软件的全新开发、增量开发以及现有软件的升级维护等活动。

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 34402—2017 汽车产品安全风险评估与风险控制指南

GB/T 39603—2020 缺陷汽车产品召回效果评估指南

GB/T 45496—2025 汽车产品召回信息缺陷评估指南

3 术语和定义

GB/T 43387—2023、GB/T 11457—2006、ISO/IEC/IEEE 24765:2010界定的以及下列术语和定义适用于本文件。

3.1

配置项 configuration item

在配置控制下被管理的项目。

注1：配置项可包括软件项、代码库、测试用例、编译器、数据、文档、物理介质和外部接口。

注2：对配置项的描述可包括：

a) 物品类型；
b) 相关的配置管理系统；
c) 负责任的所有者；
d) 归于配置管控的日期；
e) 状态信息（如开发、基线、已发布等）；
f) 与较低级别配置项的关系；
g) 变更控制记录；
h) 变更历史。

3.2

文档化 documentation

系统或软件产品生成和维护文档的过程。

注：包括如下要素：

a) 文档集合,可包括描述、定义、报告，计划或认证活动，需求，程序及结果的书面或图形信息；

b) 生成或修改文档的过程；

c) 对文档的管理过程，包括识别、获取、处理、存储和传播。

3.3

软件项 software item

软件程序中可被集成为软件整体的组成元素。

注：典型的软件项包括：

a) 软件子程序；

b) 软件模块；

c) 软件库；

d) 配置文件。

3.4

交付物 deliverables

软件全生命周期的不同阶段所产生并需提交的工作产品，如文档、计算机模型、源代码、脚本等。

3.5

软件问题 software issue

软件程序不符合定义或基本使用认知的运行表现。

3.6

历史问题规避 avoidance of solved issues recurrence

对产品曾发生问题总结反思后，通过技术和管理方法提升软件产品的稳定性，避免同样或同类问题再次发生的行为。

3.7

人工智能 artificial intelligence；AI

人工智能系统（3.8）相关机制和应用的研究与开发。

[来源:GB/T 41867—2022,3.1.2]

3.8

人工智能系统 artificial intelligence system

针对人类定义的给定目标，产生诸如内容、预测、推荐或决策等输出的一类工程系统。

注1:该工程系统使用人工智能相关的多种技术和方法,开发表征数据、知识、过程等的模型,用于执行任务。

注2:人工智能系统具备不同的自动化级别。

[来源:GB/T 41867—2022,3.1.8]

3.9

机器学习 machine learning

通过计算技术优化模型参数的过程，使模型的行为反应数据或经验。

[来源:GB/T 41867—2022,3.2.10]

3.10

人工智能安全 AI safety

不存在由于人工智能（3.7）错误、故障或功能不足引起的危害而导致不合理的风险（3.26）。

3.11

功能安全 functional safety

不存在由电气/电子系统的功能异常表现引起的危害而导致不合理的风险（3.26）。

[来源:GB/T 34590.1—2022,3.67]

3.12

预期功能安全 safety of the intended functionality; SOTIF

不存在因预期功能或其实现的功能不足引起的危害而导致不合理的风险（3.26）。

[来源:GB/T 43267—2023,3.25]

3.13

信息安全 cybersecurity

汽车的电子电气系统、组件和功能被保护,使其资产不受威胁的状态。

[来源:GB 44495—2024,3.1]

3.14

数据安全 data security

通过采取必要措施,确保数据处于有效保护和合法利用的状态,以及具备保障持续安全状态的能力。

[来源:GB/T 41479-2022,3.4]

3.15

软件缺陷 software defects

软件程序中存在的破坏正常运行能力并会对整车造成安全风险的软件问题。

3.16

质量策划 quality planning

划定软件开发的目标和交付物的特征范围，定义和管理项目或迭代目标和计划，规定必要的运行过程和相关资源以实现软件质量目标的行为。

3.17

需求分析 requirement analysis

针对某系统应提供的服务以及对其运行的约束进行挖掘和描述，以及分析、文档化及评审与确认这些服务和约束的过程。

注：软件需求可包括基本软件需求，与功能安全、预期功能安全、信息安全、人工智能安全相关的安全需求。

3.18

设计实现 design implementation

依据已定义的需求，分析、研究、评估和选择明确且安全的解决方案，并且按照方案构建并转化为实际软件、功能、产品或服务的过程。

3.19

集成 integration

将多个产品组件整合成为更复杂的产品组件或完整产品的过程。

3.20

验证 verification

评价系统或部件,以确定开发周期中的一个给定阶段的产品是否满足在阶段的开始确立的需求的过程。

注1:在设计和开发中,验证是指对某项指定活动的结果进行检查的过程,以确定该活动是否符合明确的需求。

注2:"验证过的"一词用来表示相应的状况。

3.21

确认 validation

在开发期间或结束时对系统或部件进行评价,提供证据证明允许直接与最终用户交互的最终产品，在其运行目标环境中满足使用期望的过程。

注1:在设计和开发中,确认关系到检查产品是否符合用户要求的过程

注2:确认一般是在规定的操作条件下对最终产品进行的。在早期阶段,这样做是必要的。

注3:"确认过的"一词用来表示相应的状况。

注4:如果有几种不同的预期用途,可进行多种确认。

3.22

变更管理 change management

通过识别变更请求、分析评估其对技术、质量、业务及安全的影响、制定并批准变更实施方案、跟踪执行过程、验证变更结果、记录变更文档并最终关闭变更的系统化流程。

3.23

释放管理 release management

通过对软件发布前的构建、验证确认、配置和文档化等过程进行检查，并评审软件质量成熟度、版本兼容性、部署策略及回滚机制，从而决策软件是否可以发布的过程。

3.24

配置管理 configuration management

通过配置项识别、储存控制、版本控制、基线控制和审核活动，建立并维护配置项的完整性和一致性，实现产品依其状态可被唯一识别并获取、不同版本的关联和差异可被追溯的过程。

