



汽车软件标准体系研究报告

全国汽车标准化技术委员会
智能网联汽车分技术委员会
资源管理与信息服务标准工作组
2023年11月

目 录

前言.....	IV
特别鸣谢.....	V
1 汽车软件标准体系建设思路.....	1
2 汽车软件标准体系研究内容.....	2
2.1 操作系统.....	2
2.1.1 车载操作系统.....	2
2.1.1.1 发展历程.....	2
2.1.1.2 特点.....	4
2.1.1.3 发展现状.....	5
2.1.1.4 标准需求.....	6
2.1.1.5 标准体系框架及分类说明.....	8
2.1.2 车控操作系统.....	12
2.1.2.1 发展历程.....	12
2.1.2.2 特点.....	13
2.1.2.3 发展现状.....	14
2.1.2.4 标准需求.....	17
2.1.2.5 标准框架及分类说明.....	19
2.1.3 系统服务中间件.....	20
2.1.3.1 发展历程.....	20
2.1.3.2 特点.....	21
2.1.3.3 发展现状.....	22
2.1.3.4 标准需求.....	24
2.1.3.5 标准体系框架及分类说明.....	26
2.2 应用软件.....	27
2.2.1 车控域应用软件.....	29
2.2.1.1 发展历程.....	29
2.2.1.2 特点.....	29
2.2.1.3 发展现状.....	30
2.2.1.4 标准需求.....	36
2.2.1.5 标准体系框架及分类说明.....	37
2.2.2 智驾域应用软件.....	41
2.2.2.1 发展历程.....	41
2.2.2.2 特点.....	42
2.2.2.3 发展现状.....	43
2.2.2.4 标准需求.....	44
2.2.2.5 标准体系框架及分类说明.....	44
2.2.3 座舱域应用软件.....	46
2.2.3.1 发展历程.....	46
2.2.3.2 特点.....	47
2.2.3.3 发展现状.....	48

2.2.3.4 标准需求	48
2.2.3.5 标准体系框架及分类说明	49
2.2.4 整车应用软件	50
2.2.4.1 发展历程	50
2.2.4.2 特点	52
2.2.4.3 发展现状	52
2.2.4.4 标准需求	53
2.2.4.5 标准体系框架及分类说明	54
2.2.5 手机及网联应用软件	57
2.2.5.1 发展历程	57
2.2.5.2 特点	58
2.2.5.3 发展现状	59
2.2.5.4 标准需求	62
2.2.5.5 标准体系框架及分类说明	62
2.2.6 云端应用软件	65
2.2.6.1 发展历程	65
2.2.6.2 特点	67
2.2.6.3 发展现状	70
2.2.6.4 标准需求	72
2.2.6.5 标准体系框架及分类说明	72
2.3 数据服务软件	74
2.3.1 汽车数据服务现状	74
2.3.1.1 概述	74
2.3.1.2 采集现状	75
2.3.1.3 开发现状	77
2.3.1.4 功能现状	78
2.3.2 汽车数据服务标准需求分析	81
2.3.2.1 汽车数据服务采集标准现状分析	81
2.3.2.2 汽车数据服务开发标准现状与需求分析	88
2.3.2.3 汽车数据服务功能标准现状与需求分析	88
2.3.2.4 汽车数据服务通信与安全标准现状与需求分析	88
2.3.3 数据服务软件标准体系框架及分类说明	91
2.4 工具软件	92
2.4.1 概述	92
2.4.2 需求分析软件	95
2.4.2.1 发展历程	95
2.4.2.2 特点	96
2.4.2.3 发展现状	97
2.4.2.4 标准需求	97
2.4.2.5 标准框架及分类说明	98
2.4.3 系统设计软件	99
2.4.3.1 发展历程	99
2.4.3.2 特点	99
2.4.3.3 发展现状	100

2.4.3.4 标准需求	101
2.4.3.5 标准框架及分类说明	102
2.4.4 开发软件	103
2.4.4.1 发展历程	103
2.4.4.2 特点	104
2.4.4.3 发展现状	104
2.4.4.4 标准需求	105
2.4.4.5 标准框架及分类说明	106
2.4.5 测试验证软件	106
2.4.5.1 发展历程	106
2.4.5.2 特点	107
2.4.5.3 发展现状	108
2.4.5.4 标准需求	109
2.4.5.5 标准框架及分类说明	110
2.4.6 项目管理软件	110
2.4.6.1 发展历程	110
2.4.6.2 特点	111
2.4.6.3 发展现状	112
2.4.6.4 标准需求	113
2.4.6.5 标准框架及分类说明	113
2.4.7 后市场工具软件	114
2.4.7.1 发展历程	114
2.4.7.2 特点	115
2.4.7.3 发展现状	116
2.4.7.4 标准需求	117
2.4.7.5 标准框架及分类说明	119
2.5 质量与度量	119
2.5.1 研究背景	119
2.5.1.1 汽车软件质量模型	120
2.5.1.2 汽车软件过程能力度量	134
2.5.2 汽车软件质量与度量类标准需求分析	143
2.5.2.1 汽车软件质量模型	143
2.5.2.2 汽车软件过程能力度量	144
2.5.3 标准体系及工作建议	146
3 附录	149

前言

为贯彻落实《汽车产业中长期发展规划》和《深化标准工作改革方案》要求，深化推进《国家车联网产业标准体系建设指南（智能网联汽车）》（2023版）的规划，满足市场和创新发展需要，聚焦汽车软件前瞻、重叠、空白领域，建立适应我国技术和产业发展需要的、先进的、合理的、可持续进步的汽车软件标准体系，构建跨领域协同的智能网联汽车软件技术标准体系，发挥标准的引领性和规范性作用，推进智能网联汽车产业基础高级化、产业链现代化，建设社会主义现代化智能网联汽车强国。基于我国智能网联汽车产业发展现状，加强汽车软件标准体系研究，按照前瞻布局、急用先行的思路，以汽车产品为核心，整车厂牵头，协同产业链上下游合作单位共同制定汽车软件标准体系框架。共识决策可以凝聚各方力量，共同推动这些技术的研发和应用。通过共同努力，可以加快技术的成熟度和推广速度，促进行业的创新和发展。

在本研究项目编制过程中，各起草单位参阅了大量资料，并借鉴了行业的部分素材，鉴于篇幅有限，这里不一一列举，仅作诚挚的感谢！

在此，再次衷心感谢参与研究报告编写的特别评审专家、单位、组织及个人。

特别鸣谢

在本报告完成后，来自高校和行业的八位权威专家作为特别评审为本报提出了宝贵的意见做了指导，他们是：高振海、周时莹、孙航、李丰军、张春晖、杨国青、杜强、朱红儒。

对他们的贡献，项目组表示衷心感谢！

组织指导：全国汽车标准化技术委员会智能网联汽车分技术委员会

参与单位：中国第一汽车集团有限公司、中国汽车技术研究中心有限公司、中国软件评测中心（工业和信息化部软件与集成电路促进中心）、东软睿驰汽车技术（沈阳）有限公司、华为技术有限公司、中兴通讯股份有限公司、重庆长安汽车股份有限公司、北京汽车研究总院有限公司、上海（商汤）临港绝影智能科技有限公司、大众汽车（中国）投资有限公司、斑马网络技术有限公司、泛亚汽车技术中心有限公司、赛迪（浙江）汽车检测服务有限公司、国汽智控（北京）科技有限公司、吉利汽车研究院（宁波）有限公司、小米汽车科技有限公司、东风汽车集团有限公司技术中心、阿里云计算有限公司、米其林（中国）投资有限公司、上海机动车检测认证技术研究中心有限公司、博世汽车部件（苏州）有限公司、上汽大通汽车有限公司、广州汽车集团股份有限公司汽车工程研究院、中国汽车工程研究院股份有限公司、中电金信软件（上海）有限公司、中科软科技股份有限公司、维克多汽车技术（上海）有限公司、北京赛目科技股份有限公司、国家新能源汽车技术创新中心、北京车和家汽车科技有限公司、北京地平线信息技术有限公司、智马达汽车有限公司

编写人员：高长胜、吴嘉杰、刘斌、王伟、苏煜、李岩、吴含冰、祁帅、马涛、陈晓、鞠伟男、宁友良、王冠翎、赵洋、李雪萍、赵秋俊、王勇、崔长军、吕汉伟、王琳、缪敬、李栋、王禹寒、王宁、张志运、白智敏、谢正天、徐宗雷、梁海强、孔晓霜、郭茜茜、刘曰、杨小明、李政、范开伟、余来星、张静

1 汽车软件标准体系建设思路

智能网联汽车软件正处于快速发展阶段，在操作系统软件、应用软件、数据软件、工具软件和质量与度量上都涌现出诸多的创新技术，需要制定相应的标准和规范，来提高汽车软件整体的安全性、互操作性和可复用性。

汽车软件标准体系的总体思路为：适应我国智能网联汽车在新发展阶段的新趋势新需求，围绕智能化和网联化两大主线创新协同发展，兼顾产品研发、生态建设和用户需求，同步推进技术创新发展和基本安全保障，统筹国内国际标准法规制定协调，构建科学合理、开放创新、协调兼容的汽车软件标准体系，该标准体系总体分为 5 大核心部分，分别是操作系统软件、应用软件、数据软件、工具软件和质量与度量。该标准技术架构以操作系统软件为载体，上层部署应用软件和数据软件，覆盖智控、智驾、智享和云端的应用软件，再通过工具软件和质量度量进行支撑，完整呈现汽车软件标准体系的技术逻辑，其标准体系逻辑框架如图 1、图 2 所示：



图 1 汽车软件标准体系技术逻辑框架

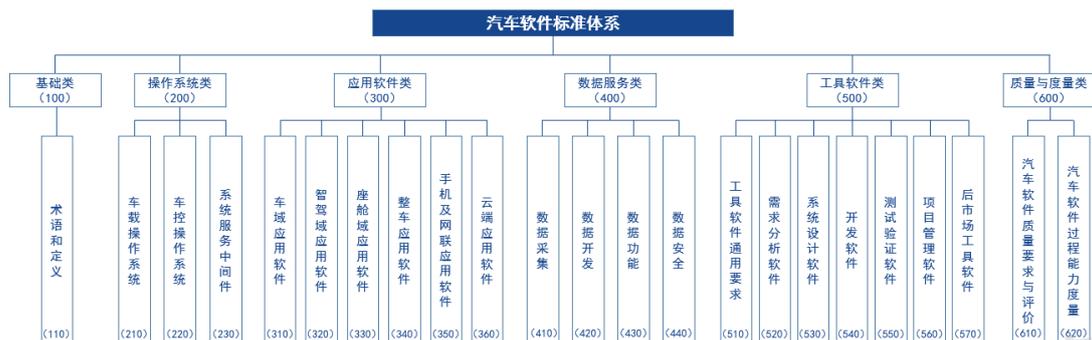


图 2 汽车软件标准体系树状架构

2 汽车软件标准体系研究内容

2.1 操作系统

2.1.1 车载操作系统

2.1.1.1 发展历程

车载操作系统是指运行于车载芯片上，管理和控制智能网联汽车车载软件、硬件资源的软件集合，为智能网联汽车提供除驾驶自动化功能实现以外的服务，包括车载信息娱乐、网联、导航、多媒体娱乐、语音等人机交互基础服务，以及 AR (Augmented Reality) 导航、盲区检测，驾驶员疲劳检测，FaceID (人脸识别)，手势识别、语音图像理解等人工智能服务。

车载操作系统是技术进步对车辆智能化的体现。最初的汽车只有单独的电话/收音机/CD 播放器，逐步采用音频解码芯片实现音乐的智能控制，此时还是以模拟电路到单片机系统开发过程，并未嵌入真正的车载操作系统。从可触摸液晶屏代替播放器开关、调节按钮实现数字化开始，由于增加了蓝牙电话功能、又集成了地图导航、倒车雷达影像等复杂的软件功能，车载操作系统开始被广泛应用，主要的车载操作系统包括 Windows CE、QNX、Linux 和 Android。

1995 年，微软开始研发 Windows CE 的新操作系统主要面向嵌入式和移动平台。Windows

CE 由于在移动端的优秀表现，因此微软也将其应用到汽车端，推出了 windows CE 汽车版 (Windows CE for Automotive)，windows CE 汽车版于 1998 年首次亮相，应用于福特，起亚等汽车中。但是随着微软在移动端落寞，2014 年 12 月，微软长期合作的福特公司宣布放弃 windows CE 改用 QNX，windows CE 汽车版也随即逐步告别历史舞台。

QNX 成立于 1980 年，是全世界第一个类 UNIX 的符合 POSIX 标准的微内核的硬实时操作系统，在过去的几十年中被广泛应用于汽车、工业自动化、国防、航空航天、医疗、核电和通信等领域，提供以嵌入式操作系统为核心的中间件和基础软件解决方案。1999 年 QNX 进入车载汽车电子目前仅中控娱乐系统已经累积出货了 1.75 亿台。在车载操作系统市场，QNX 操作系统占据 50% 以上的市场份额。

Linux 是一个基于 POSIX 和类 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统内核，遵循 GPL (GNU General Public License) 开源协议。2012 年 9 月，Linux 基金会联合 Intel、Toyota、三星、英伟达等多家企业推出了汽车端开源车联网系统 Automotive Grade Linux (AGL, 汽车级 Linux)。AGL 是一个协作开源项目，它将汽车制造商、供应商和技术公司聚集在一起，为汽车应用构建了一个基于 Linux 的开放软件平台，并允许汽车制造商和供应商重复使用相同的代码库，提升产品研发效率。AGL 最初专注于信息娱乐系统，目前已经开源的系统项目包括信息娱乐、仪表盘、平视显示器 (HUD)、远程信息处理/联网汽车、高级驾驶辅助系统 (ADAS)、功能安全和自动驾驶等，涵盖车辆中所有软件的组织。AGL 开发项目的参与者包括了丰田、铃木、戴姆勒、福特、本田、捷豹路虎、马自达、三菱汽车、日产、斯巴鲁等众多全球知名车企，当前已经有超过 150 家成员单位。

Android 具有开源开放、良好的开发环境，并且在手机端积累了大量的应用生态，在车载娱乐领域具有得天独厚的优势。Android 系统最早是基于 Linux 内核打造。2008 年，谷歌发布了 Android 1.0。Android 系统在汽车领域的尝试，从 2011 年就开始，是 Parrot 基于 Android 系统做的汽车音响系统——Asteroid。2014 年，谷歌联合奥迪、通用、现代和本田等几家全球主要汽车制造商，以及英伟达，在 CES 上宣布成立开发汽车联盟，旨在生产基于 Android 的车载娱乐

系统。2014年6月，谷歌发布了Android Auto车载操作系统。2019年，谷歌在Android Auto的基础上，进一步推出Android Automotive OS，定位于专业的车载信息娱乐系统。

2.1.1.2 特点

车载操作系统主要为车载信息娱乐服务以及车内人机交互提供控制平台，随着人们对车辆由单纯交通工具向智能移动终端转变的需求，车载操作系统需要支持更多样化的应用与服务，并且具有丰富的生态资源。

(1) 支持多样化应用

支持多样化的应用已经成为车载操作系统的重要指标，车载操作系统需要支持多样化功能实现。目前，汽车座舱除了仪表显示、空调/车窗控制等传统功能外，已经开始集成支付、娱乐、导航、信息服务等多样化的功能，在线电台、在线音乐、美食景点等互联网资源纷纷在车内向用户呈现。车载操作系统支持 TensorFlow、MXNet、Caffe、PyTorch 等多种机器学习框架所训练的人工智能模型，提供满足多种机器学习框架训练的人工智能模型的高效执行环境。其中，人工智能模型包括符合车载应用场景需要的语音识别、自然语言处理、图像分割和识别、搜索推荐等。

(2) 多生态资源

越来越多的车载操作系统采用 Android 或其他类 Linux 系统的原因是便于应用程序移植。目前手机端已经具有十分庞大的信息娱乐服务生态资源，通过采用相同或类似的操作系统，直接将功能移植到车辆智能终端上，能够避免重复开发并且快速丰富车端生态。

(3) 信息安全

智能座舱的使用不仅关乎用户的个人隐私安全与财产安全，同时也通过车内网络与底盘控制、自动驾驶等车控系统相连通。因此，车载操作系统不能简单地将手机操作系统复制到车端，而应通过深度定制达到车辆信息安全的标准。

2.1.1.3 发展现状

在车载操作系统领域，目前还没有统一的国际标准，车载操作系统主要包括 QNX、诸多基于 Linux 的定制操作系统以及基于 Android 开源项目的操作系统（其本身也基于 Linux）。

(1) QNX: 非开源、安全实时

QNX 是一款微内核、嵌入式、非开源、安全实时的操作系统。QNX 是微内核架构，内核一般只有几十 KB，驱动程序、协议栈、文件系统、应用程序等都在微内核之外的、受内存保护的内存空间内运行，可实现组件之间相互独立，避免因程序指针错误造成内核故障。因其内核小巧，运行速度极快，具有独特的微内核架构，安全和稳定性高，不易受病毒破坏系统，是全球首款通过 ISO 26262 ASIL-D 安全认证的实时操作系统，常用于安全稳定性要求较高的数字仪表中，已匹配全球超过 45 个汽车品牌，并应用于 1.75 亿辆汽车。QNX 在车载操作系统市场的占有率超过 75%，在更注重生态和内容的车载娱乐系统占有率也超过 60%。不过，QNX 的开放性不足导致应用生态缺乏。

(2) Linux: 开源、功能强大

作为一款开源、高效、灵活、功能强大的操作系统，Linux 的最大优势是具备很强的定制开发灵活度。例如，特斯拉在 Linux 基础上开发出了完全适配旗下车辆的车载系统；阿里的 AliOS 也是基于 Linux 开发，目前已经应用在上汽荣威、上汽名爵等多款车型上。2014 年，Linux 基金会赞助并发布了开源 AGL (Automotive Grade Linux) 规范 1.0 版本，它是首个开放式车载信息娱乐软件规范。AGL 是一个协作开源项目，由 Linux 基金会管理，将汽车制造商、供应商和科技公司聚集在一起，以加速开发和使用完全开放的智能网联汽车软件堆栈。AGL (Automotive Grade Linux) 目前最新发布代码版本库 UCB14。UCB 14 包括操作系统、板级支持包、中间件、应用程序框架和应用程序 API。

(3) Android: Linux 的发行版

Android系统是基于Linux内核开发的最成功的产品，开源，定制灵活，应用可移植性强，应用生态丰富。目前，国内厂家在车载信息娱乐应用中主要采用Android系统，尤其是各大互联网巨头、自主品牌和造车新势力纷纷基于Android进行定制化改造，推出自己的汽车操作系统，例如，百度小度车载OS、比亚迪DiLink、蔚来NIOOS、小鹏XmartOS等。

表 1 国内汽车行业相关研究成果

序号	名称	机构	状态
1	《智能网联汽车 车载操作系统技术要求及试验方法》	NTCAS	撰写中
2	《智能网联汽车接口标准体系研究》	NTCAS	撰写中
3	《车载操作系统架构研究》	NTCAS	已发布
4	《车载操作系统总体技术要求研究》	NTCAS	已发布
5	《车用操作系统测试评价研究》	NTCAS	已发布
6	《智能网联汽车新型电子电气架构标准化需求研究》	NTCAS	已发布

2.1.1.4 标准需求

车载操作系统主要应用于车机中控系统，对于安全性和可靠性的要求处于中等水平，该类操作系统发展迅速，种类繁多，操作系统的生态分裂比较严重。目前车载操作系统的架构、功能、性能和安全等要求缺乏统一标准，导致车厂在选择车载操作系统时，无据可依。车载操作系统供应链相关方在互操作和产品研发测试时，以定制化为主，很难实现可复制规模化推广。

在车载操作系统架构层面，对于单系统、多系统、专用的车载操作系统（如仪表系统、中控系统、T-box 系统、可信执行环境系统）尚无统一的架构分类和功能模块定义。

在车载操作系统单系统方面（图 3），宏内核和微内核共存，缺乏统一的内核基础能力定义；车载操作系统可提供的语音、账户管理、驾驶辅助、增强的人机交互等基础服务，尚未统一其定义及技术功能和性能要求，不便于车厂基于自身应用需求选择适合的车载操作系统。

在车载操作系统多系统方面（图 4），目前虚拟化（Hypervisor）、容器（Container）等技术方案共存，功能要求不清晰，系统间通信要求不明确，车厂很难根据自身对多系统的安全性和灵活性等需求选择合适的多系统方案。

在兼容性方面，目前没有标准的南向北向接口定义，导致操作系统生态的割裂。车载操作系统对硬件、整车信号资源和服务、应用程序等的抽象尚无统一的接口要求，需要标准化，使得操作系统在调用硬件资源时，无需了解硬件的内部逻辑及资源分配情况，只需要按照标准的接口进行参数配置即可实现对硬件的调用，从而实现对于不同硬件的适配，保证系统具备较高移植性。同时，面向应用程序的接口需要标准化，使得应用程序在调用操作系统的基础服务时，无需了解其内部逻辑，只需要按照统一的接口进行参数配置即可实现应用程序与操作系统的交互，从而进一步实现应用程序与操作系统的解耦，促进应用生态发展。

在功能安全方面，不同类型的车载操作系统应用场景对车载操作系统的功能安全要求及ASIL (Automotive Safety Integration Level) 等级要求不同，需要进行规范和标准化，明确功能安全涉及的典型故障和安全机制要求。

在信息安全方面，需要对车载操作系统的系统安全、数据安全、通信安全、应用安全、多系统安全技术要求进行规范。

在车载操作系统测试方法标准化方面，目前对于车载操作系统各功能和性能尚未有标准化的测试方法指导，需要对测试项目、方法和测试通过要求进行标准化。

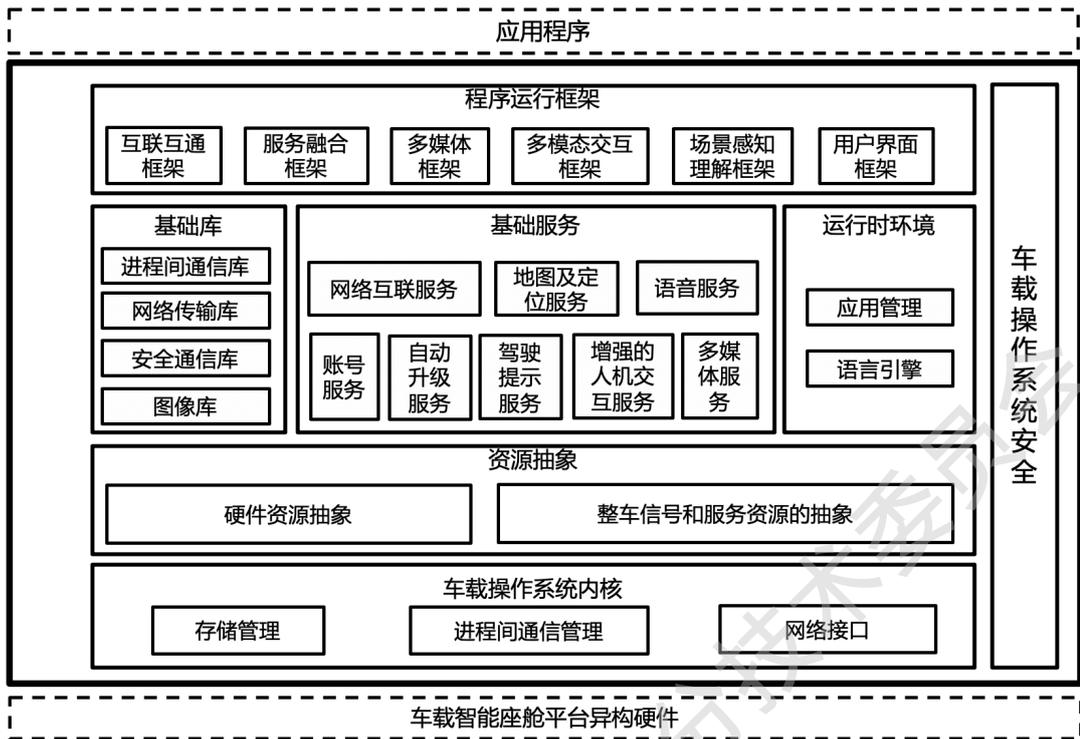


图3 车载操作系统单系统参考架构

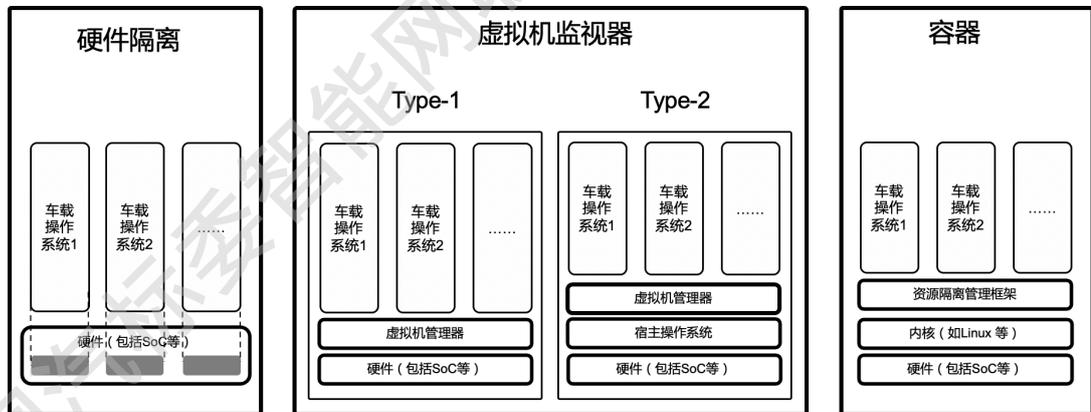


图4 车载操作系统多系统参考架构

2.1.1.5 标准体系框架及分类说明

操作系统体系架构一般由内核、资源抽象、基础库、运行时环境、协议栈、基础服务和程序运行框架组成，在部署形态上可分为单系统和多系统，多系统的实现技术包括硬件隔离、虚拟机管理器和容器。操作系统还要包含贯穿各层的功能安全和信息安全相关设计。综合操作

系统应用场景及分层解耦的需要，操作系统标准体系框架定义为“基础通用”、“技术要求及评价方法”、“系统架构”、“面向应用程序的接口”、“系统服务”、“基础系统”、“面向硬件的接口”、“功能安全”、“信息安全”、“配置方法”等部分（见图5）。

1. 基础通用

基础通用类标准主要包括术语和定义标准、系统间通信协议标准等。

2. 系统架构

针对操作系统架构中需要进行统一化定义的层级，进行了规范和定义，为建设开放的操作系统生态提供有效的标准保障。可进一步细分为面向应用程序的接口标准、系统服务功能标准、基础系统的标准及面向硬件的接口标准。其中：

（1）面向应用程序的接口标准将操作系统面向应用程序的接口标准化，使得应用程序在调用操作系统的基础服务时，无需了解其内部逻辑，只需要按照统一的接口进行参数配置即可实现应用程序与操作系统的交互，从而进一步实现应用程序与操作系统的解耦，促进应用生态发展。

（2）系统服务功能标准将操作系统中应支持的基础服务进行标准化，包括系统服务、通信服务、数据存储服务、故障诊断服务等。

（3）基础系统的标准分为单系统和多系统，单系统是对内核、资源抽象、基础服务、运行时环境、基础库、协议栈、程序运行框架等各层级的功能要求和性能要求进行标准化；多系统是对硬件隔离、虚拟管理器、容器三类架构的功能要求和性能要求进行规范和标准化。

(4) 面向硬件的接口标准将操作系统面向车规级芯片的接口标准化，使得操作系统在调用硬件资源时，无需了解硬件的内部逻辑及资源分配情况，只需要按照标准的接口进行参数配置即可实现对硬件的调用，从而实现对于不同硬件的适配。

3. 技术要求和评价方法

技术要求用于规范操作系统的功能要求和性能要求，定义操作系统应满足的功能要求和性能要求，为操作系统的分类及评价提供标准依据。

评价方法基于操作系统的技术要求标准建立相应的针对操作系统功能和性能的评价方法，以实现多维度、全方位的评价操作系统的功能及性能水平。

4. 安全要求

安全类标准包括功能安全和信息安全，分别定义操作系统的功能安全标准和信息安全标准。操作系统功能安全标准的主要目的是确保智能网联汽车操作系统功能运行的可靠性，并在操作系统发生软件功能发生失效后仍能最大程度地保证车辆安全运行。操作系统信息安全标准的主要目的是防范针对车辆操作系统安全漏洞的攻击、侵入、破坏和非法使用，最大限度地避免由于操作系统安全漏洞带来的车辆安全隐患。

5. 配置方法

配置类标准主要用于操作系统配置方法的统一化和规范化，定义操作系统配置方法，即组成操作系统的软件模块的配置参数和配置方法、以及不同软件模块之间的协同方法。随着车用操作系统的发展，尤其是面向智能网联汽车的操作系统的集成化的发展趋势，操作系统的功能日益复杂，随之操作系统的配置方法也趋于复杂，因此，建议统一的操作系统配置标准，有利于操作系统配置工具链的建立，也能有效提高操作系统开发效率。

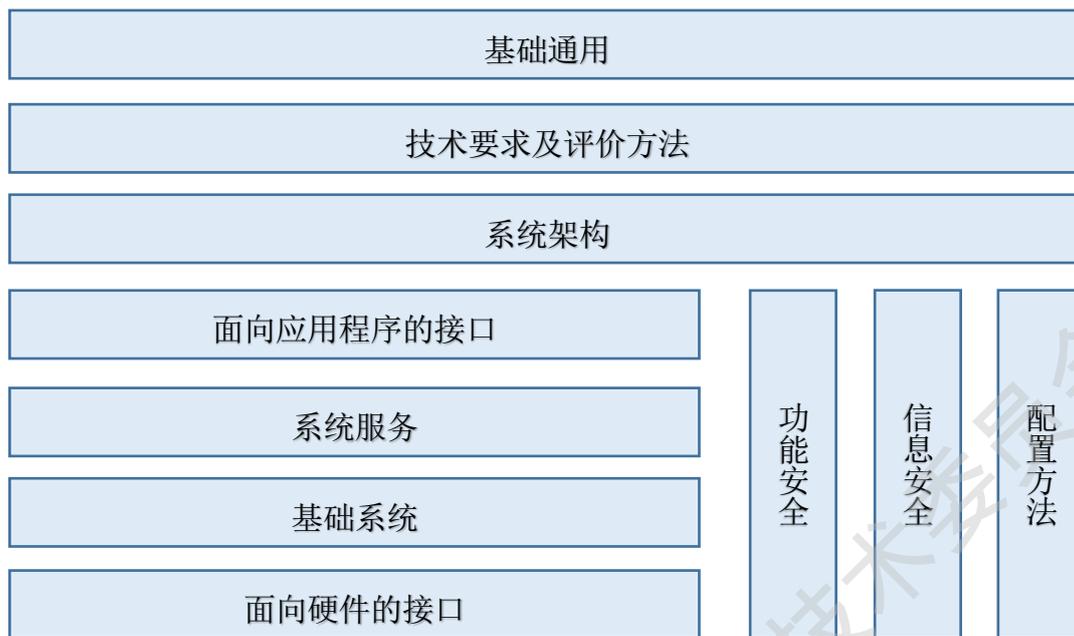


图5 操作系统标准体系框架

表2 车载操作系统标准体系

标准项目及分类		标准类型	标准描述	备注
操作系统（200）				
车载操作系统（210）				
210-1	车载操作系统技术要求和试验方法	国标推荐	定义了车载操作系统的总体架构，功能及性能要求，并规范功能和性能的验证方法，确保车控操作系统测试验证方法的统一同时为车载操作系统标准体系中其他部分标准的制定提供基础支撑。	
210-2	车载操作系统多系统技术要求	团标	明确了车载操作系统多系统的技术要求及部署方式，确保车载多系统技术的概念及技术转态的统一	
210-3	车载操作系统多系统间通信技术要求	团标	规定了车载操作系统多系统间通信的方式及技术要求，确保多系统间通信技术评价依据的统一	
210-4	基于人工智能的车载操作系统技术要求	团标	规定了基于人工智能的车载操作系统应满足的技术要求，包括提供训练模型的执行环境等功能要求、AR 导航、手势识别、图像理解等性能要求，提升基于人工智能的车载操作系统的服务质量	
210-5	车载操作系统硬件接口要求	团标	规定了车载操作系统硬件接口规范，确保车载操作系统硬件生态的统一	

210-6	车载操作系统应用接口要求	团标	规定了车载操作系统应用编程接口规范，确保车载操作系统软件生态的统一
210-7	车载操作系统功能安全技术要求	团标	定义了车载操作系统功能安全等级要求建议与设计方法，旨在规范功能安全相关开发要求及方法。
210-8	车载操作系统可信执行环境技术要求	行标	定义了车载操作系统对于可信执行环境的支持及技术标准，确保车载操作系统可行执行环境技术能力及指标的一致性

2.1.2 车控操作系统

2.1.2.1 发展历程

车控操作系统是指用于控制汽车各种功能的软件系统。它的发展历程可以大致分为以下几个阶段：

(1) 早期的机械控制阶段，汽车最初的控制系統由机械元件实现，如油门、刹车、离合器等。这些机械元件的设计和制造需要耗费大量的人力和物力，而且容易出现故障。

(2) 电子控制阶段，随着电子技术的发展，汽车控制系统开始采用电子元件，如传感器、执行器等。该方式能够提高控制精度和可靠性，但仍然需要人工进行调整和维护。

(3) 网络化控制阶段，随着互联网技术的普及，汽车控制系统进一步实现网络化，车控操作系统也出现了 AUTOSAR (Automotive Open System Architecture) 等标准通信中间件。该方式实现车辆与外部环境的信息交换和数据共享，提高了驾驶体验和安全性。

(4) 自主控制阶段，近年来，人工智能技术的兴起使得汽车控制系统开始向自主控制方向发展。自主控制系统能够学习和适应不同的驾驶环境和驾驶者行为，实现更加智能化的驾驶体验。总之，随着技术的不断发展，汽车控制系统不断创新和升级，为驾驶者提供更加安全、便捷和智能的驾驶体验。

AUTOSAR是一种基于开放标准的汽车软件架构，它旨在促进汽车电子系统的开发和集成。AUTOSAR软件的发展历史可以大致分为以下几个阶段：在初期阶段（2003-2006年），AUTOSAR最初于2003年成立，由欧洲汽车制造商协会（ACEA）和汽车电子工程师协会（SAE）共同发起。在这一阶段，AUTOSAR主要关注汽车电子系统的标准化和模块化设计，推出了第一个版本的

AUTOSAR标准。标准化阶段（2006-2010年），AUTOSAR在2006年发布了第二个版本的标准，并开始向全球范围内推广。在这一阶段，AUTOSAR不断完善标准规范和工具链，以支持更多的汽车制造商和供应商使用AUTOSAR。标准实施及应用阶段（2010-2015年）：在AUTOSAR标准得到广泛应用的同时，AUTOSAR开始关注基于标准的实际应用。AUTOSAR推出了一系列的应用指南和最佳实践，帮助汽车制造商和供应商更好地应用AUTOSAR标准。开放平台阶段（2015年至今）：随着汽车电子系统的不断发展和互联网技术的普及，AUTOSAR开始向开放平台方向发展。AUTOSAR推出了一系列开放平台标准和工具，支持更加灵活和高效的汽车软件开发和集成。总之，AUTOSAR标准是汽车电子系统开发和集成的重要标准之一，它经历了从标准化到应用再到开放平台的发展过程。随着汽车电子系统的不断创新和发展，AUTOSAR标准也在不断升级和完善，以应对日益复杂的汽车电子系统需求。

车用 Linux 操作系统是指基于 Linux 内核的汽车软件操作系统，它具有开放性、灵活性和可定制性等特点，是汽车电子系统中的重要组成部分。最初，车用 Linux 操作系统主要用于嵌入式系统和网络设备等领域。在这一阶段，Linux 内核被移植到了汽车电子系统中，但仅仅是作为一个实验性质的项目。在 2005 年左右，Linux 开始在汽车电子系统中得到更广泛的应用。一些汽车制造商和供应商开始将 Linux 作为车载娱乐系统、导航系统等应用的操作系统。在这一阶段，Linux 开始逐渐成为车用操作系统的标准之一。随着汽车电子系统的不断发展和互联网技术的普及，车用 Linux 操作系统开始向开放平台方向发展。在这一阶段，Linux 开始支持更多的汽车应用，如自动驾驶、车联网等。同时，开源社区也在不断改进和完善 Linux 内核，以支持更广泛的汽车应用。总之，车用 Linux 操作系统是汽车电子系统中的一个重要组成部分，它经历了从实验性质到标准化再到开放平台的发展过程。随着汽车电子系统的不断创新和发展，车用 Linux 操作系统也在不断升级和完善，以应对日益复杂的汽车电子系统需求。

2.1.2.2 特点

车控操作系统是指用于控制汽车电子系统的软件操作系统，它是汽车电子系统中的重要组成部分。车控操作系统采用纵向分区、横向分层式架构，并在逻辑层次上包含系统软件和功能软件框架，是车载智能计算基础平台安全、实时、高效的核心和基础。系统软件创建复杂嵌入式系统运行环境，可以实现与 Classic 和 Adaptive 两个平台的兼容和交互。功能软件根据中国智能网联汽车应用特点，以及各类辅助驾驶/自动驾驶功能的核心共性需求，明确定义和实现各共性子模块，并进行通用模块定义和实现。车控操作系统主要负责处理和控制在汽车各个部件的信号和数据，如发动机、变速器、制动系统、安全气囊等，以实现汽车的正常行驶和安全性能。

车控操作系统具有以下几个特点：

(1) 实时性

车控操作系统需要保证对汽车各个部件的控制是实时的，以确保汽车的正常行驶和安全性能。

(2) 高可靠性

车控操作系统需要具备高可靠性，以确保汽车的稳定性和安全性。

(3) 高安全性

车控操作系统需要具备高安全性，以应对各种潜在的安全威胁，如黑客攻击、恶意软件等。

(4) 可扩展性

车控操作系统需要具备良好的可扩展性，以支持不同类型和不同品牌的汽车电子系统。

目前，市场上主要的车控操作系统包括 AUTOSAR、车用 Linux 操作系统、车用 QNX 操作系统等。这些车控操作系统都具备高实时性、高可靠性和高安全性等特点，并且支持不同类型和不同品牌的汽车电子系统，以满足汽车行业的不断发展和创新需求。

2.1.2.3 发展现状

安全车控操作系统国外发展较早，目前已经开展了一系列的标准化工作，国内目前主要处于跟随状态。2003年，宝马、博世、大陆、戴姆勒、通用、福特、标志雪铁龙、丰田、大众等9家企业作为核心成员，成立了一个汽车开放系统架构组织（简称AUTOSAR组织），致力于建立一个标准化平台，以减少汽车软件设计的复杂度，提高灵活性和开发效率。AUTOSAR以OSEK/VDX为基础，但涉及的范围更广。车控操作系统多基于经典AUTOSAR，适用于控制类 and 安全性较高的应用场景。AUTOSAR组织发展至今，得到了越来越多的行业认可，目前已有超过180家的车、零部件、软件、电子等领域的成员。AUTOSAR标准平台由于采用开放式架构和代码开源方式，目前已经成为国际主流的标准软件架构，它不仅提高了开发效率，降低开放成本，同时保障了车辆的安全性与一致性。目前基于AUTOSAR标准平台，拥有完整的汽车软件解决方案的企业主要有Vector、KPIT、ETAS、DS以及被大陆收购的伊莱比特和被西门子收购的MentorGraphics。此外，宝马、沃尔沃等汽车厂商都相继推出了基于AUTOSAR标准平台的车型。

智能驾驶操作系统将会成为自动驾驶汽车发展的核心竞争力之一，由于安全车控操作系统相对成熟，且智能驾驶操作系统部分包含安全车控操作系统，所以本文提到的车控操作系统主要是指智能驾驶操作系统。AUTOSAR组织为应对自动驾驶技术的发展推出了Adaptive AUTOSAR架构。

目前普遍采用的车控操作系统底层内核主要有Linux、QNX。Linux最初是作为通用操作系统而设计开发的，但提供了一些实时处理支持，这包括大部分POSIX标准中的实时功能，支持多任务、多线程，具有丰富的通信机制等。Linux也提供了符合POSIX标准的调度策略。QNX是一种商用的遵从POSIX规范的类Unix实时操作系统，其主要特点是符合分布式、嵌入式、可规模扩展的硬实时操作系统。QNX遵循POSIX.1（程序接口）和POSIX.2（Shell和工具）、部分遵循POSIX.1b（实时扩展）。QNX的微内核结构是它区别于其它操作系统的显著特点。

表3 国内汽车行业相关研究成果

序号	名称	机构	状态
1	《智能网联汽车 车控操作系统技术要求及试验方法》	NTCAS	撰写中

2	《车控操作系统架构研究》	NTCAS	已发布
3	《车控操作系统总体技术要求研究》	NTCAS	已发布
4	《车用操作系统测试评价研究》	NTCAS	已发布
5	《车控操作系统功能软件架构及接口规范》	CSAE	已发布
6	《智能网联汽车 车控操作系统功能安全要求》	CSAE	已发布

