

# 道路车辆功能安全-ISO26262 标准

<b>一、ISO26262-1 适用范围和主要内容</b>	4
<b>二、ISO26262-2 功能安全管理</b>	5
<b>三、ISO26262-3 概念阶段</b>	7
1、项目定义.....	7
2、项目的安全生命周期 .....	8
3、项目的危险分析和风险评估 .....	8
4、功能安全概念 .....	11
<b>四、ISO26262-4 系统级产品开发</b>	14
1、系统级产品开发启动 .....	14
2、技术安全需求制定 .....	15
3、系统设计.....	16
4、项目集成和测试 .....	19
5、安全确认.....	25
6、功能安全评估.....	26
7、产品发布.....	26
<b>五、ISO26262-5 硬件级产品开发</b>	27
1、硬件级产品开发初始化 .....	27
2、硬件安全需求规范拟定 .....	27
3、硬件设计 .....	28
4、硬件体系指标评估 .....	30
5、随机硬件故障对安全目标影响评价 .....	31
6、硬件集成和测试 .....	32
<b>六、ISO26262-6 软件级产品开发</b>	34
1、软件级产品开发启动 .....	34
2、软件安全需求规范拟定 .....	35
3、软件体系设计 .....	36
4、软件单元设计和实现 .....	39
5、软件单元测试 .....	42
6、软件集成和测试 .....	44
7、软件安全需求验证 .....	46
<b>七、ISO26262-7 生产运行</b>	46
1、生产 .....	46
2、运行、服务（保养和维护）和关闭 .....	48
<b>八、ISO26262-8 支持过程</b>	49
1、分布式开发接口 .....	49
2、安全需求规范和管理 .....	51
3、配置管理 .....	53
4、变更管理 .....	54
5、验证 .....	55
6、文档 .....	57
7、可信的软件工具 .....	58
8、软件组件证明 .....	62
9、硬件组件证明 .....	64
10、论证证明 .....	67

<b>九、ISO26262-9 面向汽车安全完整性等级(ASIL)和安全的分析.....</b>	<b>70</b>
1、考虑 ASIL 裁剪等级分解要求 .....	70
2、要素共存标准 .....	73
3、关联故障分析 .....	74
4、安全分析 .....	76
<b>十、ISO26262-10 指南 .....</b>	<b>78</b>

# ISO26262-1 适用范围和主要内容

ISO26262 是 IEC61508 对 E/E 系统在道路车辆方面的功能安全要求的具体应用。它适用于所有提供安全相关功能的电力、电子和软件元素等组成的安全相关系统在整个生命周期内的所有活动。

安全在将来的汽车研发中是关键要素之一，新的功能不仅用于辅助驾驶，也应用于车辆的动态控制和涉及到安全工程领域的主动安全系统。将来，这些功能的研发和集成必将加强安全系统研发过程的需求，同时，也为满足所有预期的安全目的提供证据。

随着系统复杂性的提高，软件和机电设备的应用，来自系统失效和随机硬件失效的风险也日益增加，ISO26262，包括其导则，都为避免这些风险提供了可行性的要求和流程。

系统安全可以从大量的安全措施中获得，包括各种技术的应用（如：机械，液压，气动，电力，电子，可编程电子元件）。尽管 ISO26262 是相关与 E/E 系统的，但它仍然提供了基于其他相关技术的安全相关系统的框架。

ISO26262：

-提供了汽车生命周期（管理，研发，生产，运行，服务，拆解）和生命周期中必要的改装活动。

-提供了决定风险等级的具体风险评估方法（汽车安全综合等级，ASILs） -使用 ASILs 方法来确定获得可接受的残余风险的必要安全要求。 -提供了确保获得足够的和可接受的安全等级的有效性和确定性措施。

功能安全受研发过程（包括具体要求，设计，执行，整合，验证，有效性和配置），生产过程和服务流程以及管理流程的影响。

安全事件总是和通常的功能和质量相关的研发活动及产品伴随在一起。ISO26262 强调了研发活动和产品的安全相关方面。

ISO 26262 主要用于安装在最大毛重不超过 3.5 吨的乘用车上的一个或多个 E/E 系统的安全相关系统。ISO26262 唯一不适用于为残疾人设计的特殊目的车辆的 E/E 系统。系统研发早于 ISO26262 出版日期的，也不在标准的要求之内。ISO26262 表述了由 E/E 安全相关系统，包括这些系统的互相影响，故障导致的可能的危险行为，不包括电击，火灾，热，辐射，有毒物质，可燃物质，反应物质，腐蚀性物质，能量释放及类似的危险，除非这些危险是由于 E/E 安全相关系统故障导致的。

ISO26262 对 E/E 系统的标称性能没有要求，对这些系统的功能性性能标准也没有什么要求（例如：主被动安全系统，刹车系统，ACC 等）

ISO26262 主要包括以下几个部分：

Part 1: 定义

Part 2: 功能安全管理

Part 3: 概念阶段

Part 4: 产品研发：系统级

Part 5: 产品研发：硬件级

Part 6: 产品研发：软件级

Part 7: 生产和操作

Part 8: 支持过程

Part 9: 基于 ASIL 和安全的分析

Part 10: ISO26262 导则

## ISO26262-2 功能安全管理

ISO26262 是 IEC61508 对 E/E 系统在道路车辆方面的功能安全要求的具体应用。它适用于所有提供安全相关功能的电力、电子和软件元素等组成的安全相关系统在整个生命周期内的所有活动。

那么，为什么遵照 ISO26262 就能设计出符合功能安全要求的产品呢？ISO26262 是通过什么方式来保证产品能够符合功能安全的要求的呢？下面我们就来具体看看 ISO26262 在产品研发上的具体思路。

ISO26262 系列标准分为 10 本，从 ISO26262-1 到 ISO26262-10，分别从功能安全管理，概念，系统级研发，软硬件的研发，生产和操作等方面对产品的整个生命周期进行了规范和要求。从而使得产品在各个生命周期都比较完善的考虑了其安全功能。

一个好的产品，要靠一整套好的管理体系来实现，并可靠的生产出来。ISO26262 给出了一套这样的管理方法、流程、技术手段和验证方法，称之为安全管理生命周期，框架如下：