3.25

评审管理 review management

通过对交付物的审查，提前识别和解决问题，减少因问题/失效引起额外成本的活动。

3.26

风险 risk

伤害发生概率和伤害严重程度的组合。

[来源:GB/T 43387—2023,3.11]

3.27

风险评估 risk assessment

确定危险事件或情形的严重性与发生可能性的综合水平等级的过程。

[来源：GB/T 43387—2023,5.13]

3.28

风险控制 risk control

减小或避免危险事件或情形发生的策略与措施。

[来源：GB/T 43387—2023,5.14]

3.29

风险评估对象 risk assessment object

可能存在故障或失效的汽车软件缺陷或开发过程。

[来源：GB/T 34402—2017，2.7，有修改]

3.30

结构化走查 structured walk-through

由有能力人员依据标准要求对系统或其中部分的需求、设计或实现进行系统性审查的行为。

4 质量策划

4.1 总则

为保证软件质量与缺陷预防，汽车产品生产者、软件及相关组件供应商、系统供应商以及供应链上下游中其它为车辆提供软件产品或服务的组织应建立并运行软件质量安全管理体系并对软件需求、设计实现、集成、验证和确认、释放管理、软件升级及维护的实现过程进行质量策划，实施软件安全管理、计划管理、过程指标制定、历史问题规避、状态报告监控、风险管理、配置管理、评审管理、问题管理、变更管理等关键质量保证活动。

4.2 软件安全管理

组织应基于对国内外适用软件产品相关标准分析进行质量策划，如涉及安全应包括，但不限于功能安全、预期功能安全、信息安全、人工智能安全、数据安全等内容的策划（如安全管理目标、安全管理活动，对应交付物及评审计划），管理范围覆盖产品生命周期的各个阶段（需求、设计、实现、验证、确认、生产、运维），主要内容应包含安全目标定义、安全需求管理、危害分析与风险评估、安全导向的设计与实现、安全验证与确认、生产运维与支持过程的安全策划、工具链与供应链安全等。

4.3 计划管理

项目应策划在软件全生命周期中对需求、设计、开发、集成、验证等活动的具体执行和监控方法。项目应根据要求，在项目立项启动后，更新并发布相关计划。如采用迭代开发方式，应按迭代制定计划。

软件项目计划应包括：

a) 项目目标和所需交付物的特征范围，以满足法律法规、验证/认证、诊断、维护要求，及客户的需求；

b) 项目角色、职责和所需技能，并与项目人员映射关系；

c) 分解的子任务及任命授权；

d) 项目资源：设备，工具，人力等明细及可用保证；

e) 时间规划，包括分解的子任务时间跨度、里程碑、阶段目标；

f) 沟通计划；

g) 软件验证计划；

h) 软件开发及验证所需环境/场地的可用保证；

i) 维护计划；

j) 评审计划；

k) 软件发布计划。

也宜包括：

a) 风险规划；

b) 对指定项目人员的培训规划。

其中：

开发过程应与项目生命周期相匹配，项目活动应涵盖所有过程。

软件验证计划应分级描述，应包括软件单元验证、对机器模型的验证、集成验证及系统验证与确认。软件验证计划应描述验证活动的具体执行方法，并制定回归验证准则。

组织应制定并发布沟通计划。沟通计划应描述沟通形式、沟通方法及沟通的内容。

4.4 过程指标制定

项目应规划和监控软件开发过程指标，用于支持验证和确认以及释放管理。

软件指标宜包括：

a) 单元验证覆盖率；

b) 编程规范符合度；

c) 当前和计划的微处理器（包括CPU、GPU）的负载；

d) 当前和计划的闪存/RAM/EEPROM 内存使用率；

e) 需求评审率、需求实现率；

f) 缺陷密度、测试通过率等；

g) 问题关闭率；

h) 功能安全相关指标（如安全目标ASIL等级达成，安全机制有效性等）；

i) 信息安全相关指标(（如安全目标CAL等级达成，安全机制有效性等）；

j) 预期功能安全相关指标（如未知场景识别能力、危害触发抑制能力等）；

k) 人工智能安全相关指标（如模型可靠性、数据集、算法认知缺陷等 ）；

l) 数据安全相关指标（如可靠性、保密性、完整性等）。

4.5 历史问题规避

项目应收集相关的监管信息、召回信息、客户反馈及项目过往发生的问题，在新项目开发初期制定问题规避计划，针对所有历史问题改进的结果需要逐一验收闭环。

4.6 状态报告监控

项目应编制项目进度报告，状态报告应显示实际项目活动的时间，负责人和进度，并保持对项目计划的更新。组织在任何情况下发现阶段性目标无法完成时，都应立即通知所有相关方，说明目标无法完成的原因以及项目将受到的影响。当项目进度落后时状态报告要包含追赶计划及相应措施。