表4 国际操作系统类标准统计

发布日期	标准名称	部门/机构	相关内容
1993	可移植操作系统接口 (POSIX) 第一部分: 系统应用程序接口 (API)	国际标准化组织	标准定义了操作系统应该为应用程序提供的接口标准, 是为要在各种 UNIX 操作系统上运行的软件而定义的一系列 API 标准
2004	IEEE 标准信息技术标准化应用环境简介 (AEP) - POSIX (R) 实时和嵌入式应用支持	美国电气电子工程师学会	适用于使用标准所提供的服务和实用程序来开发和执行实时或嵌入式应用程序
2008	信息技术 POSIX 标准符合性的测试方法规范和测试方法实现的要求和指南	中国国家标准化管理委员会	本标准旨在供测试方法规范和测试方法实施的开发者和使用者使用。
2022	Classic AUTOSAR R22.11	AUTOSAR 协会	Classic AUTOSAR 标准平台是为硬实时和安全要求严格的嵌入式系统提出的 AUTOSAR 解决方案。Classic AUTOSAR 将微控制器上的软件抽象为三个软件层: 应用程序、运行时环境(RTE)和基本软件(BSW)。其中 BSW 分为三个主要层: 服务层、ECU 抽象层和微控制器抽象层。应用与应用之间, 以及应用于 BSW 之间的通信都是经过 RTE 完成数据交换, 因此做到了应用与硬件的完全独立。
2022	Adaptive AUTOSAR R22.11	AUTOSAR 协会	Adaptive AUTOSAR 标准平台是为高性能计算的 ECU 提出的解决办法, 用于自动驾驶等。Adaptive AUTOSAR 平台为 AUTOSAR 应用实现了运行环境 ARA。使用两种接口完成数据交换: 服务和 API。平台由功能集群组成, 这些集群按服务和自适应 AUTOSAR 基础进行分组如下。 Adaptive AUTOSAR 解决了新一代汽车高性能需求、连接性和持续软件无线(OTA)更新带来的新市场需求, 它作为多个供应商的软件集成平台, 解决了 Classic AUTOSAR 经典架构的局限性, 其为灵活性而设计的, 以便在运行时支持软件更改。 Adaptive AUTOSAR 构建在 POSIX

			操作系统之上，由不同的功能模块组成，这些模块被划分在服务模块和基础模块上,它的的通信是面向服务类型的，会将网络绑定到 DDS 或者 SOME/IP 使用以太网与其它 ECU 通信。
--	--	--	--

2.1.2.4 标准需求

车控操作系统是智能网联汽车的基础软件部分，运行于智能网联汽车车载智能计算基础平台之上，支撑智能网联汽车驾驶自动化功能实现和安全可靠运行的软件集合，分为安全车控操作系统和智能驾驶操作系统。安全车控操作系统，面向传统车辆控制领域，如动力系统、底盘系统和车身系统等；该类操作系统对实时性和安全性要求极高，生态发展已经趋于成熟。智能驾驶操作系统，面向智能驾驶领域，应用于智能驾驶控制器，该类操作系统对安全性及可靠性要求较高，同时对性能及运算能力的要求也较高，目前在全世界范围内都处于研究发展的初期，生态尚未完备。因此本文提到的车控操作系统主要是指智能驾驶操作系统。

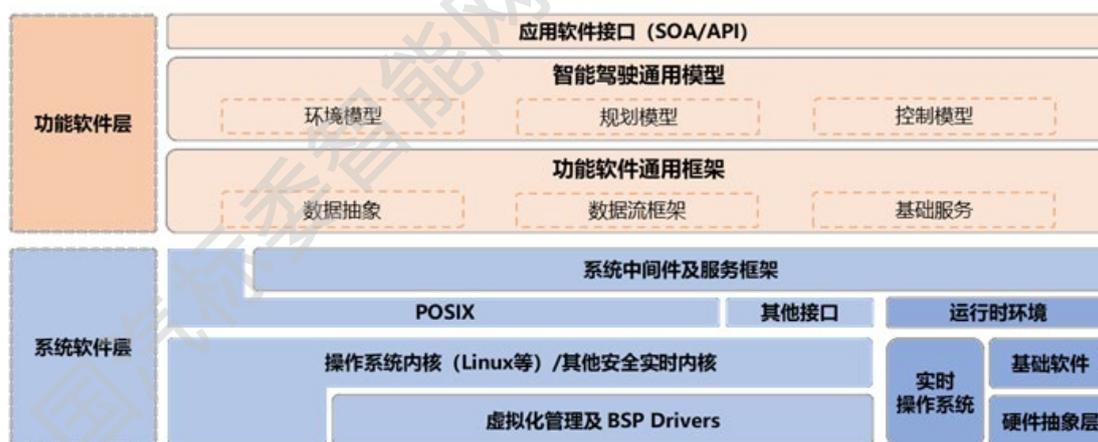


图 6 车控操作系统参考架构

2021 年汽标委在参考并吸收国内外主流车控操作系统架构的基础上，提出基于“中国方案”的车控操作系统参考架构（见图 6）。车控操作系统介于硬件平台层和应用软件层之间，由系统软件和功能软件两部分组成。系统软件是嵌入式系统运行环境的软件集合，车控操作系统中主要支撑驾驶自动化功能实现，包括内核、虚拟化管理和中间件等；功能软件是

车控操作系统中面向智能驾驶核心共性需求形成的智能驾驶共性服务软件集合，支撑自动驾驶自动化功能开发，包括数据抽象、功能软件通用框架、智能驾驶通用模型和应用软件接口等。

车控操作系统目前种类较多，各厂家提供的系统无法兼容，在技术方案上，以微内核操作系统为主，近年来，Safety Linux 由于其较好的富生态能力，也开始在智能驾驶场景部署。面对这种现状，需要通过车控操作系统的标准化实现产业链的分工协同，提高开发效率，保障软件平台的安全可信，减少对接成本，构建统一生态。

车控操作系统术语和定义的标准化能够统一车用操作系统相关的基本概念，为车用操作系统标准体系中其他部分标准的制定提供基础支撑。车用操作系统间通信协议标准化则为实现操作系统间的通信建立统一规范，从而为不同车用操作系统间的兼容融合提供标准保障。

架构是操作系统的顶层设计，明确了对系统实现的约束条件，为维护决策提供依据，以及便于在较高层面上实现复用。因此，将车控操作系统的架构和技术要求进行标准化有助于实现操作系统的行业共识，降低主机厂、零部件供应商等之间的沟通成本，实现操作系统软件复用，提高开发效率。车控操作系统的技术要求包括功能要求和性能要求等，通过技术要求和测试方法的标准化，可以评价车控操作系统的基本功能和性能能否满足产品设计要求。

功能安全和信息安全是车控操作系统产品可靠安全运行的必要组成部分。车控操作系统为智能汽车自动驾驶功能提供运行环境，其功能安全要求是保证整个系统功能安全的基础和核心。车控操作系统是单域核心攻击目标，对于车控操作系统的攻击，可以直接影响到智能网联汽车的功能安全和人身安全，也会涉及到个人隐私数据和包括测绘等敏感数据。对于车控操作系统的安全要求类的标准制定，可以设定操作系统安全基线，最大限度保护智能网联汽车的安全运行，数据的安全存储和处理。

车控操作系统技术作为智能网联汽车的支撑技术，车控操作系统模块间接口标准化主要是为了为智能驾驶等应用提供标准化的运行环境和服务，满足不同操作系统之间以及操作系统内部不同层次之间的实现通信，高效实现和互操作，实现操作系统解耦，满足跨平台、跨车型、可扩展等要求。

随着车控操作系统的发展，尤其是面向智能网联汽车的操作系统的集成化的发展趋势，操作系统的功能日益复杂，随之操作系统的配置方法也趋于复杂，因此，建立统一的操作系统配置标准，有利于操作系统配置工具链的建立，也能有效提高操作系统开发效率。

2.1.2.5 标准框架及分类说明

表5 车控操作系统标准体系

标准项目及分类		标准类型	标准描述	备注
操作系统（200）				
车控操作系统（220）				
220-1	车控操作系统间通信要求	国标推荐	定义了车控操作系统间的通信标准，确保车内多个操作系统之间能互相通信及兼容并存	
220-2	车控操作系统技术要求及测试方法	国标推荐	定义了车控操作系统的总体架构、功能及性能要求，并规范功能和性能的验证方法，确保车控操作系统测试验证方法的统一	
220-3	车控操作系统评价指南	团标	定义车控操作系统评价标准，确保车控操作系统评价体系的一致性	
220-4	车控操作系统功能软件架构及接口规范	团标	定义了车控操作系统功能软件的架构及应用的编程接口要求，确保车控操作系统基础能力及接口的统一	
220-5	车控操作系统虚拟化管理技术要求及测试方法	团标	定义了车控操作系统虚拟化技术的支持要求及对应的技术项测试验证方法，确保车控操作系统虚拟化技术标准 and 验证方法的统一	
220-6	车控操作系统中间件技术要求及测试方法	团标	定义了车控操作系统中间件的技术要求及验证方法，确保车控操作系统中间件的功能及验证方法的一致	
220-7	车控操作系统开发工具链技术要求及测试方法	团标	定义了车控操作系统开发工具链的技术标准及验证规范，确保工具链技术要求及测试验证方法的一致	
220-8	车控操作系统内核技术要求与测试方法	团标	定义了车控操作系统内核功能及性能要求，并规范内核技术的验证方法，确保内核功能、性能要求及测试方法的一致性	
220-9	车控操作系统面向硬件的接口规范	团标	规定了车控操作系统硬件接口规范，确保车控操作系统硬件生态的统一	
220-10	功能软件和系统软件层间接口规范	团标	规定了车控操作系统功能软件和系统软件层的编程接口规范，确保车控操作系统功能软件生态的统一	

220-11	车控操作系统功能安全技术要求	行标	定义了车控域操作系统功能安全等级要求建议与设计方法，旨在规范功能安全相关开发要求及方法。
220-12	车控操作系统配置方法	团标	规定了车控操作系统配置标准，包括车控操作系统软件模块的配置参数和配置方法、以及不同软件模块之间的协同方法，提高操作系统开发效率

2.1.3 系统服务中间件

2.1.3.1 发展历程

系统服务中间件是指位于基础软件平台（即车载操作系统和车控操作系统）和应用软件之间，主要用于支持应用软件的通用化基础功能的开发和使用，可实现车内各功能服务之间、车云之间共享通信、诊断、计算等资源的服务软件单元的集群，随着汽车 E/E 架构的演进而发展。汽车 E/E 架构处于分布式架构阶段时，功能主要靠堆砌控制器来实现，软硬件紧密耦合，不存在系统服务中间件层，控制器之间耦合也较多，新技术和新功能的应用及更新受制于 E/E 架构限制；汽车 E/E 架构当前处于域控架构阶段，以太网作为骨干网通信，域控承载域内的路由功能，各域内通过软件 SOA 实现系统服务中间件，支撑智驾、大数据、FOTA 以及跨域功能融合；进入 2023 年后，汽车 E/E 架构演进到准中央计算架构阶段，整车绝大部分功能由 HPC 来实现，区域控制器负责区域下的软件逻辑实现，全车通过软件 SOA 实现系统服务中间件，支撑软硬解耦及功能动态配置；2026 年之后，汽车 E/E 架构将进入中央计算架构阶段，骨干网络将由车载以太网和高速无线网络组成，车云一体化系统服务中间件将支撑远程云端车辆控制及车辆功能部署的无关化。

为了满足车企架构转型的需求，车企内外部生态建设的需要，也为了满足汽车行业生态建设的需求，AUTOSEMO 推出了车端开放的分布式服务框架 ASF (AUTOSEMO Service Framework)，以构建本土化基础软件生态，促进产业链协作和可持续发展。当新型 E/E 架构的硬件架构逐步明确后，软件将会逐步标准化，从软件层次划分情况，各家整车厂已经在逐步向统一方向发

展。目前明显的趋势是跨域协同，车云互联，以及软硬件分离。虽然，AUTOSAR在标准基础软件开发平台对上层提供统一封装的硬件能力和标准服务组件，但是没有对跨核、跨域功能的汇总，使用者仍需要完成业务整合汇总的工作，对开发者的开发效率会产生一定影响。ASF使用标准化基础软件开发平台和操作系统提供的接口，提供更多整车业务层面需要的功能，并封装成基础系统服务与整车系统服务，例如整车级日志服务、整车级诊断服务等，将各个节点的服务汇总成整车级，为开发者提供基于整车统一视图的服务功能及接口。可以帮助 OEM 更好的聚焦于策划与开发附加值更高的应用软件，软件服务企业的分工逐渐细化，促进汽车软件生态的发展。

2.1.3.2 特点

系统服务中间件是指车端各 ECU (Electronic Control Unit) 与云端生态能力之间，基于 SOA (Service-Oriented Architecture) 服务化的设计思想，构建支撑整车智能化应用场景的平台化软件，赋能千人千面个性化体验。核心软件特点包括：横向-跨域，针对不同的 E/E (Electrical/Electronic) 架构（尤其在域控集中式架构下），通过解决车内横向跨域的问题，构建整车集中的软件平台，是车云一体架构中车端面向云端能力融合的基础。纵向-跨车云，基于车内功能统一整合和可开放的能力，通过与云端的打通实现车云之间数据和服务的交互，从而充分利用云端的计算能力和生态服务能力提升智能化。基于 SOA 的设计思想将云端和车端能力以服务化的方式进行封装，进而简化应用的开发效率以及体现服务自由组合能力实现场景灵活可变的用户体验。

系统服务中间件需要进行能力抽象、分层、权限控制、分模态组合、分应用的提供车云一体服务，支撑云、管、端一体的 SOA 平台架构。技术特点主要表现在以下方面：

(1) 位置透明

使用者不需要关心服务在车端还是云端，由平台自动完成服务的消费和响应；车端服务能够在云端访问，云端服务同时能够在车端访问，形成车云一体化服务。

(2) 可重用

软件组件抽象实现，方便地提供服务，能够被不同的上层软件抽象和使用，不同的车辆能够使用云端同一个软件组件提供的能力服务，车辆内部也能够很容易地实现跨域访问，无需重复地构建。

(3) 易扩展

实现了组件的接口与实现分离，便于进行实现的重构而不影响服务的消费者；也可以方便地进行云端、车端能力的扩展，通过服务发现的机制自动加入到 SOA 平台而不需要平台重构就可以向消费者提供服务；还可以通过服务编排的机制根据不同的场景进行服务的调度和协调。

(4) 易使用

服务消费者只关心抽象服务的使用，而不用关心具体使用哪些车辆服务，这些服务是来自车端还是云端、也不用关心车辆服务的组合和协调调度，更不用关心服务的实现机制和原理。

(5) 安全可控

通过访问权限暴露相关服务接口的机制，有效地控制了非法的访问，并且系统的安全机制和分层管控方法，更加方便地进行安全策略的控制和调整，保证了整车 SOA 平台的安全可靠。

系统服务中间件解放了开发者的思维，提升了开发者对车辆的认识，可以让不同的角色聚焦到自己熟悉的领域，而屏蔽不必要的实现和部署细节，降低了上层应用对底层能力的依赖，解耦了不同功能组件的相互的依赖等。

2.1.3.3 发展现状

随着电子电气性能和可靠性等因素的发展，E/E 架构逐步向中央集中式结构演进，实时性要求高的计算在本地中央计算平台处理，实时性要求较低或需要与外部协同的计算通过车云

一体平台交到云端进行处理。国内外主机厂、造车新势力、零部件供应商等都在着力打造其专属的操作系统加系统服务中间件的基础软件平台。

国际企业中，2019年6月，基于汽车智能化和汽车新商业模式需求，大众汽车宣布成立 Car Software Org 软件部门，专门从事自主软件平台 VW.OS 的研发。2021年，在 Car Software Org 的基础上，成立 Cariad 软件子公司。计划于 2025 年，旗下所有新车型均搭载 VW.OS，以实现车辆传统控制系统、ADAS 系统、车载娱乐/互联系统、能源管理系统和车辆舒适系统的全面整合。VW.OS 是集成了 POSIX OS、AUTOSAR、VW API 的一体式 SOA 平台，用于大众全新端到端架构（E3），其中，E3 1.1 版本用于 MEB 平台，E3 1.2 版本用于奥迪和保时捷的 PPE 平台，E3 2.0 版本未来的 SSP 平台，适用于集团旗下所有品牌，支撑大众自主研发的操作系统 VW.OS，并连接至大众汽车云 VW.AC。此外，其他国际车企也相继推出自己的操作系统，例如，特斯拉的 Tesla.OS、丰田的 Arene、奔驰的 MB.OS 等。

在国内，2022年，由 AUTOSEMO 执行办公室发起、一汽、东风、蔚来、小鹏、广汽、吉利、长城、长安、东软睿驰、零束等 20 家创始会员单位参与，结合当下车用操作系统主要技术形态及 AUTOSEMO 技术生态，提出了“AUTOSEMO（2022）智能汽车软件生态框架”，包含了车端软件、云端软件及相关技术。会员单位也相继发布自己的系统软件并应用到量产车辆中，包括一汽的 FAW.OS、小鹏的 XPilot、理想的 Li OS、广汽的普赛 OS、上汽零束的 SOA 软件平台、东软睿驰的 openVOC 开放技术框架等。

其主要价值体现如下：

(1) 整车级软件层面：基于整车软件平台抽离通用化且平台功能可复用性强的基础功能，并集中在一个软件集群中为各功能服务所调用，用于服务在整车平台的能力扩展，并实现整车各系统之间的协同，保证整车软件平台的整体性并进行统一管控。

(2) 平台级软件层面：基于 AUTOSAR 服务框架扩展，向应用层提供更多基于服务开发需要的功能，接口更加丰富和灵活，新增了一个运行在 AUTOSAR CP 和 AP 之上的软件层，对上提供统一视图的操作接口。应用可以调用系统服务中间件的接口，也可以调用 AUTOSAR 标准

的接口，并且可以基于原子服务及系统服务提供的功能进行组合，实现服务的级联，针对不同异构系统分别提供软件包。

(3)车云一体级软件层面：通过打通车云之间的通讯链路，可以帮助 OEM 提升车云高效协同，在复杂的网络环境下可自由切换，并提高数据传输效率，降低数据传输成本，同时在获取脚本后，通过已封装的服务可立即解析和执行，无需繁琐的升级流程，高效的服务解析能力，可提高客户体验和满意度，车云一体通过分布式引擎能力，保障场景引擎在服务自由调度过程中平稳运行，并且有效降低频繁调度的总线带宽负荷，从而提高运行稳定性和安全性。

表 6 国内汽车行业相关研究成果

序号	名称	机构	状态
1	《中国汽车基础软件发展白皮书》	AUTOSEMO	2023 年发布 4.0 版本
2	《车载 SOA 软件架构技术规范》	AUTOSEMO	2021 年发布 1.1 版本
3	《ASF 软件架构技术规范》	AUTOSEMO	2022 年发布 1.0 版本
4	《车云一体软件架构技术规范》	AUTOSEMO	2022 年发布 1.0 版本
5	《整车 SOA 架构设计与软件平台框架定义》	CAAM	撰写中
6	《整车服务框架技术规范》	CAAM	撰写中
7	《车云一体服务化通讯协议》	CAAM	撰写中
8	《车云一体架构规范》	CAAM	撰写中

2.1.3.4 标准需求

系统服务中间件使用基础软件平台的接口，提供更多整车业务层面需要的功能，并封装成基础系统服务与整车系统服务，例如整车级日志服务、整车级诊断服务等，将各个节点的服务汇总成整车级，为开发者提供整车统一视图的服务功能及服务接口。系统服务中间件架构包括车端系统服务、车云一体动态服务框架、云端系统服务三个部分，车端系统服务又由原子服务框架、SOA 增强型服务、系统级基础服务、整车级系统集成服务组成。如图 7 所示：

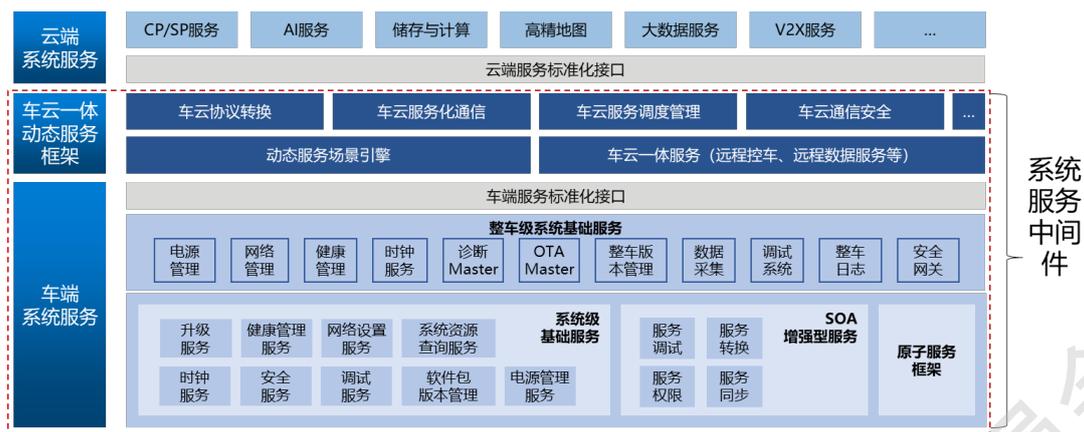


图7 系统服务中间件架构

随着整车 E/E 架构演进，以操作系统为核心向软件平台发展过程中衍生出系统服务中间件，由于汽车行业内各企业都在打造其专属的软件平台，为了更好的实现横向跨域、纵向跨车云的需求，急需对系统服务中间件提出基础的技术要求及试验方法。

系统服务中间件的定位是为应用软件提供系统服务，所以需要对其内部的各个组成部分的服务接口进行标准化，包括原子服务接口、系统级基础服务接口、整车级系统集成服务接口、云端系统服务接口、车云一体服务接口等，以达到应用部署位置透明、可重用、易扩展等目标。

系统服务中间件作为服务软件集群，分布式部署在车端及云端，车内也部署在不同的控制器中，所以各系统服务在车内各控制器及车云间的通信协议需要进行标准化，包括整车日志采集协议、整车数据采集协议、整车版本采集协议、整车 OTA 管理协议、整车诊断管理协议、整车时钟服务协议、整车健康管理协议、整车网络管理协议、整车电源管理协议及车云一体服务化通信协议等，以实现横向跨域、纵向跨车云的需求。

各系统服务还需要从一般要求、功能安全、信息安全、性能、配置等方面提出功能要求的相关标准，包括整车日志采集、整车数据采集、整车版本采集、整车 OTA 管理、整车诊断管理、整车时钟服务、整车健康管理、整车网络管理、整车电源管理等，使开发者专注于业务设计和开发，为业务创新和软件快速迭代提供支撑。

2.1.3.5 标准体系框架及分类说明

表 7 系统服务中间件标准体系

标准项目及分类		标准类型	标准描述	备注
操作系统（200）				
系统服务中间件（230）				
230-1	系统服务中间件技术要求及试验方法	国标推荐	明确操作系统中应支持的系统服务的技术要求和试验方法，包括标准化集合、安全要求等，确保车用操作系统功能要求的一致性	
230-2	原子服务技术要求及试验方法	行标	规定了整车原子服务接口规范及试验方法，确保车用操作系统整车生态的统一	
230-3	车端系统服务技术要求及试验方法	行标	规定了车端系统级服务接口规范及试验方法，确保车用操作系统整车生态的统一	
230-4	车云一体服务化技术要求与试验方法	行标	规定了车与云间的通信协议的技术要求与试验方法，确保车与云之间能互相通信及兼容并存；规定了云端系统服务接口规范及试验方法，确保车用操作系统车云一体生态的统一	
230-5	整车电源管理技术要求与试验方法	团标	定义整车电源管理功能的技术要求及试验方法，确保整车电源管理功能及验证方法的一致；规定了车用操作系统间配电管理的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-6	整车网络管理技术要求与试验方法	团标	定义整车电源管理功能的技术要求及试验方法，确保整车电源管理功能及验证方法的一致；规定了车用操作系统间网络唤醒管理的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-7	整车健康管理技术要求与试验方法	团标	定义整车健康管理功能的技术要求及试验方法，确保整车健康管理功能及验证方法的一致；规定了车用操作系统间健康监控的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-8	整车时钟服务技术要求与试验方法	团标	定义整车时钟服务功能的技术要求及试验方法，确保整车时钟服务功能及验证方法的一致；规定了车用操作系统间时钟同步的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-9	整车诊断管理技术要求与试验方法	团标	定义整车诊断管理功能的技术要求及试验方法，确保诊断电源管理功能及验证方法	

				的一致；规定了车用操作系统间诊断管理的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-10	整车 OTA 管理技术要求与试验方法	团标		定义整车 OTA 管理功能的技术要求及试验方法，确保整车 OTA 管理功能及验证方法的一致；规定了车用操作系统间软件升级控制的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-11	整车版本管理技术要求与试验方法	团标		定义整车版本管理功能的技术要求及试验方法，确保整车版本管理功能及验证方法的一致；规定了车用操作系统间版本采集的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-12	整车数据采集技术要求与试验方法	团标		定义整车数据采集功能的技术要求及试验方法，确保整车数据采集功能及验证方法的一致；规定了车用操作系统间数据采集的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-13	整车调试系统技术要求与试验方法	团标		定义整车调试系统功能的技术要求及试验方法，确保整车调试系统功能及验证方法的一致	
230-14	整车日志功能技术与试验方法	团标		定义整车日志功能的技术要求及试验方法，确保整车日志功能及验证方法的一致；规定了车用操作系统间日志采集的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	

2.2 应用软件

随着汽车应用软件的发展历程经历了电子化、多媒体化、智能化和全面智能网联化的发展，可以概括为以下几个阶段：

90年代初期，传统的嵌入式系统阶段，汽车应用软件主要是嵌入在汽车的各个电子控制单元中，用于管理和控制汽车的基本功能，如刹车系统、车身电子等。这些软件通常是针对特定功能的专用代码，往往是闭源和定制化的。

2000年，车载信息娱乐系统阶段，随着技术的进步和用户需求的变化，车载信息娱乐系统登上历史舞台。这些系统提供了更多的车内娱乐和信息服务，如音频播放、导航、蓝牙连接等。

此阶段的软件开发开始更加注重用户界面和用户体验，并且开始引入第三方应用软件和互联网服务。

2014年，互联网和智能化阶段：随着互联网和人工智能技术的普及，汽车应用软件逐渐向互联网和智能化方向发展。智能手机的普及使得车辆与互联网相连，并且能够实现远程控制、车辆定位、实时交通信息等功能。此外，人工智能技术的应用也增强了车辆的智能化和自主化能力，如语音识别、智能导航和自动驾驶等。

2023年，进入软件定义汽车阶段：软件定义汽车（SDV）是未来发展的方向，将更多的功能和特性通过软件定义和控制实现。这意味着汽车的功能不再完全依赖于硬件，而是通过软件的配置、更新和交互来实现。软件定义汽车可以实现更高级别的自动驾驶、车辆个性化定制、人机界面灵活性等。

在整个发展历程中，汽车应用软件逐渐从传统的嵌入式系统过渡到基于互联网和智能化的软件，为车辆带来更多的功能和服务，并提供更好的用户体验。未来，软件定义汽车将成为主流趋势，驱动汽车行业的持续创新和进步。根据应用软件功能及未来技术发展趋势，整个应用软件分别车控域应用软件、智驾域应用软件、智能座舱应用软件、整车应用软件、手机及车联网应用软件和云控应用软件6大部分，应用软件整体逻辑架构如图8。



图8 应用软件标准逻辑框架

2.2.1 车控域应用软件

2.2.1.1 发展历程

车控域应用软件是指用于控制和管理汽车各个部件的电子控制系统。车辆控制软件主要包括五大部分：动力系统、底盘系统、车身系统、电池管理系统、手机智能控车系统。随着汽车电子化技术的不断发展，汽车控制器系统也经历了多个阶段的演进。

1) 传统机械控制系统：在汽车电子化技术未普及之前，汽车的控制系统主要采用机械和液压控制方式。例如，刹车系统采用机械或液压方式控制刹车片与车轮的接触，而发动机则通过机械方式控制进气量和点火时间。

2) 电子控制单元的出现：20世纪80年代，电子控制单元（ECU）开始在汽车控制系统中得到广泛应用。ECU是一种可以控制多个汽车系统的微型计算机，它可以读取传感器的数据，计算出最佳控制方式，并通过执行器来控制相关系统的操作。

3) 系统集成和网络化：随着ECU的应用，汽车控制系统开始向更加集成化和网络化的方向发展。例如，车辆动力总成控制系统集成了发动机、变速器等多个控制系统，通过CAN总线进行互联和协同控制。

4) 智能化和自动化：近年来，随着人工智能和机器学习技术的应用，汽车控制系统开始向智能化和自动化的方向发展。例如，自动驾驶技术利用传感器、摄像头和雷达等技术实现车辆的自动驾驶，而智能制动系统可以根据路况和驾驶员行为进行智能制动控制，提高了驾驶的安全性和舒适度。

2.2.1.2 特点

车控域应用软件运行在广义操作系统之上，具体负责功能实现，主要面向动力系统和底盘电控。车控域应用软件具有以下特点：

(1) 高实时性：车控域应用软件通常需要实时响应驾驶员指令并实时控制车辆状态，因此需要具有高实时性，以确保软件在实时环境下运行。许多针对发动机和行驶安全的系统必须具有良好的“实时性”，即控制器的响应必须与实际的物理过程相一致。在对非常快的物理过程进行调节时，如发动机控制或者汽车行驶动力学系统控制，计算速度必须非常快，当前高端车控芯片的主频可达300MHz。在很多情况下对可靠性的要求会很高，尤其是与安全相关的功能。此外，还要采用一个复杂的诊断系统监控软件和电子系统的运行。

(2) 可靠性：车控域应用软件需要确保高度的可靠性，以保证软件在各种工况下均能正常运行，避免由于软件故障导致车辆非预期行驶造成车辆事故。

(3) 安全性：车控域应用软件的安全性是极其重要的，包括功能安全、信息安全等，为此在开发过程中应遵循相应标准，以保证开发出的软件的安全性。为了顺应高阶自动驾驶的需要，底盘技术实现汽车底盘的电动化、智能化、包括线控制动技术、线控转向技术、线控悬架技术、线控油门/驱动技术等四大线控技术。这也意味着高度的智能化和电气化，假如车辆电气系统出现故障，在没有传统机械结构保底的情况下，很容易就会发生失控，因此对功能安全要求更高。

(4) 复杂性：车控域应用软件中包括大量的接口信号、复杂的逻辑状态和先进的控制算法等，从而满足相应的功能和性能要求，因此具有复杂性。在分布式架构下，制动、转向、驱动为独立子系统，由单独的 ECU 分别控制，由于底盘运动执行信号来自驾驶员，各子系统协同需求较低，EPS、ABS、ESP、线控驱动等子系统在分布式架构下可独立应用。随着整车智能化程度的提升，各子系统难协同、网络结构复杂、软硬件耦合关系强。

2.2.1.3 发展现状

早期车控域应用软件的开发模式还是瀑布式的模型，虽然瀑布模型可以按工序将问题化简，将功能的实现与设计分开，便于协作，即采用结构化的分析与设计方法讲逻辑实现与物理实现分开。虽然可以保证质量、明确分工、用于迭代模型且模块化、标准化。但各个阶段划分

完全固定，阶段之间产生大量文档极大的增加工作量；模型开发是线性的，用户只有在整个过程末期才能看到开发成果，增加了开发风险；同时面对用户的需求变更不利于进行修改。后续随着V字开发模型的出现，解决了瀑布模型中的严格分离很难实现的困境；软件回溯较为方便快捷；测试提前，可以更加提早发现问题；问题的可追溯性强，测试过程与设计过程一一对应。同时在应对客户需求变更方面也比瀑布模型更加适合。

但随着目前汽车日新月异的需求和快速更新迭代的节奏，V模型的项目反应速度也不能满足了。伴随着汽车ECU的发展，车内网络总线技术，软件开发方式也在发生一些新的变化。并且车联网、OTA、域控制器普及之后，汽车软件开发模型逐渐向Agile模型转变。但目前应用最为广泛的依旧是V模型。

针对车控域应用软件的V模型，在ISO 26262、ASPICE等都在这方面给出了较详细的介绍和规定。架构设计上三类软件区别不大，分硬件架构设计和软件架构设计。在硬件架构上，由传统的分布式EE架构，逐步向域架构以及中央化架构转变。在软件架构上目前依旧是以Signal-Oriented Architecture（面向信号的开发方式）为主，但也有部分向Service-Oriented Architecture（面向服务的开发方式）发生转变，主要是因为其具有松耦合，接口标准化、易扩展等特性。

随着SOA架构汽车软件的发展，Arm公司发布的开源软件架构SOAFEE（Scalable Open Architecture for Embedded Edge, SOAFEE）-面向嵌入式边缘的可扩展开放架构，将云原生的技术的一些优点引入汽车软件中。其中主要核心是将容器化技术引入车上，汽车的应用软件将在容器中运行，让应用软件更容易满足ISO-26262的相关要求。关于容器化技术的相关标准有由Linux基金会管理的OCI（Open Container Initiative）标准。OCI规范主要包含3个部分：the Runtime Specification (runtime-spec)运行时规范，the Image Specification (image-spec)镜像规范与the Distribution Specification (distribution-spec)分发规范。

在软件设计方面，车控域应用软件大多基于模型开发，应用较为广泛的是MATLAB自带的MAAB规范，不同版本的MATLAB细微处可能会有些不同，此外还有常用的有MISRA C-汽车制造

业嵌入式C编程标准、MISRA C++-汽车制造业嵌入式C++编程标准和GB/T 28169 嵌入式软件 C语言编码规范。

软件测试主要包括，单元测试、可靠性测试、HIL测试、集成测试、功能性测试等。上述三类软件在测试过程中涉及的内容可能会略有不同，同时测试的指标也有所不同；目前没有统一的测试标准规范。

表 8 车控域应用软件类标准统计

发布日期	标准名称	部门/机构	相关内容
2018	ISO 26262 Road vehicles - Functional safety	国际标准化组织	<p>1、安全要求：ISO 26262 要求在整个软件开发生命周期中，包括要求分析、设计、实现、验证和确认等各个阶段都要考虑安全性。开发人员需要遵循一定的安全要求来确保软件的安全性和可靠性。</p> <p>2、安全文档：ISO 26262 要求开发人员编写安全文档来记录整个软件开发过程，包括需求分析、设计、实现、验证和确认等各个阶段。这有助于开发人员更好地管理和追踪软件开发过程，以确保软件的安全性和可靠性。</p> <p>3、风险分析：ISO 26262 要求开发人员进行风险分析，以确定可能导致汽车电子系统故障的原因和影响。在软件工具开发中，开发人员需要考虑软件工具对</p>

			<p>汽车电子系统的影响，并采取相应的安全措施来减少风险。</p> <p>4、软件测试：ISO 26262 要求对软件进行测试以验证其安全性和可靠性。在软件工具开发中，开发人员需要设计和实现有效的测试策略来确保软件的安全性和可靠性。</p>
2016	GB/Z 33103 道路车辆 车用嵌入式软件开发指南	国家标准化 管理委员会	<p>1、软件生存周期：汽车软件设计准则、项目计划、完整性、需求规格说明、设计、编程、测试、产品支持。</p> <p>2、软件质量计划：管理职责、教育与经验、软件开发的人的因素、软件质量保证、文档编制要求、分包。</p>
2017	Automotive SPICE Process Assessment / Reference Model	德国汽车工 业联合会 (VDA) 质量管理中 心 (QMC)	<p>ASPICE 将汽车软件开发过程划分为以下过程：</p> <ol style="list-style-type: none"> 1、软件需求分析 2、软件架构设计 3、软件详细设计和单元构建 4、软件单元验证 5、软件集成和集成测试 6、软件合格性测试
2012	MISRA C:2012 Guidelines for the use of the C language in critical	汽车工业软 件可靠性协 会（英国）	<p>1、代码静态分析：MISRA C 规范提供了针对 C 语言编程的静态代码分析规则，以确保编写的代码符合 MISRA C 规范。</p>

	systems		<p>2、自动化测试工具：MISRA C 规范要求 在软件开发过程中进行自动化测试，以 验证代码是否符合 MISRA C 规范。</p> <p>3、静态检查工具：MISRA C 规范要求 在开发过程中使用静态检查工具进行代码 检查。这些工具可以检测代码中的违规 操作，并为程序员提供修复建议。</p> <p>4、代码生成工具：MISRA C 规范要求使 用代码生成工具生成符合规范的代码。</p> <p>5、软件验证工具：MISRA C 规范要求进 行软件验证以确保代码符合规范。</p>
2008	MISRA C++	<p>汽车工业软 件可靠性协 会（英国）</p>	<p>1、类型和运算符的正确使用：MISRA C++标准强制执行对变量和运算符的正确 使用。</p> <p>2、内存管理：汽车软件工具通常需要使 用动态内存分配和释放，这可能导致内 存泄漏和其他内存相关的问题。MISRA C++标准规定了对内存管理的最佳实践， 帮助开发人员避免这些问题。</p> <p>3、异常处理：异常处理是汽车软件工具 开发中的重要问题之一。MISRA C++标 准提供了关于异常处理的最佳实践，帮 助开发人员正确地使用异常处理机制， 以确保代码的可靠性和安全性。</p>

			<p>4、多线程和并发编程：现代汽车软件工具通常需要使用多线程和并发编程。</p> <p>MISRA C++标准提供了对这些主题的最佳实践和指导，帮助开发人员编写高质量、可靠的多线程代码。</p>
2011	GB/T 28169 嵌入式软件 C 语言编码规范	国家标准化 管理委员会	<p>本标准对 C 语言编码规范提出了要求，主要包括：总体要求、内存空间管理类、中断处理类、系统接口类、硬件系统初始化类、软件模块初始化类、版面书写类、声明定义类、控制语句类、类型转换类、指针、数组使用类、运算处理类、函数使用类、冗余类、程序效率类。</p>
2011	GB/T 28172 嵌入式软件 质量保证要求	国家标准化 管理委员会	<p>1、一般要求：嵌入式软件质量保证综述、质量保证的要素、开发文档、开发项目的问题报告。</p> <p>2、嵌入式软件开发的质量保证要求：嵌入式软件开发项目生存周期的阶段划分、质量保证过程涉及的角色及职责、嵌入式软件开发的主要质量保证活动、嵌入式软件开发中各阶段的质量保证要求、其他要求。</p>
2011	GB/T 28171 嵌入式软件 可靠性测试方法	国家标准化 管理委员会	<p>本标准规定了嵌入式软件可靠性测试的测试目的、测试环境、测试内容、测试</p>

			方法。
2020	GB/T 38628-2020 《信息安全技术 汽车电子系统网络安全指南》	国家标准化 管理委员会	1、软件架构：给出了汽车电子系统网络安全活动框架，以及在此框架下的汽车电子系统网络安全活动、组织管理和支撑保障等方面的建议。指导相关人员在从事汽车电子系统的设计开发、生产、运行和服务等过程中满足基本的网络安全需求
2020	MAAB	MathWork	1、建模规范：明确 Simulink 和 Stateflow 建模中的一些基本规则；用于提高模型的可读性、实现仿真和验证、提高代码生成效率，确保生成代码的健壮性
2009	Google C++ Style Guide	Google 开 源	1、编写规范：提高代码的可读性、规范性、安全性
2001	PEP8	Python 社区 开源	1、编写规范：提高代码的可读性、规范性、安全性

2.2.1.4 标准需求

(1) 软件架构：目前的标准（如AUTOSAR）主要介绍了软件的整体架构，很少聚焦于应用软件的局部架构，尤其是国内的标准在应用软件的架构设计方法及要求等方面很少涉及。

(2) 软件设计：国际上有MISRA C语言编码规范、MATLAB建模规范等标准，国内多是参考国际标准，且更新速度较慢，时效性和先进性较差。此外，随着软件的发展，当前的编程语言已不再局限于C，诸如C++等语言越来越多的在车载软件中应用，然而目前还没有针对这些语言编码规范的国标。

(3) 功能及性能特性、技术要求：目前国内外标准通常是针对汽车某个系统或总成提出技术要求，尚未在相应功能的软件层面提出相关技术要求。随着“软件定义汽车”的趋势发展，当前的标准已经不能满足要求，应从汽车软件实现的功能、性能指标、技术要求等方面制订相应的标准，指导软件开发。

(4) 测评方法：对于通用的软件及嵌入式软件，有一些国标介绍了验证与确认方法、可靠性测试方法等，但是目前缺少针对汽车软件的测试方法标准。此外，标准中较少提到测试指标的合理值或推荐值，有待后续完善。

(5) 开发流程：国内外相关标准比较多，内容也比较全面，但开发模式多为传统的V流程，在当前这个汽车软件快速变革的阶段可能存在不适用的情况，越来越多的汽车企业采用敏捷开发模式，然而这种开发模式尚未形成标准。

2.2.1.5 标准体系框架及分类说明

车控域应用软件标准体系横向以传感器抽象层、策略与算法层、执行器抽象层、资源与应用层四个层次为基础，纵向以功能安全和网络安全通用规范技术为支撑，形成“四横两纵”的核心技术架构，完整呈现标准体系的技术逻辑，明确各项标准在车控域应用软件产业技术体系中的地位和作用（见图9）。