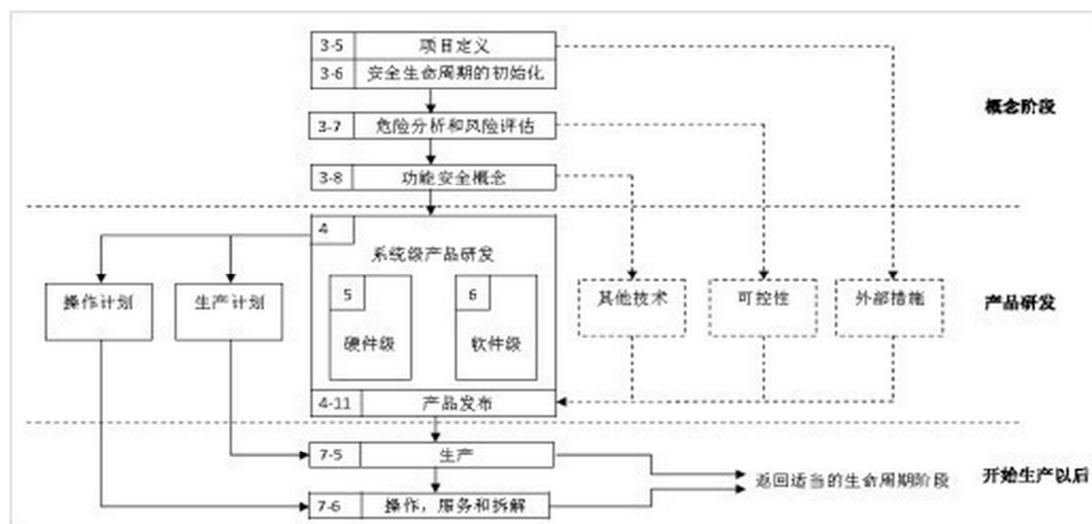


图 1 项目安全生命周期

那么各部分又有什么具体含义和措施呢？下面就来分别说明：

### 1、 项目定义：

项目定义，是对所研发项目的一个描述，是安全生命周期的初始化任务，其包括了项目的功能，接口，环境条件，法规要求，危险等内容，也包括项目的其他相关功能，系统和组件决定的接口和边界条件等。

### 2、 安全生命周期的初始化

基于项目定义，安全生命周期要对项目进行区分，确定是新产品研发，还是既有产品更改。如果是既有产品更改，影响分析的结果可以用来进行安全生命周期的拼接。

### 3、危险分析和风险评估

安全生命周期初始化之后，就要按照 ISO26262-3 的第七条款来进行危险分析和风险评估，危险分析和风险评估的流程要考虑暴露的可能性，可控性和严重性，以便确定项目的 ASIL 等级。接下来就是为每一个风险设立安全目标，并确定合适的 ASIL 等级。

### 4、功能安全概念

基于安全目标，功能安全概念就要考虑具体的基本架构。功能安全概念就是对定位到每个项目元素中的功能安全要求的具体化和细化。超出边界条件的系统和其他技术可以作为功能安全概念的一部分来考虑。对其他技术的应用和外部措施的要求不在 ISO26262 考虑的范围之内。

### 5、系统级产品研发

有了具体的功能安全概念之后，接下来就是按照 ISO26262-4 的系统级研发了。系统级研发的过程基于技术安全要求规范的 V 模型。左边的分支都是系统设计和测试，右边的分支是集成，验证，确认和功能安全评估。

### 6、硬件级产品研发

基于系统的设计规范，硬件级的产品研发要遵循 ISO26262-5 的要求。硬件研发流程应符合 V 模型概念左侧分支的硬件设计和硬件要求。硬件的集成和验证在右侧分支。

### 7、软件级产品研发

基于系统的设计规范，软件级的产品研发应遵循 ISO26262-6 的要求。软件研发流程应符合 V 模型概念中左侧分支的软件需求规范和软件设计架构设计的要求。软件安全需求中的软件集成和验证在右侧分支中。

### 8、生产计划和操作计划

其包括：生产和操作计划，相关的需求规范，系统级产品研发的开始等。ISO26262-7 的第 5 条款和第 6 条款给出了生产和操作的具体要求。

### 9、产品发布

产品发布是产品研发的最后一个子阶段，该项目也将完成，具体要求在 ISO26262-4 的第 11 条款中。

### 10、产品的操作、服务和拆解

产品的操作、服务和拆解应符合 ISO26262-7 的第 5 条款和第 6 条款中，对产品的生产、操作、服务和拆解的相关要求。

### 11、可控性

在危险分析和风险评估中，要考虑司机和处于危险中的其他人可以采取措施来控制危险情况的能力。如何提供对可控性的有效性证明不在 ISO26262 的范围之内。

### 12、外部措施

参考项目以外的，在项目定义中被描述的措施（参加 ISO26262-3 的第 5 条款），以便减小项目的危险结果。外部危险降低措施不但可以包括附加的车载设备，

如：动态稳定控制器防爆轮胎等，也可以包括非车载装置，如：护栏，隧道消防系统等。这些外部措施在进行危险分析和风险评估的时候应该被考虑到，但如何为这些外部措施的有效性提供证明不在 ISO26262 的范围之内，除非是 E/E 设备。但要注意的是，没有明确安全例证的外部措施是不完整的。

### 13、其他技术

其他技术是指那些不在 ISO26262 范围之内的，不同于 E/E 技术的设备。如：机械和液压技术。这些都要在功能安全概念的规范中加以考虑或者在制定安全要求时加以考虑。

通过以上这些具体的生命周期的各个阶段和标准中对每个阶段所必须考虑的措施、方法和具体技术的要求，将各个阶段的要求和如何满足要求的措施都进行逐一落实，这样才能设计出、制造出满足功能安全要求的安全产品。

#### 5.4.2 安全文化

7 组织应建立，执行和维持一个持续改进的过程，基于在：

- 1) 从其他项目的安全生命周期执行过程中学习的经验的，包括 现场经验；
- 2) 在以后的项目中的改进应用

## ISO26262-3 概念阶段

我们来具体看一下在概念阶段，ISO26262-3 对于项目定义、安全生命周期初始化和危险分析和风险评估的定义和要求。

### 5、项目定义

首先是项目定义阶段。项目定义，也就是对要进行研发的产品进行一个定义，进行一个描述。主要有两个目的：一个是定义和描述项目；一个是对项目有一个足够的理解，以便能够很好的完成安全生命周期中定义的每一个活动。

基于以上目的，要对项目进行明确、准确、正确的定义，就需要获得一些基本信息，ISO26262 中给出了一些建议如下：

#### 1、项目信息：

- a) 项目的目的和功能
- b) 项目的非功能性要求，如操作要求、环境限制等
- c) 法规要求（特别是法律和法规），已知的国家和国际标准等
- d) 类似功能、系统或元素达到的行为
- e) 对项目预期行为的构想
- f) 已知的失效模式和风险在内的项目缺陷造成的潜在影响

#### 2、项目的边界条件以及相关项目之间的接口条件：

- a) 项目的所有元素
- b) 项目对其他项目或项目环境元素的相关影响
- c) 其他项目，元素和环境对本项目的要求
- d) 在系统或者包含的元素中，对功能的定位和分配
- e) 影响项目功能时，项目的运行情况

有了以上这些基本的信息，就可以对要进行的项目给出一个比较明确和具体的项目定义，明确项目的要求，从而使得对项目有一个足够的理解，能够指导后续工作，来很好的完成安全生命周期中定义的每一个活动。