状态报告应包括：

a) 项目活动和成果总结；

b) 风险管理清单；

c） 项目进度和当前问题。

项目状态报告宜包括：

a) 根据项目活动分解出的任务进度；

b) 需求规格的完成情况及趋势；

c) 软件维护状态；

d) 报告期间的其他问题和建议。

4.7 风险管理

4.7.1 应根据软件实现计划中详述的开发过程进行风险管理。

4.7.2 应编制和维护风险管理清单，应描述已识别的项目/产品风险，并根据项目计划保持更新。

4.7.3 风险管理清单应包括：

a) 具有唯一标识的风险描述；

b) 风险评估的结果；

c) 风险优先级，例如高风险、中风险或低风险；

d) 风险处置措施；

e) 处置措施实施的跟踪状态。

4.7.4 在每次发布风险管理清单之前，应邀请所有相关方进行风险评估。

4.7.5 风险管理的交付物：风险清单。

4.8 配置管理

4.8.1 项目应在软件开发初始阶段制定及发布配置管理计划，并根据项目计划进展保持更新，该计划定义了用于管理配置项目及其接口的过程、方法和工具。

4.8.2 配置管理计划应包括以下内容：

a) 职责；

b) 基线标准；

c) 控制配置项目更改的过程；

d) 配置系统、工具和存储库；

e) 配置管理系统的位置和访问机制；

f) 规定的储存、处理和交付（包括文档和检索）机制；

g) 控制修改和发布的机制；

h) 定义命名约定；

i) 定义配置项存储年限及其备份恢复机制。

4.8.3 配置管理计划通常应支持处理产品/软件变体，这些变体可能是由不同的应用程序参数集或其他原因引起的。

4.8.4 配置管理计划应定义：

a) 产品发布分类和编号方案；

b) 构建环境（各方都应使用指定且一致的构建环境）；

c) 确定产品发布的内容；

d) 从配置的项目构建版本的方法。每个版本应从配置的项目构建，以确保完整性；

e) 定义和生成发布文档的方法；
f) 软件发布的传递机制和介质。

4.8.5 应识别配置项，定义配置项属性，记录并发布于配置项清单。

4.8.6 应建立配置管理机制（包括控制配置项并行修改的机制，如分支合并的策略、检入检出机制），用于控制已识别的配置项（含配置项属性），保证配置项的每个版本都根据定义的标准获得批准。

4.8.7 应根据配置管理计划建立基线，确保基线的完整性和一致性。

4.8.8 配置管理的交付物包括：
a) 配置管理计划；
b) 配置项清单。

4.9 评审管理

4.9.1 应建立评审流程及评审计划，对项目交付物及交付过程实施评审。
4.9.2 评审策划时应考虑：
a) 评审要素；
b) 评审成员；
c) 评审形式；
d) 准入准出标准。
4.9.3 评审可选择以下形式：
a) 检查；
b) 结构化走查；
c) 专项评审（基于标准、领域等）。
4.9.4 必要交付物可按以下原则选择：
a) 最关键的部分；
b) 用户使用最多的部分；
c) 有缺陷的成本最高的部分；
d) 最容易出错的部分；
e) 理解最少的部分；
f) 更改最频繁的部分。
4.9.5 应根据评审流程和计划对必要交付物及交付过程进行不同形式的评审。评审记录应根据约定向相关方报告，作为该项目评审活动的执行证明。
4.9.6 组织应将所有评审发现的问题进行汇总分析以发现共性问题，进行必要的过程改进，以避免问题重复发生。

4.10 问题管理

4.10.1 组织应建立问题管理流程，管理软件生命周期各阶段发生的问题。以唯一的标识，跟踪和推动问题的解决，并保证问题到根本原因的可追溯性。
4.10.2 项目中发生的问题宜包括，但不限于以下类型：
a) 管理类问题；
b) 技术类问题；
c) 安全类问题；
d) 复盘的问题；
e) 流程不符合。
4.10.3 宜按照不同的问题类型执行不同的问题状态模型。
4.10.4 为有利于问题的推进，应将问题列入问题清单，并记录问题的相关信息。

4.10.5　所有软件问题的当前信息应在软件版本说明中记录或引用，问题的信息应包含：

a)　问题描述（包括发生日期、版本信息、详细场景等）；

b)　优先级；

c)　负责人；

d)　原因及影响分析；

e)　整改措施及验证状态；

f)　截止日期；

g)　适宜时，宜包括复现环境（含软件版本、硬件版本、配置参数版本）。

注：问题优先级应基于问题的严重度、发生概率、探测度、易恢复度来判断。

4.10.5　当现有机制无法解决问题时，应启动升级流程，将问题提交至更高层级处理。

4.11　变更管理

4.11.1　项目应建立变更流程，以管理项目中的变更请求。以唯一的标识，在变更请求清单中记录和推动变更请求的关闭。如变更是由问题触发，需与问题唯一标识进行关联。

4.11.2　变更请求的来源通常有：

a)　对内部挖掘需求的变更；

b)　来自相关方或用户的需要；

c)　软件问题引发的变更。

4.11.3　变更请求的内容应包括：

a)　变更来源；

b)　变更前后变化点描述；

c)　变更影响的描述（详见4.7.7）；

d)　变更实施计划。

4.11.4　应对变更请求进行影响评估，并将评估结果提交给变更委员会，以便制定变更方案并进行决策。

4.11.5　影响分析宜包括，但不限于：

a)　技术或项目需求；

b)　突破项目或合同要求的影响；

c)　对发布计划的影响；

d)　成本影响；

e)　质量影响；

f)　功能性影响；

g)　系统资源影响（如：RAM、ROM、EEPROM内存、CPU负载和（静态）电流消耗等）；

h)　对关联模块/系统的影响；

i)　安全性影响；

j)　制造影响；

k)　环境影响；

l)　法律、法规影响。

4.11.8　应针对变更影响分析结果定义变更的优先级，变更实施的优先级通常可归类为：

a)　在当前版本基线紧急变更；

b)　在后续版本基线实施变更。

4.11.10　应制定变更验证和确认计划，以确保变更不引入新的问题。变更验证和确认宜包括变更功能验证和确认，相关功能回归验证及确认和性能验证及确认。

4.11.11　所有变更的当前和历史状态应在软件版本说明中记录或引用，软件版本应具有可追溯性。

5　过程质量保证

5.1 总则

为确保汽车软件质量安全，规避软件缺陷及召回风险，应在软件全生命周期实施质量保证活动。对软件开发的策划、需求分析、设计实现、集成、验证确认、释放管理、升级与维护等关键过程进行管理。