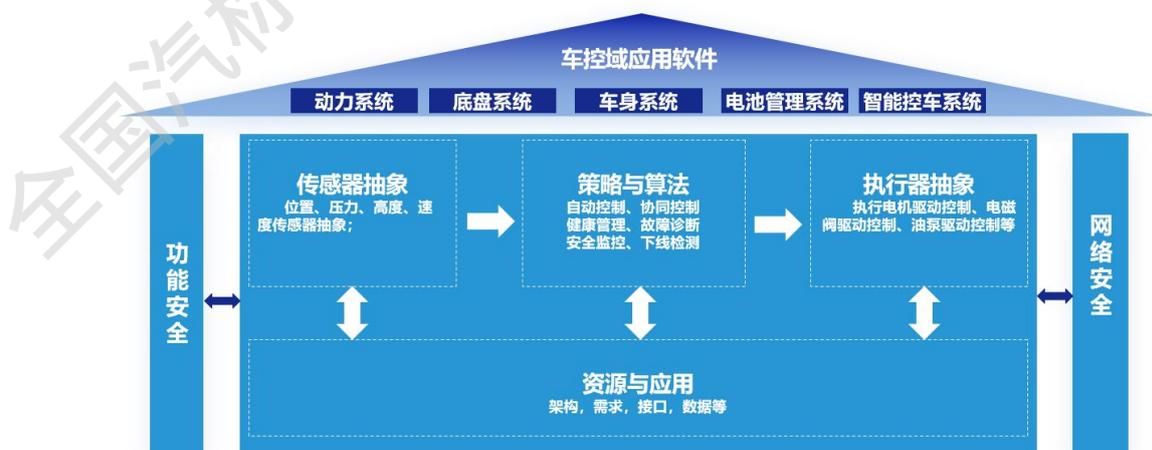


图9 车控域应用软件技术逻辑框架

按照车控域应用软件技术逻辑架构，综合考虑不同功能、产品和技术类型、各子系统之间的交互关系，将车控域应用软件标准体系划分为两个层级。其中，第一层级规定了车控应用软件标准体系的基本分类，即基础、通用规范、产品与技术应用三个部分；第二层级根据标准内容范围和技术等级，细分形成9个二级分类，从而形成了逻辑清晰、内容完整、结构合理、界限分明的标准体系框架，见图10。

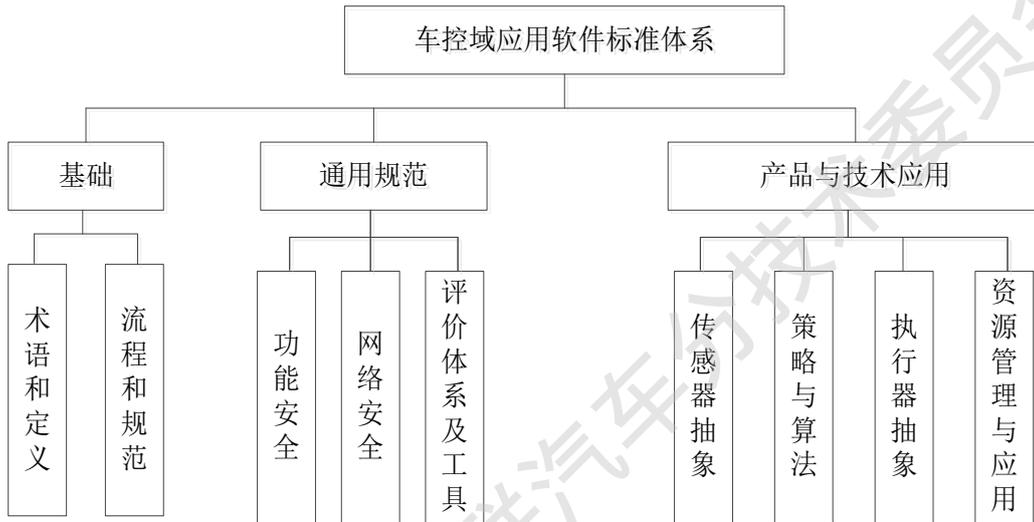


图10 车控域应用软件标准体系

车控域应用软件类标准体系主要包括基础、通用规范、产品与技术应用等三个部分，共有标准项目23个：

1. 基础

基础标准主要用于统一车控域汽车软件领域相关概念，厘清标准化对象及边界，建立标准化对象的统一表达方式，包括流程和规范。

(1) 流程和规范

目前大多数都是依照ISO26262等相关标准的要求去开发以及V字开发流程，该部分内容包括车控域应用软件需求分析、开发设计、测试验证和维护等流程。

2. 通用规范

通用规范标准侧重于提出适用于车控域汽车软件的通用要求和共性评价准则，主要包括功能安全、网络安全、评价体系及工具等部分。

(1) 功能安全

功能安全标准用于确保电子电气系统故障（包括软件、硬件、系统故障）等功能异常的情况下，车辆能够安全运行、不会引发安全风险，主要包括车控域应用软件功能安全等级要求建议与设计方法，旨在规范功能安全开发要求及方法。

(2) 网络安全

汽车网络安全标准基于车联网复杂环境，以车端为核心运用纵深防御理念保护其免受网络攻击或缓解网络安全风险，主要包括车控域应用软件网络与数字安全等级要求与设计方法，旨在规范网络与数字安全开发效率方法。

(3) 评价体系及工具

评价体系及工具标准规范了车控域应用软件中评价体系的关键要素，创建测试过程、测试方法、测试工具和测试标准，并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求。

3. 产品与技术应用

产品与技术应用标准主要涵盖传感器抽象、策略与算法、执行器抽象、资源管理与应用等车控域应用核心产品与技术应用。

(1) 传感器抽象

包含传感器信号解析、网络信号定义与解析，信号融合、处理与诊断，旨在规范车控软件输入信号设计。

(2) 策略与算法

规定车控软件典型功能规范与算法，旨在提升车控软件应用功能规范性。

(3) 执行器抽象

包含执行器驱动控制算法、一致性与可靠性设计，旨在提高执行器控制性能与可靠性

(4) 资源管理与应用

资源管理与应用侧重于规范平台架构、需求设计等车控域应用软件核心共性资源的功能、性能及应用要求，旨在规范车控软件架构接口与需求设计，提高车控软件开发通用性及质量。

表9 车控域应用软件标准体系

标准项目及分类		标准类型	标准描述	备注
应用软件（300）				
车控域应用软件（310）				
310-1	车控域应用软件编码规范	团标推荐	本标准包括车控域应用软件需求分析、开发设计、测试验证和维护等流程、旨在提高产品开发流程的规范性、效率性和质量。	
310-2	车控域应用软件功能安全技术要求	团标推荐	本标准包括车控域应用软件功能安全等级要求建议与设计方法，旨在规范功能安全相关开发要求及方法。	
310-3	车控域应用软件信息安全技术要求	团标推荐	本标准包括车控域应用软件信息安全要求与设计方法，旨在规范信息安全相关开发要求及方法。	
310-4	车控域应用软件功能测试规范	团标推荐	包含测试过程、测试方法、测试工具和测试标准，并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求	
310-5	车控域应用软件编码测试规范	团标推荐		
310-6	传感器抽象接口设计规范	团标推荐	包含传感器信号解析、网络信号定义与解析，信号融合、处理与诊断，旨在规范车控软件输入信号设计；	
310-7	传感器抽象设计规范	团标推荐		
310-8	驱动软件技术要求	团标推荐	包含执行器驱动控制算法、一致性与可靠性设计，旨在提高执行器控制性能与可靠性；	
310-9	执行器抽象接口设计规范	团标推荐		
310-10	执行器抽象设计规范	团标推荐		
310-11	车控域应用软件数据埋点设计规范	团标推荐	包含车控软件典型功能规范与算法，旨在提升车控软件应用功能规范性	
310-12	车控域应用软件健康管理设计规范	团标推荐		
310-13	车控域应用软件 EOL 设计规范	团标推荐		

310-14	车控域应用软件故障处理设计规范	团标推荐	包含车控软件架构、需求设计规范，旨在规范车控软件架构接口与需求设计，提高车控软件开发通用性及质量；
310-15	车控域应用软件动力系统功能规范	团标推荐	
310-16	车控域应用软件接口设计规范	团标推荐	
310-17	车控域应用软件网络接口设计规范	团标推荐	
310-18	车控域应用软件诊断接口设计规范	团标推荐	
310-19	车控域应用软件平台软件需求规范	团标推荐	
310-20	车控域应用软件电子电气需求规范	团标推荐	

2.2.2 智驾域应用软件

2.2.2.1 发展历程

智驾域软件是利用汽车上各式各样的传感器和电子控制系统，采用人工智能、通信、软件算法等技术实现的一系列的软件功能，从而改善出行体验、提高出行效率、保障出行安全。自动驾驶的基本过程基本上三个步骤：感知、决策、控制。关键技术是自动驾驶的软件算法与模型，通过融合各个传感器的数据，不同的算法和支撑软件计算得到所需的自动驾驶方案。目前两种主流的技术路线就是：1、以摄像头为主导的多传感器融合方案，例如：特斯拉等；2、以谷歌、百度为代表的以激光雷达为主导，其他传感器为辅助的技术方案。根据国家标准《汽车驾驶自动化分级》，将汽车驾驶自动化分为0-5级：

- 1) L0级应急辅助：车道建立预警、前碰撞预警、自动紧急制动等。

-
- 2) L1级部分驾驶辅助：自适应巡航控制、半自动泊车等
 - 3) L2级组合驾驶辅助：集成式自适应巡航控制、高速公路辅助、驾驶人触发换道、全自动泊车、遥控代客泊车等。
 - 4) L3级有条件自动驾驶：交通拥堵自动驾驶、高速公路自动驾驶等。
 - 5) L4级高度自动驾驶：远程代客泊车、最后一公里代客泊车、无人驾驶出租车、城区自动驾驶等。
 - 6) L5级完全自动驾驶：全场景无人驾驶。

随着汽车智能化的发展，行业已实现了应急辅助、部分驾驶辅助阶段和组合驾驶辅助的规模化量产，正处于由组合驾驶辅助向有条件自动驾驶的过渡阶段。中国的自动驾驶技术处于高速发展阶段，正在不断的迭代升级中，2007-2015年L1级别的功能已经实现商业化并广泛普及；2014-2020年L2级别的组合驾驶辅助系统已经成熟并广泛使用；2025年将计划实现L3级别的自动驾驶，目前技术已能够实现，还处于测试阶段；2030-2040年计划实现L4/L5级别的自动驾驶。

2.2.2.2 特点

智能驾驶经过近几年的高速发展，硬件从分布式ECU架构发展到域控制EE架构，正向中央集中式EE架构发展，硬件的集中也带了软件的集中化，智驾域的软件也呈现出如下特征：

(1) 高集成：因硬件的集中化，原本分布在各个ECU的软件都集中到域控制内，在一个域控制器内部集成感知融合、规划决策、控制交互等核心软件，同时还包括操作系统、基础库和中间件等，因此具有高集成的特点。

(2) 高安全：智驾软件的安全性非常重要，必须满足功能安全和信息安全的相关标准，从而保证驾乘人员的人身安全和数据安全。

(3) 高体验：智驾软件发展到当前阶段，实现相应的功能是基本要求，提升用户的驾乘体验是现阶段的重点，因此在软件设计时需充分考虑用户体验，而体验评价也作为软件验收的一个重要指标。

(4) 快迭代：随着OTA技术在车载软件中的成熟，智驾软件已经可以像手机一样便捷的进行OTA升级，软件开发方可以便捷的进行问题修复和新功能推送。快速迭代已经成为智驾软件高速进化的主要推手。

2.2.2.3 发展现状

智驾软件发展到当前阶段，已经不在仅仅是ECU上的简单的几百至几千行代码，目前已经涵盖操作系统、基础软件（基础库、分布式通信、基础服务）、算法设计（感知、定位、规划等）、云服务（仿真、高精地图等）等，智驾软件已经如同智能手机一样，是一个真正的软件系统。

目前主流车控操作系统基本都兼容OSEK/VDX和AUTOSAR这两类汽车电子软件标准，它们定义了操作系统的技术规范主流的车控操作系统都兼容这两类标准，当前主流的AUTOSAR分为两个平台，即Classic平台和Adaptive平台，分别对应传统控制类车辆电子系统与对应自动驾驶的高性能类车载电子系统。在国内，依托国家“核高基”课题，i-soft公司也开发了符合AUTOSAR标准的操作系统和基础软件，并成功应用与自主品牌和新能源量产车型，总体而言，我国的车载OS仍处于跟随海外发展阶段。

智驾软件通常也采用了模块化设计原则，为了给开发提供更多便利，中间件成为智驾软件的一个基础软件，用于在应用与操作系统、数据库之间，解决分布式环境下数据传输、数据访问、应用调度、系统构建和系统集成、流程管理等问题，帮助对接整车OEM、底层OS与核心硬件厂商，使标准化产品能向不同层级的用户提供个性化的接口调用。

智驾软件的算法主要包括感知、决策、执行三个层次，大体上可以分为传统算法和深度学习算法，传统算法多为基于规则的解决方案，以Mobileye为代表，深度学习算法为近年来人工智能的发展而得到了重视，处于快速发展阶段，国内的地平线、百度等较为突出。

2.2.2.4 标准需求

(1) 软件架构：汽车正加快智能化进程，传统通讯网络及软件架构设计中扩展性差、升级和移植成本高等问题日益凸显。为了实现软件定义汽车，智能汽车软件架构需向SOA转型升级，相关的标准尚未形成。

(2) 软件设计：智驾软件的开发主要集中为MATLAB、C、C++、python等，相关的代码规范和开发标准主要参考其他行业标准，未对智驾软件本身的特殊性做相关要求，有待完善后形成对应的标准。

(3) 功能及性能特性、技术要求：相关的国际/国内标准正在不断完善中。

(4) 测评方法：智驾软件的评测方法一方面是借鉴其他软件行业和其他车载软件的评测方法，各厂家也结合智驾软件的特点做了一些特殊的评测，但未有相对完整的评测方案，有待进一步完善。

(5) 软件安全：智驾软件区别于座舱软件，对安全性要求较高，目前参照的标准并未做特殊区分，有待后续完善。

2.2.2.5 标准体系框架及分类说明

智驾类应用软件的分类原则也是划分为两个层级。其中，第一层级规定了智驾域应用软件标准体系的基本分类，即基础、通用规范、产品与技术应用三个部分；第二层级根据标准内容范围和技术等级，细分形成9个二级分类，如图11。

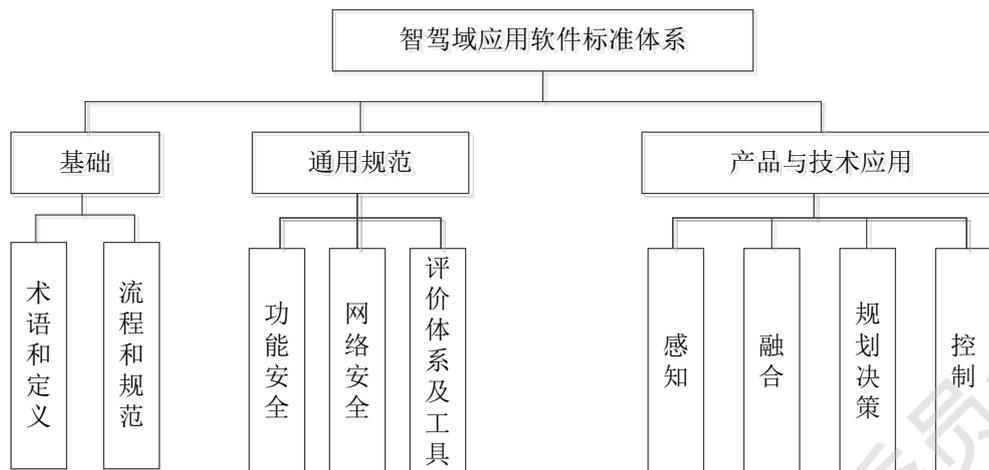


图11 智驾域应用软件标准体系

标准分类说明部分参考车控应用软件，包括软件编码规范、功能安全技术要求、信息安全技术要求等。

表 10 智驾域应用软件标准体系

标准项目及分类		标准类型	标准描述	备注
应用软件（300）				
智驾域应用软件（320）				
320-1	智驾域应用软件编码规范	团标推荐	本标准包括智驾域应用软件需求分析、开发设计、测试验证和维护等流程、旨在提高产品开发流程的规范性、效率性和质量。	
320-2	智驾域应用软件功能安全技术要求	团标推荐	本标准包括智驾域应用软件功能安全等级要求建议与设计方法，旨在规范功能安全相关开发要求及方法。	
320-3	智驾域应用软件信息安全技术要求	团标推荐	本标准包括智驾域应用软件信息安全要求与设计方法，旨在规范信息安全相关开发要求及方法。	
320-4	智驾域应用软件功能测试规范	团标推荐	包含测试过程、测试方法、测试工具和测试标准，并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求	
320-5	智驾域应用软件编码测试规范	团标推荐		
320-6	智驾感知软件设计规范	团标推荐	包含智驾软件在感知，融合，规划决策，和控制方面典型功能与算法设计规范	
320-7	智驾融合软件设计规范	团标推荐		
320-8	智驾规划决策软件设计规范	团标推荐		
320-9	智驾类控制软件设计规范	团标推荐		

2.2.3 座舱域应用软件

2.2.3.1 发展历程

汽车座舱作为与驾乘人员直接接触的空间生态，更易被用户感知到，因此是车企在寻求差异化、品牌化发展当中的重点布局领域。智能座舱主要涵盖座舱内饰和座舱电子领域的创新与联动，从消费者应用场景角度出发而构建的人机交互(HMI)体系。当前智能座舱主要满足座舱功能需求，在原有的基础上，对现有的功能或是分散信息进行整合，提升座舱性能，改善人机交互方式，提供数字化服务。随着汽车产业的智能化变革，汽车座舱也逐渐由电子化向智能化发展，成为整车智能化发展过程中的核心构成。智能座舱软件的发展演进主要包含以下几个阶段：

1) 功能座舱阶段：上世纪80年代，传统供应商博世联合英特尔开发出CAN总线系统并用于车内ECU的数据通信，标志着汽车座舱生态的开始。早期座舱信息多服务于驾驶者，显示基本驾驶信息，2001年宝马引入了中央显示屏，车载信息娱乐系统(IVI)成为智能座舱最具代表的系统，提供包括车载导航、蓝牙、媒体播放等功能。在这个阶段，座舱的便利功能主要依靠将电子设备“搬”进车内来实现，设备的操作基本都是物理按键形式，后期逐步加入触控和语音控制等方式，但是功能之间的整合并不合理，体验也不够流畅。

2) 智能座舱阶段：随着IT及消费电子产业的繁荣，用户可以充分享受平板电脑、智能手机等智能终端带来的丰富、优质的交互体验；用户对于这种体验形成习惯后，对车载交互体验也提出了新需求，大尺寸中控屏开始成为标配，通过融合“视觉”、“语音”等模态的感知数据，做到更精准、更智能、更人性化的交互逐渐取代了“物理按键交互”的模式。

智能座舱是整合液晶仪表、信息娱乐系统、HUD、流媒体后视镜等多种电子部件的复杂系统，能够实现多屏互动以及语音识别、手势识别等多模态人机交互，并将融入 OTA 技术、云

端智能等功能。智能座舱系统通过独立感知层，能够拿到足够的感知数据，例如车内视觉（光学）、语音（声学）以及方向盘、刹车踏板、油门踏板、档位、安全带等底盘和车身数据，利用生物识别技术（车舱内主要是人脸识别、声音识别），来综合判断驾驶员(或其他乘员)的生理状态（人像、脸部特征等）和行为状态（驾驶行为、声音、肢体行为），做到“理解”人。随后根据具体场景，推送交互请求，如提供咨询信息，提供车辆状态信息，提供“车对人”主动交互，降低驾驶员在驾驶过程中“人对车”的交互负担，改善交互体验。

4) 移动生活空间阶段：未来汽车使用场景将更加丰富化、生活化，基于车辆位置信息，融合信息、娱乐、订餐、互联等功能为消费者提供更加便捷的体验。与其他“空间”不同，车辆自带可移动的属性，可以通过网联功能轻松实现线上与线下体验的无缝连接，为消费者提供更加便捷的体验。

2.2.3.2 特点

智能座舱应用层软件是指为实现具体座舱功能，给用户可提供感知服务而开发的软件，例如仪表盘显示、车载信息娱乐系统、抬头显示系统等。智能座舱应用层软件具有以下特点：

(1) 多样化的交互方式：为了满足不同用户的需求和应用场景，智能座舱应用软件需要具备多种交互方式，如触摸屏、语音、手势等。这些交互方式需要考虑驾驶员的安全和便利性，必须具备简单易用的操作界面和操作方式，以防止驾驶员在使用过程中分心或者产生误操作。同时，操作界面和操作方式需要与车辆的其他系统进行协同工作，以提供更加全面和优质的服务。

(2) 界面美观和易于定制：智能座舱应用软件需要具备美观和易于定制的界面，以满足不同的需求和用户喜好，提高用户的满意度和使用体验。例如，仪表盘显示系统需要具备美观的显示效果和易于定制的显示内容，车载娱乐系统需要具备多样化的主题和界面风格，驾驶员辅助系统需要具备易于调整和配置的界面等。

(3) 需要满足不同功能安全级别要求：当前智能座舱是整合液晶仪表、信息娱乐系统、HUD、流媒体后视镜等多种电子部件的复杂系统，因此对于不同子系统会有不同的功能安全级别的要求。比如通常液晶仪表会要求满足功能安全ASIL-B等级，而信息娱乐系统满足QM等级即可。

(4) 信息安全要求高：智能座舱应用层软件需要确保车辆系统的信息安全，例如避免被黑客攻击、保护车主的隐私等。智能座舱应用层软件需要采用多种安全技术，如加密、认证、权限控制等，以确保车辆系统的信息安全。

2.2.3.3 发展现状

在软件定义汽车大背景下，基于SOA架构，软硬解耦成为必然趋势。智能座舱应用软件产品也正从碎片化向模块平台化逐步演进，通过软件平台的标准化、模块化、可复用，可显著缩短软件开发周期，简化开发流程，同时应用软件与服务可根据不同需求定制，为用户提供差异化功能及体验。

目前，高端智能座舱已经可以实现的功能主要包括：个性化，即可以智能识别驾驶员身份，并自动实现车内相关配置；可以与消费电子端APP实现账号打通；支持包括语音、手势在内的多种交互方式，可实现一定程度的自然语义识别和交互；支持驾驶员监控，监测驾驶员疲劳状况，并给出相应提示和建议；支持无线CarPlay、CarLife互联方案，同时支持手机无线充电功能；基于电动车使用场景，含有电池低电量提醒并智能导航附近充电桩，整合ADAS高级驾驶辅助系统，并集成高清360°环视及驾驶员监控系统。

在进行智能座舱应用软件开发时，改善车辆的用户界面和用户体验（UI/UX）是最优先被考虑的事情，通过酷炫的界面为用户提供沉浸式的使用体验。目前行业内多选用Kanzi、Qt等跨平台人机界面（HMI）开发工具和开发框架实现统一智能座舱平台的设计、开发。

2.2.3.4 标准需求

伴随座舱应用软件快速发展，对其标准化工作提出迫切需求，主要体现在以下几个方面：

- (1) 数据和接口格式标准化需求：智能座舱应用软件的发展需要开放的生态，数量繁多的软件产品如果数据格式不统一、接口不兼容，就会导致数据互联互通困难，形成数据流通壁垒，造成大量的数据烟囱和信息孤岛。
- (2) 关键技术、典型应用的标准化需求：当前，部分软件头部企业为提升各自的生态主导力，分别形成了所在领域的事实标准，以提升企业用户对其产品的依赖度，培育形成各自的封闭生态圈，会严重制约行业发展。
- (3) 软件文档标准化需求：软件文档标准化可以提高软件的可维护性、可读性、可测试性、可重用性和质量，从而降低软件开发和维护的成本，提高软件的稳定性和可靠性，为软件开发提供坚实的基础。
- (4) 软件测评标准化需求：软件测试评价是软件工程的一个重要活动，是获取软件质量数据、掌握软件状态的重要途径。软件测评标准化有助于提高软件测试工作质量和工作效率，进而促进软件研发过程的规范和产品质量的提升。

软件成本度量标准化需求：随着汽车行业软件开发模式的变革，需要提供科学的成本控制依据和规范的成本量化方法，以便应用在软件项目预算、招投标、项目计划等活动中。

2.2.3.5 标准体系框架及分类说明

座舱类应用软件的分类原则也是划分为两个层级。其中，第一层级规定了智驾域应用软件标准体系的基本分类，即基础、通用规范、产品与技术应用三个部分；第二层级根据标准内容范围和技术等级，细分形成9个二级分类，如图12。

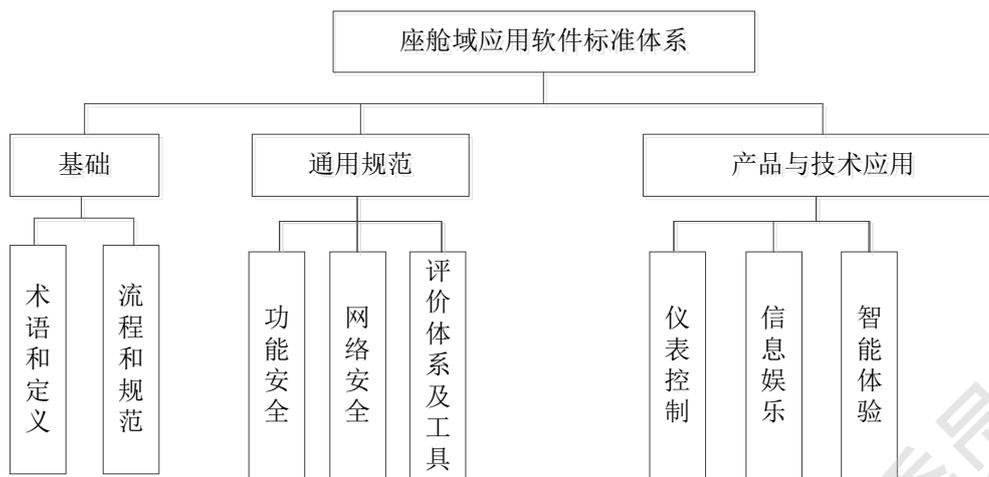


图12 座舱域应用软件标准体系

表11 座舱域应用软件标准体系

标准项目及分类		标准类型	标准描述	备注
应用软件（300）				
座舱域应用软件（330）				
330-1	座舱域应用软件编码规范	团标推荐	本标准包括座舱域应用软件需求分析、开发设计、测试验证和维护等流程、旨在提高产品开发流程的规范性、效率性和质量。	
330-2	座舱域应用软件功能安全技术要求	团标推荐	本标准包括座舱域应用软件功能安全等级要求建议与设计方法，旨在规范功能安全相关开发要求及方法。	
330-3	座舱域应用软件信息安全技术要求	团标推荐	本标准包括座舱域应用软件信息安全要求与设计方法，旨在规范信息安全相关开发要求及方法。	
330-4	座舱域应用软件功能测试规范	团标推荐	包含测试过程、测试方法、测试工具和测试标准，并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求	
330-5	座舱域应用软件编码测试规范	团标推荐		
330-6	座舱仪表控制软件设计规范	团标推荐	包含座舱软件在仪表控制，信息娱乐，智能体验面向功能与算法设计规范	
330-7	座舱信息娱乐软件设计规范	团标推荐		
330-8	座舱智能体验软件设计规范	团标推荐		

2.2.4 整车应用软件

2.2.4.1 发展历程

随着汽车技术的不断进步，越来越多的车辆将配备各种智能化的应用软件，如智能驾驶辅助系统、车载娱乐系统、车联网系统等。同时，随着高性能计算平台的兴起，整车级应用软件也得到更多关注，该类软件的特点为实现跨域部署，支持应用软件的升级、安全、诊断和软件管理等功能，包括整车OTA软件、整车安全及加密应用软件、整车级应用软件诊断软件、跨域应用管理软件等。其中整车级应用软件的发展趋势还包括增加了更加个性化和定制化的功能，以满足消费者对汽车智能化和互联化的需求。其中整车安全及加密软件、整车级应用软件诊断软件和跨域应用管理软件还在发展初期，本章节重点介绍OTA应用的特点及发展现状。

OTA 全称为 Over-The-Air Technology(空中下载技术)。OTA 软件是指汽车各类应用软件的集合，OTA技术则是通过移动通信网络接口来实现对汽车控制器的远程管理，主要作用就是远程升级和修复车内的软件。诊断软件是指用于对汽车故障进行诊断和排查的软件工具。诊断软件可以通过OBD(On-Board Diagnostic)接口与汽车的电子控制单元(ECU)进行通信，读取车辆的故障码和传感器数据，以帮助快速定位和排除故障。

随着汽车电子控制系统的不断升级和智能化程度的提高，OTA技术和诊断软件也经历了多个阶段的演进，目前也已将OTA技术和诊断软件逐渐融合，汽车制造商可以通过OTA技术将诊断软件和相关数据下发到车辆上，实现远程诊断和故障排查，提高客户服务效率和用户体验。

1) OTA起步阶段：2000年代初，汽车OTA起步阶段，主要应用于音响、导航、车载电话等功能的升级，该阶段主要依赖于CD、DVD等媒介进行升级。

2) OTA发展阶段：2010年以后，随着汽车电子控制系统的不断升级和智能化程度的提高，OTA技术得到了快速发展，并开始应用于车载电子控制单元(ECU)的升级和更新。

3) OTA升级阶段：2015年以后，汽车OTA开始应用于整车级别的升级和更新，包括底盘控制、车身控制、驾驶辅助和安全系统等。同时，OTA技术也开始应用于汽车的远程诊断、故障排查和数据采集等方面。

4) OTA智能阶段：未来OTA技术将继续向智能化、数字化方向发展，汽车OTA将成为实现车

辆智能化、自动化、互联化的重要途径。同时，OTA技术也将应用于汽车的人工智能、自动驾驶等领域，为用户提供更加智能化、个性化的出行服务。

2.2.4.2 特点

整车级应用软件为基于SOA架构下的应用通用模块，它具有以下特点：

(1) 复杂性：整车级应用软件需要处理大量的数据和复杂的算法，以实现汽车的各种定制化功能，如OTA软件、整车应用级监控管理软件、整车应用诊断等。

(2) 可靠性：整车级应用软件需要具备高度的可靠性，同时基于应用软件的逻辑去实现相关的安全监控功能，以确保车辆在各种条件下的正常运行。

(3) 可扩展性：整车级应用软件需要具备良好的可扩展性，以便将来可以添加新的功能和更新软件。

OTA诊断软件主要应用在汽车的故障诊断、车辆控制、软件升级上，面向整车各个功能域。

OTA诊断软件具有以下特点：

(1) 高效的远程诊断能力：OTA诊断软件可以实现远程获取车辆的故障信息，并且能够通过网络将信息传输至汽车制造商或者服务提供商的服务器上，便于后续对车辆的故障进行排查。

(2) 安全的数据传输能力：OTA诊断软件在数据传输过程中采用了安全加密技术，以保证数据的安全性和防止数据泄露。

(3) 灵活的软件升级能力：OTA诊断软件可以实现远程升级车辆的软件系统，以使车辆具有更好的性能和更多的功能。

2.2.4.3 发展现状

整车级应用软件的发展趋势是向智能化、互联化和电动化方向发展。当前处于发展初期，随着高算力芯片的产生，电子电气架构的集中化，越来越多的应用软件集中部署到高算力平台上。整车级应用软件也会承担更多跨域的整车级功能。

随着智能网联技术的发展，OTA与诊断软件成为汽车软件必不可少的部分。同时诊断软件作为OTA软件的基础功能，为OTA软件提供ECU升级刷写功能与版本检测功能。

汽车OTA软件是汽车行业中使用远程升级技术来改善车辆性能和增加新功能的一种软件。可以通过OTA的方式提高用户体验、增加车辆功能、降低成本、在线修复软件缺陷。自2012年，特斯拉推出了搭载OTA技术的车型，并开始逐步通过OTA技术升级车辆系统，自此以后，OTA技术开始被广泛应用于汽车行业，当前主流的汽车都带有OTA功能。OTA软件相较于传统的线下软件升级，OTA能更方便的软件更新、更快速的漏洞修复、更便捷的功能扩展。随着OTA软件在各车型上的搭载，汽车远程升级软件变的越来越频繁，汽车OTA软件在发展过程中面临着多重挑战，如法律法规、隐私保护、系统稳定性等问题。联合国世界车辆法规协调论坛(UN/WP.29)，发布了R156法规，欧盟在2022年7月正式实施，并将于2024年7月对新车型强制实施；中国工业和信息化部2022年4月发布了《关于开展汽车软件在线升级备案的通知》，同时中国工业和信息化部2022年正在制订《汽车软件升级通用技术要求》，不久将会强制实施。

2.2.4.4 标准需求

传统汽车整车开发模式是一种V型开发模式。V型左侧涵盖需求分析，右侧对应模块测试，可在软硬件模型完整构建前完成集成测试方案设计，并有效保证测试方法与对应模块的兼容性，高效定位测试问题。但传统V型开发模式中“整车-系统-子系统-软硬件”的开发设计顺序局限于有明确需求导向的整车开发，难以适应软件定义汽车功能快速迭代的需求，需建立适软件定义汽车的相关标准体系。

目前行业内针对汽车软件的标准如下：

AUTOSAR:汽车开放系统架构标准、ISO 26262:功能安全标准、MISRA C: C语言编码标

准等。

(1) 软件架构设计标准：目前的标准（如AUTOSAR）主要介绍了软件设计的整体架构，缺乏软件动态静态架构设计的详细规范、设计原则、设计模式等，以保证硬件与软件的兼容性，软件设计的一致性、可靠性、可扩展性等方面。

(2) 软件开发设计标准：目前软件开发基本参考国际MISRA C语言编码规范，但产业标准和国际标准对接存在时序性的障碍，缺乏在汽车软件架构的基础上指导各模块开发的设计标准以及模块与模块之间开发的标准接口需求以及相关技术要求），模块设计需要考虑模块之间的通信和接口方式，以及模块的数据结构和算法，例如嵌入式控制器文件系统标准接口等。

(3) 软件集成方法：各个模块集成的标准和方法是实现汽车软件系统功能的必要条件。应从软件模块集成需求、集成接口、版本管理等方面制定相应的标准指导软件集成。

(4) 软件测试验证方法：目前行业内缺少汽车软件静态分析、单元测试、软件部件测试、软件集成测试、软件系统测试、自动化测试的方法和相关标准，另外软件接口测试也无相关标准约束，同时测试标准中应增加测试通过的参考值和标准。

(5) 软件验收标准和审核机制：国内标准中缺乏软件设计过程中的审核机制和开发完成后的验收标准和流程，例如软件全流程设计中，软件代码、模型的可靠性，软件的可扩充性、可移植性，软件的可修性以及软件的可测试性等等。包括数据库设计的逻辑性、安全保密性、是否有备份和恢复策略等审核机制。

(6) 软件迭代和监控策略：汽车软件无规范化的迭代计划和监控策略，可以根据收集各大 OEM 的迭代计划和周期，制定一套标准的软件更新计划，在迭代过程中制定详细的标准来对每个迭代周期进行监控和评估，以确保迭代结果符合预期。

2.2.4.5 标准体系框架及分类说明

整车应用软件的分类原则也是基于软件架构及开发流程、代码规范、能力等级和法规等方面进行分类（见图13）。

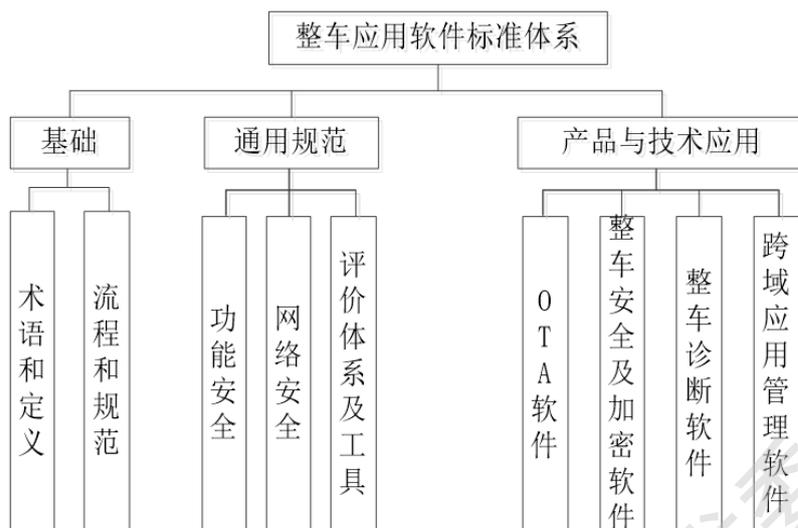


图13 整车应用软件标准体系

整车应用软件类标准体系主要包括基础、通用规范、产品与技术应用等三个部分：

1. 基础

基础标准主要用于统一整车汽车软件领域相关概念，厘清标准化对象及边界，建立标准化对象的统一表达方式，包括流程和规范。

(1) 流程和规范

目前大多数都是依照ISO26262等相关标准的要求去开发以及V字开发流程，该部分内容包括整车应用软件需求分析、开发设计、测试验证和维护等流程。

2. 通用规范

通用规范标准侧重于提出适用于整车汽车软件的通用要求和共性评价准则，主要包括功能安全、网络安全、评价体系及工具等部分。

(1) 功能安全

功能安全标准用于确保电子电气系统故障（包括软件、硬件、系统故障）等功能异常的情况下，车辆能够安全运行、不会引发安全风险，主要包括整车应用软件功能安全等级要求建议与设计方法，旨在规范功能安全开发要求及方法。

(2) 网络安全

汽车网络安全标准基于车联网复杂环境，以车端为核心运用纵深防御理念保护其免受网络攻击或缓解网络安全风险，主要包括整车应用软件网络与数字安全等级要求与设计方法，旨在规范网络与数字安全开发效率方法。

(3) 评价体系及工具

评价体系及工具标准规范了整车应用软件中评价体系的关键要素，创建测试过程、测试方法、测试工具和测试标准，并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求。

3. 产品与技术应用

产品与技术应用标准主要涵盖OTA软件、整车安全及加密软件、整车诊断软件、跨域应用管理软件等整车应用核心产品与技术应用。

(1) OTA软件

(2) 整车安全及加密软件

(3) 整车诊断软件

(4) 跨域应用管理软件

表12 整车应用软件标准体系

标准项目及分类		标准类型	标准描述	备注
应用软件 (300)				
整车应用软件 (340)				
340-1	整车应用软件开发流程规范	团标推荐	本标准包括整车应用软件需求分析、开发设计、测试验证和维护等流程、旨在提高产品开发流程的规范性、效率性和质量	
340-2	整车应用软件功能安全技术要求	团标推荐	本标准包括整车应用软件功能安全等级要求建议与设计方法，旨在规范功能安全相关开发要求及方法。	

340-3	整车应用软件信息安全技术要求	团标推荐	本标准包括整车应用软件信息安全要求与设计方法，旨在规范信息安全相关开发要求及方法。
340-4	整车应用软件总体架构及要求	团标推荐	在整车应用软件系统结构、行为和属性的高级抽象基础之上展开的全面的系统设计，其主要内容包括概要设计和详细设计
340-5	整车应用软件编码技术要求	团标推荐	规范整车应用软件编码规范，括文件组织、注释、声明、语句、命名约定、编程原则等规则
340-6	OTA 应用软件设计规范	团标推荐	规定 OTA 应用软件、整车安全及加密软件、整车诊断软件、跨域应用管理软件的接口及功能设计规范
340-7	整车安全及加密软件设计规范	团标推荐	
340-8	整车诊断软件设计规范	团标推荐	
340-9	跨域应用管理软件设计规范	团标推荐	

2.2.5 手机及网联应用软件

2.2.5.1 发展历程

手机和网联服务应用软件是手机智能化和汽车网联智能化的一个产物。这些应用软件极大丰富了汽车的功使用体验。

1) 无手机和网联服务应用时代；在汽车诞生后的前几十年，车机的功能比较单一，基本都是机械式的操控，也没有网络连接能力，只有一些卡带式音乐播放和机械导航等功能。

2) 移动互联互通的阶段；随着互联网的出现，更多的网络通信方法层出不穷，车机系统向网联化方向发展，汽车有了一定的网络连接和互联互通的能力，汽车增加了诸如蓝牙、usb 等连接能力；同时随着智能手机的发展，诸如投屏，蓝牙音乐和手机导航等功能越来越丰富，这些手机和网联服务应用软件通过标准化的蓝牙和 usb 协议，可以大幅度提升汽车的使用体验。但是受限于汽车车载系统的限制，功能和使用领域局限于少数几个方面。

3) 与车辆融合的阶段；随着数字化转型的深入，汽车行业“以用户为中心”的市场战略和运营策略也在加快落地。汽车消费者从兴趣、意向、试驾、购买、用车/服务体验，到置换\增购\转介绍的整个过程中，涉及线上和线下多个数字化营销触点，其中就包括重要性日益

提升的手机和网联服务应用软件。虽然尚未成为承载获客的第一落地点，但是凭借极高的私域运营价值和车主全生命周期的覆盖能力，手机和网联服务应用软件正向车企的一站式服务平台演进。提升手机和网联服务应用软件的用户体验，能有效的帮助整个线索转化链条的效率的提升。

服务、社区和商城是手机和网联服务应用软件最核心的三大组成部门：服务是用户使用手机和网联服务应用软件的核心诉求，社区则承担了增加用户粘性，提升用户活跃的重要功能，商城一方面是通过金粉商城提升用户粘性，另一方面可通过车辆本身和汽车周边商品售卖实现营销变现。

2.2.5.2 特点

目前手机及网联服务类应用软件主要用于提升和丰富用户的座舱驾驶体验，主要功能基本都依赖网络，故这类软件主要有以下特点：

(1) 高度依赖网络：手机及网联服务类应用的功能，都需要通过网络获取相关的功能和数据。由于网络的传输速度和稳定性存在不确定性，通常数据读取功能响应的速度是数百毫秒甚至秒级，相对与车控域软件，手机及网联服务类应用的实时性和可靠性要求较低，故此类应用通常不涉及对实时性、可靠性均要求较高的功能，诸如车辆驾驶等。

(2) 与车辆系统低耦合：手机及网联服务类应用作为车机系统的生态功能软件，主要是运行在用户手机中，其软件的功能通常不影响车辆驾驶操控相关的功能，也不会影响车辆的正常行驶功能，车辆基本功能的使用不依赖手机和网联服务应用。