## 6、项目的安全生命周期

那么，有了项目定义之后，就要确定项目的安全生命周期，对项目的安全生命周期进行初始化，也就是开始对项目的安全生命周期进行细化。而要进行细化，就要区分是项目是新产品研发还是既有产品的改造。

如果是全新的设备研发，则相关工作就得从安全生命周期的开始做起，项目定义之后就是项目危险分析和风险评估。

如果是既有产品的改造，那么从项目定义开始的这些流程都可以使用一些既有的文件对整个过程进行定制。

现有产品升级改造，就要注意以下一些问题：

1. 要做一个产品和使用环境的分析，以制定出预期更改，并评估这些更改产生的影响。
  - a) 对项目的更改包括设计更改和执行更改。设计更改应该是由需求规范、功能和性能的增加或者成本的优化所致，执行更改不能影响项目的规格和性能，但可以影响执行特征。执行更改可以由软故障更改，使用新的研发成果或生产工具所致。
  - b) 如果配置数据和校准数据的更改会影响到产品的行为，则更改须考虑这些数据。
  - c) 对产品环境的更改应该是由产品要使用的新的目标环境或由于其他相关产品或元素升级而引发。
2. 要表述清楚产品使用的前后条件的差别，包括：
  - a) 操作条件和操作模式
  - b) 环境接口
  - c) 安装特征，如：在车辆内部的位置，车辆的配置和变化等
  - d) 环境条件的范围，如：温度，海拔，湿度，震动，EMC 和汽油标号等。
3. 要明确给出产品变更的描述以及影响的范围。如果不能明确产品的变更和对环境数据影响的改变，则相关影响的分析数据都要进行记录。
4. 影响到的服役产品，需要进行升级的，要进行逐一列出。
5. 定制的相关安全活动应符合各个应用生命周期阶段的要求，包括：
  - a) 定制应基于影响分析的结果。
  - b) 定制的结果应包括在符合 ISO26262-2 的安全计划中。
  - c) 影响到的产品须返工，包括确认计划和验证计划。

确定了以上这些基本信息之后，对所要进行的产品研发或者设备更改工作就有了一个清晰明确的定义，对产品的预期使用功能、环境，以及与相关设备的接口也有了一个明确的定义，接下来就可以进行危险分析和风险评估了。

## 7、项目的危险分析和风险评估

在概念阶段，ISO26262-3 给出了对危险分析和风险评估的要求。

危险分析和风险评估的目的和之前的 ISO13849, IEC62061 等的标准一样,都是为了将设备存在的危险识别出来,并根据危险的程度按照一定的原则对其进行分类,从而针对不同的风险设定具体的安全目标,并最终减小或消除风险,避免未知风险的发生。

也正是因为这样,危险分析、风险评估和 ASIL 等级的确定只是和避免风险有关的安全目标相关。通过对危险情况的系统评估,考虑引发危险的影响因素——伤害的严重性,暴露于危险中的可能性和危险的可控性,来确定安全目标和 ASIL 等级。而这三个指标都是针对产品的功能行为的,所以做危险分析和风险评估时,并不一定先要知道设计细节。

无内部安全机制的项目应在危险分析和风险评估过程中进行评估,拟实施或在以前的项目中已经实施的安全机制不在危险分析和风险评估考虑。在一个项目中,提供充分独立的外部评估措施是非常有效的。例如,如果有足够独立的证据证明,电子稳定控制系统可以通过增加控制来减少在底盘系统的故障影响。此举的目的是证明要实施或已经实施的项目的安全机制成立为功能安全概念的一部分

危险分析和风险评估的第一步是情形分析和危险识别,即通过相关的情况分析将产品存在的风险识别出来。这就要考虑可能引发危险的操作环境和操作模式,并且要考虑在正确使用时和可预见的误使用时的情况。基于这样的考虑,我们应该通过大量的技术来系统分析,注意以下一些方面:

1. 准备一个用来进行评估的操作情况清单
2. 系统的确定清单上的危险。主要可以通过诸如:头脑风暴,检查列表,历史记录, FMEA, 产品矩阵,以及相关的领域研究等技术手段进行。
3. 风险应该用在车辆上可以被观察到的条件或影响来进行定义或描述
4. 在相关操作条件和操作模式下危险事件的影响应该被明确说明。如: 车辆电源系统故障可能导致丧失引擎动力,丧失转向的电动助力以及前大灯照明。
5. 如果在风险识别中识别出的风险超出了 ISO26262 的要求范围,则需给出合适的相应措施。当然,超出 ISO26262 的风险可以不必分类分级。

完成风险的识别之后,就要对这些风险进行适当的分级,以便设定相应的安全目标,并按照不同的风险等级来采取合理的措施加以避免。

风险的分类主要是通过 3 个指标来考量,即: 危险发生时导致的伤害的严重性、在操作条件下暴露于危险当中的可能性(危险所在工况的发生概率)、危险的可控性。

首先,来看伤害的严重性。这里的伤害是指危险事件发生时,对所有被卷入事件中的人的伤害,包括车上的司机和乘客,骑自行车的人,行人,其他车辆上的人员。伤害的严重性可以分为 4 个等级,即: S0, S1, S2, S3 (对于伤害严重性的详细描述可以参考 ISO26262-3 中附录 B 的内容,这里只做分级说明)。如下表:

级别	S0	S1	S2	S3
描述	无伤害	轻微或有限的伤害	严重或危及生命的伤害 (可以幸存)	危及生命的伤害(可能不能幸存)或致命伤害

其次,来看在操作条件下暴露于危险中的可能性。可能性被分为 5 个等级,即: E0, E1, E2, E3, E4, 具体分级见下表。至于暴露值是选 E1 还是选 E2, 主要看车辆在目标市场正常、

合理的使用情况。这里要注意的是，评估暴露于危险中的可能性并不考虑在车上安装了多少个要评估的产品，且假设了所有的车上都安装了这个产品。对于那种认为不是每辆车都安装的产品，其相应的暴露在危险中的可能性会减小的说法也是错误的。

级别	E0	E1	E2	E3	E4
描述	几乎不可能	可能性非常低	可能性低	中等可能性	可能性高

这里，E0 只用于在风险分析中一些建议性的情况，通常如果一个危险，人员暴露其中的可能性是 E0 级，则无需考虑 ASIL 等级。

再次，来看可控性。即危险事件能被司机或者其他交通参与人员进行控制并减小或者避免伤害的可能性。在 ISO26262 中，可控性被分为 4 个等级，即：C0，C1，C2，C3。但要注意，使用这个分级的条件是司机处于正常状态，即：不疲劳，有驾照，按照交通规则行驶，当然，其中要考虑可预见的误操作和误使用。四个级别为：