当软件采用瀑布、螺旋、增量、敏捷、双态等开发模式时，开发模式中的每个子循环仍然可参照策划、需求分析、设计实现、集成、验证和确认、释放管理、软件升级等关键过程的要求。

5.2 需求分析

5.2.1 项目应设计和构建一个适用于软件需求规格管理的体系。

5.2.2 项目需要管理所有软件需求，包括相关方需求和组织内部需求。

5.2.3 项目应建立外部需求与软件需求追溯性。

5.2.4 应通过分析保证所有软件需求的完整性，正确性和可验证性。

5.2.5 应基于软件需求识别和定义机器学习需求。

5.2.6 针对机器学习需求，应建立机器学习需求和软件需求之间的追溯性。保证对机器学习模型的测试能力。

5.2.7 应识别和说明机器学习需求相关的数据特征及其预期分布。

5.2.8 项目应根据软件需求和机器学习需求制定软件实现计划，以确定需求的交付节奏。软件实现计划的内容应和所有相关方达成一致。

5.2.9 项目应在每个软件交付中针对每个需求定义并维护实际的实施及验证状态。

5.2.10 软件需求分析的交付物包括：

a) 软件需求；
b) 机器学习业务需求；
c) 机器学习数据需求；
d) 软件需求追溯矩阵；
e) 软件实现计划。

注：对于实施敏捷的组织，产品待办列表可等同于软件需求，针对待办列表需要定义 DoR（就绪定义）和 DoD（完成定义）。

5.3 设计实现

5.3.1 软件设计实现

5.3.1.1 项目应基于明确的软件需求，设计软件架构。

5.3.1.2 对合理验证和部署过的软件重用可作为满足需求定义的方法。项目需要对软件重用的合理性和应用限制进行分析，并获得所有相关方的认可。对重用软件的管理应记录在软件实现计划中。

注 1：用于重用的软件可以是库文件，软件模块，软件组件，源代码（含开源代码）和商用现成产品等。

注 2：在使用开源代码的时候，需要评估许可证风险、安全风险、知识产权风险、法律风险等。如果发现存在相关风险，需及时处理，以使得开源软件的使用符合相关要求。

5.3.1.3 项目应依据需求定义来开发软件架构。识别、设计和文档化软件组件、软件组件间接口和软件的外部接口。软件应通过结构化方法逐层分解，直至不宜再分的软件项。

5.3.1.4 软件架构设计宜包括：

a) 描述整体软件结构（框图）；
b) 描述操作系统，包括任务结构；
c) 识别所需的软件项；
d) 识别软件项之间的关系和依赖关系；
e) 定义每个软件项之间的接口；
f) 识别内部和外部的代码作为重用软件的描述；
g) 动态行为描述；

h) 时序要求（如上电、下电等特殊场景）。

注：对于实施敏捷的组织：为避免技术债务以及返工的产生，可在项目初期开展必要的架构设计工作。

5.3.1.5 宜通过软件详细设计描述每个最低级别软件项和接口。软件项的大小和复杂度应得到控制。

5.3.1.6 软件详细设计应包括：

a) 结构化设计（可以表示为原型，流程图，实体关系图，伪代码等）；
b) 输入/输出数据的格式；
c) 相关软件组件/单元的动态行为；
d) 数据命名约定；
e) 异常处理机制。

5.3.1.7 应当确保需求定义与架构设计的一致性和双向可追溯性，并确保架构设计与软件详细设计的一致性和双向可追溯性。

5.3.1.8 应开发硬件/软件接口规范，该规范应明确软件应如何配置，访问和控制特定的硬件，硬件/软件接口规范应包括，但不限于：

a) 复位和看门狗定时；
b) 中断机制；
c) 内存映射；
d) 数字、模拟输入/输出；
e) 通信接口；
f) 显示接口。

5.3.1.9 应对软件性能进行分析，并将可能的极端结果编制成最坏情况分析报告。最坏情况分析报告应证明该软件能够在最坏的运行条件下满足软件需求。最坏情况分析报告应包括：

a) 中断延迟；
b) ISR 时序；
c) 微处理器（包括 CPU、GPU）的负载/带宽百分比-包括中断（标称和最坏情况）；
d) 最大/标称中断速率；
e) 调度程序计时和任务计时；
f) 微处理器负载；
g) 闪存大小（已用和可用）；
h) RAM 内存大小（已用和可用）；
i) EEPROM 内存大小（已用和可用）。

5.3.1.10 应制定并遵守软件开发使用语言的编码规范，以开发实现软件详细设计的软件组件。

5.3.1.11 应制定并遵守软件模型化开发的设计规范，以开发符合需求规格的软件模型。

5.3.1.12 应依照软件详细设计、编码规范或者软件模型，编写或生成符合要求的软件。

5.3.1.13 项目应制定软件单元验证计划，以证明软件设计的符合性。

注：单元验证的可能技术包括静态分析、代码审查、单元测试等。

5.3.1.14 项目应选用单元验证措施，并验证软件单元。

5.3.1.15 项目应对所有源代码进行基于工具的静态分析，以检查是否符合编码标准。

5.3.1.16 静态分析应明确包含以下内容：

a) 被测代码模块的唯一标识；
b) 使用的工具（包括所有版本信息）；
c) “运行”中包含的所有排除项的明细清单以及相关理由；
d) “运行”中发现的所有不合规情况的明细列表以及相关的理由或恢复计划；
e) 源代码典型错误，标准类、质量类、安全类。