(3) 围绕提升用车体验：手机及网联服务应用提供了完善的智能车联服务，支持通过手机查看车辆的各种信息，并且能远程操控车辆的各种能力。围绕用户可能遇到的各种问题，提供诸如维修保养，道路救援、停车违章等用车服务，提供在线客服，问答，手册等售后服务。探索潜在用户，提供预约试驾，个性化定制、车型展示等购车服务，提供金融计算，征信查询

等金融服务。方便用户和潜在用户了解和使用车辆。

(4) 活跃的社区体验：手机及网联服务应用的社区功能可以增加车主日常的活跃度和粘性，通过丰富的社区内容，提升用户发布用车体验、用车生活并积极分享自己的用车经历和方法，参与社区的各种活动等，提升用户的活跃度，吸引和增加潜在用户。

(5) 丰富的商城内容：商城是售后变现的一个重要渠道，在通过各种方式增加用户粘性，可以积极推广各种车辆周边商品，刺激用户进行售后消费。通过积分方案的设定，如通过积分对话和积分抵扣等功能，增加用户粘性，提升用户营收。

2.2.5.3 发展现状

但随着移动互联网的发展，汽车向智能化的转向，汽车网联模式日渐丰富，手机和网联服务应用软件与车机系统的交互日渐加深。由于手机和网联服务应用软件的固有特点，其开发过程与车机系统耦合度低，通常前期分别进行开发，后期进行集成测试。由于手机和网联服务应用兴起的时间比较晚，大部分手机和网联服务应用软件的开发模型都采用Agile模型（敏捷开发模型）。

由于目前绝大多数的手机软件开发平台都是基于Andorid系统或者IOS系统进行开发，所以大部分功能定义都是遵循谷歌和苹果的软件开发标准。具体的编码规则根据其使用的语言遵循不同的标准。

目前手机和网联服务应用软件测试主要包括，单元测试、可靠性测试、集成测试、功能性测试等。上述软件在测试过程中涉及的内容可能会略有不同，同时测试的指标也有所不同；目前没有统一的测试标准规范。

表 13 手机和网联服务类应用软件类标准统计

发布日期	标准名称	部门/机构	相关内容
2018	ISO 26262 Road vehicles -	国际标准化	1、安全要求：ISO 26262 要求在整个软件

	Functional safety	组织	<p>开发生命周期中，包括要求分析、设计、实现、验证和确认等各个阶段都要考虑安全性。开发人员需要遵循一定的安全要求来确保软件的安全性和可靠性。</p> <p>2、安全文档：ISO 26262 要求开发人员编写安全文档来记录整个软件开发过程，包括需求分析、设计、实现、验证和确认等各个阶段。这有助于开发人员更好地管理和追踪软件开发过程，以确保软件的安全性和可靠性。</p> <p>3、风险分析：ISO 26262 要求开发人员进行分析，以确定可能导致汽车电子系统故障的原因和影响。在软件工具开发中，开发人员需要考虑软件工具对汽车电子系统的影响，并采取相应的安全措施来减少风险。</p> <p>4、软件测试：ISO 26262 要求对软件进行测试以验证其安全性和可靠性。在软件工具开发中，开发人员需要设计和实现有效的测试策略来确保软件的安全性和可靠性。</p>
2017	Automotive SPICE Process Assessment / Reference Model	德国汽车工业联合会 (VDA) 质量管理中心 (QMC)	<p>ASPICE 将汽车软件开发过程划分为以下过程：</p> <ol style="list-style-type: none"> 1、软件需求分析 2、软件架构设计 3、软件详细设计和单元构建 4、软件单元验证 5、软件集成和集成测试 6、软件合格性测试
2012	MISRA C:2012 Guidelines for the use of the C language in critical systems	汽车工业软件可靠性协会 (英国)	<ol style="list-style-type: none"> 1、代码静态分析：MISRA C 规范提供了针对 C 语言编程的静态代码分析规则，以确保编写的代码符合 MISRA C 规范。 2、自动化测试工具：MISRA C 规范要求软件开发过程中进行自动化测试，以验证代码是否符合 MISRA C 规范。 3、静态检查工具：MISRA C 规范要求使用静态检查工具进行代码检查。这些工具可以检测代码中的违规操作，并为程序员提供修复建议。 4、代码生成工具：MISRA C 规范要求使用代码生成工具生成符合规范的代码。 5、软件验证工具：MISRA C 规范要求使用软件验证工具以确保代码符合规范。
2008	MISRA C++	汽车工业软件可靠性协会 (英国)	<ol style="list-style-type: none"> 1、类型和运算符的正确使用：MISRA C++ 标准强制执行对变量和运算符的正确使用。 2、内存管理：汽车软件工具通常需要使用动态内存分配和释放，这可能导致内存泄漏和其他内存相关的问题。MISRA C++ 标准规定了对内存管理的最佳实践，帮助开发人员避免这些问题。

			<p>3、异常处理：异常处理是汽车软件工具开发中的重要问题之一。MISRA C++标准提供了关于异常处理的最佳实践，帮助开发人员正确地使用异常处理机制，以确保代码的可靠性和安全性。</p> <p>4、多线程和并发编程：现代汽车软件工具通常需要使用多线程和并发编程。MISRA C++标准提供了对这些主题的最佳实践和指导，帮助开发人员编写高质量、可靠的多线程代码。</p>
2020	GB/T 38628-2020 《信息安全技术 汽车电子系统网络安全指南》	国家标准化管理委员会	1、软件架构：给出了汽车电子系统网络安全活动框架，以及在此框架下的汽车电子系统网络安全活动、组织管理和支撑保障等方面的建议。指导相关人员在从事汽车电子系统的设计开发、生产、运行和服务等过程中满足基本的网络安全需求
2020	MAAB	MathWork	1、建模规范：主要讲的是 Simulink 和 Stateflow 建模中的一些基本规则；用于提高模型的可读性、实现仿真和验证、提高代码生成效率，确保生成代码的健壮性
2009	Google C++ Style Guide	Google 开源	1、编写规范：提高代码的可读性、规范性、安全性
2013	Google Java Style Guide	Google 开源	1、编写规范：提高代码的可读性、规范性、安全性
2001	PEP8	Python 社区开源	1、编写规范：提高代码的可读性、规范性、安全性
2019	GB/T 37729-2019 《信息技术 智能移动终端应用软件（APP）技术要求》	国家标准化管理委员会	<p>1、功能性要求：APP 应该满足预期的功能要求。</p> <p>2、性能效率要求：APP 应该具有良好的性能效率，如启动时间短、资源利用率高。</p> <p>3、兼容性要求：APP 应该能够在不同品牌、型号和操作系统的移动终端上运行。</p> <p>4、易用性要求：APP 应该具有友好的用户界面和使用体验。</p> <p>5、可靠性要求：APP 应该具有高可靠性和稳定性，避免崩溃、死循环等问题。</p> <p>6、安全性要求：APP 应该具有高水平的安全性，防止数据泄露、被篡改等安全问题。</p> <p>7、维护性要求：APP 应该具有易维护性和可修改性，方便后续升级和补丁更新。</p> <p>8、抗风险要求：APP 应该具有抵御各种风险的能力，如自然灾害、政治法律风险等。</p>
2023	GB/T 42884—2023 《信息安全技术 移动互联网应用程序（App）生命周期安全管理指南》	国家标准化管理委员会	标准提供了移动互联网应用程序(App)生命周期阶段管理过程和风险监测管理过程的安全管理指南。标准适用于 App 提供者对 App 的开发、运营等生命周期安全管理，App 分发平台管理者和移动智能终端厂商等可参考使用。
2023	苹果 IOS 开发规范	苹果开源	1、编写规范：提高代码的可读性、规范

			性、安全性 2、规范功能实现：符合苹果上带你的上架规则
--	--	--	--------------------------------

2.2.5.4 标准需求

(1) 软件架构：目前主流的手机平台主要包括 Android 和 IOS，手机 APP 的开发主要是遵从软件平台规定。当前手机 APP 并没有统一的软件架构。目前主流座舱系统均为定制版 Android 系统或者 Linux 系统，网联服务应用软件主要是围绕这些平台系统进行的开发，除了适配平台系统的特定功能外，并没有统一的软件架构。

(2) 软件设计：目前手机及网联服务类应用软件的实现方式众多，原生代码、跨平台的、套壳 H5 的等等，种类各不相同，使用的开发方式和开发语言也不尽一致。

(3) 功能及性能特性、技术要求：目前手机及网联服务类应用软件的功能、性能等都各不相同，不同汽车厂家根据自己的车辆的功能差异，所设计的软件也不尽相同。所以这方面没有统一的标准。

(4) 开发流程：国内外相关标准较多，内容也较为全面，越来越多的汽车企业采用敏捷开发模式，然而这种开发模式尚未形成标准。

2.2.5.5 标准体系框架及分类说明

手机和网联服务类应用软件标准体系横向以车控、公共能力、云店营销、三个层次为基础，纵向以功能，基础能力封装和基础库的技术框架。完整呈现标准体系的技术逻辑，明确各项标准在车控域应用软件产业技术体系中的地位和作用。

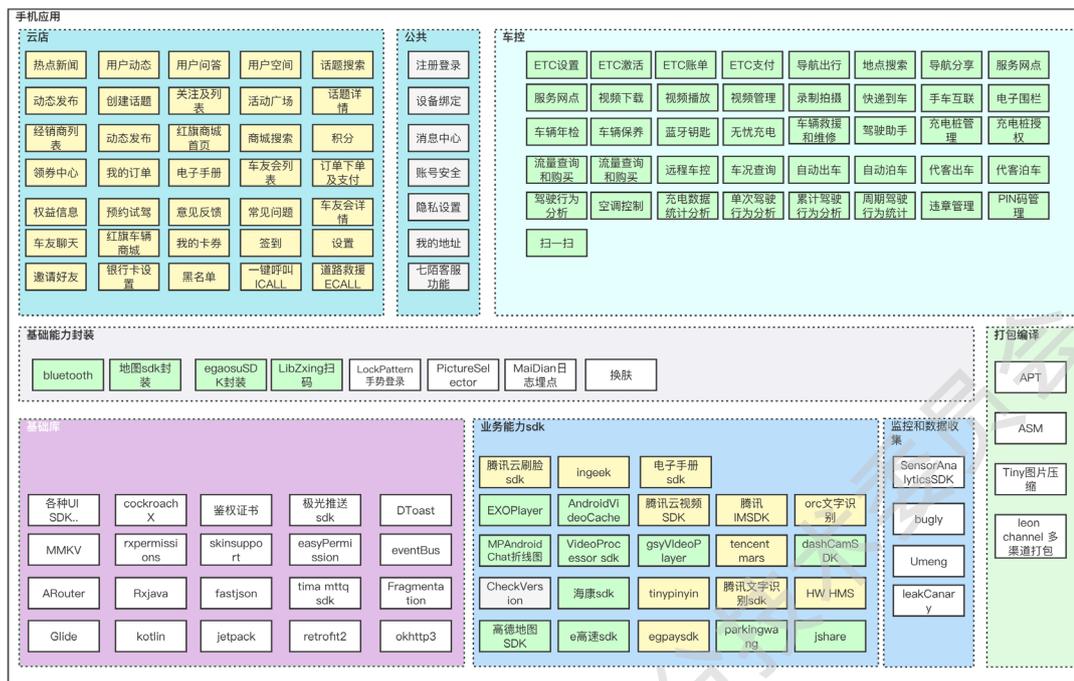


图 14 手机和网联服务类应用软件系统架构

按照手机和网联服务应用软件逻辑架构(见图14)，综合考虑不同功能、产品和技术类型、各子系统之间的交互关系，将车控域应用软件标准体系划分为四个部分。其中，第一部分规定了手机和网联服务应用的基础软件库和业务能力开发工具包的范围和选用的标准；第二部分基础能力封装是定义对第一部分的进行统一标准的封装；第三部分业务层对第一部分和第二部分的读取和使用的标准，第四部分是手机和网联服务应用软件业务数据上报和处理的标准。

(1) 第一部分：基础库和业务能力 sdk

提供应用软件的基本的网络、图片加载、数据处理、页面处理等能力。业务能力 sdk 主要提供应用软件的业务所需的各种第三方组件和开源组件。

(2) 第二部分：基础能力封装

对基础库和业务 sdk 进行统一的封装，以统一的支持业务层的能力调用

(3) 第三部分：业务层

主要是各种业务的实现。根据产品的业务需要、根据功能架构的划分分成不同的模块进

行开发。

(4) 第四部分：业务日志收集和数据监控

主要用于在业务监控，可以方便查看应用具体的运行现状和业务数据收集。方便以后进行持续优化。

表14 手机及网联服务类应用软件

标准项目及分类		标准类型	标准描述	备注
应用软件（300）				
手机及网联服务类应用软件（350）				
350-1	手机及网联服务类应用软件基础库和业务能力 sdk 的集成标准	团标	手机及网联服务类应用软件基础库和业务能力 sdk 的集成标准	
350-2	手机及网联服务类应用软件基础库和业务能力 sdk 的软件安全要求	团标	手机及网联服务类应用软件基础库和业务能力 sdk 的软件安全要求	
350-3	手机及网联服务类应用基础能力列表	团标	手机及网联服务类应用基础能力列表	
350-4	手机及网联服务类应用软件开发流程技术要求	团标	手机及网联服务类应用软件开发流程技术要求	
350-5	手机及网联服务类应用软件总体架构及要求	团标	手机及网联服务类应用软件总体架构及要求	
350-6	手机及网联服务类应用软件编码技术要求	团标	手机及网联服务类应用软件编码技术要求	
350-7	手机及网联服务类应用软件安全要求	团标	手机及网联服务类应用软件安全要求	
350-8	手机及网联服务类应用软件性能要求及测试方法	团标	手机及网联服务类应用软件性能要求及测试方法	
350-9	数据服务技术要求	团标	手机及网联服务类应用数据服务技术要求	
350-10	开发流程技术要求	团标	手机及网联服务类应用软件开发流程技术要求	
350-11	日志收集的要求	团标	手机及网联服务类应用软件日志收集的要求	
350-12	日志的信息安全要求	团标	手机及网联服务类应用软件日志的信息安全要求	
350-13	日志存储和使用的要求	团标	手机及网联服务类应用软件日志存储和使用的要求	
350-14	数据监控的要求	团标	手机及网联服务类应用软件数据监控的要求	

2.2.6 云端应用软件

2.2.6.1 发展历程

汽车行业云端软件系统是一种基于云计算和物联网技术的服务平台，用于实现汽车远程诊断、远程监控和智能管控。它可以对汽车的各项数据进行收集、分析和处理，从而实现车辆的状态监测、故障预警、车辆定位、驾驶行为分析、车况评估等功能；它还可以与车主手机 APP、维修保养平台等应用程序进行集成，提供更便捷和全面的管理服务；同时，云端智慧运维体系的构建也是云端软件系统的重要组成部分：它通过构建基于云、微服务架构，开放的，可以自主掌控的智能网联平台来支撑包括车辆的运营管理、手机的远程控车、车况的监控诊断、用户的行为分析等智能网联云平台的业务实现。

汽车行业云端软件的发展分三个阶段：

1) 初步探索阶段（1990 年-2005 年），在这一阶段，汽车行业云端软件还处于摸索和试验的阶段。本阶段主要是一些汽车制造商在尝试将计算机技术应用到车辆管理中来。大多数软件都只具备单向传输的数据分析能力，例如可以进行车辆故障检测和数据记录等功能，但无法进行实时处理和反馈。

2) 应用拓展阶段（2005 年-2012 年），在这一阶段，云计算和物联网技术的逐渐成熟，为汽车行业云端软件的扩展提供了可行性基础。从本质上来说，这个发展阶段意味着更多的厂商开始采取综合计算、产品信息化与智能化相结合的方式，使用云服务平台、物联网技术进入云端软件发展领域。同时，在此前开发的云端软件基础上，出现了基于“车联网”框架下的智能云端软件，专注于远程监控、智能管控和移动互联等方面的全方位服务，推动了软件技术越来越深入人们的生活。

3) 产业集成阶段（2012 年至今），在这个阶段，汽车行业云端软件真正地走上了一个新的阶段，即“产业集成”阶段。这个阶段是呈现行业巨头联盟、互动共生并行之势，有一系列全球性企业加快了整个行业云计算的步伐，例如，丰田公司、通用汽车、福特汽车、奥

迪等诸多大型汽车制造商均在此时推出了基于“车联网”的智能云端软件。同时，无论是在云端服务还是逐渐规模化的应用场景中，更多都是走向人工智能等前沿技术的升级。

我国云端软件发展随着移动网络的发展分为三个阶段：

第一阶段（2009-2013年）：2G/3G网络时代，功能型车载信息服务

2009年，上汽通用汽车将OnStar命名为安吉星并正式引入中国，在国内率先开启了车联网应用的前瞻探索。这一阶段的车联网主要是以汽车厂商为主导，通过在汽车上安装T-Box（Telematics Box）或者智能手机等终端设备，利用网络将汽车与云端服务器连接起来，实现对汽车状态、位置、行驶数据等信息的采集和传输，并提供基于云端计算和大数据分析的各种信息服务。这一阶段是国内车联网的启蒙时期。当时国内汽车产品大多处在车载离线导航的时代，智能手机尚未普及，大多数人还处在一个月30M流量的2G时代，鲜有人将汽车和网络联系在一起。这一时期的车联网“网络化”很低，只能做到GPS定位导航以及知道车主在何处遭遇碰撞、碰撞严重程度如何、安全气囊是否打开，并基于此提供主动介入的道路救援服务。

第二阶段（2014-2019年）：4G网络时代，智能座舱网联服务

随着移动互联网、物联网、大数据、人工智能等技术的发展，车联网进入了一个新的阶段，即智能网联服务阶段。这一阶段的车联网不仅仅是实现车与云的连接，而且通过更先进的通信技术和传感技术，实现车与车、车与路、车与人等多方面的智能信息交互和协同控制，从而提高汽车的自动驾驶能力和智能出行服务能力。2014年，移动网络正式迎来4G时代。4G网络时代视频数据实时性好，互联网应用拓展多，无线应用空间大，车联网正式迎来高速发展期。这一阶段的车联网主要围绕车载信息服务、移动娱乐与消费、智能驾驶三方面，比如接入云服务，推进以“云”为中心的车辆应用与服务；开发基于空中更新

（Over the Air, OTA）技术的系统自动更新服务；推出车内支付系统、个性化车联网保险服务等。这个阶段也是云运维的起步和初期发展阶段。这一阶段的运维工作建设主要集中在基础设施层面，包括对服务器、网络、存储的监控管理等。同时，云端运维体系也能够进

行初步的自动化的流程实现，如自动化部署、自动化测试、自动化监控等。运维工作主要集中在应用层面，包括应用开发、应用部署、应用运维等。

第三阶段（2020-2025年）：5G-6G 卫星网络时代，智慧出行服务

随着5G、云计算、区块链等技术的成熟和应用，车联网已经进入一个更高级的阶段，即智慧出行服务阶段。这一阶段的车联网不仅仅是实现车与X的智能信息交互和协同控制，而是通过更广泛的网络连接和数据共享，实现车与社会各方面的深度融合和价值创造，从而提供更多元化、个性化、智慧化的出行服务和生活服务。这一阶段的车联网发展已上升到国家战略高度。在5G网络时代下，5G的高传输、低时延、高稳定等特性将满足未来对车联网的更高要求。5G车联网可以真正实现车内、车与人、车与车、车与路、车与服务平台的全方位网络连接，与此同时，随着卫星互联网被列入新基建，卫星网络将在6G时代大放异彩，卫星车联网服务即将落地。卫星车联网是用地面通信和卫星通信双通道模式，实现车载终端的全时全域覆盖，服务于特种车联网需求和船联网需求。发展至今，车联网已经从最初的车载信息系统，发展为了依托新一代网络通信技术与汽车、电子、道路交通运输等领域深度融合的新兴产业形态。在这个过程中，所产生的问题也不断增加，云运维系统也随之进入了一个更智慧化的时期。这个阶段运维领域的发展趋势主要是以智能化运维为主，比如 AI0ps、智能诊断、智能预测等。运维工作主要集中在数据层面，包括数据采集、数据分析、数据挖掘等。

2.2.6.2 特点

云端软件系统覆盖云、网、车、路四端，按照汽车全生命周期和用户使用场景，可划分为研发、控制、运营、安全和服务五个类别。研发方面，智能网联云服务赋能数据价值挖掘，在研发环节助力智能驾驶辅助和自动驾驶的开发与应用。控制方面，智能网联云服务拓展软硬件控制能力，增强车企和服务商在汽车智能驾驶和智能座舱场景下的服务能力，提升用户体验。运营方面，智能网联云服务通过强大的数据打通、数据分析和数据预测能力增强

车企和服务商的运营能力。安全方面，智能网联云平台作为中枢神经增强汽车信息安全和驾驶安全。服务方面，智能网联云平台扩张车联网的服务能力，尤其在车联网支付服务、智能汽车商业化运营服务方面发挥重要作用。

近年来，汽车行业云端软件在智能化驱动下发展迅速，并得到广泛应用。

(1) 汽车行业云端软件功能特点

- 1) 高实时，实时监测管理：通过车辆的实时监控，可以及时发现异常情况并及时处理，提高管理效率。例如，在车辆运输过程中，可以实时监控车辆的位置、状态以及物流信息等，从而为调度、路线规划等提供参考。
- 2) 高兼容，信息共享协作：汽车行业云端软件实现了信息共享和合作，各个环节都可以及时地传递信息，促进合作，提高效率。例如，在物流管理方面，货主、承运商、仓库等相关单位可以实现信息共享，从而优化物流流程。
- 3) 高性能，增强系统性能：汽车行业云端软件可以提高整个系统的性能和工作效率，从而更好地服务于客户需求。例如，在公交管理中，通过人脸识别、语音助手等技术应用，可以提高管理效率和用户满意度。
- 4) 高安全，提高安全保障：车辆与云端建立的联系，使得汽车鉴权、车辆追踪、安全控制等变得更加精准和高效，实现了汽车运营的智能化和安全保障。

(2) 汽车行业云端软件技术特点

- 1) 大数据分析技术：通过对运输、维护和计费等方面的数据进行收集、存储和分析，可以为企业提供更好的决策支持，并帮助企业更好地降低成本和提升服务质量。
- 2) 智能识别技术：针对汽车行业特有的信息展示和流传特性，采用多种智能识别如模式识别、自然语言处理等技术，从而显著提高管理效率和服务质量。
- 3) 移动化领域技术：支持手机聚合支付、移动端预约挂号以及车辆机器人驾驶等领域开发，拓宽汽车行业各种应用场景，进一步提高汽车行业的智能化水平。

- 4) 云服务能力：利用云计算技术的速度、灵活性和弹性等优势，打造出解决方案，如数据仓库、数据挖掘等应用。

(3) 汽车行业云端软件运维特点

- 1) 灵活性和可扩展性：汽车行业的云平台包括基础设施即服务（IaaS）、平台即服务（PaaS）和软件即服务（SaaS），这些平台可以进一步划分为公有云、私有云和混合云。这样的分类和组合为汽车行业的发展提供了极大的灵活性和可扩展性。
- 2) 高度的性能和稳定性：智能车联云平台需要支持百万车辆接入，同时处理海量数据管理和调度，这对云平台的性能、稳定性和安全性提出了极高要求。因此，高度的性能和稳定性是汽车行业云端运维的重要特点。
- 3) 可用性和安全性：汽车行业的远程运维服务能力标准专注于保障智能网联汽车在不同场景下的可用性和安全性。这涉及到在运行过程中保障车辆持续执行业务工作的能力，以及在运行时段以外保障车辆达到可执行业务工作所需状态的能力。
- 4) 高网络带宽需求：随着车载以太网应用场景的增多，对带宽的需求也越来越高，尤其是对于自动驾驶车辆而言，10G+端口的需求将会逐渐增加。这也意味着汽车行业云端运维面临着更高的网络带宽需求。

(4) 汽车行业云端软件应用场景

- 1) 物流管理：在物流配送过程中，可以通过云控技术实现货物追踪和定位、物流监控等功能，有效提高物流效率。
- 2) 车联网：汽车行业云端软件支持车载设备与云端数据交换和处理，为下一代汽车带来更好的驾驶体验和服务质量。
- 3) 公交管理：公交系统可以采用云控技术进行车辆调度、计费信息收集等功能，使公交运营管理变得简单、快捷、高效。
- 4) 汽车制造：在汽车制造领域，可以利用云控技术分析生产链上每一道工序的数据，以此提升汽车生产质量和效率。

汽车行业云端软件正在快速发展，并且随着新兴技术的不断涌现，未来的发展前景也将变得更加广阔。

2.2.6.3 发展现状

当今的汽车生产和销售业态已经发生了很大变化，互联网应用和智能技术的快速发展改变了整个行业的面貌。同时，越来越多的厂商开始关注并深入研究汽车云端软件，将其应用到整个生产、销售和售后服务流程中。

汽车云端软件是基于云计算和物联网技术而开发的，它能够通过近距离无线通信或者是远程网络连接实现对汽车进行监管、控制和管理。目前，全球范围内许多品牌的汽车行业都在积极投入和开发相关的云端软件，以便更好地服务消费者，提升企业的竞争力。

汽车行业云端软件的当前发展现状如下。

(1) 市场需求推动了汽车行业云端软件的快速发展

随着中国经济不断发展，汽车行业也得到了迅速的提升。据有关数据显示，目前中国的出行市场规模已经达到 75000 亿，而随着消费者购买力和对汽车智能化需求不断提高，汽车行业面临的管理压力也日益加大，如何实现优质服务和高效管理成为了当前需要解决的问题。

在此背景下，互联网、大数据、人工智能等新型技术应用于汽车行业，创造了汽车行业云端软件，通过对整个供应链进行管控，解决了传统车企长期存在的库存积压、零部件过多等诸多痛点问题。在这种背景下，汽车行业云端软件应运而生，得到了广泛的市场认可和推广。预计到 2025 年，中国的汽车零部件智能供应链市场规模将达到 1.6 万亿元，具有广阔的市场空间。

(2) 汽车行业云端软件应用场景不断扩大

目前，汽车行业云端软件的应用场景远远不限于单一领域，而是涵盖了整个汽车产业的各个环节，包括管理、服务、销售等。尤其是以下几个方面：

汽车租赁管理：在现代化城市中，汽车共享作为新型出行方式已经成为越来越多人的选择，而汽车行业云端软件可以实时监测车辆位置，智能分配订单，并进行费用计算，有效提高车辆利用率和利润水平。

车辆物流管理：目前，在我国的物流市场上，“最后一公里”始终是一个难题，智能化的车辆管理系统可以通过前置仓库、调度中心，以及分配物流路径等类似的方法缓解这种情况。同时在调度过程中，根据实时交通情况，智能进行路线优化，从而实现快速车辆派遣和高效配送。

汽车售后服务：除了车辆出行过程中的各项运营管理之外，汽车行业云端软件应用于售后服务领域，能够帮助车主通过在线平台预订维修、保养等服务，优化车主体验。同时，在维修过程中软件还可以监测车辆故障、诊断问题，并结合技术资源和数据信息，提供专业的解决方案，提高客户满意度。

智慧停车系统：当前城市停车难的情况已经超出人们的承受极限，因此开发智慧停车系统是行业内正在大力推进的一项工作。通过汽车行业云端软件的管理模式，可以在网络上对停车位进行实时监控并管理，使车位利用率得到最大化。同时，这种方式对于道路拥堵情况的缓解也具有重要意义。

(3) 安全漏洞问题日益严重

随着云端软件应用的不断扩大和深入，在安全方面的风险也日益显现。例如黑客攻击、数据泄漏等问题，都需要云端软件提供更加完善的安全保障。汽车厂商和相关技术企业需要加强技术创新和研发，推出更为安全可靠的云端软件系统。

(4) 智慧运维构建的诉求更加迫切

现如今各种智慧运维平台如雨后春笋般涌现，但由于行业内缺乏统一的标准，导致行业内大部分平台在技术、能力上存在较大差异，进而各大车厂运维能力参差不齐，其安全性和可靠性难以得到保障。因此，构建智慧车辆平台等一系列研究与探索工作变得极其迫切。这些平台需要支持百万车辆接入，同时处理海量数据管理和调度，这对云平台的性能、稳定性

和安全性提出了极高要求。高度的性能和稳定性是汽车行业云端运维的重要特点。此外，随着车载以太网应用场景的增多，对带宽的需求也越来越高，尤其是对于自动驾驶车辆而言，10G+端口的需求将会逐渐增加。这也意味着汽车行业云端运维面临着更高的网络带宽需求。总之，构建智慧车辆平台的一系列研究与探索变得极其迫切，以满足智能网联汽车行业快速发展带来的挑战。

总体来说，在汽车行业向着智能化迈进的背景下，云端软件作为重要的技术手段和应用平台，将持续发挥着越来越重要的作用。随着技术的不断进步和应用场景的拓展，未来云端软件有望为汽车行业带来更多便利和先进的管理模式。同时，在云端软件的发展过程中，安全问题也应引起广泛关注，各方应共同努力，构建更加稳健、安全的云端软件生态环境。

2.2.6.4 标准需求

云端软件在标准建设方面需要在设计、服务、技术、管理等方面进行完善。设计方面需要完善架构设计、系统设计、模块设计及系统接口设计方面尤其是端云接口、云云接口、系统内接口的要求；服务方面需要完善场景和质量等方面的要求；技术方面需要完善安全、性能、埋点及中间件使用方面的要求；管理方面需要完善开发流程、编程规范、测试规范、建设规范等方面的要求；汽车智慧运维平台体系的构建标准需求包括跨领域协同开放的智能网联汽车技术标准体系、应用服务安全标准、车辆运行状态监控系统和智能生产管理系统等方面。

2.2.6.5 标准体系框架及分类说明

车联网云端软件是一个规模庞大且复杂的体系，如图15所示，以满足不同用户及对象的产品需求。站在车厂云的角度，一方面要对接设备终端进行数据分析、处理和交互，另一方面要满足用户终端的各种指令和管理操作，复杂的业务有时候除了车厂云自身的业务提供支撑

外还要联合三方云共同提供服务，甚至车厂云本身内部也不是一个单一系统，存在系统间的数据对接和协同。

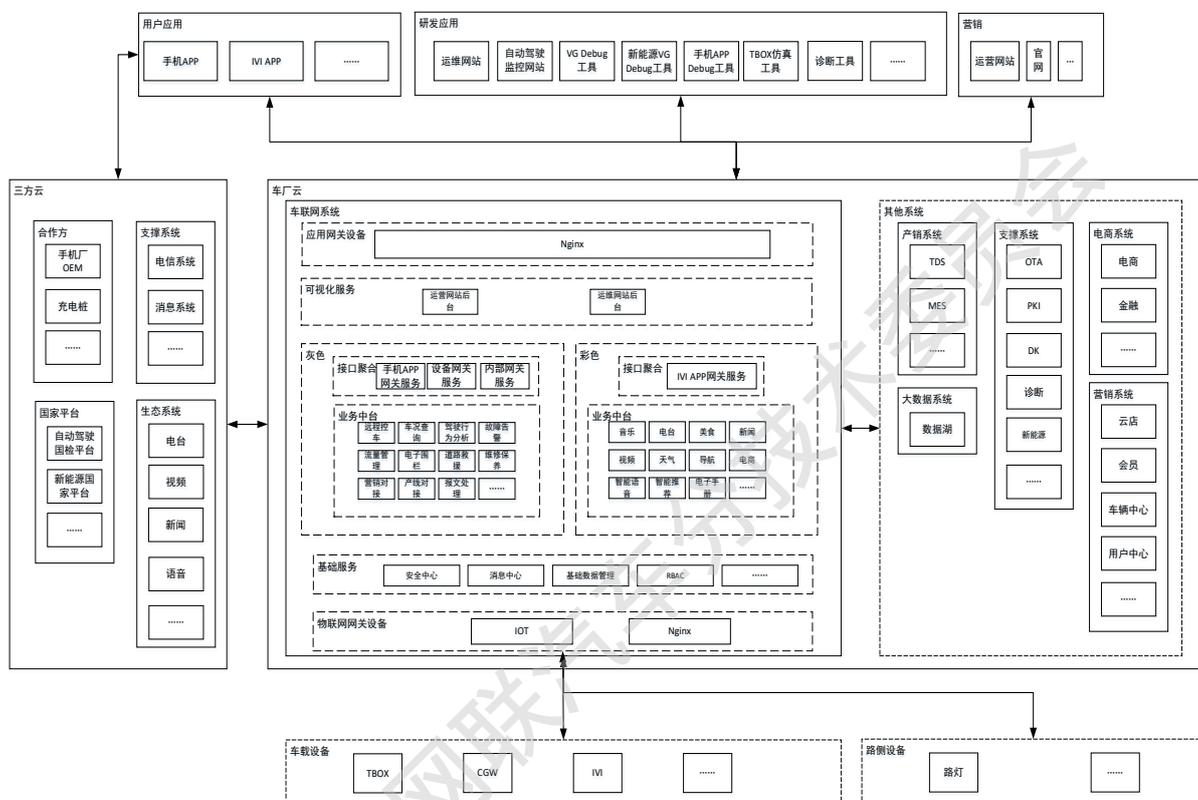


图15 车联网云端软件系统

目前车联网云端软件系统一般具备产销系统、控车系统、生态系统、语音系统、推荐系统、电商系统、营销系统、支撑系统、监测系统、检测系统、大数据系统等多个系统和几百个模块。

表15 云端软件标准分类

标准项目及分类	标准类型	标准描述	备注
应用软件（300）			
云端应用软件（360）			
360-1	云端应用软件开发流程要求	团标	规定了车联网云端应用软件开发流程的要求，包括需求分析、系统设计、开发环境搭建、编码规范、测试流程、安全性要求、部

			署和升级等方面的内容，为开发人员提供完整的开发指导，并确保云平台的功能完整性、可靠性。
360-2	云端应用软件总体架构及要求	团标	明确车联网云端应用软件的总体架构、技术要求、管理要求，保证云平台高效开发及稳定运行。
360-3	云端应用软件编码技术要求	团标	规定了车联网云端应用软件的实际编码过程和技术细节。明确了编码的规范、标准和实践，以确保开发团队在编码过程中遵循统一的标准和规范，减少错误和漏洞，提高软件的质量和安全性。
360-4	云端应用软件安全要求	团标	明确了车联网云端应用软件中应具备的安全机制，包括用户身份认证、访问控制、数据加密、系统防护等方面的机制。为开发团队提供一个详细的安全要求指导，以确保车联网云端应用软件的安全性。
360-5	云端应用软件性能要求及测试方法	团标	明确车联网云端应用软件的性能要求，包括响应速度、处理能力、网络传输速度等方面的要求，并提供相应的测试方法和工具。
360-6	云端应用软件运维流程要求	团标	规范了云端应用软件运维流程的具体要求和流程。概述运维流程的各个环节，包括监控预警、问题流转、问题解决、质量管理等全闭环流程，以提升车联网云平台运维效率。

2.3 数据服务软件

2.3.1 汽车数据服务现状

2.3.1.1 概述

随着汽车“智能化”“网联化”“电动化”“共享化”地发展，汽车的核心能力也逐渐从机械拓展到软件和服务。随着各类智能化应用层出不穷，数据逐渐成为车企的核心资产，数据价值随着技术发展而显著增强，构建有效的数据服务发挥最大化的数据价值，逐渐成为企业的核心竞争力。如何高效采集获取数据，如何存储和处理海量数据，如何挖掘数据价值提升用户体验，就成为汽车数据服务的关键挑战。本节将从数据服务的采集、开发、功能三个层面介绍分析汽车数据服务的发展现状。

汽车数据服务标准体系框架如图 16 所示：

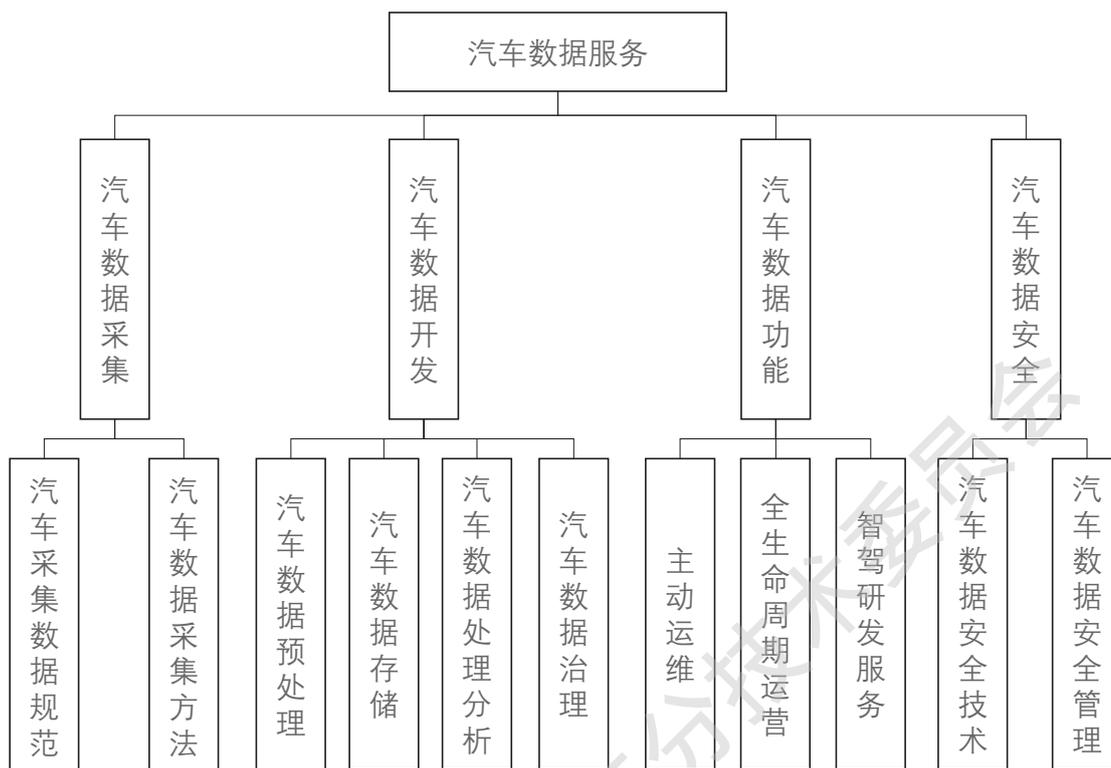


图 16 汽车数据服务标准体系框架

2.3.1.2 采集现状

近年来，车辆智能化、网联化功能越来越丰富，水平也越来越高，这就需要采集海量的数据以支持功能的实现，更好的服务用户。行业的这一变革，导致汽车对数据的需求逐年提升。根据 Garner 估计，每一部自动驾驶联网车辆每天至少产生 4TB 数据，每年约产生数百 PB 的数据，OEM 和出行服务商未来需处理的数量估计可达 ZB 级。海量数据的背后，蕴藏着巨大的应用潜力

现阶段，数据采集被应用到更多场景，数据采集需求呈现爆发式增长。在这一背景下，目前数据采集工作可为车辆的智能网联提供强大的支撑，但也存在诸多挑战。问题归纳如下：

1、数据采集标准不统一。面对海量的信号，开发平台的不同，各企业拥有不同采集体系。这就导致面对同一类信号，数据采集的周期、频率、触发条件等均有不同。比如前车碰撞预警功能，采集发生事故时该项功能是否能降低车损，对于车险理赔具有经济价值，但是各车企对于该项功能的采集并未形成统一格式，对车险理赔的判断标准会有产生不同结果。

2、采集数据项不统一。在研的国家标准标准《智能网联汽车数据通用要求》提出了智能网联汽车数据分级分类要求，将车联网数据分为车辆基本数据、感知数据、决策数据、运行数据、其他数据；同时要求数据根据危害程度和重要程度实施分级管理。目前，数据采集项各企业有不同的分类分级，行业按照统一的标准进行数据分类分级，有利于采集策略的统一，提升数据采集效率。

3、数据应用无法满足需求。数据采集未能充分考虑、理解需求方对数据的应用目标，导致采集的数据难以满足需求，无法输出有效的经济价值，形成人力物力的浪费。数据采集应以价值驱动，以用促建，将数据采集策略与应用场景、需求紧密结合，发挥数据更大的价值。

4、数据需求量增大，成本增大。科学技术的快速迭代，汽车将承载越来越多的智能网联功能，信号的采集频率由秒进入毫秒，采集维度的指数增长，这势必会导致一辆汽车将产生大量的数据，将这些数据采集上云，无论是人力资源还是存储成本都将是一个巨大的挑战。目前市场主流的车联网平台架构仍然是应用层：为联网用户提供各种车辆服务业务；网络层：提供信息传输服务；传输层：负责数据的采集。在日新月异的需求下，传统车联网平台架构已无法满足需求，亟需提升车企车云全链技术栈能力，整合运用高阶汽车、软件、数据库知识，建立新一代的数据采集车联网平台架构并贯穿车辆全生命周期。同时，统一数据应用标准，具体采集频率、长度、最小计量单元根据实际应用需要、流量限制进行了一定平衡，力争在最小的流量费用上实现更高质量的数据采集和传输。

5、数据采集质量有待提升。数据采集存在数据丢失、乱序、跳变、延时等问题，这些问题形成机制复杂，在分析解决数据采集质量问题时候，难度较大。行业内部对于采集的数据质量缺乏具体统一的评价体系，难以实现对数据采集质量状态好坏的评估。