级别	C0	C1	C2	C3
描述	通常可控	简单可控	正常可控	很难控制或不可控

其中，C0 通常用于不影响车辆安全操作的情况。如果一个危险的可控性被评为 C0，则对其没有 ASIL 要求。

由此，根据以上的三个参数，即可确定风险分析中每个风险相应的 ASIL 等级（汽车安全完整性等级），具体确定方法如下表：

Table 4 — ASIL determination

严重性等级	危险可能性等级	可控性等级		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

ASIL 等级分为 A、B、C、D 四个等级，ASIL A 是最低的安全等级，ASIL D 是最高的安全等级。除了这四个等级 QM 表示与安全无关。

在风险分析过程中，要确保对每个危险事件，根据 S、E、C 和具体的操作条件和模式确定的 ASIL 等级不低于其安全目标的要求。同时，相似的安全目标也可以合并为一个安全目标，但要达到的 ASIL 等级应该是合并项目中最高的。如果安全目标可以被分解到具体的状态中，那么每个安全目标也要转换成达到安全目标的具体安全状态下的具体要求。

安全目标及其属性（ASIL）应按照 ISO26262-8:2011，第 6 条款规定。

要注意的是，危险分析、风险评估和安全目标都要进行审核，以保证对条件和危险分析完整，符合项目定义，并与相关的危险分析和风险评估一致。

由此，完成概念阶段的危险分析和风险评估，形成减少和防止危险发生的安全目标，并通过验证审核。

## 8、功能安全概念

做完危险分析和风险评估之后，在概念阶段，ISO26262-3 还给出了功能安全概念这个阶段。其主要目的是通过前面的危险分析和风险评估之后得出的安全目标来确定具体的功能安全要求，并将它们分配到初步的设计架构，或者外部减少危险的措施当中去，以确保满足相关功能安全要求。

为了符合功能安全目标，功能安全概念给出了一些基本的安全机制和安全措施，以便于功能安全要求被很好的分配到系统架构的元素中去。这些主要的机制和措施如下：

- 故障检测和失效缓解措施
- 安全状态转换
- 故障容错机制。即：故障不会直接导致违背安全目标，或者保持系统出于安全状态（降级或者没有降级）
- 故障检测和为了将暴露时间减小到可接受的程度的司机警示装置
- 逻辑仲裁：不同功能触发的多任务请求应该通过逻辑仲裁来选择最合适的控制

基于以上这些机制和措施，再根据之前的项目定义、危险分析和风险评估、安全目标的设定，以及考虑来自外部的一些预想架构、功能、操作模式及系统状态等，就可以开始考虑将功能安全要求进行适当的分配，指定 ASIL 等级，并将其合理的分配到子系统当中了。安全目标和功能安全要求的层次结构如下表所示：