5.3.1.17 验证软件单元是否满足软件详细设计所描述的功能及非功能性需求。

5.3.1.18 涉及安全的软件产品，应确保在设计和实现过程中考虑其模块化、封装性、可理解性、可维护性等基本特征。

5.3.2 机器学习设计实现

5.3.2.1 对于机器学习相关的软件，需开发包含机器学习要素的特定架构，包括预处理、后处理以及机器学习模型的超参数；超参数需明确阈值和初始值。

5.3.2.2 应确保机器学习架构与机器学习需求之间的一致性和双向可追溯性。

5.3.2.3 应定义机器学习的训练数据集，用于对机器学习模型进行训练，针对机器学习的实现部分，建议增加价值观对齐管控的关键环节。

5.3.2.4 应根据机器学习架构创建机器学习模型，并使用机器学习训练数据集对模型进行训练。

5.3.2.5 应确保机器学习训练数据集与机器学习数据需求之间的一致性和双向可追溯性。

5.3.2.6 验证过程的目的是验证机器学习模型在训练和部署之后具备集成条件

5.3.2.7 对机器学习模型的验证计划应包括：

a) 机器学习测试数据集中各场景的分布和频率;
b) 每个测试数据点的预期测试结果。

5.3.2.8 应确保应用机器学习测试数据集对已训练的机器学习模型和已部署的机器学习模型进行测试。

5.3.2.9 应确保机器学习数据需求与机器学习测试数据集之间的一致性和可追溯性。

5.3.2.10 涉及安全的机器学习相关软件，应确保机器学习架构开发过程中考虑其鲁棒性、可靠性、可控性、可解释性、可预测性等基本特征。

5.3.2.11 涉及安全的机器学习相关软件，应确保数据集开发过程的准确性、完整性、独立性、代表性、时效性等基本特征。

5.3.3 设计实现的交付物

交付物应包括：

a) 软件架构；
b) 软件详细设计；
c) 最坏情况分析报告；
d) 硬件/软件接口规格；
e) 软件模型设计规范；
f) 验证计划；
g) 单元验证措施；
h) 机器学习架构；
i) 机器学习训练数据集；
j) 机器学习模型；
k) 机器学习验证数据集；
l) 验证报告。

5.4 集成

5.4.1 软件集成的目的是将单独或已集成的软件项整合，直至成为与软件架构设计一致的完整软件。集成活动可分层逐步实行。

5.4.2 项目应当制定软件集成计划和集成验证计划, 集成策略应匹配所有的软件组件的集成。

5.4.3 将各软件组件集成到目标软件系统。

5.4.4 将已部署的机器学习模型集成到目标软件系统。

5.4.5 设计集成验证措施，然后验证集成的软件组件，以证明软件架构设计和软件详细设计。

注 1：硬件/软件接口规格是集成验证措施的输入。

注 2：特定的集成验证应证明系统项目之间的接口符合系统架构设计给出的规范。

注 3：系统集成验证可以使用环境模拟（例如硬件在环模拟、车辆网络模拟、数字样机）来支持。

注 4：对于实施敏捷的组织：软件集成应实现自动化集成，以及必要的自动化集成测试。

5.4.6 集成的交付物包括：

a) 集成包；
b) 集成计划；
c) 集成验证计划；
d) 集成验证措施和结果。

5.5 验证确认

5.5.1 项目应为每次软件交付开发并向利益相关方提供验证计划，验证计划应包括：
a) 验证的目的、验证的前提条件；
b) 验证的时间安排；
c) 验证的步骤；
d) 预期结果。

5.5.2 应根据软件的类型，选择合适的验证措施，比如模拟验证、台架验证、泛化验证等。

5.5.3 应确保软件需求与要使用的验证措施的唯一标识符的一致性和可追溯性。

5.5.4 应审查和分析所使用的验证措施，以确保验证措施对每一项待验证需求的充分性，并记录判断依据。

5.5.5 对机器学习相关软件应验证数据集是否符合数据集需求（包括安全需求），是否符合数据集的预期目标。

5.5.6 应对用于交付的软件执行全面验证，以确保软件在合理应用环境下的有效性。

5.5.7 应为每次软件交付制定验证报告，以总结所执行验证的总体范围和结果，确认本软件交付满足客户预期。

5.5.8 验证报告应包括：
a) 验证日志（项目、使用的软件/硬件版本、测试执行日期、负责的测试工程师）；
b) 验证指标（如执行/未执行/失败的用例数量）；
c) 详细验证结果/判断依据；
d) 验证结论。在摘要中提供验证措施执行的所有必要信息，使相关方能够判断结果；
e) 适用时，定义行动建议或计划（如果有）。