6、数据采集灵活性不够。现有的数据采集方案往往通过车载通讯终端 T-BOX 固件中的采集功能或自行编写的采集程序进行车辆数据采集。通常采集程序所采集到的信息是固定且直接固化到车载终端上的，随着车联网的发展，数据采集项目将根据应用呈现更多变化。同时采集信息解析配置固定，无法匹配新车型项目或者总线变化。行业内部需要一种灵活的数据

采集方式和高效的车云数据协同。

7、数据采集安全和隐私保障。能够实现自动驾驶的汽车需要实时采集车辆道路目标物数据，比如行人、车辆、车牌、道路交通状况。这部分数据涉及到数据安全和隐私，对于这部分数据的采集需要严格按照国家相关法律法规执行。另外，涉及到人脸面部特征信息、车辆经纬度等个人用户敏感信息，按照《中华人民共和国个人信息保护法》规定应进行脱敏处理。同时，也要明确数据采集权责定义、数据归属权的划分，制定数据传输的安全规范和操作标准。

综上，当前汽车数据服务采集正处于不断发展和完善的阶段。隐私和安全问题以及数据标准化和整合仍然是需要解决的挑战。

2.3.1.3 开发现状

随着数据成为重要的新“生产资料”，对数据的开发和管理能力提出了新的挑战。通过构建数据开发与管理平台，构筑打通全数据链路的闭环系统，建立海量数据的获取、存储、计算、使用能力。在此基础上，云服务的海量存储和 AI 大数据挖掘和计算能力能够有效挖掘有价值的场景。

汽车数据服务的开发包括前端技术和后端技术。前端技术是汽车数据开发中至关重要的一环，它决定了用户体验和交互效果。在前端技术的选择上，目前比较流行的技术栈包括 Vue、React 和 Angular 等。后端技术主要负责汽车数据的获取、处理和存储等任务。为了提高汽车数据处理的性能和稳定性，选择合适的后端技术非常重要。常见的后端技术包括 Java、Python、PHP 等。选择哪种后端技术需要考虑数据量、并发量、可扩展性等因素。此外，由于目前汽车数据量级越来越大，成本越来越高，数据存储技术也是汽车数据后端开发中非常重要的一部分，它负责汽车数据的存储和管理。选择合适的数据库可以提高汽车数据处理能力和性能。常见的数据存储技术包括 MySQL、Oracle、MongoDB 等。选择哪种数据存储技术需要考虑数据量、数据类型、数据结构、性能需求等因素，建议选择分布式数据存储技术，以提高可扩展性和性

能。

数据开发管理包括数据预处理、数据存储、数据处理、数据可视化等各个阶段，以及数据调度、元数据管理、数据生命周期管理、数据安全等相关组件。数据预处理需要对采集数据进行系统化的清洗与分类分级，需要进行数据质量的度量和评价。数据存储是底层支撑技术，需要满足数据使用的高吞吐量、高可用性等要求。数据处理负责从数据中提取和挖掘信息，目前AI的发展使得数据挖掘的深度和维度大幅提升，从而支撑了更加智能化的汽车数据服务功能。数据可视化通过空间、时间等维度呈现数据，帮助数据治理者理解数据背后的规律和发展趋势。当前面临的主要问题有六个方面：

1. 数据融合问题。不同的采集方式上传的数据可能存在不同的频率、命名不一致、数据关联等问题；

2. 数据生命周期管理问题。数据缺乏明确的分类分级标准，以及不同不同分类分级标准下的生命周期管理。特别是汽车数据量大，对存储资源的占用很高。缺乏数据保存时间标准、归档和销毁标准、流程等；

3. 数据治理问题。数据质量不高，缺乏统一标准、统计口径不一致；缺乏数据管理流程；数据需要分层设计，提升复用率；

4. 统一数据平台问题。目前还存在烟囱式开发、重复建设的情况，需要统一的汽车数据平台来统一数据开发和治理；

5. 数据资产目录问题。需要统一的数据资产目录提升数据可见性；

6. 数据服务平台问题。通过API、数据集、数据应用等多种方式提升数据可用性。

2.3.1.4 功能现状

2.3.1.4.1 主动运维服务

传统汽车的运维是被动的，一般在发生故障后再到店由专业技术人员解决。随着智能网

联汽车的发展，软件类问题增多，且问题越来越复杂难以定位。主动运维的数据服务通过主动收集车辆数据并进行分析，可以对故障进行提前预警，并在云端有效收集与分析高频故障场景，大幅提升远程诊断和远程修复救援速度。对于车企来说一方面提升车辆安全性，另一方面也提升了售后服务效率。对用户来说能够更有效地保养车辆，降低故障影响保障行车安全。其中对于新能源车来说，动力电池是核心部件也是故障高发的重灾区，动力电池的安全是国家主管部门和消费者重点关注问题。电池安全与寿命管理为典型的主动运维服务。

电池系统在运行过程中不可避免地要经历苛刻的环境，如高温、低温、大倍率充放电等，监测和控制电池运行行为对于保证电池系统可靠性必不可少。

1) 热失控预警

电动车最大的安全隐患——电池起火，根据对电池指标的监测可以提前识别热失控风险并预警，防止电池热失控。预警能力还将随大数据和人工智能算法模型的持续优化而得到进一步的提升，预警的查全率和差准率将得到持续优化。

2) 电池寿命估算

基于云端大数据，以及电化学机理研究，构建云端电池 SOH 预测模型，准确地估算电池状态和老化程度。

3) 充电管理

实时评估电池状态并与电池负荷过重的状态加以比较，借由评估可实时优化电池组充电以便让充电更有效率，同时降低电池组长期损坏的风险。

2.3.1.4.2 全生命周期运营

汽车行业的数字化基于用户体验，实现用户全生命周期数字化服务。通过数字化平台，提升用户购车、用车的全过程的服务体系，涵盖车内数字化生活和后市场服务。平台通过数字化，进行用户画像描绘、事件发现、环境解析，还原用户的真实行为，感知用户的真实体验，洞察用户的真实需求，为场景化智能服务提供支撑。让服务方与用户实时连接，并通过线上线

下协同和智能化技术，提升服务体验。

通过用户画像总结个体/群体特征与规律，对产品研发、服务改善及市场竞争提供指导，帮助产品和品牌为用户提供更精细、更贴心的服务，同时可以为主机厂进行个性化服务、预测性维修保养、生态链开发、客户保留提供依据。

除了更加精准的用户服务以外，还可以根据用户驾驶行为数据开展更加精准的二手车评估和车险定价等业务。

2.3.1.4.3 智能驾驶研发服务

智能驾驶技术需要持续不断地积累大量的感知、决策和执行数据，并通过仿真模型和算法的训练实现算法的持续迭代。因此，有必要为智能驾驶的开发和迭代升级提供感知模型训练和仿真测试的数据服务。

感知模型训练主要包括在对采集到的海量繁杂数据进行挖掘、标注基础上，进行模型训练的过程。数据标注通过人工以及智能化工具，对传感器所捕捉的图像、视频、路牌文本等各类信息目标检测和识别模型训练基于构建的业务数据集，结合特定的模型算法，进行模型的训练与优化。数据服务可以有针对性地挖掘车端和云端的难例场景，形成难例数据集，持续提升标注算法和感知算法。在感知模型训练数据服务发展过程中，还面临着两个主要挑战：一是需要清洗海量数据中的敏感数据和无关数据，节省存储资源，保证合规。二是标注工作量大且质量参差不齐，影响模型的训练精准度。

在模型训练的基础上，仿真测试数据服务可以构建丰富的场景库，建立全面系统的模型模拟实际车路场景，对智能驾驶功能进行仿真测试，既弥补了实车路测的局限性，又提高了开发效率，是智能驾驶商用落地和迭代升级的加速器。仿真场景库的数量与质量决定了仿真测试的效果。目前仿真测试数据服务面临着三个主要挑战：一是仿真场景库覆盖度不足，且不同仿真库间因未形成标准化格式，互不兼容、无法通用，提升场景库的质量是行业的迫切需求。二是仿真测试所涉里程多、场景多，耗时大，仿真效率需要提升。三是仿真测试目前与实车路测

偏差仍然较大，如何精准模拟传感器、车辆动力学并集成到仿真测试平台，提高仿真的逼真度是仿真测试过程中的重要技术难点。四是仿真评价指标不完善，影响识别判断仿真测试过程中的问题，不能有效迭代优化仿真测试算法和场景库。

2.3.2 汽车数据服务标准需求分析

2.3.2.1 汽车数据服务采集标准现状分析

2.3.2.1.1 国内数据服务采集相关法规

1) 《中华人民共和国数据安全法》

2021年6月10日，第十三届全国人民代表大会常务委员会第二十九次会议通过《中华人民共和国数据安全法》（下称《数据安全法》），2021年9月1日实施。《数据安全法》是中国实施数据安全监督和管理的一部基础法律，适用于中华人民共和国境内开展的数据处理活动及其安全监管。该法体现了坚持总体国家安全观，聚焦数据安全领域的突出问题。确立了数据分类分级管理、数据安全风险评估、监测预警、应急处置、数据安全审查等制度，明确了相关主体的数据安全保护的义务和权益，同时鼓励数据依法合理有效利用，促进以数据为关键要素的数字经济发展。

2) 《汽车数据安全若干规定（试行）》

2021年7月5日，国家发展和改革委员会、工业和信息化部、公安部、交通运输部、国家互联网信息办公室五部门联合发布《汽车数据安全若干规定（试行）》，自2021年10月1日起施行。

《规定》倡导汽车数据处理者在开展汽车数据处理活动中坚持车内处理、默认不收集、精度适用、脱敏处理的原则。作为我国汽车行业数据安全领域的重要法规，《规定》确立了汽车行业数据安全及治理的基本框架，不仅就汽车数据处理者、汽车数据处理、个人信息和敏感个人信息等重要概念在《数据安全法》等上位法律法规相关定义的基础上作出了汽车行业上的

明确和细化，也提出了落实网络安全等级保护制度、汽车数据依法合理有效利用、告知用户数据收集的形式和用途、取得个人同意或符合法律规定、合法处理敏感个人信息、按规定风险评估并提交报告以及重要数据依法境内存储等几个方面的具体要求。

3) 《车联网（智能网联汽车）网络安全标准体系建设指南》

2021年6月21日，工信部就《车联网（智能网联汽车）网络安全标准体系建设指南》（以下简称《指南》）公开征求意见。《指南》提出了车联网（智能网联汽车）网络安全标准体系框架、重点标准化领域及方向，包括总体与基础共性、终端与设施安全、网联通信安全、数据安全、应用服务安全、安全保障与支撑六大类标准。在数据安全标准方面，《指南》建议从通用要求、分类分级、出境安全、个人信息保护、应用数据安全五类规范智能网联汽车、车联网平台、车载应用服务的数据安全和个人信息保护要求。

数据安全标准主要规范智能网联汽车、车联网平台、车载应用服务等数据安全和个人信息保护要求，包括通用要求、分类分级、出境安全、个人信息保护、应用数据安全等五类标准。

通用要求主要用于规范车联网（智能网联汽车）可采集和处理的数据类型、范围、质量、颗粒度等通用要求，包括数据最小化采集、数据安全存储、数据加密传输、数据安全共享等标准。

分类分级标准主要用于指导车联网（智能网联汽车）数据分类分级保护，制定数据分类分级的维度、方法、示例等，明确重要数据类型和安全保护要求。

数据出境安全标准主要用于规范车联网（智能网联汽车）行业依法依规落实数据出境安全要求，包括数据出境安全评估要点、评估方法等标准。

个人信息保护标准主要用于规范车联网（智能网联汽车）用户个人信息保护机制及相关技术要求，明确用户敏感数据和个人信息保护的场景、规则、技术方法，包括匿名化、去标识化、数据脱敏、异常行为识别等标准。

应用数据安全标准主要用于规范车联网（智能网联汽车）相关应用所开展的数据采集和处理使用等活动，包括车联网平台、网约车、车载应用程序等数据安全标准。

4) 在研标准《智能网联汽车 数据通用要求》

2020 年标准正式启动，2021 年下达标准计划号。2022 年 10 月发布征求意见稿。此标准是汽标委首项数据合规类标准，将作为未来主管部门管理汽车的重要今年，重点关注个人信息与重要数据的全生命周期数据处理要求。

标准要求企业建立汽车数据安全管理体系，要求个人信息应获得用户同意后可使用。要求对于车外个人信息应进行匿名化处理，并规范了匿名化处理的指标。对于重要数据的收集、存储、使用、传输、删除、出境提出要求，对个人信息和重要数据的存储和传输提出了加密要求。

此外，中国汽车工程学会在研团体标准《智能网联汽车测试场景数据车载采集平台搭建要求及方法》也规定了测试场景下的数据采集要求。

2.3.2.1.2 国际数据服务采集法规

1) 2013 年美国马萨诸塞州地方法院《机动车所有人维修权法案 An Act Relative to Automotive Repair (2013, Chapter 165)》

2012 年 7 月 31 日，美国马萨诸塞州立法机构颁布了维修权法案(以下简称《维修权法案》)，该《维修权法案》于 2013 年 11 月 26 日签署成为法律。2014 年初，汽车售后市场行业协会、汽车维修平等联盟、汽车制造商联盟和全球汽车制造商协会签署了一份基于马萨诸塞州的谅解备忘录法律，这将使汽车制造商承诺在所有 50 个州满足马萨诸塞州法律的要求。该措施旨在要求汽车制造商允许人们访问车辆数据进行维修。该《维修权法案》生效后，制造商将在 2022 年前为新车型安装标准的开放数据平台。该平台将使车主和独立机械师能够远程访问车辆信息（无线传输的数据通常直接发送到远程服务器）。

本规定适用于：该《维修权法案》适用于全美国汽车制造商。该《维修权法案》规定，汽车所有者拥有车辆的“维修权”，汽车制造商应当将服务信息和工具同时共享给特许经销商和其他独立维修店。

该《维修权法案》在 13 年通过时，将“信息通讯技术 (telematics)”信息，即汽车通过无线传输技术发送给制造商的信息，排除在了需要共享的范围外。但在 2020 年马萨诸塞州的投票中，这一排除也被推翻，汽车制造商将建立开放数据平台进行使用。

《维修权法案》第 2 章第 (f) 条规定，除了经销商诊断和维修汽车所必需的技术信息且该信息不会被其他独立维修机构获取外，其他的由车辆发出或产生的信息（信息通讯，或其他远程、信息诊断、服务信息等）将不适用该法案。

第 (f) 条注明的上述“信息”包括但不限于“自动安全气囊部署和碰撞通知、远程诊断、导航、被盗车辆定位、远程解锁车门、向公共安全应答点传送紧急情况 and 车辆位置信息以及任何其他整合车辆定位技术和无线通信的服务”

2) 2016 年欧盟《一般数据保护条例 2016/679》

2016 年 4 月 27 日欧洲议会和理事会通过了关于保护自然人处理个人数据和此类数据的自由移动的《一般数据保护条例 2016/679》（下文简称 GDPR），并废除《通用数据保护条例》。该条例于 2018 年 5 月 25 日开始强制执行。GDPR 是欧盟隐私法和人权法的重要组成部分。它还解决了欧盟和 EEA 地区以外的个人数据转移问题。GDPR 的主要目的是提高个人对数据的控制和权限，并简化国际业务的监管环境。GDPR 2016 有 11 个章节，关于一般规定，原则，数据主体的权利，数据控制人员或处理器的职责，个人数据转移到第三国，监管机构，会员国之间的合作，违反的责任，责任或处罚权利和杂项最终规定。

GDPR 是在 28 个欧盟成员国统一实施生效的，这将使 28 个欧盟及欧洲经济共同体成员国的隐私保护法更具有一致性和现代性。该条例是欧盟秉承着“顾客优先”的态度出台的个人数据保护新规，面向所有收集、处理、储存、管理欧盟公民个人数据的企业，限制了这些企业收集与处理用户个人信息的权限，旨在将个人信息的最终控制权交还给用户本人。

GDPR 适用于全自动化个人数据处理、半自动化个人数据处理，以及形成或旨在形成用户画像的非自动化个人数据处理。但不适用于自然人在纯粹个人或家庭活动中进行的个人数据处理，也不适用于有关主管部门为预防、调查、侦查、起诉刑事犯罪、执行刑事处罚、防范及

预防公共安全威胁而进行的个人数据处理。

GDPR 规定个人信息处理要遵循合法性、公开性、透明性、目的明确、数据处理最小化、信息准确性、存储限制性、数据完整性以及保密性等原则。数据保护原则不应适用于匿名信息，即与已识别或可识别的自然人无关的信息，或与以无法识别或不再可识别数据主体的方式匿名的个人数据有关的信息。GDPR 中“个人数据”指的是任何已识别或可识别的自然人（“数据主体”）相关的信息；一个可识别的自然人是一个能够被直接或间接识别的个体，特别是通过诸如姓名、身份编号、地址数据、网上标识或者自然人所特有的一项或多项的身体性、生理性、遗传性、精神性、经济性、文化性或社会性身份而识别个体。GDPR 中“处理”是指对个人数据或个人数据集执行的任何操作或一组操作，无论是否通过自动方式，例如收集、记录、组织、结构化、存储、改编或更改、检索、咨询、使用、通过传输、传播或以其他方式提供的披露、对齐或组合、限制、删除或破坏；GDPR 中“假名化”指的是在采取某种方式对个人数据进行处理后，如果没有额外的信息就不能识别数据主体的处理方式。此类额外信息应当单独保存，并且已有技术与组织方式确保个人数据不能关联到某个已识别或可识别的自然人。该定义与《个人信息保护法》中“去标识化”定义类似，但 GDPR 规定该类数据需要单独保存且需要由技术或管理约束确保不能再识别到已识别或可识别的自然人。

除此之外，GDPR 中还提出了“控制者”、“处理者”等概念，还列明了数据主体拥有“访问权”、“更正权”、“删除权”、“限制处理权”“可携带权”、“反对权”等权利以及其豁免条款。

欧盟 GDPR 条例具有域外效应。也就是说，GDPR 赋予了欧盟在个人信息安全方面的域外管辖权。主要受影响的企业为以下四类：设立在欧盟境内的企业（控制者、处理者）；未在欧盟境内设立，但向欧盟境内的数据主体（自然人）提供产品和服务的企业（控制者、处理者）；未在欧盟境内设立，但涉及监控欧盟境内数据主体（自然人）行为的企业（控制者、处理者）；未在欧盟境内设立，但在欧洲成员国法律通用的地方设立的企业（控制者、处理者）

GDPR 约束的数据包括：

个人数据：可以通过某个标识直接或间接识别某一自然人的信息。不管是采用自动化手段还是人工进行归类的数据，包括按时间顺序排列的包含个人数据的记录集合。已经被匿名化的个人数据，取决于用已有标识来识别特定个体的困难程度。

敏感个人数据：也被称为“特殊种类的个人数据”。包括揭示种族或民族出身、政治观点、宗教或哲学信仰、工会成员的个人数据。包括遗传数据和经过处理可以唯一识别个体的生物特征数据。不包括涉及刑事定罪和罪行的个人数据，但该类数据的处理和保存有特殊要求。

总结来说，GDPR 不仅适用于位于欧盟境内的企业组织机构，也适用于位于欧盟以外的企业组织机构，无论机构所在地位于哪里，只要其向欧盟数据主体提供产品、服务或者监控相关行为，或处理和持有居住在欧盟境内的数据主体的个人数据，都将受到条例的监管。条例同样适用于“数据控制者”和“数据处理者”。如果是数据处理者涉案，数据控制者也无法免除责任，GDPR 规定控制者需要承担更多的责任，以确保和数据处理者之间的合同能够严格遵守 GDPR 的规定。

3) 2018 年欧盟《非个人数据自由流动框架》

2018 年 11 月 14 日欧洲议会和理事会正式通过关于非个人数据在欧盟自由流动框架的第 2018/1807 号条例《非个人数据自由流动框架》。该法规的主要目的是提高非个人数据在单一市场中的跨境流动性；确保主管当局出于监管目的的请求和接收数据访问权的权力不受影响；使数据存储或其他处理服务的专业用户更容易切换服务提供商和移植数据，同时不会给服务提供商造成过度负担或扭曲市场。

该法案适用于在欧盟范围内处理个人数据以外的电子数据，即：

- (1) 作为服务提供给居住或设立在欧盟的用户，无论服务提供商是否在欧盟设立；
- (2) 由居住或在欧盟设立的自然人或法人出于自身需要进行数据处理；
- (3) 对于混合数据集，即由个人数据和非个人数据组成的数据集，该条例适用于数据集的非个人数据部分。如果个人数据和非个人数据有着千丝万缕的联系，则该条例的适用不影响 GDPR 的适用。

该法案允许非个人数据在欧盟内存储和处理数据，排除了未提及的限制，推动废除欧盟成员国当地立法中对数据本地化的限制。

2.3.2.1.3 数据服务采集标准需求分析

数据采集是数据处理活动的首要环节。采集过程的质量，从根本上决定了产生数据生命周期的质量。在采集阶段的主要问题在于采集的必要性、数据格式、精度、范围、和合理的用户授权等方面行业内衡量标准不统一。

在采集阶段过程进行标准化定义有助于：

1) 提出行业的数据采集过程规范，保证数据的产生过程科学有序，合理合法。为国家监管部门提供监管依据。

2) 完善公共数据开放共享机制。从源头解决数据流通的前提条件。推进公共数据归集整合、使有序流通和共享成为可能。

3) 建立健全数据流通交易规则，梳理“可控可计量”方案。探索“原始数据不出域、数据可用不可见”的交易范式，在保护个人隐私和确保数据安全的前提下，分级分类、分步有序推动部分场景数据流通应用。探索建立数据用途和用量控制制度，实现安全、规范的数据使用模式。

针对智能网联汽车车载软件，对数据采集的标准化需求主要体现在采集数据的定义与标准化。在采集阶段，现存主要问题在于数据格式、频率、精度、范围等方面行业内衡量标准不统一，而数据价值直接取决于其相关属性。因此有必要首先对车载数据进行定义，并进一步针对不同分类车载数据的格式进行标准化，这将有利于全行业降低数据流动过程中，对数据进行评估/处理等等成本。

除采集数据格式外，目前市场上数据采集设备多样，建议定义数据采集设备规范，以便于更好支持跨设备采集，规范相应市场，更好地服务于规范的数据采集活动。数据采集活动可能会对车辆正常性能产生影响，建议规范数据采集的性能开销指标和对应测试方法。

2.3.2.2 汽车数据服务开发标准现状与需求分析

目前已经有数据管理相关标准，例如《GB/T 36344-2018 信息技术 数据质量评价指标》、《GB/T 39400-2020 工业数据质量 通用技术规范》、《GB/T 39445-2020 公共信用信息数据元》等，但是专门针对汽车领域的数据服务开发标准较少。由于各企业都是参考软件开发类标准，自行开发汽车相关数据，因此缺少统一的、规范的、针对汽车数据服务的开发类标准，需要规范汽车数据服务开发的各个环节（如预处理、存储、处理与分析）的技术要求，从而指导汽车数据的开发过程。随着数据成为车企越来越重要的资产，需要就汽车数据治理制定统一的数据治理和数据管理标准，明确不同类型数据的质量指标和生命周期管理规则。

2.3.2.3 汽车数据服务功能标准现状与需求分析

目前企业各自开展数据服务功能开发，缺乏统一的汽车数据服务标准。可从数据服务功能的功能要求、性能指标、接口等方面制定标准。

2.3.2.4 汽车数据服务通信与安全标准现状与需求分析

随着智能化、互联化发展的趋势，汽车已经成为智慧城市建设的重要组成部分之一。而在这个过程中，汽车数据服务通信（V2X）和汽车网络安全问题愈来愈受到关注。

2.3.2.4.1 汽车数据服务通信标准现状与需求分析

汽车数据服务通信是指车辆之间、车辆和基础设施之间，以及车辆与云平台等之间的信息交互。它可以帮助车辆实现诸如自动驾驶、交通流优化、车队协作等功能。目前，V2X通信主要有以下两种方式：

① 车对车通信（Vehicle-to-Vehicle，简称V2V）

V2V通信是指车辆之间的通信，通过车载设备进行信息交换，包括车辆间的协作行驶、车

队行驶等。例如，当车辆遇到交通堵塞时，它可以通过车载设备向附近的车辆发送信息，提醒它们采取相应的措施减缓拥堵。

② 车对基础设施通信 (Vehicle-to-Infrastructure, 简称 V2I)

V2I 通信是指车辆与道路基础设施之间的通信，包括交通信号灯、路况信息等。通过这种方式，车辆可以获取当前路段的实时信息，帮助驾驶员优化行驶路径，并提高安全性。

除了 V2V 和 V2I 外，V2X 还包括 V2P (Vehicle-to-Pedestrian) 即车辆与行人之间通信等。

为了保证 V2X 通信的稳定性、可靠性和安全性，需要制定统一的通信标准。目前国内外的相关机构和组织正在积极推进相关标准的制定和完善。例如，欧洲车联网协会 (ERTICO) 制定了 Vehicle-to-Anything (V2X) 通信体系结构和标准规范等。

除此之外，目前还存在许多不同的 V2X 通信标准，如 IEEE 802.11p、LTE-V 等。这些标准在技术特性、适用场景、通信协议等方面存在差异，需要根据实际需求进行选择。

关于汽车数据服务通信的需求分析有以下几个方面：

① 提高效率：汽车数据服务通信可以帮助车辆之间实现信息交流和协作，提高行驶效率，缓解交通拥堵，降低出行时间和成本。

② 增强安全性：通过汽车数据服务通信，车辆可以获取实时路况、交通拥堵状况等，提前规避风险，增强行车安全性。

③ 促进智能化：汽车数据服务通信可以为车辆提供自动驾驶、交通流优化、车队协作等功能，促进汽车智能化发展。

④ 改善用户体验：汽车数据服务通信可以为驾驶员提供更多种类的信息和服务，改善用户体验。

2.3.2.4.2 汽车数据安全标准现状与需求分析

随着车联网技术的发展，汽车数据安全问题也日益突出。尤其是近年来，由于智能汽车

使用软件和互联技术的增加，使得汽车系统面临越来越多的安全威胁，如黑客攻击、恶意软件等。因此，制定汽车数据安全标准是十分必要的。

在过去的几年里，汽车行业中出现了一些严重的网络安全事件，这些事件引起了人们的广泛关注，也促使汽车行业开始加强汽车网络安全方面的投入和应对措施。目前，国际上出现了一些针对汽车网络安全的标准和规范，以确保汽车系统的安全性。

在 2016 年，ISO/SAE J3061 出台并发布，作为一种针对汽车安全标准，提供了一套流程以确保汽车系统的安全性，包括风险评估、威胁建模、安全性验证等内容。这个标准逐渐成为汽车行业中通用的汽车信息安全管理体系标准，并且被认为是未来汽车安全领域的参考标准。

除此之外，欧洲车联网协会（ERTICO）也提出了相应的安全标准，如 EU-SEC（European Secure Software Defined Radio）项目、CERT4ITS（Computer Emergency Response Teams for Intelligent Transport Systems）项目等。这些标准和规范提供了一个框架，帮助汽车制造商开发出更安全的汽车系统。

目前，国内外正在积极推进汽车数据安全标准的制定和完善。例如 2022 年 10 月发布的 GB/T 41871—2022《信息安全技术 汽车数据处理安全技术要求》，从汽车数据处理活动通用安全、车外数据安全、座舱数据安全和管理安全四个方面对汽车数据处理者提出要求，也规定了不适用该标准的特例汽车数据处理活动，是对《汽车数据安全管理办法（试行）》条文的进一步细化。还有正在研制的 GB《汽车整车信息安全技术要求》，也对车辆数据的安全保护提出了技术要求和测试方法。正在研制中的 GB/T《智能网联汽车 数据通用要求》规定了汽车数据的个人信息保护要求、重要数据保护要求、审核评估及试验要求。

关于汽车数据安全标准的需求分析有以下几个方面：

① 防范网络攻击：由于汽车系统使用软件和互联技术的增加，使得汽车系统面临越来越多的网络攻击，需要制定相应的安全标准来防范这些攻击。

② 保护隐私安全：车辆通过数据服务通信可以获取大量个人信息，因此需要制定相应

的隐私保护措施和安全标准，确保个人信息不被泄露。

③ 提高安全性：汽车是人们生命财产的重要财产之一，需要确保其正常运行和安全性。

制定相应的数据安全标准可以提高车辆的安全性。

④ 促进创新发展：制定统一的数据安全标准可以为汽车行业创新发展提供保障，促进汽车智能化和互联网化进程。

2.3.3 数据服务软件标准体系框架及分类说明

表16 数据服务软件标准体系

标准项目及分类		标准类型	标准描述	备注
数据服务软件（400）				
数据采集（410）				
410-1	智能网联汽车 车用数据格式及编码	国标推荐	规范汽车数据格式与编码，可用于规范汽车采集数据格式等	
410-2	汽车数据采集方法要求	行标	规范汽车数据采集过程与方法，包括对采集设备的要求，对采集性能开销的要求与试验方法	
数据开发（420）				
420-1	汽车数据预处理技术要求	行标	规范汽车数据清洗和预处理的基本方法和技术要求	
420-2	汽车数据存储技术要求	行标	规范汽车数据存储的技术要求和试验方法	
420-3	汽车数据处理与分析技术要求	行标	规范汽车数据处理和数据分析的技术要求和试验方法	
420-4	汽车数据质量规范	行标	定义汽车数据模型和数据质量评价指标，对汽车数据质量管理提供指导	
420-5	汽车数据治理体系要求	行标	汽车企业对数据实施治理的指南	
数据功能（430）				
430-1	主动运维功能定义与技术要求	行标	主动运维功能定义与技术要求	
430-2	主动运维功能接口规范	行标	规范主动运维功能与车端的接口	
430-3	全生命周期运营功能定义与技术要求	行标	全生命周期运营功能定义与技术要求	

430-4	全生命周期运营功能接口规范	行标	规范全生命周期运营功能与车端的接口	
430-5	智能驾驶研发服务功能定义与技术要求	行标	智能驾驶研发服务功能定义与技术要求	
430-6	智能驾驶研发服务功能接口规范	行标	规范智能驾驶研发服务功能与车端的接口	
数据安全（440）				
440-1	汽车数据处理安全技术要求	国标推荐	汽车数据进行传输、存储和出境等处理活动的技术要求	GB/T 41871 — 2022 《信息安全技术 汽车数据处理安全技术要求》，已发布
440-2	智能网联汽车 数据通用技术要求	国标推荐	规定了汽车数据的一般要求、个人信息保护要求、重要数据保护要求、审核评估及试验要求等	已报批
440-3	汽车数据安全管理体系规范	国标推荐	汽车企业建立实施数据安全管理体系的指南	汽标委 SC34 已启动研究
440-4	汽车整车信息安全技术要求	国标强制	规定了汽车信息安全管理体系要求、信息安全一般要求、信息安全技术要求、试验方法及同一型式判定，其中数据安全要求章节要求车辆应保护存储在车内车辆关键数据、敏感个人信息、安全日志、密钥等	已报批

2.4 工具软件

2.4.1 概述

随着汽车行业的数字化、网联化和智能化发展，汽车在研发、生产、销售等过程中所用到的工具也在飞速发展。从早起的扳手等硬件工具，到现在的总线监听类软件工具，汽车所涉及到的工具也发生了质的变化。基于当前的智能网联汽车环境，其所涉及的工具类软件体系贯穿着汽车软硬件研发的各个环节。复杂度较高的汽车软件系统在开发过程中涉及芯片厂商、中间件供应商、Tier1、OEM及服务外包等众多角色，不同团队采用不同流程、方法和标准，导致软件兼容性较差，软件质量参差不齐。将汽车工具类软件细致划分品类，在涵盖多种功能模块、跨越多个开发层次、适应多类供应商平台的同时，兼顾安全性、平台型和规范性要求，从

而在开发过程的不同环节使用对应的汽车工具类软件，可进一步提升开发效率，有助于共同完成对整车汽车电子软硬件产品的开发。

根据应用场景及功能特性不同，整个工具类软件可分为需求分析软件、系统设计软件、开发软件、测试验证软件、项目管理软件和后市场工具软件6大部分，工具软件类整体逻辑框架如图17所示。



图17 工具软件类标准逻辑框架

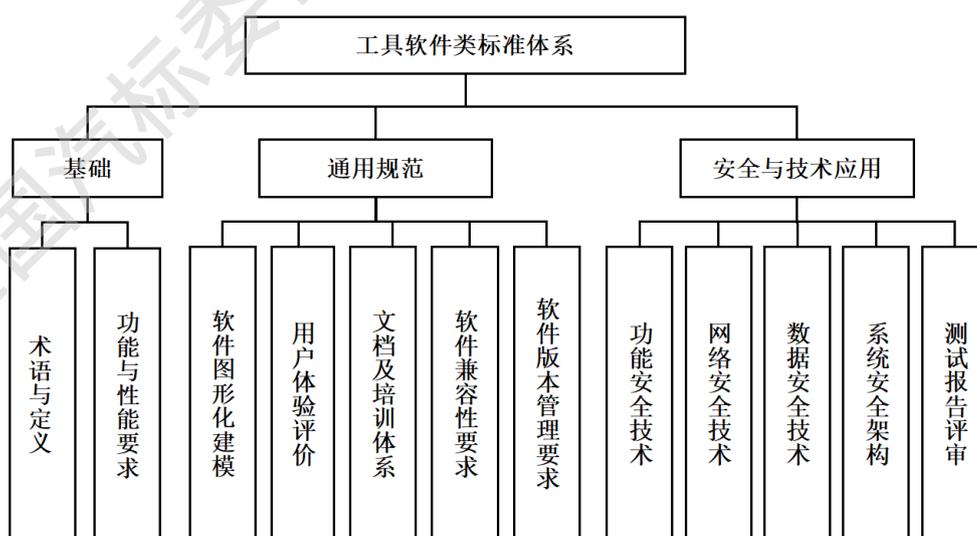


图18 工具软件类标准体系

工具软件通用要求标准框架及分类说明见表17。

表17工具软件通用要求标准体系

标准项目及分类		标准类型	标准描述	备注
工具软件（500）				
工具软件通用要求（510）				
510-1	汽车软件工具分类与术语定义	行标	分别针对汽车全生命周期所涉及的工具进行分类，并针对工具的使用和定义规范其术语和定义	
510-2	工具软件功能及性能技术要求及测试方法	团标	各类工具软件所通用的，制定相关标准规范以规定汽车工具类软件应具备的基础功能要求，包含记录文档的版本号、变更信息、著作人、主要内容以及注意事项等；同时规定需求分析软件的性能和效率要求，包含计算速度、内存占用、响应时间等方面的要求	
510-3	工具软件功能安全技术要求及测试方法	团标	各类工具软件所通用的，制定汽车工具类软件功能安全相关标准及对应测试方法，确保软件开发工具能满足功能安全对工具的要求	
510-4	工具软件网络安全技术要求及测试方法	团标	各类工具软件所通用的，制定汽车工具类软件网络安全相关标准及对应测试方法，为保障开发者、企业和最终用户的利益，并支持汽车软件的网络安全需求，具体可从漏洞扫描及修复、安全分析、测试、应急响应计划等方面开展工作	
510-5	工具软件数据安全技术要求及测试方法	团标	各类工具软件所通用的，制定汽车工具类软件数据安全相关标准及对应测试方法，规定敏感数据的存储、传输和处理方式，包括合适的加密算法、密钥管理、访问控制等，以保护数据的机密性和完整性；同时在软件使用过程中需要保护用户的个人隐私信息，包括数据采集、使用和销毁等方面的规定，规定对个人和敏感数据的处理方式，包括存储、传输访问控制，以符合相关隐私法律	
510-6	工具软件系统安全架构规范	团标	各类工具软件所通用的，制定汽车工具类软件系统安全架构相关标准规范，确定安全架构的设计原则和安全防护机制，包括身份认证、防火墙等，以确保系统的安全性	
510-7	工具软件图形化建模规范	团标	各类工具软件所通用的，制定相关标准规范以规定汽车工具类软件的图形化建模要求，具体基于面向对象设计的建模，从基元	

				模型、元模型、模型实例等方面开展相关工作	
510-8	工具软件文档及培训体系规范	团标		各类工具软件所通用的，制定相关标准规范以规定汽车工具类软件的文档和培训要求，具体包括用户手册、技术文档、培训资料等方面，可有助于用户理解和正确使用工具软件	
510-9	工具软件版本管理通用要求	团标		各类工具软件所通用的，制定相关标准规范以规定后期维护管理的相关要求，定义软件开发工具更新和维护细则以及软件更新的方式，以满足市场需求和用户期望，从而保障软件升级过程用户及产品工具的安全	
510-10	工具软件功能及性能技术要求及测试方法	团标		各类工具软件所通用的，制定相关标准规范以规定汽车工具类软件应具备的基础功能要求，包含记录文档的版本号、变更信息、著作人、主要内容以及注意事项等；同时规定需求分析软件的性能和效率要求，包含计算速度、内存占用、响应时间等方面的要求	

2.4.2 需求分析软件

2.4.2.1 发展历程

需求分析软件是指管控项目需求工程的系统工具，需求工程作为工程的开端，包括需求的开发过程和需求的管理过程。需求开发指的是从需求收集、分析、评估到审批等一系列产生需求的活动，通过需求开发获取能用于系统开发的研发需求。需求管理是与系统开发过程中控制和维持需求约定的活动，包括变更控制、版本控制、需求跟踪、需求状态跟踪等，目的是确保各方对需求理解的一致性，实现需求与最终产品之间的双向跟踪。需求分析软件发展历程可分为以下几个阶段：

(1) 2000年前，计算机在中国的使用率还非常低，需求管理工具普遍都是使用纸质记录模式，无论是需求的时效性和分析的高效性都比较落后，需求管理需要依赖大量的纸质文档进行跟踪记录。

(2) 2000-2010年，随着计算机在中国的逐步引入，office 相关工具被企业接受并逐步普及，excel 表格的需求管理模式逐步在各大公司中使用，需求过程也逐步工程化，需求能实现

搜索与跟踪，需求分析也较为便捷，但在时效性和便捷性上还得依赖大量的人工。

(3) 2010年-2015年，随着计算机的普及以及工程行业对需求工具需求的日益增加，开始出现PLM产品生命周期管理模式的系统，如：Dassault Systemes、Oracle Agile PLM、PTC Integrity、perforce等。PLM按照传统的瀑布模型开发模式设计，能够实现需求的全面管控，但是过程比较严格，无法实现开发过程的快速迭代和敏捷管理。

(4) 2015年-2020年，随着互联网行业对汽车传统工业的冲击，汽车产品向智能化产品的转型，汽车控制器的开发逐步传统的瀑布模式研发转型敏捷模式，对需求过程要求精细化、高效性管理，主机厂的控制器研发部门开始积极的采用互联网行业普遍使用的敏捷开发管理工具，如：Atlassian JIRA、易软天创-禅道、西门子-Polarion等，能够实现需求的快速开发、版本的快速迭代，从而加快控制器的研发过程。但是在多个控制器软件协同开发方面，依然存在诸多不便。

(5) 2020年后，各大行业对需求管理工具的依赖越来越高，为了寻求一款完没匹配本公司业务需要的需求工具，各大厂商纷纷内部研发需求工具，如华为 DevCloud、腾讯 TAPD、中兴 RDCloud 等等。但是，作为汽车主机厂，既要满足单控制器的快速迭代要求，又要满足整车控制器自上而下的协同开发要求，仅依赖单项目的敏捷开发管理工具无法满足汽车行业的需求工程管理需要。

2.4.2.2 特点

汽车智能化的大趋势下，“软件定义汽车”成为产业共识，软件将深度参与到汽车定义、开发、验证、销售、服务等过程中。相较于IT行业的软件研发，汽车软件研发存在需求工程庞大、需求分析复杂、需求参与方多、集成难度大、需求验收难等问题，这也形成了汽车需求分析软件独有的特点：

(1) 需求管控难度大：汽车软件研发属于一个大型的集成项目，需求分析层面覆盖车控、驾驶、座舱、动力等众多专业域，包含控制器数目较多，涉及多方渠道的供应商；

(2) 需求分解跟踪难度大: 汽车软件需求分析呈现多层次、逐级分解的特点, 从用户需求、产品需求、架构需求、系统需求、至软件需求, 涉及用户、产品、架构师、系统工程师、软件工程师等多类角色;

(3) 相互依赖性强: 汽车软件对其支撑的硬件属于强依赖关系, 相较于互联网软件对服务器的弱依赖, 汽车控制器硬件的变更基本都会导致软件的变更, 软硬件相互依赖;

(4) 协议需求统一化: 需求相较于互联网软件依赖的标准网络协议, 汽车软件存在很多非标准、非平台化的协议, 需要将各类协议以需求的方式纳入管控范围。

2.4.2.3 发展现状

汽车研发从传统汽车转向智能化汽车的过程中, 需求工具也从传统的瀑布模式管理转向敏捷迭代管理。明确各阶段的工具的优缺点后, 针对汽车行业软件需求工具具有以下特点: 单个控制器的需求管理上升到多控制器联动的需求集管理, 又从多控制器联动的需求集管理期望能实现整车集成的需求管理。需要需求工具实现整车软件开发过程中的需求分层分级管理, 又要实现对各层级需求的精细化、快速迭代管理, 从而达到整车-零部-软件的分布与集成的协同开发, 最终实现整车软件研发的数字化运营。

目前, 行业中比较热门的JIRA、禅道、TAPD、DevCloud等敏捷开发管理工具, 虽然能实现单控制器的敏捷开发, 但是对整车层面的开发管理管控非常薄弱, 市面上基本找不到一款能够完全匹配智能化汽车软件需求管理的工具, 自研或者在现有主流管理工具上二次开发成为目前汽车软件需求管理工具的必经之路。