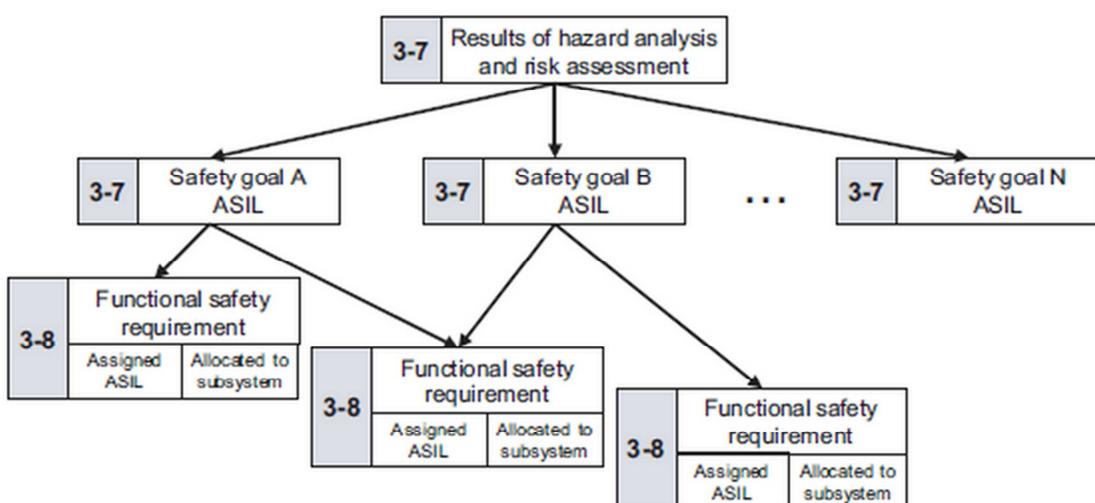


图 安全目标和功能安全要求的层次结构

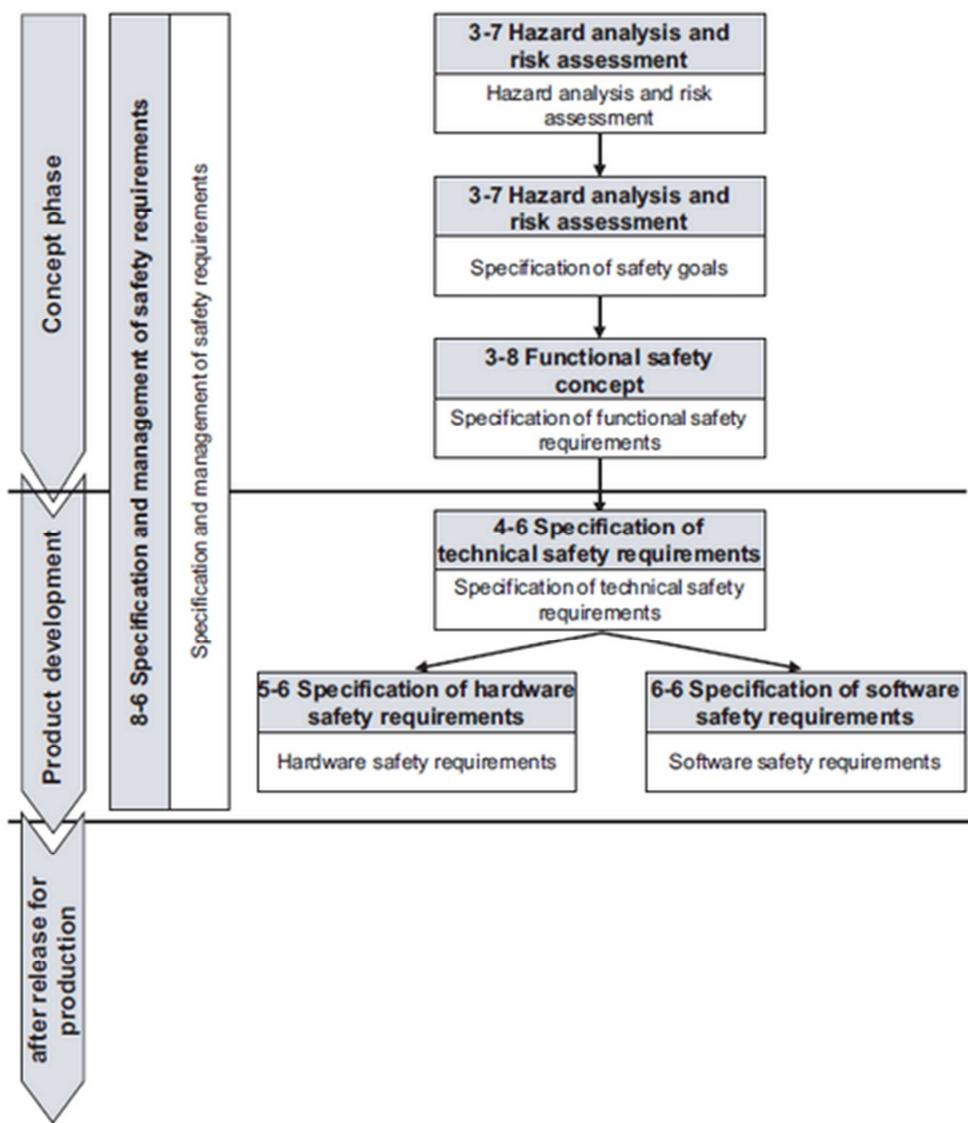


图 安全要求结构

在功能安全概念中，ISO26262 从功能安全要求的来源和功能安全的分配两个方面给出了一些建议和要求，具体如下：

1. 功能安全要求的来源：

- 功能安全要求应该从安全目标和安全状态来获得，并考虑预想架构、功能概念、操作模式和系统状态等。
- 要为每个安全目标设定至少一个功能安全要求。
- 每个功能安全要求都要考虑以下内容：
  - 操作模式
  - 故障容错时间间隔
  - 安全状态，过渡到安全状态是否符合设备要求
  - 急停操作间隔
  - 功能冗余

这项活动可以通过安全分析（如 FMEA, FTA, HAZOP），以制定一套完整有效的功能性安全要求的支持。

- d) 警示和降级
- e) 如果安全状态不能通过立即关闭来达到，则需指定一个紧急操作。
  - i. 这些动作应该在功能安全概念中详细描述
  - ii. 驾驶员或者陷入危险中的人可以使用的手段或者控制要在功能安全概念中详细描述
- 2. 功能安全要求的分配：
  - a) 研发安全架构概念
  - b) 功能安全要求分配
    - i. 功能安全要求的分配应该基于项目预想架构的元素进行。
    - ii. 分配过程中，ASIL 和功能安全要求考虑的内容信息都要继续传承。
    - iii. 如果多个功能安全要求被分配到同一个架构元素，则这个架构元素应以这些功能安全要求的最高 ASIL 等级进行研发。
    - iv. 如果项目由超过一个的系统组成，则对于每个独立系统和他们的接口的功能安全要求都要从考虑预想系统架构的功能安全要求中获得，而这些功能安全要求也都需要被分配到系统中去。
    - v. 如果 ASIL 等级需要被拆解，则要符合 ISO26262-9 第五条款的要求。
    - vi. 如果安全要求被分配到其他技术的元素中，则无需考虑 ASIL 等级。
  - c) 如果功能安全概念依赖于其他技术的元素，则应考虑以下环节：
    - i. 靠其他技术执行的功能安全要求应该从其相应的元素中获得并分配到元素中去。
      - ii. 明确与其他技术的接口的相关功能安全要求。
      - iii. 有其他技术执行的功能安全要求要确保有具体的措施。
  - d) 依赖于外部风险降低措施的功能安全概念应满足如下要求：
    - i. 应用于外面风险降低措施的功能安全要求应该从相应的外部风险降低措施中获得并分配到其中去。
      - ii. 明确与外部风险降低措施的接口的功能安全要求
      - iii. 如果外部风险降低措施由 E/E 系统构成，则功能安全要求可以用 ISO26262 来进行评估。
    - iv. 必须确保由外部风险降低措施执行的功能安全要求的正确执行。
  - e) 功能安全概念应该按照 ISO26262-8 第九条款的要求来验证与安全目标的一致性和符合性。
  - f) 项目安全确认的原则应该详细的写在功能安全概念中。
  - g) 功能安全要求的审核应该阐明功能安全要求符合安全目标。

由此，按照流程完成以上的这些分析和审核之后，即完成了功能安全概念的阶段，最终会形成功能安全概念的结果，和通过审核的功能安全要求。

## 附录 A 概念阶段概述

## 附录 B 危险分析和风险评估

给出了危险分析和风险评估的一般解释

对于这种分析方法，风险（R）可在危险事件被描述为一个函数（F），与危险事件的发生频率（f），即通过的人的及时反应避免特定的伤害或损害能力，损害或损伤的可控性（C），以及潜在的严重程度（S）有关。

$$R = F(f, C, S)$$

发生的频率 f 有以下几个因素确定，一个需要考虑的因素是危险事件的频繁度和在危险事件涉及的人数。在 ISO26262 中，这个的度量是简化成暴露于危险中的可能性（E）。另一个因素是，有可能导致危险事件（故障率， $\lambda$ ）的产品故障率，故障率的特点是硬件随机故障和系统性故障：

$$f = E \cdot \lambda$$

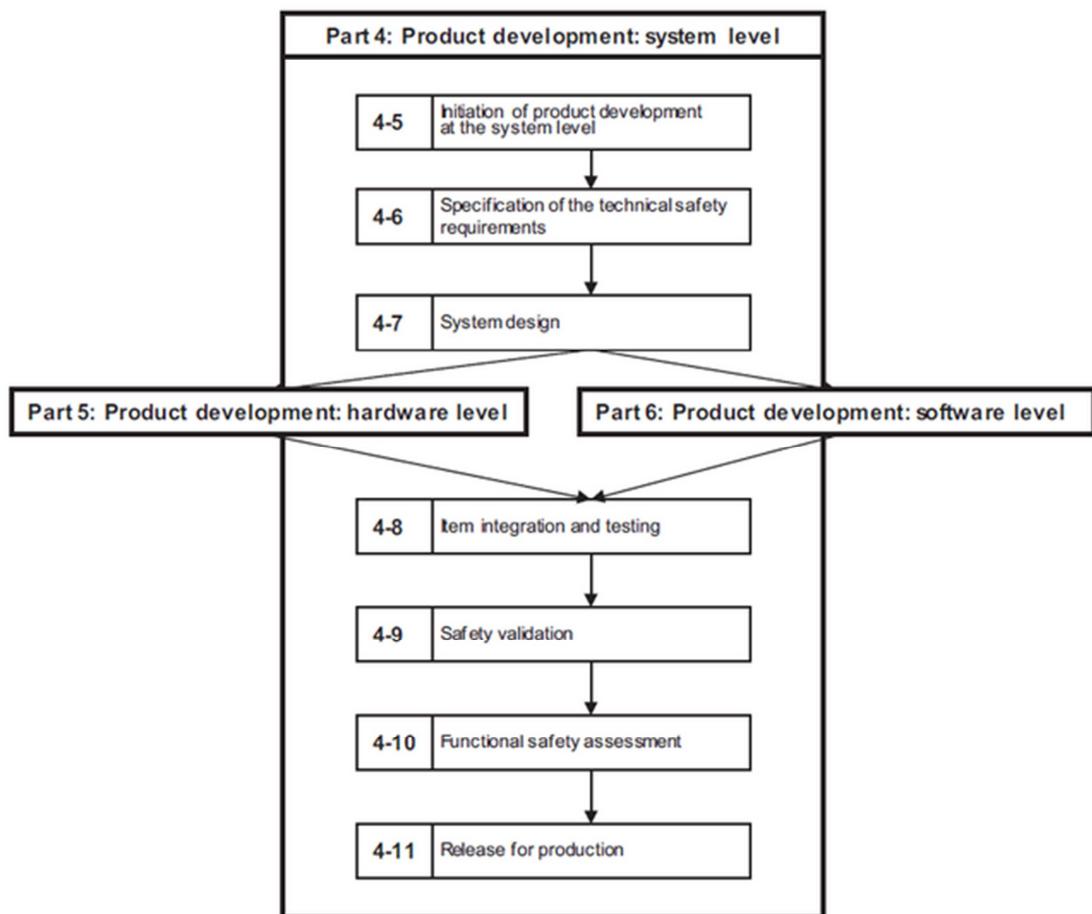
危险分析和风险评估用来设置功能安全要求，这样项目风险是可以避免的。ASILs 等级导出的危害分析和风险评估确定出项目的功能安全要求最小集合。