5.5.9 验证和确认的交付物包括：
a) 验证计划；
b) 验证措施；
c) 验证报告。

5.5.10 接收软件交付物时，接收方应对交付物进行验证，检查交付物是否符合预期目标和质量安全要求，并及时与软件提供方沟通验证情况。必要时可进行风险评估。

软件安全验证方向应包含但不限于以下事项：
a) 软件组件完整性与防篡改验证：验证软件交付物未被篡改、植入恶意代码，完整性校验值与官方一致，保障软件本体安全；
b) 算法决策逻辑与采信依据验证：验证算法的感知、融合、决策逻辑的合理性，核查验证数据集、采信标准是否全面，测试环境、场景、评价标准是否与实车运行场景相适应；
c) 功能安全与异常容错验证：验证软件在输入异常、硬件故障、环境干扰下的容错能力，是否会触发非预期车辆行为；
d) 网络安全攻防与漏洞验证：针对车载软件开展漏洞挖掘、渗透测试，验证是否存在被远程入侵、控制的风险；
e) 数据安全与隐私合规验证：验证软件采集、传输、存储车辆及用户数据的安全性，符合数据安全法规要求。

5.6 释放管理

5.6.1 项目应制定释放计划，针对每次释放确定其目的并定义释放标准、识别所要求的文档和释放对象。

5.6.2 应准备释放要求的相关文档，以反映当前需释放的交付软件的真实状态。
5.6.3 应依据上述文档对每个交付软件进行释放评审。
5.6.4 针对释放评审发现的问题，项目立即采取纠正措施。
5.6.5 应针对评审通过待交付的软件版本创建软件版本说明，用于描述本次交付软件的情况。
5.6.6 软件版本说明应包括：

a) 构建环境，例如工具及其版本；
b) 软件版本号；
c) 硬件部件号；
d) 资源包信息（如有）；
e) 需求实现和验证状态；
f) 现存问题；
g) 硬件/软件的依赖项；
h) 对相关方的影响。

5.6.7 所有交付软件均应至少附有以下文档，所有这些文档都必须反映所交付的软件：

a) 软件版本说明；
b) 软件验证报告；
c) 每个需求定义的开发和验证状态。

5.6.8 应按照项目计划约定的交付日期，以软件实现计划定义的约定状态交付软件。交付的软件应已通过约定的验证活动得以确认。交付应在指定位置以商定的格式完成。
5.6.9 释放管理的交付物包括：软件包、资源包（如果有）、软件版本说明、释放评审报告。

5.7 升级与维护

5.7.1 项目应制定软件维护计划。该计划应详细说明项目应如何确保必要的人员和设备，以便在软件的整个生命周期内支持软件维护活动。
5.7.2 应决定维护的职责分工，如是否应由软件开发组织或第三方进行支持和维护，必要时签订服务协议。
5.7.3 软件的生命周期中应定义维护计划。
5.7.4 应按照软件维护计划在生命周期对软件进行维护，包括必要的软件升级。
5.7.5 软件升级及维护的交付物：升级计划、软件维护计划。

6 关键过程评审

6.1 总则

为有效对软件质量和缺陷进行过程管理，应在汽车软件实现过程的 5 个关键过程开展评审活动，分别是：软件项目策划、需求分析、设计实现、软件集成、验证确认。

关键过程评审的评审组人员构成宜包括：

a) 软件质量经理；
b) 软件项目管理经理；
c) 软件集成经理；
d) 软件架构师；
e) 功能安全专家、预期功能安全专家、信息安全专家、人工智能安全专家、数据安全专家。

关键过程评审可根据软件版本的迭代周期按以下规则做适当合并：

a) 迭代周期不大于1个月：软件项目策划和需求分析评审合并开展，设计实现、软件集成和验证确认评审合并开展；
b) 迭代周期在1至3个月之内：软件项目策划和需求分析评审合并开展，设计实现评审单独开展，软件集成和验证确认评审合并开展；

c) 迭代周期在3个月及以上：5个过程独立开展。

6.2 关键过程评审对象

关键过程评审的对象是软件开发过程中每个阶段的交付物和软件成熟度，评审结果将作为判断软件是否能顺利进入下一个阶段的决策依据。

关键过程的交付物宜包括，但不限于：

a) 软件开发与测试计划；
b) 软件质量管理计划；
c) 软件配置管理计划；
d) 软件进度报告；
e) 软件需求规范；
f) 软件架构设计；
g) 软件详细设计；
h) 软硬件接口规范；
i) 最坏情况分析报告；
j) 软件集成策略；
k) 软件测试用例；
l) 软件测试报告；
m) 软件版本说明；
n) 风险问题清单；
o) 软件质量报告；
p) 软件配置管理报告；
q) 变更控制记录；
r） 软件安全管理报告；
s） 安全管理计划。

注：针对软件配置管理、问题管理、风险管理、变更管理和安全管理的决策分析与解决贯穿整个软件全生命周期，在每个关键节点都须对相关内容进行评估。

6.3 软件项目策划过程评审

在项目策划阶段结束前，对软件的迭代计划，迭代目标，质量保证计划，测试计划，配置计划等进行评估，确保项目迭代目标和计划设置合理，和所有相关方达成一致。

6.4 需求分析过程评审

在需求分析阶段冻结前，对需求的挖掘分析，分解分配，评估握手，完整性和一致性进行评估，确保所有需求（包括功能性需求及和非功能性需求）已完成评估和分解工作，所有相关方对需求的理解一致无偏差。

6.5 设计实现过程评审

设计实现阶段结束前，对架构设计，详细设计，软硬件接口，源代码，测试用例，测试策略等进行评估，确保所有需求的设计文档、编码，以及测试用例已按照需求规格中的要求完成设计工作。

对于融合安全要求的需求定义须开展专项评审，包括对功能安全、预期功能安全、信息安全、人工智能安全、数据安全的分析文档。

6.6 软件集成过程评审

软件集成阶段结束前，对集成策略，集成结果等进行评估，确保各软件项的集成符合软件架构设计和软件详细设计，集成测试结果符合测试准出条件。

6.7 验证和确认过程评审

验证和确认结束前，对测试报告，软件版本说明，迭代目标和交付范围完成情况等进行评估，确保软件所有需求满足功能性、非功能性以及安全设计的要求，满足客户的期望，满足软件释放的条件。