2.4.2.4 标准需求

需求分析类工具, 主要围绕汽车软件的需求分析过程, 实现整车-零部件-软件的多层级分解的管理策略, 业务层面需求管理目标要求工具能够从整车、控制器以及软件多层面进行

需求管理。业务层对于需求分析工作的要求，从需求分析阶段开始，在架构设计、详细设计、软件开发、软件测试、系统集成等多阶段都有相对应的需求文献记录，每一个阶段的需求分解内容，在一定程度上相关联但又不完全一致。对于技术层面的解析由浅入深。在工具实现层面，需要做到用户需求-产品需求-架构需求-系统需求-软件需求的双向可追溯，需求状态实时可追踪，并且实现数字化大屏展示。需求分析类软件在标准上应有以下要求：

(1) 功能性要求：标准体系应规定汽车需求分析工具应具备需求分析类工具的基础功能要求，包含记录需求文档的版本号、变更信息、著作人、主要内容以及注意事项等等，并且支持一定格式的文档上传，并与相关功能开发内容和测试内容建立关联关系，保证其可追溯性。

(2) 性能和效率要求：标准体系应规定汽车需求分析工具软件的性能和效率要求，例如计算速度、内存占用、响应时间等方面的要求。这有助于确保软件在实际应用中具有良好的性能和效率；

(3) 安全性和可靠性要求：标准体系应规定汽车需求分析工具软件的安全性和可靠性要求，包括数据安全、防止错误和故障的能力、可靠的结果和报告生成等方面的要求；

(4) 用户界面和用户体验要求：标准体系应规定汽车需求分析工具软件的用户界面和用户体验要求，包括界面设计、交互方式、可定制性、可访问性等方面的要求。这有助于提供用户友好的使用体验和操作便捷性；

(5) 文档和培训要求：标准体系应规定汽车需求分析工具软件的文档和培训要求，包括用户手册、技术文档、培训资料等方面的要求。这有助于用户理解和正确使用工具软件。

2.4.2.5 标准框架及分类说明

表18 需求分析软件标准体系

标准项目及分类	标准类型	标准描述	备注
工具软件（500）			

需求分析软件（520）				
520-1	需求分析软件用户体验测评准则及评价规程	团标	规定需求分析软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
520-2	需求分析软件兼容性通用要求	团标	规定需求分析软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	

2.4.3 系统设计软件

2.4.3.1 发展历程

回顾国内发展现状，一些知名的汽车系统设计软件代表了系统设计软件的发展历程。主要包括汽车电子系统仿真软件、汽车电子控制单元（ECU）开发工具、汽车网络通信协议分析工具等。其他系统设计层面的工具研发商，例如 CAD 相关领域的相关技术企业等等，在移动 CAD 系统架构、移动场景用户交互、图形数据云存储、跨平台 CAD 云协作、轻量化图形显示处理等方面拥有核心技术。除此之外，随着各家整车企业着力推进信息化建设，主要是为了满足整车制造业务所需，强调过程的标准化和结果的可预知，并初步建立了面向现代制造的信息化业务架构体系。“十三五”期间，在信息化建设基本完成的基础上，各家正在努力推进数字化转型，着眼于以数据赋能业务，开展创新探索，并取得一些初步的成效。其他公司则不断推出新的产品和技术，以满足汽车制造商和供应商在系统设计的需求，并且也在不断加强与汽车制造商和供应商的合作，共同推动汽车电子技术的发展。这些软件供应商应用范围主要集中在整车设计、车身设计、底盘设计和动力系统设计等方面。

2.4.3.2 特点

针对目前汽车系统设计类工具软件，可总结出以下特点：

(1) 多领域综合性强：汽车系统设计软件通常涵盖多个领域，可集成不同学科模型与工

具。它们提供综合的工具和功能，以支持整个汽车系统的设计和开发，从而实现多学科协同设计和分析，同时有助于确保不同系统之间的一致性和协调性；

(2) 在计算机上模拟和验证车辆的性能、行为和功能，同时此类工具软件也具备自动化和优化功能，能够通过算法和优化方法，自动化地生成和优化设计方案。这有助于提高设计效率和产品质量，并满足特定的性能和要求，使得设计团队可以在实际制造之前进行预测和优化，加快开发周期并降低成本；

(3) 标准化和兼容性强：汽车系统设计软件通常严格遵循行业标准和规范，确保与其他系统和组件的兼容性，且需提供数据管理和版本控制功能，以便于实现不同供应商和厂商之间的集成和协作，提升管理设计数据和文档效率。

2.4.3.3 发展现状

目前，汽车系统设计软件在国内外都有较为广泛的发展，并且随着汽车工业的进一步发展，这类软件的需求及应用市场也在不断增长扩大。国外的一些知名汽车系统设计软件供应商，如安波福、西门子 PLM 软件、达索系统等，它们在汽车行业软件领域有着广泛的应用和较高的市场份额。当前国外的汽车系统设计软件在性能、功能和技术创新方面暂处于领先地位，它们通常具有先进的模拟仿真、虚拟样机、碰撞模拟等功能，能够提高设计效率和产品质量。在国际标准方面，国际上的汽车系统设计软件通常符合 ISO 26262 功能安全标准、ISO 9001 质量管理标准等行业标准，以确保软件在设计、开发和测试过程中的安全性和质量。与此同时，国外软件供应商与汽车制造商和供应商之间还有着密切的合作关系，以满足不断增长的技术需求，并在新技术领域如自动驾驶和电动车等进行创新。

回顾国内发展现状，一些知名的汽车系统设计软件供应商包括浩辰科技、上海汽车集团股份有限公司、北京金顶科技等，其应用范围主要集中于整车设计、车身设计、底盘设计和动力系统设计等方面。浩辰软件是从「芯」出发，持续丰富国产 CAD 应用场景，不断拓展软件技术与应用边界。目前，浩辰 CAD 看图王在各平台应用市场下载量排名前列并持续保持领先地位。

位，发展成为一款国内外移动端 CAD 品类的头部应用。从电动汽车到氢能源汽车，上汽在创新端始终以这个目标为关键核心技术。作为燃料电池系统的核心，捷氢科技 PROME M3H 电堆完全自主设计开发，58 个一级零部件全部国产化，自主化程度和国产化率均达到了 100%，并且各关键参数均处于国际领先地位，用实力领跑行业。北京金顶科技是一家专注于汽车电子系统设计和开发的软件公司。其主要产品包括汽车电子系统仿真软件、汽车电子控制单元 (ECU) 开发工具、汽车网络通信协议分析工具等。金顶科技的软件产品已经被广泛应用于国内外的汽车制造商和供应商中。国内的汽车系统设计软件虽已取得一定的技术成果，但在一些关键技术领域与国外的领先软件相比仍存在一定差距。随着国内汽车市场的快速增长，汽车系统设计软件的需求也在不断增加，用户对软件的功能性、性能和易用性等方面有着更高的要求。汽车系统设计软件在国内外都面临着持续发展的机遇和挑战，未来的发展将更加注重模拟仿真、自动驾驶、电动化和智能化等方面，以满足不断增长的汽车市场需求和技术创新的要求。模拟仿真在汽车系统设计软件中的应用将更加广泛，通过虚拟样机和仿真分析，可以在设计阶段就发现并解决问题，提高产品质量和减少开发成本。随着电动车和智能汽车的快速发展，汽车系统设计软件也将更加关注电动驱动系统、电池管理系统、智能感知和控制等方面的设计和优化，以实现系统更高层次的电动化和智能化。在数据驱动设计层面，大数据和人工智能技术的应用将进一步推动汽车系统设计软件的发展，通过分析海量数据和智能算法，优化设计和改进决策过程。

2.4.3.4 标准需求

汽车系统设计类工具软件在标准体系要求层面可分为以下几个方面：

- (1) 功能性要求：标准体系应规定汽车系统设计工具软件应具备的基本功能和特性，例如模型创建和编辑、仿真和模拟、分析和优化、数据管理和版本控制等方面的功能要求；
- (2) 格式和交换标准：标准体系应规定与汽车系统设计相关的数据格式和交换标准，以确保不同工具软件之间的数据互操作性。这包括模型、图形、文档和其他设计数据的格式和交换

标准；

(3) 性能和效率要求：标准体系应规定汽车系统设计工具软件的性能和效率要求，例如计算速度、内存占用、响应时间等方面的要求。这有助于确保软件在实际应用中具有良好的性能和效率；

(4) 安全性和可靠性要求：标准体系应规定汽车系统设计工具软件的安全性和可靠性要求，包括数据安全、防止错误和故障的能力、可靠的结果和报告生成等方面的要求；

(5) 用户界面和用户体验要求：标准体系应规定汽车系统设计工具软件的用户界面和用户体验要求，包括界面设计、交互方式、可定制性、可访问性等方面的要求。这有助于提供用户友好的使用体验和操作便捷性；

(6) 标准符合性和认证要求：标准体系应规定汽车系统设计工具软件的标准符合性和认证要求，包括软件开发过程符合性、功能验证和认证等方面的要求。这有助于确保软件符合相关的行业标准和规范；

(7) 文档和培训要求：标准体系应规定汽车系统设计工具软件的文档和培训要求，包括用户手册、技术文档、培训资料等方面的要求。这有助于用户理解和正确使用工具软件。

2.4.3.5 标准框架及分类说明

表19 系统设计软件标准体系

标准项目及分类		标准类型	标准描述	备注
工具软件（500）				
系统设计软件（530）				
530-1	系统设计软件用户体验测评准则及评价规程	团标	规定系统设计软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
530-2	系统设计软件兼容性通用要求	团标	规定系统设计软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	

2.4.4 开发软件

2.4.4.1 发展历程

汽车软件开发工具的发展历程可以追溯到 20 世纪 60 年代。当时的汽车制造商开始意识到计算机技术对于汽车开发的重要性，开始着手开发自己的计算机辅助设计和计算机辅助制造软件，即 CAD/CAM 软件。而随着计算机技术的不断发展，汽车软件开发工具也经历了多个阶段的变迁，主要可根据发展历程概括为命令行工具、可视化工具、IDE 集成开发环境和云开发工具共四类：

(1) 命令行工具：在 20 世纪 60 年代至 80 年代初期，软件开发工具非常有限，人们主要使用命令行工具来进行软件开发，这些工具通常是基于终端的，用户需要手动输入命令并接收响应。由于是手工编写的，没有专门的工具支持。这种方式效率较低，容易出错，难以保证软件的质量和可靠性；

(2) 可视化工具：20 世纪 80 年代至 90 年代初期，随着计算机图形技术的发展，各厂商开始使用可视化工具来进行软件开发，用户被允许通过图形界面实现交互和编写代码，大大地提高了开发效率；

(3) IDE 集成开发环境：20 世纪 90 年代至今，随着电子设备在汽车中的应用越来越广泛，软件的复杂性也在增加。为了提高开发效率和代码质量，集成开发环境（IDE）开始在汽车软件开发中应用。IDE 提供了更好的代码编辑、调试和版本控制功能，使得软件开发变得更加便捷和高效；

(4) 云开发工具：近年来随着云计算技术的不断发展，汽车制造商也开始尝试使用云开发工具来进行软件开发，开发人员在云端进行开发、测试和部署，一定程度上提高了开发效率和灵活性。云原生平台能提供容器化、微服务架构等技术，使得软件开发更加敏捷和灵活，同时

也提高了软件的安全性和可靠性。

总而言之，随着汽车电子技术的发展，汽车软件开发工具也在不断创新和演进。从传统编程环境到集成开发环境和云开发工具等，均有效推动了汽车软件开发的效率和质量的提升，也为智能化汽车的实现提供了支持。

2.4.4.2 特点

在汽车软件开发过程中，需要面对各种复杂的问题，对其安全性、可靠性、可维护性和系统性能均提出较高要求，开发软件类工具普遍具有以下特点：

(1) 可定制化程度高：汽车软件开发工具通常用于支持特定的汽车设计和制造需求，因此通常具有高度定制化的特点，以满足根据不同使用者的需求进行定制开发；

(2) 开发复杂度高：汽车软件开发工具通常需要支持多种不同的汽车设计和制造过程，涉及多种编程语言融合、复杂的集成和测试验证等工作；

(3) 系统安全性高：汽车软件开发工具通常需要进行严格的安全测试和认证，以确保工具的安全性和可靠性；

(4) 维护及兼容性强：汽车行业的软件开发工具需要支持多种不同的硬件平台和操作系统，以便满足使用者的需求，且同时需要能够方便地进行维护和更新，以确保软件的长期稳定运行。

2.4.4.3 发展现状

随着汽车智能化、网联化、电动化、共享化的趋势，汽车软件的需求和复杂度不断增加，传统的分布式电子电气架构已经难以满足未来智能汽车产品的开发需求。汽车软件开发工具也正朝着集中化、通用化、平台化、面向服务化的方向演进。

目前，国外汽车开发软件公司占据主导地位。SAP、Oracle、Microsoft 等软件公司旗下的汽车开发软件占据着汽车市场的主要份额。这些软件产品广泛应用于汽车生产、研发、管理

等领域。而国内汽车开发软件虽处于发展初期，但汽车开发软件市场尚未完全被占领，随着国内汽车制造业的发展，许多本土汽车开发软件公司也在逐渐发展壮大。例如华为、百度、阿里巴巴、腾讯等互联网企业已陆续进入汽车领域，推出了自己的汽车软件开发平台，其开发工具也在不断完善和更新。此外，一些传统的汽车厂商也在积极探索汽车软件开发工具的研发，比如比亚迪，吉利等。虽然国内汽车软件开发工具的发展相对滞后，但随着我国汽车产业的快速发展，国内汽车软件开发工具的研发正在加速推进，且主要体现在以下几个方面：

(1) 云原生技术的应用：随着云原生技术的普及，越来越多的汽车软件开发工具开始采用容器化、微服务架构等技术，以提高软件的灵活性、可扩展性和可靠性；

(2) 自动化测试工具的推广：为了确保汽车软件的高质量和可靠性，越来越多的汽车软件开发工具开始采用自动化测试工具，以减少人工测试的误差和成本；

(3) CI/CD（持续集成、持续交付和持续部署）的完善：随着汽车软件规模的增加，CI/CD已经成为汽车软件开发中不可或缺的技术。GitLab、Jenkins 等工具被广泛应用于汽车软件开发中，以确保软件的稳定和可靠；

(4) 智能化工具的引入：随着人工智能技术的发展，越来越多的汽车软件开发工具开始引入智能化工具，如代码推荐、智能诊断等，以提高开发效率和软件质量。

总的来说，汽车开发工具软件市场仍处于快速增长阶段，未来随着汽车电子化程度的进一步提高，汽车开发工具软件的需求将会继续增加，逐步走向智能化、自动化和高效。

2.4.4.4 标准需求

参与汽车软件开发的人数众多，为了能有统一的指导标准为了完成从系统设计到生成可执行文件，以实现车辆各项功能部署到各个 ECU 当中并正常运行，部分固化的标准需要被抽象封装成工具，保证在整个系统开发过程当中，开发人员有通用可行的技术手段。

汽车开发软件类工具在标准体系要求层面可分为以下几个方面：

(1) 软件质量保证：包括软件开发过程、编码规范、软件测试等方面的要求，以确保开发工具的质量；

(2) 系统安全性：软件开发工具需要考虑网络安全、功能安全、数据安全和用户界面安全等方面的安全，以确保工具软件的权益和客户的安全；

(3) 数据交互标准：指定数据交换格式、数据标识符、数据传输协议等。这有助于确保软件工具链之之间、软硬件之间的数据交互和共享；

(4) 文件规范：制定一套文件规范，包括源代码库的目录结构、文档规范和操作手册等，以便于开发者和客户更好地了解和和使用软件系统；

(5) 软件更新：定义软件开发工具更新和维护的要求，以及软件更新的方式，以满足市场需求和用户期望，并保障软件与用户的安全。

2.4.4.5 标准框架及分类说明

表20 开发软件标准体系

标准项目及分类		标准类型	标准描述	备注
工具软件（500）				
开发软件（540）				
540-1	开发软件用户体验测评准则及评价规程	团标	规定开发软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
540-2	开发软件兼容性通用要求	团标	规定开发软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	

2.4.5 测试验证软件

2.4.5.1 发展历程

现如今汽车测试验证类软件工具已经得到了广泛的应用，并还在不断发展和壮大。汽车测试验证类软件的发展历程可概括总结为三个阶段：

(1) 初期阶段：20 世纪 80 年代，汽车控制系统开始使用微处理器和集成电路来实现高级功能，出现了第一批汽车验证软件工具，主要用于手动测试 ECU；

(2) 中期阶段：20 世纪 90 年代，汽车电子技术迅速发展，出现了越来越多的 ECU，测试需求变得更加复杂，因此汽车验证软件工具开始自动化，以提高测试效率和精度；

(3) 现代阶段：随着汽车电控系统的智能化、网络化和安全性的要求越来越高，汽车验证软件工具也在不断升级和改进。

目前验证工具种类更加齐全、功能越来越集成化、自动化程度越来越高、测试效率越来越高。例如，引入了MBD模型驱动开发、虚拟化仿真、云计算等先进技术，使汽车验证软件工具更加灵活、可靠、高效和易用。

2.4.5.2 特点

测试验证类软件工具是指在汽车软件测试验证过程中使用的各种软件，包括测试诊断类软件、测量标定类软件等。汽车软件测试验证软件需要覆盖软件开发的全生命周期，以验证软件开发软件的各个阶段是否达到质量要求，通常具备以下特点：

(1) 多功能覆盖面广：汽车验证软件工具可以用于多种类型的验证和测试，覆盖软件开发的全生命周期，需要满足单元测试、集成测试、整车测试的各个阶段。从功能性角度，测试验证工具需要满足诊断、通信、性能、仿真、信息安全、标定、自动驾驶场景、HMI、售后服务等；

(2) 自动化程度高：它可以自动执行测试过程，无需手动干预。这可以提高测试效率并减少人为错误，软件代码量和功能的增加以及敏捷迭代的开发方式，没有自动化测试技术，会导致高昂的人力成本和时间成本，现有测试验证软件多实现自动执行测试过程，无需手动干预

从而提高测试效率并减少人为错误；

(3) 高效可靠性强：测试验证类工具通常可在短时间内完成大量测试，并且可以同时测试多个 ECU，同时确保测试结果的准确性和一致性，并通过验证结果来确定 ECU 是否按照规格要求运行；

(4) 交互体验良好：汽车验证软件工具通常具有用户友好的图形用户界面，并附带详细的文档和培训资料，使用户能够轻松地了解和使用该工具，且该工具可实时监视 ECU 的性能和状态，并生成报告和日志，以帮助用户跟踪问题和改进测试流程。

2.4.5.3 发展现状

现如今汽车测试验证类软件工具已经得到了广泛的应用，并还在不断发展和壮大。随着引入 MBD 模型驱动开发、虚拟化仿真、云计算等先进技术，汽车测试验证软件工具变得更加灵活可靠、高效易用。当前的发展状况可总结为以下几点：

(1) 在测试需求方面，随着汽车技术的不断创新，新车型的开发和上市周期缩短，对测试验证的要求也将越来越高，同时随着自动驾驶技术的兴起，对于安全性和可靠性的测试需求也进一步增加，由此汽车制造商和供应商对测试验证软件的需求也会不断增长；

(2) 模拟和仿真技术在汽车测试验证中的应用得到了显著的改进和发展。通过使用更精确的物理模型、更强大的计算能力和智能优化算法，测试验证软件能够更准确地模拟真实的驾驶环境和复杂的交通场景。这使得汽车制造商能够更好地评估汽车系统的性能和安全性能，模拟与仿真的相关技术有待进一步改进提升；

(3) 数据驱动的测试方法在汽车测试验证领域中被广泛应用。通过收集和分析大量的实际驾驶数据，测试软件能够生成更真实和复杂的测试场景，并能够发现潜在的问题和改进方向。数据驱动的测试方法有效地提高了测试的效率和准确性，并且能够更好地满足日益增长的测试需求。

虽然总体上取得一定的进展，但未来还面临一定挑战：首先现有软件的通用性较差，测试实施的工作强度大，重复性工作较多，难以适应产品型号多样性的特点；其次当面对复杂的系统组成，现有软件无法对各部分设备进行独立的测试验证，设备质量认证困难导致重复投资严重；除此之外，现有技术无法完整地搭建出被测设备的外围环境，测试的充分性难以保证，导致软件质量问题频发；最后，测试工装系统的搭建离不开开发人员的支持，测试的可信度受到质疑。汽车测试验证类软件工具已经成为现代汽车制造中不可或缺的重要组成部分，随着汽车电控系统的发展，其未来的发展前景非常广阔。

2.4.5.4 标准需求

汽车软件测试验证工具标准是指在汽车软件开发过程中所使用的软件测试验证工具必须符合一定的标准要求，以确保汽车软件在测试验证中的可靠性。随着汽车电子技术的发展和应用范围的扩大，汽车软件的质量和安全性变得越来越重要。因此，汽车制造商和供应商需要遵守一些严格的标准和规范，以确保其产品的质量和可靠性。软件测试验证工具的标准一般是基于测试的内容，如代码静态测试满足 MISRA C、MISRA C++；诊断服务满足 ISO14229；信息安全满足 ISO21434 等。

汽车测试验证类工具软件在标准体系要求层面可分为以下几个方面：

(1) 统一的开发工具标准：随着汽车电子化和智能化程度的提高，汽车软件测试验证工具的种类和数量越来越多，如何统一标准，使得不同厂商和供应商的软件可以相互兼容、互操作，是当前的一个重要需求；

(2) 高效的工具链集成：汽车软件测试验证需要用到多种工具，如单元测试、集成测试、网络通信、诊断服务、标定测量等。如何将这些工具集成在一个平台上，使得工具之间可以互相协同工作，提高测试效率，也是一个重要的需求；

(3) 安全可靠的工具链：汽车软件测试验证需要保证软件的安全性和可靠性，因此需要工具链支持安全可靠的编码和测试，如静态代码分析、动态测试等；

(4) 支持开放标准和协议：为了实现汽车软件测试验证工具的互联互通和共享，需要测试验证工具支持开放标准和协议，如 AUTOSAR 标准和 CAN 协议等。

2.4.5.5 标准框架及分类说明

表21 测试验证软件标准体系

标准项目及分类		标准类型	标准描述	备注
工具软件（500）				
测试验证软件（550）				
550-1	软件用户体验测评准则及评价规程	团标	规定测试验证软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
550-2	软件兼容性通用要求	团标	规定测试验证软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	
550-3	测试验证软件测试报告评审规范	团标	制定测试验证软件测试报告评审相关标准规范，针对详细的测试报告编写和评审的指导，具体包括测试结果的整理与汇总、缺陷统计分析、测试执行的回顾和总结等，以便有效地向相关利益相关者沟通测试进度和质量信息	

2.4.6 项目管理软件

2.4.6.1 发展历程

项目管理软件的发展历程可追溯至上世纪50年代的计划编制和控制类工具，随着计算机技术发展，项目管理类软件逐渐出现并演变为今天的形式。项目管理软件发展历程中的重要节点如下：

(1) 发源阶段：20世纪50年代，Gantt图表为最早的项目管理工具，该工具以时间轴显示项目的进度和任务依赖关系。之后60年代出现的关键路径法（CPM），程序评审及技术评估技术（PERT）提供了更精确的时间和资源管理，推动了项目管理软件的进一步发展；

(2) 早期阶段：20世纪80年代，个人计算机开始普及，随之早期的项目管理软件开始出现，如Microsoft Project、Primavera等，这类软件提供了图形界面和更多的功能以帮助项目经理进行计划、进度跟踪和资源管理等操作；

(3) 中期阶段：20世纪90年代，互联网技术快速发展，项目管理软件开始转向Web平台，如Lotus Notes、Open Workbench等，这类软件通过客户端/服务器模型的协同工作平台，以浏览器形式访问，实现在线协作和信息共享，极大提高了项目管理的效率；

(4) 近期阶段：现如今互联网技术趋于成熟，新能源汽车时代来临，项目管理软件基于已有成果研究向敏捷项目管理演变，敏捷项目管理软件开始涌现，如JIRA、Trello等，这类软件具备灵活性、可定制性、问题跟踪、任务管理、敏捷项目管理、资源管理、报告和分析等特点。

2.4.6.2 特点

项目管理类软件工具专门为汽车项目所设计开发，具体涉及到项目计划与跟踪、工作分配和资源管理、团队协作和沟通、风险管理等多个方面，可以帮助团队集中精力于项目本身，更高效地管理和协调团队，同时提高项目的质量和准确性，从而实现更好的项目管理和交付。

汽车项目管理软件通常具备以下特点：

(1) 系统可追溯性强：汽车项目管理软件可以帮助项目负责人或项目团队成员创建项目计划，包括定义项目范围、制定时间表、目标和里程碑等。通过跟踪任务进程和计划的完成情况，团队可以更好地管理和协调项目；

(2) 分配管理合理化：汽车项目管理软件可以帮助用户分配任务、分配人员和资源，以确

保每个任务都得到适当的分配。可视化资源负载图可以帮助团队成员更好地管理项目资源和时间，从而保证项目按时交付：

(3) 协作沟通便利：汽车项目管理软件通常提供协作和沟通工具，以便团队成员之间进行实时交流、分享文件、笔记和讨论。这些工具可以提高团队关系、协作效率和工作效率；

(4) 风险管理强化：汽车项目管理软件可以帮助用户识别潜在的风险，记录和分析风险，并采取措施进行管理和减轻风险，从而降低项目失败的可能性；同时项目管理软件还可以帮助用户跟踪项目进展、预算和时间表，从而更好地控制项目成本和风险。

2.4.6.3 发展现状

当今项目管理软件已经变得更加强大和全面，它们通常具备多项目管理、资源管理、风险管理、报表生成、可视化分析等功能，并且更加注重用户体验和移动端的支持。但面临汽车产业智能网联化发展，仍具备如下挑战：

(1) 复杂性管理：项目管理软件需要应对越来越复杂的项目需求和流程。随着项目规模和复杂性的增加，软件需要能够满足更多的定制化需求，同时保持用户友好性和易用性；

(2) 大数据管理：项目管理软件需要处理大量的项目数据，包括任务、资源、进度、成本等。确保数据的准确性、完整性和一致性是一个挑战。此外，软件还需要能够与其他系统和工具进行数据整合，以实现信息的无缝流动和协作；

(3) 安全性管理：随着项目管理软件使用的普及，数据安全和隐私保护变得更加重要。在商业和技术安全的大需求下，软件开发商需要关注数据的安全性和保护措施，以保护用户敏感信息免受未经授权的访问或泄露；

(4) 移动性应用：随着移动技术的进步和远程工作的普及，项目管理软件需要适应用户在移动设备上的使用和远程协作的需求。提供移动应用程序和云端协作功能成为了软件开发商的挑战；

(5) 集成和生态系统：在汽车全生命周期中，工具链庞大，项目管理软件需要能够与其他

系统和工具进行集成，以实现数据共享和工作流程的连接。软件开发商需要建立健康的生态系统，吸引第三方开发者提供丰富的集成和插件。

除以上挑战外，国内项目管理软件发展较晚，且市场应用也以国外工具为主，在国产化需求的大环境下，实现国产项目管理软件的开发与成熟应用已成为汽车软件质量和过程管理的迫切需求。

2.4.6.4 标准需求

目前项目过程管理类标准已经比较成熟，但还没有标准对项目管理工具提出统一的要求和规范，标准的缺失导致市面上的管理类工具功能各异，未针对汽车软件管理过程的特点形成高效的过程管理工具汽车软件项目管理工具标准是指在汽车软件全生命周期过程中所使用的管理工具必须符合一定的标准要求，以确保汽车软件的质量、安全和可靠性。

汽车项目管理类工具软件在标准体系要求层面可分为以下几个方面：

(1) 支持多平台：针对市场管理工具类别的多样性，且与管理工具交互的其他平台的多样性，不同工具间的接口兼容性是影响过程效率的一大因素，因此针对这类工具形成标准化的调用接口，使得不同平台间具备更高的兼容能力；

(2) 支持定制化：项目管理工具往往需跟随固定的标准流程与管理体系以达到特定的项目过程管理功能，不同企业间的过程管理往往具有特殊化要求，因此针对某些企业的特定要求，工具需具备一定的定制化服务功能。

2.4.6.5 标准框架及分类说明

表22 项目管理软件标准体系

标准项目及分类	标准类型	标准描述	备注
工具软件（500）			

项目管理软件（560）				
560-1	项目管理软件用户体验测评准则及评价规程	团标	规定项目管理软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
560-2	项目管理软件兼容性通用要求	团标	规定项目管理软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	

2.4.7 后市场工具软件

2.4.7.1 发展历程

我国汽车后市场工具软件的发展历程总体上大致经历了五个阶段：

(1) 第一阶段：1990-1996年，汽车后市场开始阶段，又称萌芽阶段。这时的服务对象主要是公务车；

(2) 第二阶段：1997-2006年，这段时间里汽车后市场已经高速发展，服务对象以公务车和私家车为主(私家车占比15%)；

(3) 第三阶段：2007-2010年，这段时间是汽车后市场的洗牌阶段，服务对象依然是公务车和私家车，只不过两者的占比已经持平，均占50%；

(4) 第四阶段：2011-2015年，此时的汽车后市场发展趋于平缓，私家车服务占据了主导；

(5) 第五阶段：2016年至今，互联网+阶段，结合电商平台的优势以及客户渠道多的特点，汽车后市场软件工具以“互联网+平台”的形式逐渐涌出，正式进入激烈竞争时期。

后市场工具软件主要包括车辆诊断维修类软件和车辆运营监控服务类软件两大类。其中具体在车辆诊断维修类软件方面，最早的诊断维修是基于定期的维护保养，或者是车主、维修

人员通过车辆上的故障指示灯了解车辆的故障问题。诊断仪的出现，帮助企业提取到存储在车辆内部的参数和故障码，来了解车辆的故障信息，维修服务可以基于诊断结果来进行。现阶段由于网络技术的支持，可通过基于车辆的远程监测来实时的获取车辆数据，从而可根据更加完整的问题信息来展开维修保养服务。在未来，诊断的方式将不再限于通过诊断仪的本地诊断，或者是借助网联技术的云端在线诊断，而是基于大数据与故障预测模型的预测型诊断，基于数据的积累和收集，再借助云存储计算的能力，理论上可以实现在车辆部件和系统可能发生故障之前进行预测。

其次，车辆运营监控服务类软件同时也经历了多个阶段的发展。最初的车辆运营监控技术通过安装摄像头来记录车辆行驶过程中的画面。这种技术比较简单，对于车辆行驶安全和交通管理的作用非常有限。随着科技的不断发展，车辆运营监控技术开始向着系统化方向发展，不仅仅可以记录车辆行驶过程中的画面，还可以对车辆状态进行实时监控，提高车辆行驶的安全性和管理效率。现在的车辆运营监控技术已经发展到了智能化的阶段，不仅可以实时监控车辆行驶过程，还可以通过数据分析和智能算法，提供更加精准的驾驶行为分析和预测，帮助驾驶员更好地掌握车辆行驶情况，提高驾驶安全性和行驶效率。

2.4.7.2 特点

后市场工具深度融合了汽车电子、信息通信、大数据、人工智能等技术与应用，其软件具有以下特点：

(1) 应用多元化：后市场工具使用面向众多不同应用场景及用户群体，如主机厂的信息安全运营、政府的新能源车辆监控、4S 店的故障诊断与维修服务、运输车队的车辆调配、车主的驾驶行为分析等，因此后市场工具软件需支持多样化的应用与交互方式，可以通过手机 APP、web 端、专用设备等为用户提供更好的体验；

(2) 可配置性强：不同车企的不同车型其装备配置及功能匹配千差万别，汽车后市场工具

面向不同车辆需要具备兼容性及易于定制可配置的能力，以满足不同用户的需求。例如：运营监控工具根据用户的需要为不同类型的事件设置不同的响应级别，诊断工具可通过图形化的编辑配置方式为不同车辆设计对应的检测工序流程等；

(3) 高度智能化：后市场应用中会收集到大量数据，可通过对大数据的智能分析技术实现车辆状态监测、故障预警等功能。人工智能算法可以帮助诊断软件使诊断的分析过程变得快捷和方便，简化分析操作过程，提高分析诊断的精度和效率，并为可能发生的故障发出预警。应用人工智能技术，提供更加精准的驾驶行为分析和预测，帮助驾驶员更好地掌握车辆行驶情况，提高驾驶安全性和行驶效率。也可以通过对运营监控数据的分析，为主机厂、交通、运输、市政等部门提供更好的决策支持；

(4) 安全要求高：后市场应用中涉及大量用户数据，包括车辆位置、行驶轨迹、驾驶员行为、车内对话等敏感信息，因此，后市场工具软件要保证在采集、存储、传输数据时，考虑驾驶员的隐私权，合理规定数据的采集范围和使用目的，防止数据被泄露、篡改或非法使用，以保护驾驶员的隐私权。在车辆故障维修过程中，诊断工具要访问车辆控制器，会存在设备恶意访问或篡改车内重要数据及启动远程控制等功能的可能，甚至会造成人身伤害，因此后市场工具软件需要具备信息安全防护能力，避免工具身份被非法盗用。

2.4.7.3 发展现状

对于后市场工具类软件，可从汽车故障诊断工具与车辆运营监控工具两大类展开总结：

随着汽车保有量的增加，汽车后市场的需求不断增加，而汽车的维修及汽配行业又是其中最为重要的一环。其中汽车故障智能诊断工具是汽车维修和汽配行业最为核心的产品，为确定汽车技术状况或查明故障部位、原因进行检测。

随着汽车智能化的进程，基于汽车 ADAS 的传感器和以及软件大量增加，催生了大量新需求。为实现 ADAS 功能需在车身上加装大量的传感器，如摄像头、毫米波雷达、超声波雷达等，随着使用时长和次数的增加，为防止对距离和方位产生错判，驾驶辅助标定工具应运而生。

目前作为销售车辆主流的新能源汽车面临电池性能下降以及诸多潜在的安全风险等问题，消费者对其的电池寿命和安全性问题存有疑虑并提出更高要求，由此派生出市场对于电池性能检测工具的需求，通过检测工具帮助用户掌握电池的衰减情况、健康水平、评估电池寿命，从而在一定程度上缓解里程焦虑，避免车辆在充电、使用或运输过程中发生自燃等问题。

车辆运营监控工具的需求来源则主要有三个方面：一是公共安全与法制的需要，包括所有安全领域、治安、交通、公共卫生、环保、消防等方面；二是经济发展的需求，包括物流行业、运输行业、城市管理；三是个人安全与起居方便性的需要，包括普通民众、家庭、老人等方面。

车辆运营监控的技术水平不断升级，已经可以实现车辆追踪、实时监控、数据分析等多种功能，形成了以定位、监测、调度为主体的服务体系，大大提高了车辆管理效率。近年来，车辆监控行业采用了更高级别的技术，例如“车路协同”技术，可以实现车辆、道路和网络三位一体的协同，并通过道路信息和通讯设施实现交通管理、效率提升和安全保障等多种功能。

此外，人工智能技术的应用不断拓展，无人驾驶技术不断推陈出新，亦将对车辆监控行业产生深刻的影响。尤其是在车辆自动驾驶过程中，将对车辆监控行业提高精度和速度方面带来创新。

2.4.7.4 标准需求

后市场工具软件涉及范围广泛，包含车辆使用过程中的状态监测、故障诊断、健康管理、维护维修、后装产品、租赁、二手车、消费等众多领域。汽车后市场行业整体依托于汽车行业而发展，随着汽车保有量的不断上升、整车技术的不断升级以及车辆由交通工具向移动出行空间的转型，汽车后市场行业的发展动力及发展空间也越来越大。依据当前后市场行业的现状，对后市场工具软件在标准上有以下需求：

(1) 数据和接口格式标准化：后市场工具软件的发展需要开放的生态，当前不同类型的数据掌握在不同行业监管部门及企业中，而综合利用数据能够为用户提供更完善的服务，提供更好的价值体验，数量繁多的软件产品及数据，如果没有统一的数据格式与交互协议、接口不兼容，就会导致数据互联互通困难，形成数据流通壁垒。因此有必要建立统一的接口协议和格式，可有助于打通软件烟囱式信息孤岛；

(2) 系统功能性需求：汽车后市场围绕汽车售后使用过程，涵盖各项服务能力，以满足用户使用、企业监管、社会发展的新需求。UNR155（WP29）要求主机厂对车辆的网络安全状况进行持续性的监控，并在网络安全事件发生后的合理时间内做出应对，在我国，也有涉及相关要求法律法规在制定中，因此运营监控平台需具备相应的功能和技术来满足法规要求；

(3) 安全性和可靠性需求：由于汽车后市场是一个开放的生态，其发展面临着信息安全与数据安全的挑战，如个人隐私保护、防范黑客攻击、零部件仿冒等问题。为了解决这些问题，需要加强技术研发和标准制定，建立完善的监管体系和法律法规，规定汽车后市场工具软件的安全性和可靠性要求，包括数据安全、漏洞管理、身份认证和安全传输等方面的要求；

(4) 规范软件升级服务：随着智能车辆在网联化、智能化及架构技术的发展，汽车后市场工具亟需进行软件迭代升级。然而软件升级在安全管理等方面面临着较大挑战，法规层面对软件升级的安全性和可靠性提出要求，因此有必要定义软件工具开发、更新和维护阶段的要求以及软件更新的方式，以满足市场需求和用户期望，保障软件升级过程用户及产品工具的安全；

(5) 可配置工具标准化：后市场工具的需要针对不同的车型或用户需求进行配置，配置过程较为复杂且使用者存在专用技术壁垒，使用低代码配置工具能够将迭代流程自动化，以提升软件开发效率。为提高不同工具的通用性降低用户使用难度，需要定义图形化建模中最基本的元素基元模型至元模型、模型实例，制定相关标准规范。

2.4.7.5 标准框架及分类说明

表23 后市场工具软件标准体系

标准项目及分类		标准类型	标准描述	备注
工具软件（500）				
后市场工具软件（570）				
570-1	后市场工具软件用户体验测评准则及评价规程	团标	规定后市场工具软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
570-2	后市场工具软件兼容性通用要求	团标	规定后市场工具软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	

2.5 质量与度量

2.5.1 研究背景

质量管理发展过程大致分为四个阶段。第一阶段是质量检验阶段，其特点是剔除不合格的产品。这种方式无法在生产过程中起到预防、控制的作用，且全面检查周期长、成本高，不适用于大规模生产。第二阶段是统计质量控制阶段，将质量管理与数据统计相结合，在大规模生产时，利用统计抽样的方法来检验产品。这种方式的质量控制和管理局限在制造和检验两个环节，忽略了其他环节对质量的影响。第三阶段是全面质量管理阶段，其特点是以顾客满意为目标，将质量控制过程延伸到整个产品的生命周期，强调了质量控制是全体人员的责任。第四阶段是现代质量管理阶段，其特点是充分利用现代化质量管理理念以及方法。

软件质量是指“软件与明确地和隐含地定义的需求相一致的程度”。软件度量是软件质量评价的基础，是对于软件项目、过程以及产品进行数据定义、收集和分析的持续性量化过程。本章节重点研究了汽车软件的质量模型、开发模型和质量管理、开发过程能力度量的发展现状，并分析了智能网联带来的新问题和新的挑战。

2.5.1.1 汽车软件质量模型

国内外主要研究机构对于软件质量的评价方法一般为：首先建立适合待评价软件的质量模型，构造相应的评价指标体系；然后对各软件质量属性进行度量得到相应的度量值，最后对质量属性度量值进行综合，将底层软件指标聚合为高层次因素，逐步得到综合的软件质量评价结果。因此首先对软件质量模型开展研究。

2.5.1.1.1 软件质量模型发展历程

Boehm 质量模型是 1976 年由 Boehm 等提出的分层方案，将软件的质量特性定义成分层模型，如图 19 所示。

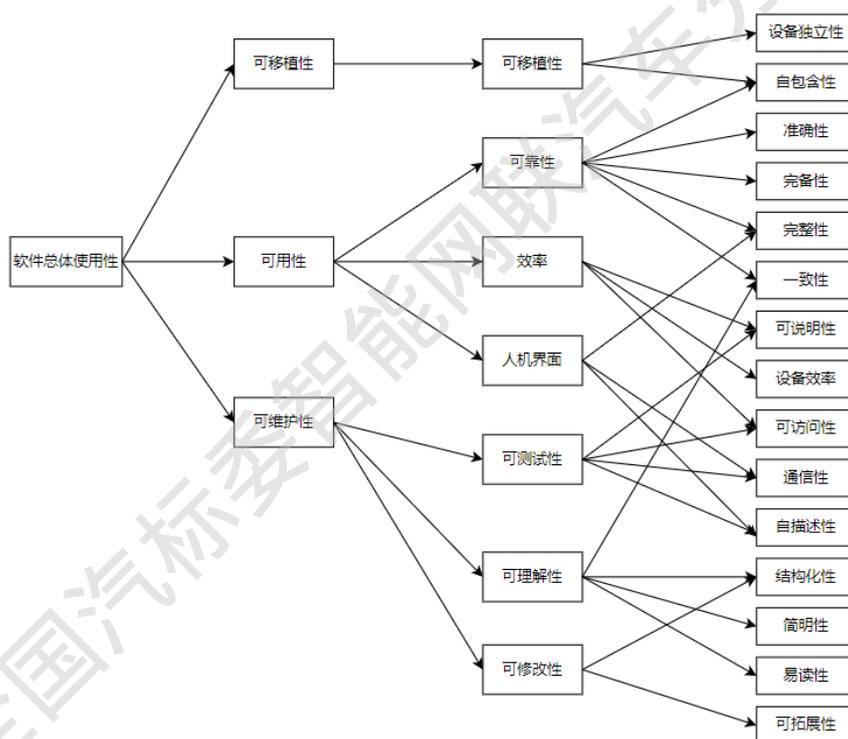


图 19 Boehm 模型

Boehm 模型中的高层属性包括可移植性、可用性和可维护性。中层属性包含 7 个质量要素，分别是可移植性、可靠性、效率、人机界面、可测试性、可理解性和可修改性。原始属性包含设备独立性、自包含性、准确性、完备性、完整性、一致性、可说明性、设备效率、可访问性、通信性、自描述性、结构化性、简明性、易读性和可拓展性。