## ISO26262-4 系统级产品开发

### 5、系统级产品开发启动

系统级产品开发启动的目标是确定和规划在系统开发各个子阶段的功能安全活动。这部分内容在 ISO26262-8 中也有描述。系统级安全活动包含在安全计划中。

系统开发的必要活动如下图所示，产品开发启动和技术安全需求说明之后是系统设计。在系统设计过程中，系统体系结构建立以后，技术安全要求被分配到的硬件和软件部分，如果合适的话，分配到其它技术。从系统架构所增加产生的需求，包括硬件，软件接口 (HSI)，对技术安全要求进行细化，依据体系结构的复杂性，对子系统的需求依次地导出。之后，硬件和软件部分进行集成和测试，然后进行装车测试。一旦到装车测试的水平，执行安全确认，以提供达到安全目标的功能安全证据。系统级产品开发启动的安全活动是计划设计和集成过程中适当的方法和措施。



## 6、技术安全需求制定

这个阶段的第一个目标是规范技术安全需求。该技术安全需求说明细化了功能安全的概念，同时考虑功能性的概念和初步的体系架构。第二个目标是通过分析技术安全需要来验证符合功能安全需求。

在整个开发生命周期，技术安全需求是要落实功能安全概念的技术要求，其用意是从细

节的单级功能安全要求到系统级的安全技术要求。

### **技术安全需求规范**

技术安全需求应符合功能安全的概念，项目的初步架构和系统相关属性：

- 1、外部接口，如通信和用户界面；
- 2、限制，例如环境条件或功能限制；
- 3、系统配置要求。

如果其他功能或要求由系统或其部件来实现，除了技术安全需求规范规定的那些功能，那么其他要求应作为他们的规范或做参考。其它要求比如：经济委员会（欧洲经委会）的规则，联邦机动车辆安全标准（FMVSS）或公司的平台战略。

技术安全需求须指明安全相关的依赖关系，系统之间或项目之间，项目与其他系统之间。

### **安全机制**

技术安全需求应指定系统或要素达到安全目标的影响因素，包括每个相关的工作模式和系统定义的状态的失效和相关因素的组合。比如，如果车辆稳定性控制的制动系统是不可用的，自适应巡航控制系统（ACC）ECU 禁用 ACC 功能。

技术安全需求规定的必须的安全机制包括：

- 1、系统本身的检测，指示和故障控制措施，包括系统或元件来检测随机硬件故障，检测系统故障的自我监控措施，包括检测和控制通信信道失效模式的措施（例如，数据接口，通信总线，无线射频链路）。
- 2、检测，指示和与该系统交互的外部设备的故障控制的措施，比如，外部设备包括其它电子控制单元，电源或通信设备。
- 3、使系统达到或维持安全状态的措施。这包括在相互冲突的安全机制的情况下优先级 和仲裁逻辑情况。
- 4、细化和实现警告和降级概念的措施
- 5、防止故障被隐藏的措施

为使项目达到或维持一个安全状态的安全机制应规定：

- 1、安全状态的切换
- 2、容错的时间间隔
- 3、如果安全状态不能立即达到，应确定应急操作的时间间隔
- 4、维持安全状态的措施

### **ASIL 分解**

按照 ISO26262-9:2011，第 5 条款

### **潜在故障的避免**

制定安全机制以防止故障被隐藏。关于随机故障，只有多点故障有可能包含潜在故障，比如，在线测试，在不同的操作模式如上电，掉电，在运行时或在额外的测试模式下，来检测潜在故障，以验证组件状态的安全机制。阀门，继电器或指示灯功能测试是这样的在线测试的例子。

识别防止故障被潜伏的安全措施的评估标准来自于良好的工程实践。潜在故障的度量，

在 ISO26262-5:2011，第 8 条款给出，提供评价标准。

适用于 ASIL 的技术安全需求应避免多点故障失效，确定多点故障检测间隔时，应考虑以下因素：

- 1、根据硬件的可靠性考虑它在体系中的角色
- 2、相应的危险事件曝光的概率
- 3、由违反安全目标的硬件随机失效概率规定量化目标值
- 4、分配的 ASIL 等级对应的安全目标

下列采取的措施依赖于时间限制：

- 定期测试运行期间，系统或元件；
- 在上电或掉电时在线测试元件；
- 维护期间测试系统或元件。

防止双点故障被潜伏的安全机制的开发应符合：

#### **产品、运行、维护和结束**

在生产，经营，维护，维修和关闭的项目或元件的功能安全性的技术安全要求在 ISO 26262-7 中规定。

#### **检验和确认**

技术安全要求应按照 ISO26262-8:2011，第 9 条，进行验证：

- a) 符合的功能安全概念，
- b) 遵守初步体系设计。

项目的安全确认标准应根据技术安全细化要求。

## **7、系统设计**

这个阶段的第一个目标是进行系统设计、开发符合项目技术安全需求规范的功能要求。第二个目标是校验系统设计和功能要求。

系统设计和基于项目技术安全需求规范的技术安全概念来源于功能安全概念。为了开发一个系统架构设计，功能性安全要求，技术安全要求和非安全相关的要求被完成。因此，在这个阶段安全和非安全相关的要求都在这个过程中处理。