6.8 评审结果

依据项目质量目标达成状态和关键过程评审风险清单产生以下评审结果：

a) 通过：整体按计划交付无风险，可直接进入下一个阶段；
b) 偏差通过：整体存在一定的延期或质量问题，无高风险，允许存在中低风险，但中风险有影响分析和改进措施，评估组成员经过充分评估后达成一致可以带风险进入下一个阶段；
c) 不通过：整体存在重大延误或高风险，会影响整车交付或整车质量，需要立即采取改善措施并重新评估通过后才能进入下一个阶段。

6.9 措施及验证

针对评估报告中的不符合项和风险项制定改进措施，确定责任人和关闭日期，验收措施改进效果，直至所有不符合项和风险项关闭。

7 软件风险评估

7.1 总则

软件问题在不同的系统及运行环境下，会对系统的质量安全产生不同的影响，本章节规定了汽车软件产品问题对系统或整车的质量安全影响的风险评估方法及风险分级的对应关系，以保证软件的关键问题或缺陷能得到充分的风险评估，并及时采取相应的风险消除措施。通过持续的风险评估，实现软件缺陷风险前置与预防控制，提升汽车软件质量，降低交付风险，降低各类安全事件发生的概率。

涉及信息缺陷的评估与认定参考 GB/T 45496。

7.2 风险评估对象

软件风险评估对象宜包括，但不限于以下环节发现的软件问题：

a) 设计开发环节中的软件问题：在汽车软件项目过程中，方案策划、需求评审、技术选型、架构设计、设计实现、代码生成等过程评审活动及关键节点评审中识别到有关软件的质量问题；
b) 测试验证环节的软件问题：指在软件研发过程中，软件单元静/动态测试、集成测试、系统测试、验收测试等活动及关键节点评审中所暴露的软件质量问题；
c) 生产制造环节的软件问题：在汽车生产制造过程中，软件封装、测试、产线的EOL检测、整车测试等活动中所暴露的软件质量问题；
d) 产品使用环节的软件问题：在产品使用过程中，从各种数据、资料、报表、文件、情报收集的有关软件的质量问题；包括但不限于市场索赔信息、用户反馈、网络舆情、网上投诉、市场质量信息报告、政府监管部门以及第三方机构反馈信息等；
e) 接收交付物时发现的软件问题：在接收软件交付物时，对交付物进行验证过程中检查出来的软件质量问题。

7.3 软件风险评估流程

为确保软件问题导致的质量安全风险得到充分识别和修复，软件风险评估应基于以下流程：

a) 软件对系统影响的评估：确定软件问题对系统或整车可能产生的系统风险；
b) 系统风险评估：根据系统风险的严重性和可能性进行综合评估，确定风险等级；
c) 风险控制：根据风险等级制定相应的对策方案；
d) 效果评估：跟踪措施实施情况和验证措施的有效性。

7.4 软件对系统影响的评估

软件问题在不同系统和运行环境下，对系统风险有不同影响。软件问题应基于在系统运行环境下对系统或整车层面的影响进行评估。

注：系统风险评估包括软件产品的部署、预期使用和可预见的滥用对个人或群体或两者以及社会的潜在后果的评估。

7.5 系统风险评估

7.5.1 系统风险严重性评估

软件问题导致的系统风险严重性按照以下风险等级说明进行分级：

a) 高：具有突发性，且不可控，可能会严重危及人身、财产安全、严重影响产品使用性能和降低产品寿命、造成重大经济损失、不符合安全、环保等法规要求,以及必然会引起用户重大投诉的问题；

b) 较高：具有突发性，且可控性降低，可能危及人身、财产安全，会造成顾客强烈不满；导致基本功能缺陷的问题；

c) 中：造成车辆行驶性能或功能下降，但可控，车辆有可能继续使用，若继续使用可能会导致高、较高的严重性等级，对顾客满意度有重要影响的问题；

d) 较低：对车辆行驶性能或功能有部分影响，但可控，车辆可继续使用，若继续使用可能会导致较高、中的严重性等级：对顾客满意度有一定影响的问题；

e) 低：对车辆安全性无直接影响：大多数情况下不需要维修或可接受的问题；对顾客满意度影响极低的问题。

7.5.2 系统风险可能性评估

软件问题导致的系统风险可能性按照以下风险等级说明进行分级：

a) 高：该问题必现；

b) 较高：该问题频发；

c) 中：该问题偶发；

d) 较低：该问题发生概率较低；

e) 低：该问题在整车发生概率极低。

7.5.3 确定综合风险水平等级

在软件质量问题导致的系统或整车质量安全风险严重性等级和发生可能性等级确定的基础上，按GB/T 34402 确定软件问题的综合风险水平等级，综合风险水平等级分为 5 级：

a) 高 （第5级）；

b) 较高（第4级）；

c) 中 （第3级）；

d) 较低（第2级）；

e) 低 （第1级）。

可能性	严重性				
	低	较低	中	较高	高
低	1	2	2	3	3
较低	2	2	3	3	4
中	2	3	3	4	4
较高	3	3	4	4	5
高	3	3	4	5	5