问性、通信性、自描述性、结构化性、简明性、易读性及可扩展性。

Boehm 模型已经囊括了软件和硬件的属性,但是最终的原始属性和前面介绍的质量要素交叉映射, 这为 Boehm 模型的广泛推广造成了影响。

McCall 模型也称为 GE (General Electric) 模型, 最初起源于美国空军, 主要面向系统开发人员和系统开发过程。1977 年, Jim A. McCall 试图通过一系列软件质量属性指标弥补开发人员与最终用户之间的沟壑, 提出了 McCall 模型, 如图 20 所示。McCall 模型指出, 特性是软件质量的反映, 因此软件属性可用于 (软件质量的) 评价准则, 通过对软件属性定量的度量就可以反映出软件的质量。



图 20 McCall 模型

McCall 模型是一个三层模型, 自顶向下分别是质量因素、质量准则和质量度量。其中的质量因素是面向管理观点的产品质量, 软件的最终用户尽管不了解软件的内部实现细节, 而是非常了解自己的需求, 用户从外部视角定义和描述软件, 形成从外部可观察到的特性, 这就是 McCall 模型中顶层的质量因素的来源。次顶层是质量准则, 开发人员从内部视角构建软件属性, 这些属性是从内部可以观察到的属性, 是决定产品质量的属性; 底层是定量地度量软件属性的质量度量。

FURPS 模型中 FURPS 是功能 (function)、易用性 (usability)、可靠度 (reliability)、性能 (performance) 及可支持性 (supportability) 五个英文词的前缀缩写, 是一种识别软件质量属性的模型。其中功能部分对应功能需求, 另外四项则是软件系统中重要的四项非功

能性需求，有时会特别用 URPS 来表示此四项非功能性需求。FURPS 可分为以下五项：

功能需求：功能集、能力、通用性、保安性。

易用性：人因、美学、一致性、说明文件。

可靠性：故障频率及严重程度、可恢复性、可预见性、准确性、修复前平均时间。

性能：速度、效率、资源消耗、吞吐量、反应时间。

可支持性：易测性、延伸性、适用性、可维护性、兼容性、可配置性、可服务性、可安装性、本地化能力、可携性。

ISO/IEC 9126 质量模型是质量特性及其使用指南纲要，为支撑软件产品的每个相关质量特性而发展出来的。在此标准中，定义了六种质量特性，并且描述了软件产品评估过程的模型。

功能性：指与软件所具有的各项功能及其规定性质有关的一组属性，包括适合性、准确性、互操作性、依从性、安全性。

可靠性：指在规定运行条件下和规定时间周期内，与软件维护其性能级别的能力有关的一组属性。可靠性反映的是软件中存在的需求错误、设计错误和实现错误而造成的失效情况，包括成熟性、容错性、可恢复性。

可用性：指根据规定用户或隐含用户的评估所作出的与使用软件所需要的努力程度有关的一组属性，包括可理解性、易学性、可操作性。

效率：指在规定条件下，与软件性能级别和所用资源总量之间的关系有关的组属性。包括时间特性、资源特性。

可维护性：指与对软件进行修改的难易程度有关的一组属性，包括可分析性、可改变性、稳定性、可测试性。

可移植性：指与一个软件从一个环境转移到另一个环境运行的能力有关的一组属性。包括适应性、可安装性、遵循性、可替换性。

ISO/IEC 9126 模型综合了 Boehm 模型和 McCall 模型的优点与缺点，站在用户、开发者、

管理者的角度，从外部质量、内部质量、使用中质量三个方面完成了质量模型的建设，从外部和内部对质量进行度量。其中，外部度量在测试和使用软件产品的过程中进行，通过观察软件产品的系统行为，完成对其系统行为的测量，得到度量的结果；内部度量在软件设计和编码过程中进行，通过对中间产品的静态分析完成，其目的是确保获得所需的外部质量和使用质量。

ISO/IEC 9126 质量模型包含了 6 个质量特性和 27 个质量子特性，特性和子特性意义映射，不存在交叉问题，但是还不完善，因此发展到后来的 ISO 25010 质量模型。

ISO/IEC 9126 质量模型包含了 6 个质量特性和 27 个质量子特性，特性和子特性意义映射，不存在交叉问题，但是还不完善，因此发展到后来的 ISO 25010 质量模型。

相较于 ISO/IEC 9126，ISO 25010 质量模型将原来的 6 个属性扩展到 8 个属性，新增的内容是安全性和兼容性，另外还对功能性、易用性和可维护性做了修改，具体内容如图 21 所示。

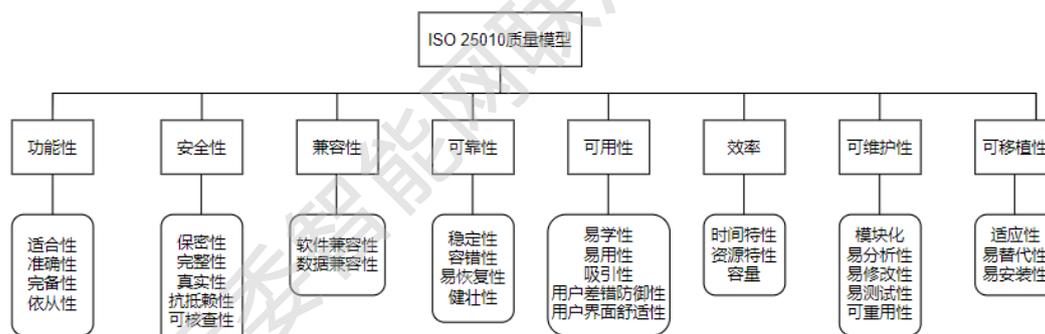


图 21 ISO 25010 质量模型

国家标准中，软件产品质量模型是 GB/T 25000.10—2016，该国标对应的国际标准为 ISO/IEC 25010—2011。软件产品质量模型将一个软件产品需要满足的质量要求总结为 8 个属性（功能性、兼容性、安全性、可靠性、易用性、效率、可维护性和可移植性），每个属性又可细分出了很多子属性。

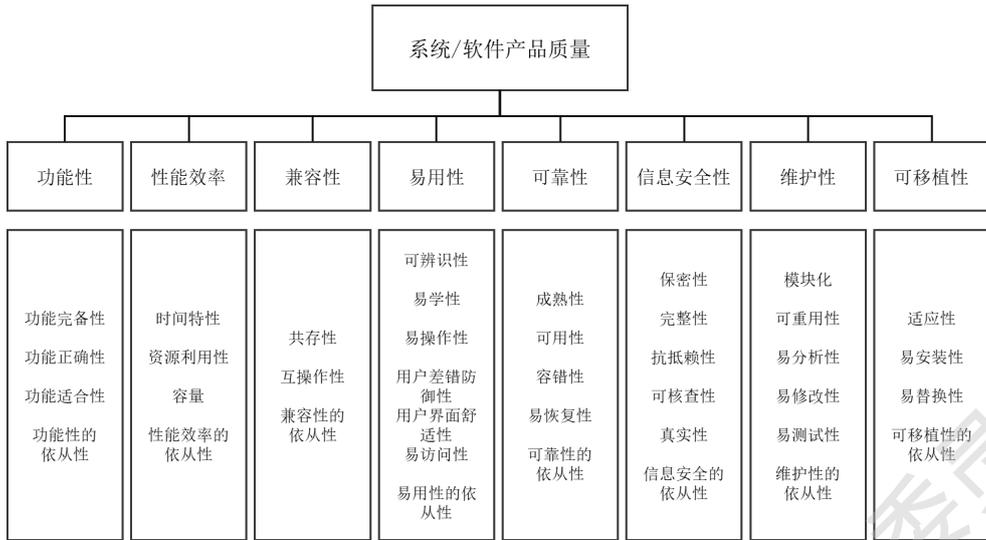


图 22 GB/T 25000 质量模型

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》于 2016 年发布。标准在 GB/T 25000. 51-2010 的基础上增加“信息安全”及“兼容性”两部分内容,要求第三方测评机构依照此标准开展测评活动。明确了:

质量要求:产品说明质量要求、用户文档集质量要求、软件质量要求(功能性、性能效率、兼容性、易用性、可靠性、信息安全性、维护性、可移植性);测试文档集要求:符合性评价细则。

总结来看,Boehm 软件质量模型试图通过一系列的属性的指标来量化软件质量,采用层级的质量模型结构,包括高层属性、中层属性和原始属性。McCall 模型通过层级的要素、标准和指标来详述产品修改、产品转移和产品运行这 3 个视角定义。ISO/IEC 9126 模型则是建立在 McCall 和 Boehm 模型之上的,同时加入了功能性要求,还包括识别软件产品的内部和外部质量属性,但后续 ISO/IEC 9126-1 被 ISO/IEC 25010 模型代替并废止。总结现有的质量模型可知,此类模型适用面较广,但应用于针对性较强的汽车软件领域时可能不够系统和全面,仍需结合汽车软件的特点制定专属的质量模型。

2.5.1.1.2 智能网联汽车安全体系对软件的新要求

L3 级别自动驾驶需要在紧急不可控情况下留给驾驶员 6-8s 以上的时间接管车辆，因此需要整体系统都具备冗余。主体软件感知、决策、规划、控制等都要有冗余备份设计，保证从故障发生到驾驶员接管这段“真空期”内功能不退出。带有冗余设计的软件需要包含大量的失效分析和故障处理设计，能够处理各种异常情况，并进行合理的保护措施。这部分设计的测试需要充分的模拟仿真、封闭场地和实际道路测试。

功能安全指不存在由电子电气系统的功能异常表现引起的危害而导致不合理的风险，其中功能异常主要表现为系统性失效及随机硬件失效，在汽车软件开发过程中实施功能安全可有效避免系统性失效的发生。功能安全标准在软件开发阶段，分别从软件开发环境、软件安全要求定义、软件架构设计、软件单元设计及实现、软件单元验证、软件集成和验证、嵌入式软件测试等方面针对不同安全等级的软件提出了不同的方法来控制软件的系统性失效。特别是为了证明软件不包含与功能安全相关的非预期功能和特性，需在各个测试阶段进行充分的验证，为了评估测试用例的完整性，在单元及集成验证阶段有结构覆盖率指标要求。同时为了降低软件工具带来的系统失效，对软件开发及测试过程中所用的工具也提出了置信度的要求。

预期功能安全旨在避免由于预期功能或其实现的功能不足导致危害所产生的不合理风险，其基本概念由 ISO 21448 给出定义。ISO 21448 作为 ISO 26262 延伸，处理在不发生硬件随机失效和系统故障情况下的功能不足问题。SOTIF 研究涉及系统功能设计改进、分析评估、验证确认和认证等多方面问题，且随着技术发展和新技术的引入不断提出新的需求。明确的问题定义和风险源分析是保障 SOTIF 的前提。从系统自身角度分析，SOTIF 问题主要源于两方面：

(1) 在车辆层对预期功能的规范不足，场景开放性、系统复杂性和专家经验的不完备性等限制均可能导致车辆行为的设计规范过程出现问题，进而难以实现理想的安全目标；(2) 预期功能实现的不足，即使对车辆层预期功能的规范足够完备，由于系统组件的性能局限和规范不足，感知、决策和控制等功能的实现可能不符合预期。如传感器、执行器存在感知、执行能力上限或易受外界环境因素干扰等性能局限；感知、决策算法可能具有鲁棒性、泛化性、可解释性、逻辑完备性、规则覆盖度等方面的问题。此外，SOTIF 危害的产生和演化依赖于特定场

景。需要通过分析方法（FTA、FMEA、HAZOP、STPA 等方法）、仿真和实际道路测试进行触发，以识别出未知的不安全情况。首先，上述规范不足或性能局限由场景中特定条件触发而导致危害行为；其次，上述危害行为最终演化为伤害是建立在当前场景包含相关风险源以及场景可控性低的情况下。因此，在进行 SOTIF 保障过程中，需要综合系统自身局限和运行场景风险以建立安全保障体系。

网络安全与数据安全对汽车软件提出的要求，包括：应具备系统镜像的防回退校验功能；重要业务应用代码应进行混淆、加花指令或加密，如使用 OLLVM 进行混淆加密；ECU 应具备日志功能，对安全活动的时间和结果进行记录；软件要支持 TLS1.2 的加密通讯；车辆控制器应对其获取，保存的有机密性要求数据（用户账号、密码，电话簿，通话记录，车外视频数据），在控制器内进行加密存储；车辆内部网络通讯应根据功能相关性，对通讯进行路由，并对无关的数据通讯进行隔离；在监控对象和安全日志信息保存完整的前提下，在攻击事件次数超出阈值的条件下，宜通过故障灯或者屏幕，告知用户需要返回大通授权的维修店，以上传安全威胁有关的日志信息；总线交互的 CAN 报文通讯应通过 SecOC 的方案进行保护；在从生产设备将密钥或证书注入控制器的过程中，生产系统应保护其内容不被泄露或篡改等。

汽车软件升级会影响智能网联汽车的各个环节参数，包括底盘域、车身域、自动驾驶域等等，与汽车安全息息相关。智能网联汽车生产供应链和制造流程复杂，在软件升级层面对安全性和合理性提出更高的要求，具体包括软件升级过程中的安全保障、升级包防篡改、功能及代码验证和应急管理安全要求，并要求提供升级流程合规性标准文件、软件识别码唯一可识别记录、生产一致性系统配置文件和兼容性确认文件等信息记录。智能网联汽车 OTA 软件升级重点围绕软件包真实完整性、软件识别码的更新及防篡改能力、升级结果确认告知、影响车辆安全及驾驶安全等车辆要求对象，针对升级包真实性完整性测试、软件识别码更新及读取测试、用户告知与确认测试、先决条件测试和升级过程安全影响相关测试等具体项目开展测试工作，从而提升 OTA 升级技术的规范性，保障车辆的全生命周期内升级的稳定性和可靠性。

传统的开发流程中，只需要软件按照功能开发和问题修复迭代即可；汽车软件开发过程

中需要考虑功能安全、预期功能安全、网络安全和数据安全，这些安全要求也形成了一定的体系标准，在软件开发过程中需要基于这些标准进行质量管理。在功能安全方面，美国早于 1996 便在 ANSI/ISA-S84.01 标准中将功能安全的概念纳入，之后国际电工委发布 IEC61508 标准，并相继发布针对众多工业领域的功能安全标准，形成功能安全标准体系，后续欧美国逐渐形成了完整的功能安全产品、系统和服务的产业链，且占据世界绝大部分市场，我国由于引入功能安全技术较晚，仍需要进一步将安全测试、评估、认证等微服务产品化，以形成高附加值的产业链；在信息安全方面，近年来国内外网络信息安全产品市场快速发展，参与厂商众多，由于行业细分领域较多，大部分企业专注于某一细分领域，导致综合型企业及较少，现亟需开展自主研发综合型的信息安全系统软件相关工作方可适应我国的实际情况。

安全体系的建议需要考虑以下因素：

安全威胁与漏洞：随着软件复杂性的增加和网络连接的扩展，汽车软件面临更多的安全威胁和漏洞。黑客攻击、恶意软件和数据泄露等问题对汽车的安全性和隐私构成威胁。软件与安全体系的融合需要识别和解决这些安全问题。

安全性评估和验证：确保汽车软件的安全性需要进行全面的评估和验证。这包括对软件设计和实现的审查、漏洞扫描、安全性测试等。软件与安全体系的融合需要建立有效的评估和验证方法，以确保软件的安全性。

安全文化和组织能力：软件与安全体系的融合需要建立强大的安全文化和组织能力。这包括培养安全意识、建立安全开发流程、加强安全培训等。汽车制造商和供应商需要在组织层面上重视软件与安全体系的融合。

更新和演进：汽车软件是一个动态的领域，需要不断进行更新和演进。软件与安全体系的融合需要考虑软件的持续更新和演进过程中的安全性。这包括远程更新、软件漏洞修复等方面的问题。

另一方面，安全体系的加入会造成两个冲击：

软件中的功能必须按照既定的项目节点要求发布，否则渗透测试的软件与量产软件是不

匹配的，会带来额外的测试费用。

软件的变更必须系统性地评估对于安全的影响，需要从 HARA、TARA 分析开始，变更的周期和工作量会增加。

2.5.1.1.3 汽车软件质量模型分析

汽车软件的质量模型在各方面均存在特性要求：质量模型应强调对实时性的要求，包括响应时间、延迟和任务调度时间等指标；应考虑对软件的可靠性和稳定性进行评估，包括对错误处理、容错能力和系统恢复能力的考虑；应该考虑对软件更新和维护的评估，包括对版本管理、更新流程和兼容性测试的考虑。安全性方面，由于汽车固有的安全属性，与汽车软件相关的有功能安全、预期功能安全、信息安全，在现有的质量模型中，信息安全体现相对比全面，功能安全及预期功能安全比较薄弱。此外，质量模型还需要考虑对软件复杂性的评估，包括对系统结构、模块化设计、接口规范和代码质量等方面的考虑。结合已有的软件质量模型及汽车软件特点，提出汽车软件质量模型如图 23 所示：

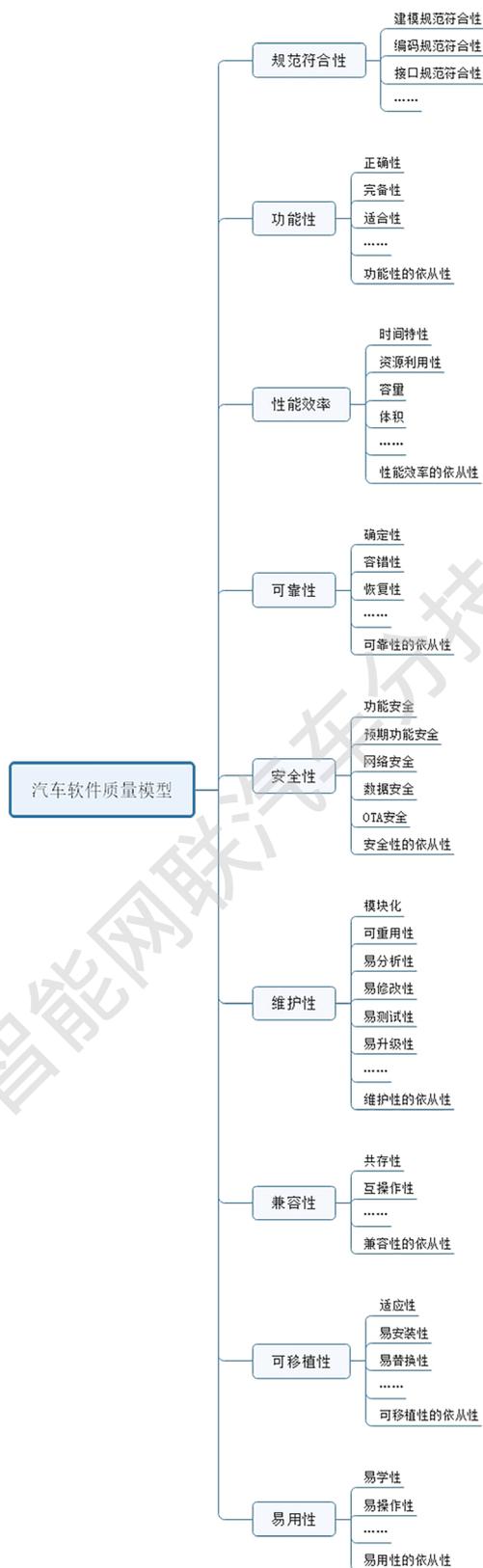


图 23 汽车软件质量模型

值得注意的是，汽车软件种类较多，功能用途差别较大，软件自身架构和复杂程度各不相

同。例如，底盘控制器和动力控制器的软件对安全要求很高，对更新迭代速度要求较低，以安全、稳定为主。而座舱应用软件允许有一定非功能缺陷在软件快速迭代中解决。不同类型的汽车软件在应用质量模型进行度量时侧重点并不一致，按照汽车软件的分类来看：

车载操作系统：

1) 规范性 车载操作系统软件开发要遵循相关的标准、约定或法规以及类似规定的程度，在质量度量时要考虑建模规范、编码规范、接口规范等的符合性要求。

2) 功能性 车载操作系统软件质量要正确、完备、适应，符合软件的开发需求。

3) 可靠性 车载操作系统软件支持多样化功能，汽车座舱除了仪表显示、空调/车窗控制等传统功能外，已经开始集成支付、娱乐、导航、信息服务等多样化的功能，这些车机软件需要可靠稳定，还需要具备一定的容错性和可恢复性。

4) 安全性 车载操作系统需要满足功能安全、信息安全等要求，车载操作系统一般和车控软件进行交互，因此需要考虑功能安全要求，此外智能座舱的使用不仅关乎用户的个人隐私安全与财产安全，同时也通过车内网络与底盘控制、自动驾驶等车控系统相连通。因此，车载操作系统不能简单地将手机操作系统复制到车端，而应通过深度定制达到车辆信息安全的标准。

5) 易用性 车载操作系统都需要进行人机交互，因此操作系统需要易学习、易操作。主要考虑界面的布局设计、操作逻辑与流程、交互信息的显示、帮助和索引等。

车控操作系统：

除功能外重点聚焦于性能、可靠性、安全性等方面。

1) 性能 包括时间特性、资源利用性和容量。

时间特性包含了响应时间、处理时间、时延、吞吐量、可用时间等方面的度量和评估，确保软件运行期间，在不同的时间点或时间段内，系统能够快速、准确和及时地响应和处理任务；

资源利用性主要关注系统资源(CPU、内存、磁盘、网络带宽等)的利用效率，针对这些方

面的质量度量评估可优化资源占用，在实现系统稳定性的同时，最大化利用系统资源。从而提高应用程序的质量和性能，支持系统高效正常运行；

容量主要关注系统在运行期间能够处理的事务量、并发连接数以及资源容量等。评估这些方面的容量能力可以了解系统的扩展潜力、性能瓶颈以及资源利用情况，从而对系统进行合理规划和优化，以满足不同负载场景的需求，确保系统能够在高负荷环境下正常运行，保证系统的可靠性和性能。

2) 可靠性 主要从系统的成熟性、可用性、容错能力和易恢复性等方面进行质量度量，确保系统能够在复杂环境下保持长时间稳定运行。

3) 安全性 重点针对系统的访问控制、加密和认证机制、安全策略、安全漏洞、安全启动、抗抵赖性、可核查性等方面进行质量度量。确保敏感信息的保密性、完整性和可用性，防止任何未经授权的使用，具备抵御攻击的能力，以使系统安全稳定运行。

中间件：

中间件在应用质量模型进行度量时侧重点通常是可靠性和安全性。此外，中间件的性能和可维护性也是应用质量模型中需要考虑的因素。

智驾域应用软件：

智驾域应用软件必须具备高度的安全性、可靠性和性能，以确保车辆在各种交通和环境条件下能够安全运行。

1) 安全性 智能驾驶软件必须具备极高的安全性，以保护乘客、其他道路用户以及周围环境的安全。这包括防止碰撞、遵守交通规则等。在遇到异常情况时，智能驾驶系统需要能够进行有效的错误处理，并采取适当的措施以保证安全。

2) 可靠性 由于智能驾驶系统可能处于各种复杂环境下，它们必须能够在各种条件下可靠地运行。

3) 性能效率 智能驾驶软件需要在实时性和精确性方面表现出色，以确保及时做出正确的决策。

4) 可维护性 智能驾驶软件需要容易维护和更新, 以保证其持续符合最新的安全标准和技术要求。

5) 功能性的依从性 智能驾驶软件必须符合相应的法规要求。

座舱域应用软件:

座舱域应用软件主要以人机交互的方式为用户提供数字化服务, 使用的场景大多涉及用户隐私, 如人像、地址、生日等, 且为了优化用户体验需进行软件更新与维护, 因此在应用质量模型度量时的重点为易用性、性能效率、安全性和维护性。

手机及互联应用软件:

1) 兼容性 软件与不同车型的兼容性。

2) 可靠性 有线连接稳定性、无线连接稳定性、手机音频在车机端输出的稳定性。

3) 安全性 手机音频, 车机音频混音策略对驾驶安全的影响、视频对安全驾驶的影响; 不同场景下音量, 明暗度对驾驶员的影响。

4) 易用性 车机界面美观度。

5) 可移植性 手机 APP 车机适配难易度。

整车应用软件:

整车应用软件在应用质量模型进行度量时主要侧重安全性、可靠性、可扩展性等。

1) 可靠性 整车级应用软件需要具备高度的可靠性, 以确保车辆在各种条件下的正常运行。

2) 可维护性 整车级应用软件需要具备良好的可扩展性, 以便将来可以添加新的功能和更新软件。

高效的远程诊断能力: OTA 诊断软件可以实现远程获取车辆的故障信息, 并且能够通过网络将信息传输至汽车制造商或者服务提供商的服务器上, 便于后续对车辆的故障进行排查。

灵活的软件升级能力: OTA 诊断软件可以实现远程升级车辆的软件系统, 以使车辆具有更好的性能和更多的功能。

3) 安全性 安全的数据传输能力, OTA 诊断软件在数据传输过程中采用了安全加密技术, 以保证数据的安全性和防止数据泄露。

云端应用软件:

云端应用软件在应用质量模型进行度量时主要侧重云端软件的可靠性、安全性、兼容性、实时性等, 保障云端软件的高性能、高安全、高兼容、高时效。

数据服务软件:

1) 功能 数据准确性和完整性, 数据服务软件必须确保所提供的数据准确无误, 并保持完整性, 以便在车辆内部系统和外部系统之间传递信息。

2) 性能效率 数据服务软件应强调实时性要求, 包括数据传输的响应时间和延迟。数据必须在实时性要求下被采集、处理和传输。

3) 可靠性 数据服务软件应具备高可靠性, 确保数据采集和传输不容易受到干扰或丢失。这包括错误处理和容错能力, 以应对可能的故障情况。

4) 安全性 数据服务软件应考虑信息安全, 包括数据的机密性、完整性和可用性。它应该采取适当的安全措施, 以保护数据免受恶意攻击和未经授权的访问。

5) 兼容性 数据服务软件需要具备可兼容、可扩展性, 以应对未来的需求变化和新增数据源的接入。数据服务软件应该考虑与不同车辆系统和数据源的兼容性, 以确保其可以在多样化的环境中无缝运行, 且容易进行升级和扩展

数据服务软件的具体需求可能因应用场景和车辆类型而有所不同。例如, 对于自动驾驶汽车, 数据服务软件需要满足更高的实时性和安全性要求, 而对于娱乐系统, 可能更关注性能和用户体验。因此, 在度量数据服务软件的质量时, 需根据具体用途和要求进行合适的权衡和重点设置。

工具软件:

结合工具类软件的系统特点及应用需求可发现, 此类软件首先须全部具备一款工具的基础功能, 涵盖各项服务能力, 以满足用户使用、企业开发、行业监管的需求; 其次在性能及效

率要求方面，应重点关注计算速度、内存占用和响应时间等性能指标；在安全性及可靠性层面，应注重保障数据安全、信息安全、漏洞管理、身份认证和安全传输等；同时在系统兼容性层面，应统一标准化数据和接口格式，实现数据互联互通，打破流通壁垒，构建开放软件生态；最后应着重关注系统用户的使用体验，具体包括界面设计、交互方式、可定制性、可访问性等方面的要求，这均有助于提高用户友好的使用体验和操作便捷性。因此，工具类软件在度量质量模型时应兼顾系统的功能性、性能效率、安全性、可靠性、兼容性和易用性等多重维度的综合考量。

2.5.1.1.4 汽车软件质量度量

汽车软件测试过程中会产生大量过程数据及结果数据，如何利用这些质量数据反馈指导汽车软件的设计和研发就显得尤为重要。汽车软件检测过程中会产生如响应时间、吞吐量、千行代码缺陷率等大量检测结果数据，同时会产生用例、缺陷、脚本等检测过程数据，这些数据统称为汽车软件质量数据。参考《GB/T25000.10-2016 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》等国内外标准，建立多层次、多维度的汽车软件质量综合分析模型，包括目标层（即汽车软件质量）、指标层（规范符合性、功能性、性能效率、安全性、易用性、可靠性、维护性、可移植性等质量指标及子指标）、度量元层（每个质量指标下的度量元集合）。借助专家评分法、模糊数学等理论，开展基于质量数据分析模型的汽车软件质量度量元评价方法研究。在度量元评价基础上，借助层次分析法、模糊数学等理论，依据汽车软件质量综合分析模型，自下而上对每个软件质量指标进行综合评价，实现各类指标的对比分析，明确薄弱环节以及变化趋势。最后，实现汽车软件质量度量。

2.5.1.2 汽车软件过程能力度量

软件生命周期是指软件从立项开始，经过开发、使用和不断修改，直到最后废弃的整个过程。软件开发模型是指软件开发全部过程、活动和任务的结构框架，通过该模型能清晰、直观

地表达软件开发全过程，明确地规定要完成的主要活动和任务，它奠定了软件项目工作的基础。汽车软件的质量与开发模型息息相关，因此对汽车软件开发模型展开研究。

2.5.1.2.1 软件开发模型

1970年温斯顿·罗伊斯（Winston Royce）提出了著名的“瀑布模型”，直到80年代早期，它一直是被广泛采用的软件开发模型。瀑布模型核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作，即采用结构化的分析与设计方法将逻辑实现与物理实现分开。将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并且规定了它们自上而下、相互衔接的固定次序。

瀑布模型的特点是：

阶段间具有顺序性和依赖性，前一阶段结束后才能开始后一阶段的工作，前一阶段的输出是后一阶段的输入；推迟实现观点，尽可能推迟程序的物理实现；强调质量保证观点，每个阶段必须完成规定的文档，每个阶段结束前完成文档以便及早改正错误。

优点：

- 1) 原理简单，容易掌握。
- 2) 各阶段间都有验证和确认环节，以便进行质量管理。
- 3) 主要用于支持结构化方法。

缺点：

- 1) 缺乏灵活性，不能适应用户的需求变化。
- 2) 缺乏演化性，返回上一级的开发需要付出十分高昂的代价。
- 3) 是线性的软件开发模型，回溯性差。

使用场合：

- 1) 适合于软件需求比较明确或很少变化，且开发人员可以一次性获取到全部需求的场合
- 2) 适合开发技术比较成熟，工程管理比较严格的场合

3) 一般用于低风险的项目, 适合开发人员具有丰富的经验。

总的来讲, V 模型是对瀑布模型的细化和完善。相对瀑布模型, V 模型的优势在于:

解决瀑布模型中严格分离很难实现的困境。

——软件回溯较为方便快捷。

——测试提前, 及早发现问题, 解决问题。

——问题追溯性更强。

——提高了开发效率/降低开发成本。

不过 V 模型和瀑布模型一样, 过程中产生大量文档, 项目反应速度也越来越不能满足当前汽车日新月异的需求和快速的更新换代的节奏。

2001 年, 由 17 位软件开发专家正式提出了 Agile(敏捷开发)概念, 并共同签署了《敏捷宣言》。敏捷软件开发是基于敏捷宣言定义的价值观念(《敏捷宣言》和《敏捷开发的十二条原则》)的一系列方法和实践的总称。在符合敏捷价值和原则的基础上, 能让开发团队拥有快速应对需求变化的能力的方法和实践均可称为敏捷开发。常用的成熟敏捷开发方法主要有 Scrum(管理实践)和 XP(极限编程, 技术实践为主), 近年来基于 Scrum 演进大规模敏捷框架 Scrum of Scrums、SAFe 和 LESS 等方法。

Scrum 是敏捷方法论中定义简洁、重要且应用最广泛的框架之一, 使软件开发中的迭代过程实现了标准化的高效管理。

1) Scrum 开发过程由若干个短的迭代周期组成, 一个短的迭代周期称为一个 Sprint, 每个 Sprint 的建议长度是 2 到 4 周。

2) 在 Scrum 中, 使用产品 Backlog 来管理产品的需求, 产品 backlog 是一个按照商业价值排序的需求列表。

3) Scrum 团队先开发对客户具有较高价值的需求。

4) 从产品 Backlog 中挑选最高优先级的需求, 在 Sprint 计划会议上经过讨论、分析和估算得到相应的任务列表, 我们称它为 Sprint backlog。

5) 每个迭代结束时, Scrum 团队将递交潜在可交付的产品增量。

6) Scrum 聚焦管理实践, 适用于任何复杂或创新性的项目, 对汽车软件的开发有良好的适用性。

总体来看, 瀑布式软件开发的主要问题在于缺乏灵活性, 无法适应日益复杂和快速变化的市场需求和技术环境, 过度注重文档和计划, 导致难以控制成本和时间。而敏捷软件开发则是为了解决瀑布式软件开发面临的这些问题, 注重以客户为中心, 快速、高效、高质量地交付软件产品。

优点:

1) 敏捷软件开发追求的是对客户需求的快速响应、迭代开发、重视反馈和持续集成, 注重开发团队自组织、高效协作和自治。

2) 敏捷方法适用于需求不确定或者项目需求不停变化的软件开发项目。

3) 敏捷方法相对于瀑布开发可以更加迅速和持续地交付价值, 降低了开发成本和时间成本, 缩短了产品推向市场的周期。

4) 敏捷开发方法基于快速反馈, 可以应用丰富的优秀技术实践, 开发效率远远高于瀑布式方法, 能够减少冗余设计, 快速完成软件迭代, 提高用户对产品的满意度。

5) 敏捷开发方法的各项技术和管理实践在自组织团队管理下要求严格的纪律性, 能够在早期开始发现更多的软件缺陷, 提升软件质量。

不足:

1) 自组织和自治的团队模式更加依赖于个人感性和经验, 对团队管理能力要求比较高。

2) 敏捷开发方法需要团队掌握多种先进的软件设计和开发、测试技术及实践, 以满足快速迭代的需要, 需要一定的学习成本。

3) 汽车软件尤其是嵌入式软件部分, 对硬件的依赖较多, 整车研发通常遵从 V 模型, 对敏捷开发方法的应用在范围和具体实践方面要考虑适用性。

2008 年, Patrick Debois 提出 DevOps 概念, 并于 2009 年发起 DevOps 运动。DevOps

在敏捷软件开发的基础上进行了拓展，包含 development 和 operations，是一组开发和运营维护最佳实践的总称。

优点：

1) 打破研发和运维团队的分隔，强调业务人员及研发人员在软件研发和服务生命周期中的协作和沟通。

2) 强调整个组织的合作，基于研发、交付过程中基础设施的自动化，实现持续集成、自动化测试、持续部署、持续交付、高效运维和持续反馈，增强研发的灵活性。

3) DevOps 开发方法适用于软件的迭代和交付速度较快或者迭代周期较长，需要运维操作比较复杂的应用场景。

不足：

1) DevOps 开发方法强调研发和运维的自动化使得软件开发流程更加高效和有序，减少手工操作，需要投入更大的人力成本和物质资源成本。

2) DevOps 可实现的程度依赖于软件应用的类型，需要依据具体情况分析部署方案。

2.5.1.2.2 汽车软件开发模型

汽车软件更侧重于安全和可靠性，如果汽车软件发生问题，可能影响汽车的正常使用，甚至造成生命财产的损失，而且车载嵌入式软件的后期维护也较为复杂。汽车行业的公司其技术部门组织架构/研发体系几乎都是参照 V 模型设置。而且很多的汽车行业标准和规范的基石都是 V 模型。比如，汽车行业的 ASPICE，ISO26262（道路汽车功能安全规范），都是参照 V 模型。V 模型过程中产生大量文档，项目反应速度也越来越不能满足当前汽车日新月异的需求和快速的更新换代的节奏。

随着智能网联化的发展，软件在研发工作中的比重越来越大，差异化的市场和用户需求变化越来越多，对汽车不同功能之间的特征交互需要深入理解，汽车软件从与硬件深度耦合走向对硬件依赖越来越低，开发方式也正由模型驱动转向数据驱动转变。快速变化和更短的

上市时间要求，对当前的软件开发模型提出了挑战。V 模型软件开发已经难以适应市场效率和质量要求，因此当前的开发模型也需要进行更新，比如对网络安全，机器学习、OTA 等相关要求的兼容。

尽管 V 模型优势不再那么明显，但其在汽车行业依旧会存在。特别是其良好的可追溯性和软件质量的把控，是不能被敏捷开发等新的开发模式所完全取代的。目前部分企业正在探索在 V 模型的整体框架下，基于软硬件研发分离的思想，实现汽车软件的“敏捷+DevOps”开发模式。另一方面，也有软件供应商选择在原有开发模型的基础上进行部分环节和流程的删减，更多以迭代开发的方式进行汽车软件研发工作，此类开发模式虽较好地适应了国内汽车软件开发环境，一定程度上缩短了产品研发周期，但在产品的质量保证方面以及问题可追溯性方面存在较大缺陷和隐患。

汽车软件开发模型应保证以下几点：

1) 汽车软件对其系统的易用性、可靠性和可追溯性提出更高要求，汽车软件多采用基于模型设计并自动代码生成的方式以保证软件的规范性和可靠性；多采用基于 V 模型的开发流程，过程中使需求分析、设计开发与测试验证相对称，以此保证软件开发的可追溯性；

2) 汽车软件通常需要与硬件设备紧密配合才能实现功能，所以需支持硬件与软件的协同开发，包括硬件与软件的代码交互以及硬件模拟器等。

2.5.1.2.3 软件过程能力度量

软件过程能力度量是通过系统地分析和研究来评定过程能力与指定需求的一致性。软件开发的风险之所以大，是由于软件过程能力低下，其中最关键的问题在于软件开发组织不能很好地管理其软件过程。

CMMI (Capability Maturity Model Integration, 能力成熟度模型集成) 是由美国卡内基-梅隆大学软件工程研究所推出的评估软件能力与成熟度的标准。它旨在帮助组织根据业务目标改进开发过程，提高开发人员的技能水平和组织的能力，以提高产品质量。

CMMI 针对不同的过程领域提供了完整的过程改进框架，包括 CMMI-DEV (面向开发)、CMMI-SVC (面向服务) 等，用于帮助组织实施并维护最佳实践。CMMI DEV V2.0 模型共有 6 个成熟度等级，包括：不完整级(0 级)、初始级(1 级)、已管理级(2 级)、已定义级(3 级)、量化管理级(4 级)、持续优化级(5 级)，每个级别代表了一定的过程成熟度。CMMI V2.0 共有 20 个过程域 (比如 Planning 域、CAR 域、CM 域、PQA 域、RD&M 域等)，用于描述实现已定义的目的和价值所需的关键活动；如果要达到某一成熟度等级，则需要满足相应的过程域要求。

SPICE 是 Software Process Improvement and Capability dEtermination 的缩写，是 ISO、IEC 共同提出的标准 ISO/IEC15504。根据此标准，行业派生出了各种更具体的规范，包括医药设备领域制订的 Meti SPICE、航天领域制订的 SPICE for SPACE。

ASPICE 全称为 “Automotive Software Process Improvement and Capacity dEtermination” (汽车软件过程改进及能力评定)，是汽车行业用于评价软件开发过程能力的模型框架。最初由欧洲多家主要汽车制造商共同制定，目的是为了指导汽车零部件研发厂商的软件开发流程，从而改善汽车软件的质量。

ASPICE V3.1 版本包含了供应商过程组，系统过程组，软件过程组，支持过程组，管理过程组，过程改进过程组，重用过程组，共 32 个过程域，其中包含了 VDA Scope 的 16 个过程域，通常会根据 VDA Scope 进行能力评估。ASPICE 共有 6 个能力等级，包括：不完整级(0 级)、已执行级(1 级)、已管理级(2 级)、已建立级(3 级)、可预测级(4 级)、创新级(5 级)，每个级别代表了一定的过程能力。

2.5.1.2.4 汽车软件开发过程能力度量

目前汽车软件开发过程能力度量采用较多的是 CMMI 和 A-SPICE，CMMI 适用于所有软件研发团队，A-SPICE 仅仅用于汽车行业的软件研发团队，CMMI 遵从并不意味着一个组织或项目自动遵守 ASPICE。尽管这两种标准在核心概念上看起来是一样的，但它们使用不同的过程评估模型，而且在流程领域的实现中也存在差距。

当前国内外主机厂主要通过 ASPICE 模型框架进行汽车软件开发过程能力的度量，也有部分主机厂支持通过 CMMI 的模型框架进行度量。国外多家主机厂要求其软件供应商达到 ASPICE CL3 级别的能力。国内部分主机厂也要求其软件供应商达到 ASPICE CL2 级别以上的能力。

国内汽车行业的大环境有异于国外，产品更新换代快，软件代码迭代速度快且要求较高。出于对时间周期上的把控，很多企业难以完全适应传统的 ASPICE 体系，故国内大多数软件开发供应商会按照企业内自行制定的标准规范来开发产品，而各个企业的标准规范之间存在有较大差异。

另一方面，敏捷开发等开发模型对软件过程能力度量并不能完全支持。例如有研究指出，Scrum 对 CMMI 中不支持的过程域有以下几个方面：过程管理类的过程域，包含了组织过程焦点、组织培训；项目管理类的过程域包含了供应商协议管理；支持类的过程域包含了过程和产品质量保证、决策分析和决定。而其他的过程域为支持或者部分支持。