#### **系统设计规范和技术安全概念**

技术安全要求的应分配给系统设计要素，同时系统设计应完成技术安全要求，关于技术安全要求的实现，在系统设计中应考虑如下问题：

- 1、系统设计的可验证性
- 2、软件硬件的技术实现性
- 3、系统集成中的执行测试能力

#### **系统架构设计约束**

系统和子系统架构应该满足各自 ASIL 等级的技术安全需求，每个元素应实现最高的 ASIL 技术安全需求，如果一个系统包含的子系统有不同的 ASIL 等级，或者是安全相关的子系统和非安全相关的子系统，那么这些系统应该以最高的 ASIL 等级来处理。

安全相关的内部和外部接口应该被定义，避免其他因素影响安全相关的接口。

#### **系统失效的避免措施**

在系统设计安全分析，根据下表和 ISO26262-9:2011，第 8 条款，找出系统故障的原因和系统故障的影响。

**Table 1 — System design analysis 系统设计分析**

Methods		ASIL			
		A	B	C	D
1	Deductive analysis <sup>a</sup> 因果分析	o	+	++	++
2	Inductive analysis <sup>b</sup> 预测分析	++	++	++	++
<sup>a</sup> Deductive analysis methods include FTA, reliability block diagrams, Ishikawa diagram. 故障树、可靠性方框图、石川图					
<sup>b</sup> Inductive analysis methods include FMEA, ETA, Markov modelling. FMEA、事件树、马尔科夫模型					

这些分析的目的是协助设计。因此，在这个阶段，定性分析很可能是足够的，在需要时可以执行定量分析。这些分析从细节的角度来识别、确定和排除系统故障的原因和影响。从内因和外因进行系统性故障识别来消除或缓解其影响。

为了减少系统故障，应采用良好的值得信赖的汽车系统的设计原则。这些包括以下内容：

- 1、可重用、可靠的技术安全概念
- 2、可重用、可靠的软件、硬件设计单元
- 3、可重用、可靠的检测控制故障机制
- 4、可重用、可靠的标准化接口

为了确保可靠的设计原则在新的项目单元的适宜性，重用之前应进行影响分析和潜在的假设条件。影响分析包括所确定的诊断，环境的约束和可行性限制，时间限制，所确定的资源的兼容性，并且在系统设计的鲁棒性。

ASIL D 规定：可靠的设计原则不再重用应该是有一定理由的。

ASIL A、B、C、D 规定：为避免高复杂性带来的故障，架构设计应该根据表 2 中的原则来展现下列的属性：模块化，层次化，简单化

**Table 2 — Properties of modular system design 模块化系统设计属性**

Properties		ASIL			
		A	B	C	D
1	Hierarchical design 分层设计	+	+	++	++
2	Precisely defined interfaces 清晰定义的接口	+	+	+	+
3	Avoidance of unnecessary complexity of hardware components and software components 避免不必要的复杂软硬件组件	+	+	+	+
4	Avoidance of unnecessary complexity of interfaces 避免不必要的复杂接口	+	+	+	+
5	Maintainability during service 后期服务的可维护性	+	+	+	+
6	Testability during development and operation 开发运行过程中的可测试性	+	+	++	++

### 运行过程中随机硬件失效的控制措施

检测、控制、减轻随机硬件故障的措施在系统设计规范和技术安全概念中给出。例如，硬件诊断功能及其软件这些措施可以用来检测随机硬件故障，直接导致随机硬件故障的情况下硬件设计即使没有检测也是失败的。

ASIL (B) C、D 规定要求：对于单点故障和潜点故障的目标值（见 ISO26262-5:2011，第 8 条款），应在项目级指定最终评估（见要求 9.4.3.4）。

ASIL (B) C、D 规定要求：由于随机硬件故障违反安全目标的评价应该作为替代方法之一（见 ISO26262-5:2011，第 9 条款）目标值应在项目级别中指定为最终评估（见 9.4.3.4）。

ASIL (B) C、D 规定要求：对于故障率和诊断覆盖率的目标值应在在单元级中指定以满足下列要求：

- a) ISO 26262-5:2011, 第 8 条款中的目标值矩阵；
- b) ISO 26262-5:2011, 第 9 条款的流程。

ASIL (B) C、D 规定要求：分布式发展（见 ISO26262-8:2011，第 5 条款），派生目标值应送交各相关方。

在 ISO26262-5:2011 第 8 和 9 条款描述中的架构限制，不能直接适用于检测设备(COTS)零部件。这是因为供货商通常不能预见在最终产品其产品的使用和潜在的安全问题。在这种情况下，基本数据，如故障率，故障模式，每故障模式下的故障率分配，内置诊断等都是为了让零部件供应商估算在整体硬件架构层的架构限制。

### **硬件和软件配置**

技术安全要求，应直接或通过进一步细化到硬件，软件或两者兼有。如果技术安全要求被分配到定制的硬件单元包括可编程的行为充足的开发过程（诸如 ASIC，FPGA 或其他形式的数字硬件）有足够的发展，应结合 ISO26262-5 和 ISO26262-6 的要求，来制定和实施。遵照分配的硬件单元的安全性要求可以依据 ISO26262-8:2011，第 13 条款。

系统的设计应符合分配和分区决策，为了实现独立，避免故障的传播，系统设计时可实现的功能和组件的划分。

### **硬件和软件接口规范 (HSI)**

软硬件接口规范应规定的硬件和软件的交互，并与技术安全的概念是一致的，应包括组件的硬件设备，是由软件和硬件资源控制支持软件运行的。软硬件接口规范应包括下面属性：

- 1、硬件设备的工作模式和相关的配置参数，硬件设备的操作模式，如：缺省模式，初始化，测试或高级模式，配置参数，如：增益控制，带通频率或时钟分频器。
- 2、确保单元之间的独立性和支持软件分区的硬件特性
- 3、共享和专用硬件资源，如内存映射，寄存器，定时器，中断，I / O 端口的分配。
- 4、硬件设备的获取机制，如串口，并口，从，主/从
- 5、每个涉及技术安全概念的时序约束

硬件和其使用的软件的相关诊断功能应在软硬件接口规范中规定：

- 1、硬件诊断功能应定义，例，检测过流，短路或过热
- 2、在软件中实现的硬件诊断功能

软硬件接口规范在系统设计时制定，在硬件开发和软件开发时被进一步细化。

### **产品运行、维护和关闭要求**

诊断功能规定应保存现场运行过程中项目或单元的监测数据，以考虑到安全结果分析和安全机制运行

为了保持安全功能，诊断功能应规定允许故障识别可以由车间员工进行服务时获得。

产品运行、维护和关闭要求应包括如下功能：

- 1、 安装说明要求
- 2、 安全相关的特殊说明
- 3、 确保系统或元件正确识别的要求，如标签
- 4、 产品的核查方法和措施
- 5、 诊断数据和售后服务要求
- 6、 关闭要求