图 2 风险评估矩阵

7.6 风险控制

7.6.1 风险控制对象软件版本已经应用于在售汽车产品时，汽车产品生产者应协同软件提供方，根据综合风险水平等级制定相应的风险控制策略与措施：

a) 综合风险水平等级为高（第5级）和较高（第4级）的，应根据相应的法律法规实施召回活动，消除车辆安全隐患；

b) 综合风险水平等级为中（第3级）的，通过分析国内外相关的召回案例，若存在类似召回案例的，应根据相应的法律法规实施召回活动；若没有类似召回案例，可自主处置；

c) 综合风险水平等级为较低（第2级）和低（第1级）的，可自主处置，如进行风险警告、提供修复该问题的新软件供升级或其他风险控制措施。

7.6.2 风险控制对象软件版本未应用于在售汽车产品或处于软件开发过程中时，汽车产品生产者应协同软件提供方，基于项目开发进度根据综合风险水平等级制定相应的风险控制策略与措施：

a) 综合风险水平等级为高（第5级）和较高（第4级）的：立即修复；

b) 综合风险水平等级为中（第3级）的：根据项目实际情况下一阶段进行修复；

c) 综合风险水平等级为较低（第2级）和低（第1级）的：软件释放前进行释放评审，基于释放评审进行修复。

7.7 措施有效性评估

应对软件风险控制的措施有效性验证、措施实施范围、措施实施计划、措施实施效果、措施的次生影响和相关措施进行管理与控制，保证风险控制效果。

8 软件缺陷管理

8.1 总则

汽车产品生产者应根据多个渠道收集和分析市场信息、顾客反馈、产品技术信息、政府产品安全监管信息等，按照第 7 章所述风险评估方法开展软件质量调查分析与评估，并根据评估结果制定解决措施与对策，进行软件缺陷认定与召回决策。软件提供方在上述过程中应配合汽车产品生产者开展相关工作。涉及信息缺陷的处置参考 GB/T 45496。

8.2 信息收集与分析

汽车产品生产者应建立信息收集与分析的渠道与工作程序，明确主要信息来源及内容，并对信息进行分类和综合性分析。信息来源包括但不限于法律法规要求、最终产品市场信息、服务热线和投诉信息、舆情监测信息、重大事故类信息、国内外同类车型召回信息、生产者内部信息、供应商报告信息、政府监管信息等。汽车产品生产者召回管理组织针对软件缺陷，应建立软件缺陷信息收集平台。

8.3 软件问题识别

汽车产品生产者召回管理组织应基于信息分析提取的共性问题，识别软件问题，从风险严重性、可能性等维度进行初步判断。软件提供方发现软件问题的，应主动向汽车产品生产者报告软件问题，进行问题识别和判断。

8.4 调查分析与评估

8.4.1 汽车产品生产者召回管理组织应确定软件提供方的调查分析责任部门，监控与督促调查分析过程，并对分析结果进行评估。

8.4.2 调查分析责任部门应基于软件问题表象，通过对开发过程包括需求分析、设计开发、集成、验证等活动的追溯，明确问题发生的根因和影响范围。

8.4.3　调查分析责任部门应按照第7章所述方法对软件问题进行风险评估，并明确风险控制措施和有效性评估。

8.5　召回决定

汽车产品生产者召回管理组织应审议调查分析结果，做出召回决定并形成记录：

a)　认定产品存在缺陷的，汽车产品生产者按相应的法律法规要求组织相关方开展召回活动，消除车辆安全隐患；

b)　认定产品不存在缺陷的，汽车产品生产者可采取服务公告、自愿性OTA升级等方式组织软件修复活动，向用户告知问题及解决方案，并根据相应的法律法规要求进行备案。

8.6　修复软件的准备

软件提供方应基于需求分析、设计实现、预验证、集成、验证和确认等过程准备软件修复，为确保软件修复质量应按照本规范进行充分的过程和产品风险评估和验证。

汽车产品生产者应进行 OTA 或返厂刷写准备，需要更换零件时，应进行零件准备。

8.7　召回计划制定

在决定实施召回后，汽车产品生产者应制定召回计划。召回计划材料的内容根据召回决策资料的内容进行整理，其中故障现象、缺陷原因、可能导致的后果等要与缺陷认定和召回决策的内容保持一致。召回计划应包含以下内容：

a)　召回车辆信息；
b)　软件缺陷描述信息；
c)　技术说明文件；
d)　软件缺陷补救措施；
e)　召回日程；
f)　召回的预期效果；
g)　软件问题修复指导；
h)　召回费用预算。

8.8　召回计划报告

汽车产品生产者应在确认产品存在缺陷之日起 5 个工作日内向市场监管部门报告召回计划，报告材料应符合市场监管部门的要求，并确保材料的客观准确性和完整性。

8.9　召回信息通知

汽车产品生产者实施召回，应以有效方式通报经营者；自完成召回计划报告之日起 5 个工作日内，以便于公众知晓的方式发布缺陷汽车产品信息和实施召回的相关信息；30 个工作日内以有效方式，告知车主汽车产品存在的缺陷、避免损害发生的应急处置方法和生产者消除缺陷的措施等事项。

8.10　召回过程管理

汽车产品生产者应保持与召回相关方的沟通，发出明确指示，并在召回备案的实施日开始针对特定范围的产品进行软件释放，进行 OTA、返厂刷写或更换零件。

汽车产品生产者按照法律法规要求向市场监管部门提交召回阶段性报告和召回总结报告。

8.11　召回效果评估

汽车产品生产者应该对召回效果进行评估，评估方法按照GB/T 39603执行。

参 考 文 献

[1] GB/T 11457—2006 信息技术 软件工程术语
[2] GB/T 43387—2023 产品召回 术语
[3] ISO/IEC/IEEE 24765:2010 系统与软件工程 生命周期过程 术语