2.5.1.2.5 汽车软件质量管理

在汽车软件开发过程中，许多软件产品却时常陷入质量低下、甚至软件不符合用户需求的旋涡。究其根源，有以下几个方面：

- 1) 软件质量保证技术(审查、复审和测试)没有贯穿到整个软件开发全过程中去。
- 2) 在于这些软件产品对其质量内涵的把握，仅仅停留在减少软件运行错误、加强软件测试，避免软件缺陷的一般性层面，而对整个软件开发生命周期的全过程质量管理，缺乏总体架构。
- 3) 测试管理的一些误区也会导致严重的质量问题。没有按照测试原则进行尽早测试、连续测试与自动化测试。是测试本省变得形式化。
- 4) 质量是全过程的，不仅是测试。质量管理者应该将质量控制与保证着眼于整个软件开发生存周期内。而事实上，质量管理者仅仅认为通过严格的测试就可以保证软件质量。

现有软件供应商多通过组装开源和商业软件组件来创建产品，对于整个软件开发生命周

期的全过程质量管理重视度不足，缺乏整体架构分析设计，且国内各主机厂面临着触碰不到供应商所提供软件底层代码的窘况，同时各供应商及主机厂所使用的软件开发工具链不健全，数据接口参差不齐，在集成多方提供的软件部件时易出现兼容性及系统完整性差等问题，导致软件系统可追性差等质量缺陷。

汽车软件质量管理是指参考行业体系标准 IATF16949、ISO26262、ISO21434 与软件体系成熟度标准 ASPICE、CMMI，建立汽车软件质量管理体系与测试体系，在汽车软件开发各个阶段开展质量控制活动，开展评审与测试活动，检查软件项目指定的交付件来决定它们与建立的标准是否符合，保证软件开发过程质量与结果质量，并基于研发工具链建设，对软件开发体系进行数字化管理。

当前汽车软件质量管理的发展现状主要包括以下几个方面：

标准和规范：汽车软件质量管理越来越重视标准和规范的应用。例如，ISO 26262 是针对汽车功能安全性的国际标准，要求制定相应的安全开发流程和评估方法。此外，汽车行业还采用了一些行业特定的规范，如 AUTOSAR（汽车软件架构）和 ASPICE（汽车软件过程能力评估）等，用于指导和评估软件质量管理实践。

测试和验证：测试和验证是汽车软件质量管理中至关重要的环节。当前的发展趋势是采用自动化测试和仿真技术，以提高测试覆盖率和效率。同时，测试方法也在不断创新，包括模型驱动测试、边界值分析、模糊测试等，以确保软件在各种情况下的正确性和可靠性。

静态分析和代码审查：静态分析和代码审查是提前发现和修复软件缺陷的重要手段。当前的发展趋势是采用静态代码分析工具，通过静态分析技术检测代码中的潜在问题和风险。代码审查也是一种常用的质量管理方法，通过团队成员之间的相互审查来发现和纠正问题。

需求管理：对需求获取、分析、管理，保证需求的一致性和可追踪性。

缺陷管理：对软件中发现的缺陷提供全生命周期的管理，覆盖从缺陷提出到修复验证的整个缺陷生命周期，实现缺陷的快速发现、全程追踪、及时消除和主动预防。

软件供应链管理：构建软件物料清单，在此基础上实现安全风险分析、许可证合规分析、

维护风险分析。

值得特别关注的是，开源软件由于其灵活性、低成本和创新性，已经成为汽车制造商及其供应链中广泛采用的关键资源。开源软件容易获得、可以灵活开发和部署，容易进行本地客户化，可以利用社区力量进行持续开发改进，与商业软件相比，没有软件授权费用，庞大的社区参与提供更安全的代码。但是使用开源软件也面临一些挑战，特别是开源软件的安全漏洞问题和许可证问题。

配置管理：确保软件版本控制和配置管理的有效性和一致性。

数据驱动的质量管理：随着智能网联汽车的兴起，汽车软件越来越依赖于数据。当前的发展趋势是利用大数据分析和机器学习技术，通过对车辆数据和用户反馈的分析，识别软件质量问题，并采取相应的改进措施。

整车验证和认证：整车验证和认证是确保汽车软件质量和安全性的关键步骤。当前的发展趋势是建立完善的整车验证流程，包括功能测试、性能测试、安全测试等，并通过认证机构的审核和认证，证明汽车软件满足相关标准和规范的要求。

有专家提出，建议按照汽车软件分类的更细颗粒度的软件管理体系。“分类保障”的目的是将公司最优的资源、最大的精力集中在重要的系统上，确保公司的系统和产品符合公司的战略发展。当前汽车行业的软件开发并没有建立相关的概念，但是汽车行业的软件同样有核心系统和其它周边系统的区分，同样需要针对不同重要性的系统建立分类的质量要求和控制措施，以合理分配公司资源，实现资源最优化、成本最优化、质量最优化。

2.5.2 汽车软件质量与度量类标准需求分析

2.5.2.1 汽车软件质量模型

2.5.2.1.1 国内外标准现状

软件质量模型主要有 McCall 模型、Boehm 模型、FURPS 模型；国际标准：ISO 9126、

ISO25010, ISO 9126 标准已经被 ISO/IEC 25010:2011 取代;

国标 GB/T 29831-29836 说明了软件 6 大特性、21 个子特性。

汽车电子软件未形成标准化质量模型, 主要是参考国际标准 ISO/IEC 25010:2011 和国家标准 GB/T 25000。

2.5.2.1.2 标准化需求分析

质量模型由特性、子特性、度量组成, 可以用于定义质量需求、确定开发和测试目标、评价软件产品等, 汽车正日益成为软件密集型的复杂系统, 汽车软件安全和可靠性越来越重要, 但缺乏相应的汽车软件质量度量指标, 需要结合现有软件质量模型与汽车软件特点行汽车软件质量模型。

2.5.2.2 汽车软件过程能力度量

2.5.2.2.1 国内外标准现状

针对于当前国内外软件质量管理标准现状, 按照通用及航天等领域的国标、国际标准、团标和行标, 和汽车行业的国标、国际标准、团标和行标进行分类:

通用国标:

GB/T 19000 系列: GB/T 19000 系列是中国通用质量管理体系标准, 包括 GB/T 19001 (ISO 9001) 等。

通用及航天领域软件相关国际标准:

ISO 9001: ISO 9001 是国际标准化组织制定的通用质量管理体系标准, 适用于各个行业的组织。

ECSS 标准: 欧洲航天标准化组织 (ECSS) 发布了一系列适用于航天软件的标准, 如 ECSS-Q-ST-80C (软件质量保证) 和 ECSS-E-ST-40C (软件工程)。

ISO/IEC 33020:2015

信息技术—过程评估—评估过程能力的过程度量框架

ISO/IEC 15504-5:2006

信息技术 - 过程评估 - 第 5 部分:典型的过程评估模型

团体标准:

CMMI: CMMI (Capability Maturity Model Integration) 是一种软件过程改进的框架, 由软件工程研究所 (SEI) 开发, 用于评估和提升组织的软件工程能力。

行业标准:

IEEE 标准: IEEE (Institute of Electrical and Electronics Engineers) 发布了许多软件工程和质量管理相关的标准, 如 IEEE 730 (软件质量保证计划) 和 IEEE 829 (软件测试文档) 等。

汽车行业的国标:

GB/Z 33013-2016 《道路车辆 车用嵌入式软件开发指南》规定了道路车辆用软件开发过程及相关要求, 并且适用于道路车辆用软件的开发, 验证和管理。

GB/T 34590.6-2022 《道路车辆 功能安全 第 6 部分: 产品开发: 软件层面》修改采用 ISO 26262, 适用于道路车辆的软件安全系统, 针对于电气/电子系统的功能安全, 规定了安全相关的开发框架, 也规定了组织应具备相应功能安全能力的开发流程要求。

汽车行业的国际标准:

ISO 26262: ISO 26262 是国际标准化组织发布的针对汽车电子系统功能安全性的标准。

汽车行业的团体标准:

AUTOSAR 是一种面向汽车电子系统的开放式软件架构标准, 旨在促进汽车软件的可重用性和互操作性。

Automotive SPICE 过程评估/参考模型

汽车行业的行业标准:

SAE 标准：SAE 国际（Society of Automotive Engineers）发布了一系列与汽车软件开发和测试相关的标准，如 SAE J3061（汽车网络安全）和 SAE J1739（设计 FMEA）等。

2.5.2.2.2 标准化需求分析

为了确保软件开发过程中的规范符合性，提供软件质量和安全性，促进互操作性和可重用性，通过汽车软件过程能力度量标准化加强汽车软件质量管理尤为重要。

国内目前的汽车软件过程质量管理存在较大缺失，目前主要是几个大域进行管理。

现状：采用整车试验和缺陷管理或质量问题推进软件过程质量管理和度量。

智能驾驶、座舱域和传统嵌入式软件的软件开发流程不一致，导致无法形成有效的、统一的度量管理。

过程质量指标包括如功能完成度、软件缺陷密度、问题管理率、测试覆盖率等，但各域软件开发差别较大。

2.5.3 标准体系及工作建议

基于汽车软件质量模型、开发模型、过程能力度量、智能网联安全新增需求等方面发展现状和需求的系统分析，以及对汽车软件质量模型、过程能力度量等方面进行已有标准的梳理，重点围绕汽车行业的特点和应用需求，梳理形成汽车软件质量与度量类标准体系，旨在凝聚行业共识，引导标准规范的制定。

表 24 汽车软件质量与度量标准体系

标准项目及分类		标准类型	标准描述	备注
质量与度量（600）				
汽车软件质量要求与评价（610）				
610-1	汽车软件质量模型	国标	定义汽车软件质量模型，由特性、子特性、度量组成，可以用于定义质量需求、确定开发和测试目标、评价汽车软件产品等。	

610-2	汽车软件规范符合性	团标	定义汽车软件建模规范、编码规范、接口规范符合性引用文件、符合性准则、符合性验证，确保汽车软件规范符合性要求的一致性
610-3	汽车软件功能性	团标	定义汽车软件功能完备性、正确性、恰当性、依从性的技术要求和测试方法，确保汽车软件功能性要求的一致性
610-4	汽车软件性能效率	团标	定义汽车软件时间特性、容量、资源利用性的技术要求和测试方法，确保汽车软件性能效率要求的一致性
610-5	汽车软件可靠性	团标	明确软件的可靠性标准化集合，确保软件可靠性要求的一致性
610-6	汽车软件安全性	国标	定义软件的功能安全、网络安全、数据安全、升级安全、融合安全等技术要求，确保智能网联汽车软件运行的安全性。
610-7	汽车软件易用性	团标	定义汽车软件的易用性要求，针对用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，提高用户友好的使用体验和操作便捷性。
610-8	汽车软件可移植性	团标	定义汽车软件的可移植性要求，汽车软件从一种硬件、软件或者其他运行环境迁移到另一种环境的有效性和效率的程度。针对可安装性与可替换性等方面要求，增强软件系统的可配置化程度，打通不同用户不同车型间的专用技术壁垒，促进软件间的相互协同，提高开发效率。
610-9	汽车软件兼容性	团标	定义汽车软件的兼容性要求，在共享相同的硬件或软件环境的条件下，汽车软件能够与其他产品、系统或者组件交换信息，和/或执行其所需的功能的程度。具体包含数据格式、交互协议与接口兼容性等方面要求，有助于打通软件烟囱式信息孤岛。
610-10	汽车软件维护性	团标	定义了质量模型及度量元选择的原则，规定了软件质量量化评价准则，为软件产品的质量模型提供了标准指标
610-11	汽车软件质量度量规范	国标	从智能网联汽车系统和软件的功能、性能、可靠性、安全性、可维护性等方面进行要求，以确保其能够满足用户需求和行业标准。
汽车软件过程能力度量（620）			
620-1	汽车软件开发过程参考模型	国标	定义汽车软件开发过程参考模型及应用方法，为汽车软件开发提供一致的参考
620-2	汽车软件敏捷开发过程参考模型	行标	定义汽车软件敏捷开发过程参考模型及应用方法，为汽车软件敏捷开发提供一致的参考
620-3	汽车软件开发运维一体化参考模型	行标	定义汽车软件开发运维一体化开发过程参考模型及应用方法，为汽车软件开发运维一体化提供一致的参考

620-4	汽车软件需求管理规范	团标	定义汽车软件需求管理的收集、分析、验证和确认的过程所应满足的要求，为汽车软件需求管理提供一致的参考
620-5	汽车软件编码规范	行标	定义不同汽车软件开发语言的编码规范，包括编码风格，编码技术约束及代码质量度量要求，旨在提高软件的可靠性，减少软件的缺陷
620-6	汽车软件集成与集成测试规范	团标	包含集成与集成测试活动要求、集成方法与工具、测试方法与工具，测试标准等，旨在确保汽车软件在开发过程中能够符合预期的功能、性能和安全性要求。
620-7	汽车软件缺陷管理规范	团标	定义汽车软件的缺陷管理等方面要求，提供覆盖软件全生命周期的缺陷管理服务，实现缺陷的快速发现、全程追踪、及时消除和主动预防。
620-8	汽车软件供应链管理规范	团标	定义汽车制造商和供应商在汽车软件开发、集成和交付等方面遵循的一系列流程和管理动作，明确了汽车软件的质量、可靠性和安全性的统一评价标准。
620-9	汽车软件配置管理规范	团标	定义管理和控制汽车软件开发、集成和交付中的配置项，定义了汽车制造商、供应商和开发者在软件生命周期过程中确保软件系统的质量、可维护性、合规性和安全性的统一评价标准。
620-10	汽车软件验证与确认规范	团标	定义汽车软件验证与确认的组织架构、作业规程、确认与验证的文件和记录要求等，为汽车软件验证与确认提供一致的参考
620-11	汽车软件过程能力成熟度模型	行标	根据汽车软件企业综合能力第二方或第三方评估需求，采用多个等级综合评估汽车软件过程能力
620-12	汽车软件开发运维一体化能力成熟度模型	行标	根据汽车软件企业综合能力第二方或第三方评估需求，通过多项能力域、能力及多级能力成熟度来描述汽车软件开发运维一体化成熟度模型

3 附录

汽车软件标准体系表

标准项目及分类		标准类型	标准描述	备注
基础（100）				
术语和定义（110）				
110-1	汽车软件专业术语和定义	国标推荐	定义了汽车软件基础通用类术语和定义，为汽车软件标准体系中各部分标准的制定提供基础支撑。	
操作系统（200）				
车载操作系统（210）				
210-1	车载操作系统技术要求和试验方法	国标推荐	定义了车载操作系统的总体架构，功能及性能要求，并规范功能和性能的验证方法，确保车控操作系统测试验证方法的统一同时为车载操作系统标准体系中其他部分标准的制定提供基础支撑。	
210-2	车载操作系统多系统技术要求	团标	明确了车载操作系统多系统的技术要求及部署方式，确保车载多系统技术的概念及技术转态的统一	
210-3	车载操作系统多系统间通信技术要求	团标	规定了车载操作系统多系统间通信的方式及技术要求，确保多系统间通信技术评价依据的统一	
210-4	基于人工智能的车载操作系统技术要求	团标	规定了基于人工智能的车载操作系统应满足的技术要求，包括提供训练模型的执行环境等功能要求、AR 导航、手势识别、图像理解等性能要求，提升基于人工智能的车载操作系统的服务质量	
210-5	车载操作系统硬件接口要求	团标	规定了车载操作系统硬件接口规范，确保车载操作系统硬件生态的统一	
210-6	车载操作系统应用接口要求	团标	规定了车载操作系统应用编程接口规范，确保车载操作系统软件生态的统一	
210-7	车载操作系统功能安全要求	团标	定义了车载操作系统的功能安全标准，确保智能网联汽车车载操作系统功能运行的可靠性	
210-8	车载操作系统可信执行环境技术要求	行标	定义了车载操作系统对于可信执行环境的支持及技术标准，确保车载操作系统可行执行环境技术能力及指标的一致性	
车控操作系统（220）				
220-1	车控操作系统接口要求+	国标推荐	定义车控操作系统面向应用和底层硬件的外部接口，以及操作系统内部各层接口，有	

			助于降低企业间适配成本，推动行业生态共建	
220-2	车控操作系统技术要求及测试方法	国标推荐	定义了车控操作系统的总体架构、功能及性能要求，并规范功能和性能的验证方法，确保车控操作系统测试验证方法的统一	
220-3	车控操作系统评价指南	团标	定义车控操作系统评价标准，确保车控操作系统评价体系的一致性	
220-4	车控操作系统功能软件架构及接口规范	团标	定义了车控操作系统功能软件的架构及应用的编程接口要求，确保车控操作系统基础能力及接口的统一	
220-5	车控操作系统虚拟化管理技术要求及测试方法	团标	定义了车控操作系统虚拟化技术的支持要求及对应的技术项测试验证方法，确保车控操作系统虚拟化技术标准 and 验证方法的统一	
220-6	车控操作系统中间件技术要求及测试方法	团标	定义了车控操作系统中间件的技术要求及验证方法，确保车控操作系统中间件的功能及验证方法的一致	
220-7	车控操作系统开发工具链技术要求及测试方法	团标	定义了车控操作系统开发工具链的技术标准及验证规范，确保工具链技术要求及测试验证方法的一致	
220-8	车控操作系统内核技术要求与测试方法	团标	定义了车控操作系统内核功能及性能要求，并规范内核技术的验证方法，确保内核功能、性能要求及测试方法的一致性	
220-9	车控操作系统面向硬件的接口规范	团标	规定了车控操作系统硬件接口规范，确保车控操作系统硬件生态的统一	
220-10	功能软件和系统软件层间接口规范	团标	规定了车控操作系统功能软件和系统软件层的编程接口规范，确保车控操作系统功能软件生态的统一	
220-11	车控操作系统功能安全技术要求及测试方法	行标	定义了车控操作系统的功能安全标准及验证方法，确保智能网联汽车车控操作系统功能运行的可靠性及验证方法的一致性	
220-12	车控操作系统配置方法	团标	规定了车控操作系统配置标准，包括车控操作系统软件模块的配置参数和配置方法、以及不同软件模块之间的协同方法，提高操作系统开发效率	
220-13	车控操作系统间通信要求	团标	定义了车控操作系统间的通信标准，确保车内多个操作系统之间能互相通信及兼容并存	
系统服务中间件（230）				
230-1	系统服务中间件技术要求及试验方法	国标推荐	明确操作系统中应支持的系统服务的技术要求和试验方法，包括标准化集合、安全要求等，确保车用操作系统功能要求的一致性	
230-2	原子服务技术要求及试验方法	行标	规定了整车原子服务接口规范及试验方法，确保车用操作系统整车生态的统一	
230-3	车端系统服务技术要求及试验方法	行标	规定了车端系统级服务接口规范及试验方法，确保车用操作系统整车生态的统一	
230-4	车云一体服务化技术要求与试验方法	行标	规定了车与云间的通信协议的技术要求与试验方法，确保车与云之间能互相通信及	

			兼容并存；规定了云端系统服务接口规范及试验方法，确保车用操作系统车云一体生态的统一	
230-5	整车电源管理技术要求与试验方法	团标	定义整车电源管理功能的技术要求及试验方法，确保整车电源管理功能及验证方法的一致；规定了车用操作系统间配电管理的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-6	整车网络管理技术要求与试验方法	团标	定义整车电源管理功能的技术要求及试验方法，确保整车电源管理功能及验证方法的一致；规定了车用操作系统间网络唤醒管理的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-7	整车健康管理技术要求与试验方法	团标	定义整车健康管理功能的技术要求及试验方法，确保整车健康管理功能及验证方法的一致；规定了车用操作系统间健康监控的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-8	整车时钟服务技术要求与试验方法	团标	定义整车时钟服务功能的技术要求及试验方法，确保整车时钟服务功能及验证方法的一致；规定了车用操作系统间时钟同步的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-9	整车诊断管理技术要求与试验方法	团标	定义整车诊断管理功能的技术要求及试验方法，确保诊断电源管理功能及验证方法的一致；规定了车用操作系统间诊断管理的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-10	整车 OTA 管理技术要求与试验方法	团标	定义整车 OTA 管理功能的技术要求及试验方法，确保整车 OTA 管理功能及验证方法的一致；规定了车用操作系统间软件升级控制的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-11	整车版本管理技术要求与试验方法	团标	定义整车版本管理功能的技术要求及试验方法，确保整车版本管理功能及验证方法的一致；规定了车用操作系统间版本采集的通信协议技术要求及试验方法，确保车内多个操作系统之间的互相通信及兼容并存	
230-12	整车数据采集技术要求与试验方法	团标	定义整车数据采集功能的技术要求及试验方法，确保整车数据采集功能及验证方法的一致；规定了车用操作系统间数据采集的通信协议技术要求及试验方法，确保车	

			内多个操作系统之间的互相通信及兼容并存	
230-13	整车调试系统技术要求与试验方法	团标	定义整车调试系统功能的技术要求及试验方法, 确保整车调试系统功能及验证方法的一致	
230-14	整车日志功能技术与试验方法	团标	定义整车日志功能的技术要求及试验方法, 确保整车日志功能及验证方法的一致; 规定了车用操作系统间日志采集的通信协议技术要求及试验方法, 确保车内多个操作系统之间的互相通信及兼容并存	
应用软件 (300)				
车域应用软件 (310)				
310-1	车控域应用软件编码规范	团标	本标准包括车控域应用软件需求分析、开发设计、测试验证和维护等流程, 旨在提高产品开发流程的规范性、效率性和质量。	
310-2	车控域应用软件功能安全技术要求	团标	本标准包括车控域应用软件功能安全等级要求建议与设计方法, 旨在规范功能安全相关开发要求及方法。	
310-3	车控域应用软件信息安全技术要求	团标	本标准包括车控域应用软件信息安全要求与设计方法, 旨在规范信息安全相关开发要求及方法。	
310-4	车控域应用软件功能测试规范	团标	包含测试过程、测试方法、测试工具和测试标准, 并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求	
310-5	车控域应用软件编码测试规范	团标		
310-6	传感器抽象接口设计规范	团标	包含传感器信号解析、网络信号定义与解析, 信号融合、处理与诊断, 旨在规范车控软件输入信号设计;	
310-7	传感器抽象设计规范	团标		
310-8	驱动软件技术要求	团标	包含执行器驱动控制算法、一致性与可靠性设计, 旨在提高执行器控制性能与可靠性	
310-9	执行器抽象接口设计规范	团标		
310-10	执行器抽象设计规范	团标		
310-11	车控域应用软件数据埋点设计规范	团标	包含车控软件典型功能规范与算法, 旨在提升车控软件应用功能规范性	
310-12	车控域应用软件健康管理设计规范	团标		
310-13	车控域应用软件 EOL 设计规范	团标		
310-14	车控域应用软件故障处理设计规	团标		

	范			
310-15	车控域应用软件动力系统功能规范	团标		
310-16	车控域应用软件接口设计规范	团标	包含车控软件架构、需求设计规范,旨在规范车控软件架构接口与需求设计,提高车控软件开发通用性及质量;	
310-17	车控域应用软件网络接口设计规范	团标		
310-18	车控域应用软件诊断接口设计规范	团标		
310-19	车控域应用软件平台软件需求规范	团标		
310-20	车控域应用软件电子电气需求规范	团标		
智驾域应用软件 (320)				
320-1	智驾域应用软件编码规范	团标	本标准包括智驾域应用软件需求分析、开发设计、测试验证和维护等流程、旨在提高产品开发流程的规范性、效率性和质量。	
320-2	智驾域应用软件功能安全技术要求	团标	本标准包括智驾域应用软件功能安全等级要求建议与设计方法,旨在规范功能安全相关开发要求及方法。	
320-3	智驾域应用软件信息安全技术要求	团标	本标准包括智驾域应用软件信息安全要求与设计方法,旨在规范信息安全相关开发要求及方法。	
320-4	智驾域应用软件功能测试规范	团标	包含测试过程、测试方法、测试工具和测试标准,并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求	
320-5	智驾域应用软件编码测试规范	团标		
320-6	智驾感知软件设计规范	团标	包含智驾软件在感知,融合,规划决策,和控制方面典型功能与算法设计规范	
320-7	智驾融合软件设计规范	团标		
320-8	智驾规划决策软件设计规范	团标		
320-9	智驾类控制软件设计规范	团标		
座舱域应用软件 (330)				

330-1	座舱域应用软件编码规范	团标	本标准包括座舱域应用软件需求分析、开发设计、测试验证和维护等流程、旨在提高产品开发流程的规范性、效率性和质量。	
330-2	座舱域应用软件功能安全技术要求	团标	本标准包括座舱域应用软件功能安全等级要求建议与设计方法,旨在规范功能安全相关开发要求及方法。	
330-3	座舱域应用软件信息安全技术要求	团标	本标准包括座舱域应用软件信息安全要求与设计方法,旨在规范信息安全相关开发要求及方法。	
330-4	座舱域应用软件功能测试规范	团标	包含测试过程、测试方法、测试工具和测试标准,并确保应用程序在开发过程中能够符合预期的功能、性能和安全性要求	
330-5	座舱域应用软件编码测试规范	团标		
330-6	座舱仪表控制软件设计规范	团标	包含座舱软件在仪表控制,信息娱乐,智能体验面向功能与算法设计规范	
330-7	座舱信息娱乐软件设计规范	团标		
330-8	座舱智能体验软件设计规范	团标		
整车应用软件(340)				
340-1	整车应用软件开发流程规范	团标	本标准包括整车应用软件需求分析、开发设计、测试验证和维护等流程、旨在提高产品开发流程的规范性、效率性和质量	
340-2	整车应用软件功能安全技术要求	团标	本标准包括整车应用软件功能安全等级要求建议与设计方法,旨在规范功能安全相关开发要求及方法。	
340-3	整车应用软件信息安全技术要求	团标	本标准包括整车应用软件信息安全要求与设计方法,旨在规范信息安全相关开发要求及方法。	
340-4	整车应用软件总体架构及要求	团标	在整车应用软件系统结构、行为和属性的高级抽象基础之上展开的全面的系统设计,其主要内容包括概要设计和详细设计	
340-5	整车应用软件编码技术要求	团标	规范整车应用软件编码规范,括文件组织、注释、声明、语句、命名约定、编程原则等规则	
340-6	OTA 应用软件设计规范	团标	规定 OTA 应用软件、整车安全及加密软件、整车诊断软件、跨域应用管理软件的接口及功能设计规范	
340-7	整车安全及加密软件设计规范	团标		
340-8	整车诊断软件设计规范	团标		
340-9	跨域应用管理软件设计规范	团标		
手机及网联应用软件(350)				
350-1	手机及网联服务类应用软件基础库和业务能力 sdk 的集成标准	团标	手机及网联服务类应用软件基础库和业务能力 sdk 的集成标准	
350-2	手机及网联服务类应用软件基础库和业务能力 sdk 的软件安全要求	团标	手机及网联服务类应用软件基础库和业务能力 sdk 的软件安全要求	
350-3	手机及网联服务类应用基础能力列表	团标	手机及网联服务类应用基础能力列表	

350-4	手机及网联服务类应用软件开发流程技术要求	团标	手机及网联服务类应用软件开发流程技术要求	
350-5	手机及网联服务类应用软件总体架构及要求	团标	手机及网联服务类应用软件总体架构及要求	
350-6	手机及网联服务类应用软件编码技术要求	团标	手机及网联服务类应用软件编码技术要求	
350-7	手机及网联服务类应用软件安全要求	团标	手机及网联服务类应用软件安全要求	
350-8	手机及网联服务类应用软件性能要求及测试方法	团标	手机及网联服务类应用软件性能要求及测试方法	
350-9	数据服务技术要求	团标	手机及网联服务类应用数据服务技术要求	
350-10	开发流程技术要求	团标	手机及网联服务类应用软件开发流程技术要求	
350-11	日志收集的要求	团标	手机及网联服务类应用软件日志收集的要求	
350-12	日志的信息安全要求	团标	手机及网联服务类应用软件日志的信息安全要求	
350-13	日志存储和使用的要求	团标	手机及网联服务类应用软件日志存储和使用的要求	
350-14	数据监控的要求	团标	手机及网联服务类应用软件数据监控的要求	
云端应用软件（360）				
360-1	云端应用软件开发流程要求	团标	规定了车联网云端应用软件开发流程的要求，包括需求分析、系统设计、开发环境搭建、编码规范、测试流程、安全性要求、部署和升级等方面的内容，为开发人员提供完整的开发指导，并确保云平台的功能完整性、可靠性。	
360-2	云端应用软件总体架构及要求	团标	明确车联网云端应用软件的总体架构、技术要求、管理要求，保证云平台高效开发及稳定运行。	
360-3	云端应用软件编码技术要求	团标	规定了车联网云端应用软件的实际编码过程和技术细节。明确了编码的规范、标准和实践，以确保开发团队在编码过程中遵循统一的标准和规范，减少错误和漏洞，提高软件的质量和安全性。	
360-4	云端应用软件安全要求	团标	明确了车联网云端应用软件中应具备的安全机制，包括用户身份认证、访问控制、数据加密、系统防护等方面的机制。为开发团队提供一个详细的安全要求指导，以确保车联网云端应用软件的安全性	
360-5	云端应用软件性能要求及测试方法	团标	明确车联网云端应用软件的性能要求，包括响应速度、处理能力、网络传输速度等方面的要求，并提供相应的测试方法和工具	
360-6	云端应用软件运维流程要求	团标	规范了云端应用软件运维流程的具体要求和方法。概述运维流程的各个环节，包括监控预警、问题流转、问题解决、质量管理等	

				全闭环流程，以提升车联网云平台运维效率	
数据服务软件（400）					
数据采集（410）					
410-1	智能网联汽车 车用数据格式及编码	国标推荐		规范汽车数据格式与编码，可用于规范汽车采集数据格式等	
410-2	汽车数据采集方法要求	行标		规范汽车数据采集过程与方法，包括对采集设备的要求，对采集性能开销的要求与试验方法	
数据开发（420）					
420-1	汽车数据预处理技术要求	行标		规范汽车数据清洗和预处理的基本方法和技术要求	
420-2	汽车数据存储技术要求	行标		规范汽车数据存储的技术要求和试验方法	
420-3	汽车数据处理与分析技术要求	行标		规范汽车数据处理和数据分析的技术要求和试验方法	
420-4	汽车数据质量规范	行标		定义汽车数据模型和数据质量评价指标，对汽车数据质量管理提供指导	
420-5	汽车数据治理体系要求	行标		汽车企业对数据实施治理的指南	
数据功能（430）					
430-1	主动运维功能定义与技术要求	行标		主动运维功能定义与技术要求	
430-2	主动运维功能接口规范	行标		规范主动运维功能与车端的接口	
430-3	全生命周期运营功能定义与技术要求	行标		全生命周期运营功能定义与技术要求	
430-4	全生命周期运营功能接口规范	行标		规范全生命周期运营功能与车端的接口	
430-5	智能驾驶研发服务功能定义与技术要求	行标		智能驾驶研发服务功能定义与技术要求	
430-6	智能驾驶研发服务功能接口规范	行标		规范智能驾驶研发服务功能与车端的接口	
数据安全（440）					
440-1	汽车数据处理安全技术要求	国标推荐		汽车数据进行传输、存储和出境等处理活动的安全要求	GB/T 41871 — 2022 《信息安全技术 汽车数据处理安全技术要求》，已发布

440-2	智能网联汽车 数据通用技术要求	国标推荐	规定了汽车数据的一般要求、个人信息保护要求、重要数据保护要求、审核评估及试验要求等	已报批
440-3	汽车数据安全管理体系规范	国标推荐	汽车企业建立实施数据安全管理体系的指南	汽标委 SC34 已启动研究
440-4	汽车整车信息安全技术要求	国标强制	规定了汽车信息安全管理要求、信息安全一般要求、信息安全技术要求、试验方法及同一型式判定，其中数据安全要求章节要求车辆应保护存储在车内车辆关键数据、敏感个人信息、安全日志、密钥等	已报批
工具软件 (500)				
工具软件通用要求 (510)				
510-1	汽车软件工具分类与术语定义	行标	分别针对汽车全生命周期所涉及的工具进行分类，并针对工具的使用和定义规范其术语和定义	
510-2	工具软件功能及性能技术要求及测试方法	团标	各类工具软件所通用的，制定相关标准规范以规定汽车工具类软件应具备的基础功能要求，包含记录文档的版本号、变更信息、著作人、主要内容以及注意事项等；同时规定需求分析软件的性能和效率要求，包含计算速度、内存占用、响应时间等方面的要求	
510-3	工具软件功能安全技术要求及测试方法	团标	各类工具软件所通用的，制定汽车工具类软件功能安全相关标准及对应测试方法，确保软件开发工具能满足功能安全对工具的要求	
510-4	工具软件网络安全技术要求及测试方法	团标	各类工具软件所通用的，制定汽车工具类软件网络安全相关标准及对应测试方法，为保障开发者、企业和最终用户的利益，并支持汽车软件的网络安全需求，具体可从漏洞扫描及修复、安全分析、测试、应急响应计划等方面开展工作	
510-5	工具软件数据安全技术要求及测试方法	团标	各类工具软件所通用的，制定汽车工具类软件数据安全相关标准及对应测试方法，规定敏感数据的存储、传输和处理方式，包括合适的加密算法、密钥管理、访问控制等，以保护数据的机密性和完整性；同时在软件使用过程中需要保护用户的个人隐私信息，包括数据采集、使用和销毁等方面的规定，规定对个人和敏感数据的处理方式，包括存储、传输访问控制，以符合相关隐私法律	
510-6	工具软件系统安全架构规范	团标	各类工具软件所通用的，制定汽车工具类软件系统安全架构相关标准规范，确定安全架构的设计原则和安全防护机制，包括身份认证、防火墙等，以确保系统的安全性	
510-7	工具软件图形化建模规范	团标	各类工具软件所通用的，制定相关标准规范以规定汽车工具类软件的图形化建模要	

			求,具体基于面向对象设计的建模,从基元模型、元模型、模型实例等方面开展相关工作	
510-8	工具软件文档及培训体系规范	团标	各类工具软件所通用的,制定相关标准规范以规定汽车工具类软件的文档和培训要求,具体包括用户手册、技术文档、培训资料等方面,可有助于用户理解和正确使用工具软件	
510-9	工具软件版本管理通用要求	团标	各类工具软件所通用的,制定相关标准规范以规定后期维护管理的相关要求,定义软件开发工具更新和维护细则以及软件更新的方式,以满足市场需求和用户期望,从而保障软件升级过程用户及产品工具的安全	
需求分析软件(520)				
520-1	需求分析软件用户体验测评准则及评价规程	团标	规定需求分析软件的用户界面和用户体验要求,具体包括界面设计、交互方式、可定制性、可访问性等方面,可有助于提供用户友好的使用体验和操作便捷性	
520-2	需求分析软件兼容性通用要求	团标	规定需求分析软件统一的数据格式与交互协议、接口兼容性等方面要求,可有助于打通软件烟囱式信息孤岛	
系统设计软件(530)				
530-1	系统设计软件用户体验测评准则及评价规程	团标	规定系统设计软件的用户界面和用户体验要求,具体包括界面设计、交互方式、可定制性、可访问性等方面,可有助于提供用户友好的使用体验和操作便捷性	
530-2	系统设计软件兼容性通用要求	团标	规定系统设计软件统一的数据格式与交互协议、接口兼容性等方面要求,可有助于打通软件烟囱式信息孤岛	
开发软件(540)				
540-1	开发软件用户体验测评准则及评价规程	团标	规定开发软件的用户界面和用户体验要求,具体包括界面设计、交互方式、可定制性、可访问性等方面,可有助于提供用户友好的使用体验和操作便捷性	
540-2	开发软件兼容性通用要求	团标	规定开发软件统一的数据格式与交互协议、接口兼容性等方面要求,可有助于打通软件烟囱式信息孤岛	
测试验证软件(550)				
550-1	软件用户体验测评准则及评价规程	团标	规定测试验证软件的用户界面和用户体验要求,具体包括界面设计、交互方式、可定制性、可访问性等方面,可有助于提供用户友好的使用体验和操作便捷性	
550-2	软件兼容性通用要求	团标	规定测试验证软件统一的数据格式与交互协议、接口兼容性等方面要求,可有助于打通软件烟囱式信息孤岛	

550-3	测试验证软件测试报告评审规范	团标	制定测试验证软件测试报告评审相关标准规范，针对详细的测试报告编写和评审的指导，具体包括测试结果的整理与汇总、缺陷统计分析、测试执行的回顾和总结等，以便有效地向相关利益相关者沟通测试进度和质量信息	
项目管理软件（560）				
560-1	项目管理软件用户体验测评准则及评价规程	团标	规定项目管理软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
560-2	项目管理软件兼容性通用要求	团标	规定项目管理软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	
后市场工具软件（570）				
570-1	后市场工具软件用户体验测评准则及评价规程	团标	规定后市场工具软件的用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，可有助于提供用户友好的使用体验和操作便捷性	
570-2	后市场工具软件兼容性通用要求	团标	规定后市场工具软件统一的数据格式与交互协议、接口兼容性等方面要求，可有助于打通软件烟囱式信息孤岛	
质量与度量（600）				
汽车软件质量要求与评价（610）				
610-1	汽车软件质量模型	国标	定义汽车软件质量模型，由特性、子特性、度量组成，可以用于定义质量需求、确定开发和测试目标、评价汽车软件产品等。	
610-2	汽车软件规范符合性	团标	定义汽车软件建模规范、编码规范、接口规范符合性引用文件、符合性准则、符合性验证，确保汽车软件规范符合性要求的一致性	
610-3	汽车软件功能性	团标	定义汽车软件功能完备性、正确性、恰当性、依从性的技术要求和测试方法，确保汽车软件功能性要求的一致性	
610-4	汽车软件性能效率	团标	定义汽车软件时间特性、容量、资源利用性的技术要求和测试方法，确保汽车软件性能效率要求的一致性	
610-5	汽车软件可靠性	团标	明确软件的可靠性标准化集合，确保软件可靠性要求的一致性	
610-6	汽车软件安全性	国标	定义软件的功能安全、网络安全、数据安全、升级安全、融合安全等技术要求，确保智能网联汽车软件运行的安全性。	
610-7	汽车软件易用性	团标	定义汽车软件的易用性要求，针对用户界面和用户体验要求，具体包括界面设计、交互方式、可定制性、可访问性等方面，提高用户友好的使用体验和操作便捷性。	

610-8	汽车软件可移植性	团标	定义汽车软件的可移植性要求，汽车软件从一种硬件、软件或者其他运行环境迁移到另一种环境的有效性和效率的程度。针对可安装性与可替换性等方面要求，增强软件系统的可配置化程度，打通不同用户不同车型间的专用技术壁垒，促进软件间的相互协同，提高开发效率。
610-9	汽车软件兼容性	团标	定义汽车软件的兼容性要求，在共享相同的硬件或软件环境的条件下，汽车软件能够与其他产品、系统或者组件交换信息，和/或执行其所需的功能的程度。具体包含数据格式、交互协议与接口兼容性等方面要求，有助于打通软件烟囱式信息孤岛。
610-10	汽车软件维护性	团标	定义了质量模型及度量元选择的原则，规定了软件质量量化评价准则，为软件产品的质量模型提供了标准指标
610-11	汽车软件质量度量规范	国标	从智能网联汽车系统和软件的功能、性能、可靠性、安全性、可维护性等方面进行要求，以确保其能够满足用户需求和行业标准。
汽车软件过程能力度量（620）			
620-1	汽车软件开发过程参考模型	国标	定义汽车软件开发过程参考模型及应用方法，为汽车软件开发提供一致的参考
620-2	汽车软件敏捷开发过程参考模型	国标	定义汽车软件敏捷开发过程参考模型及应用方法，为汽车软件敏捷开发提供一致的参考
620-3	汽车软件开发运维一体化参考模型	行标	定义汽车软件开发运维一体化开发过程参考模型及应用方法，为汽车软件开发运维一体化提供一致的参考
620-4	汽车软件需求管理规范	团标	定义汽车软件需求管理的收集、分析、验证和确认的过程所应满足的要求，为汽车软件需求管理提供一致的参考
620-5	汽车软件编码规范	行标	定义不同汽车软件开发语言的编码规范，包括编码风格，编码技术约束及代码质量度量要求，旨在提高软件的可靠性，减少软件的缺陷
620-6	汽车软件集成与集成测试规范	团标	包含集成与集成测试活动要求、集成方法与工具、测试方法与工具，测试标准等，旨在确保汽车软件在开发过程中能够符合预期的功能、性能和安全性要求。
620-7	汽车软件缺陷管理规范	团标	定义汽车软件的缺陷管理等方面要求，提供覆盖软件全生命周期的缺陷管理服务，实现缺陷的快速发现、全程追踪、及时消除和主动预防。
620-8	汽车软件供应链管理规范	团标	定义汽车制造商和供应商在汽车软件开发、集成和交付等方面遵循的一系列流程和管理动作，明确了汽车软件的质量、可靠性和安全性的统一评价标准。

620-9	汽车软件配置管理规范	团标	定义管理和控制汽车软件开发、集成和交付中的配置项,定义了汽车制造商、供应商和开发者在软件生命周期过程中确保软件系统的质量、可维护性、合规性和安全性的统一评价标准。
620-10	汽车软件验证与确认规范	团标	定义汽车软件验证与确认的组织架构、作业规程、确认与验证的文件和记录要求等,为汽车软件验证与确认提供一致的参考
620-11	汽车软件过程能力成熟度模型	行标	根据汽车软件企业综合能力第三方或第三方评估需求,采用多个等级综合评估汽车软件过程能力
620-12	汽车软件开发运维一体化能力成熟度模型	行标	根据汽车软件企业综合能力第三方或第三方评估需求,通过多项能力域、能力及多级能力成熟度来描述汽车软件开发运维一体化成熟度模型

全国汽标委智能网联汽车分技术委员会