### 系统设计验证

系统设计应遵守和具备安全概念，使用表 3 中列出的验证方法进行验证。

**Table 3 — System design verification 系统设计验证**

Methods	ASIL			
	A	B	C	D
1a System design inspection <sup>a</sup> 系统设计审查	+	++	++	++
1b System design walkthrough <sup>a</sup> 系统设计走查	++	+	o	o
2a Simulation <sup>b</sup> 仿真	+	+	++	++
2b System prototyping and vehicle tests <sup>b</sup> 系统原型和车辆测试	+	+	++	++
3 System design analyses <sup>c</sup> 系统设计分析	see Table 1			

<sup>a</sup> Methods 1a and 1b serve as a check of complete and correct implementation of the technical safety requirements.  
<sup>b</sup> 方法1a和1b作为完整和正确执行安全技术要求的检查技术  
<sup>c</sup> For conducting safety analyses, see ISO 26262-9:2011, Clause 8. 进行安全分析，见ISO26262-9:2011，第8章。

按照技术安全概念要求，将异常和不完整的情况汇总形成系统设计检测报告。在安全目标下，系统设计未覆盖的新识别的危险，应写入危险分析和风险评估报告，按照 ISO26262-8:2011，第 8 条的变更管理要求来进行。

## 8、项目集成和测试

集成和测试阶段包括三个阶段和两个主要目标如下所述：第一阶段为每个项目包含的元件的硬件和软件的集成。第二阶段是一个项目的元件的集成以形成一个完整的系统。第三阶段是项目与车辆的周围系统的集成。

集成过程的第一个目标是根据 ASIL 等级和安全需求规范测试符合各项安全要求。第二个目的是验证“系统设计”覆盖的安全要求正确地由整个项实施。项目元件的集成是在从软件硬件集成，系统集成到整车集成系统。 集成测试会在每个阶段的执行来证明系统元件正确交互。根据 ISO26262-5 和 ISO26262-6 完成硬件和软件的开发，然后按照第 8 条款（项目集成和测试）开始进行系统集成。

### 集成测试计划制定

为了证明该系统设计符合功能和技术安全要求，集成测试活动应按照 ISO26262-8:2011，第 9 条款进行。

测试目标如下：

- 1、 功能安全和技术安全要求的正确实现
- 2、 功能特此、精确度和安全机制时序的正确
- 3、 接口一致性的正确实现
- 4、 安全机制诊断和故障覆盖度的有效性

## 5、鲁棒性

集成和测试策略应该被定义,这是基于系统设计规范,功能安全概念,技术的安全概念,项目集成和测试计划,并且证明测试目标充分覆盖,集成和测试策略应涵盖电子/电气元件,并在安全的概念考虑其他技术元素。

为了使系统集成阶段化,按下列规定进行:

- 1、集成和测试计划应细化为软硬件集成和测试;
- 2、项目集成和测试计划应细化到包括系统和车辆级别的集成测试规范。应确保硬件软件验证来解决开放问题;
- 3、系统及整车级别的项目集成和测试计划应考虑车辆之间的接口、子系统(内部和外部有关项)和环境。

在规划整车级的集成和测试时,应考虑在典型和极端的车辆条件和环境下,车辆的正确行为,表4有一部分是足够的。

集成测试测试用例生成方法

Table 4 — Methods for deriving test cases for integration testing

Methods	ASIL			
	A	B	C	D
1a Analysis of requirements 需求分析	++	++	++	++
1b Analysis of external and internal interfaces 外部和内部接口分析	+	++	++	++
1c Generation and analysis of equivalence classes for hardware-software integration 软硬件集成等价类分析	+	+	++	++
1d Analysis of boundary values 边界值分析	+	+	++	++
1e Error guessing based on knowledge or experience 基于经验的错误推测法	+	+	++	++
1f Analysis of functional dependencies 功能依赖分析	+	+	++	++
1g Analysis of common limit conditions, sequences, and sources of dependent failures 公共限制条件,时间序列和故障源分析	+	+	++	++
1h Analysis of environmental conditions and operational use cases 环境条件和操作运行分析	+	++	++	+ 新版 反馈
1i Analysis of field experience 现场经验分析	+	++	++	+

## 软硬件集成测试

### 软硬件集成

根据IS026262-5所开发的软件和根据IS026262-6开发的硬件将被集成到表4的主题所列的测试活动中。

ASILs C 和 D 规定:软硬件接口(HSI)需求应在适当范围进行测试,并考虑到ASIL或相关的人机接口存在的问题。

### 软硬件测试过程中的测试目标和方法

为了检测在硬件和软件整合时,系统设计中存在的系统故障,应通过大量测试方法的应用程序来测试。

1、软硬件级的技术安全需求的正确执行应使用表5给出可行的测试方法来证明。

软硬件级的技术安全需求正确实现

**Table 5 — Correct implementation of technical safety requirements at the hardware-software level**

	Methods	ASIL			
		A	B	C	D
1a	Requirements-based test <sup>a</sup> 基于需求的测试	++	++	++	++
1b	Fault injection test <sup>b</sup> 故障注入测试	+	++	++	++
1c	Back-to-back test <sup>c</sup> 背靠背测试（比较测试）	+	+	++	++

<sup>a</sup> A requirements-based test denotes a test against functional and non-functional requirements.  
<sup>b</sup> A fault injection test uses special means to introduce faults into the test object during runtime. This can be done within the software via a special test interface or specially prepared hardware. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.  
<sup>c</sup> A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.

a 基于需求的测试是指对功能性和非功能性需求的验证。

b 故障注入测试使用特殊的手段在运行时引入到故障到测试对象。这可以在软件中通过一个特殊的测试接口或专门准备的硬件来完成。该方法经常被用于提高安全性的测试覆盖率的要求，因为在正常运行期间的安全机制不会被调用。

c 比较测试比较测试对象和仿真模型在相同的输入下的反应，以检测模型的行为和具体实现之间的差别。

2、软硬件级的功能性能，精度和安全机制的时序的正确性，应使用表 6 给出可行的测试方法证明

软硬件级的功能性能、精确度、安全机制时序的正确性验证方法

**Table 6 — Correct functional performance, accuracy and timing of safety mechanisms at the hardware-software level**

	Methods	ASIL			
		A	B	C	D
1a	Back-to-back test <sup>a</sup> 背靠背测试（比较测试）	+	+	++	++
1b	Performance test <sup>b</sup> 性能测试	+	++	++	++

<sup>a</sup> A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.  
<sup>b</sup> A performance test can verify the performance (e.g. task scheduling, timing, power output) in the context of the whole test object, and can verify the ability of the intended control software to run with the hardware.

a 比较测试比较测试对象和仿真模型在相同的输入下的反应，以检测模型的行为和具体实现之间的差别。

b 性能测试可以在整个测试对象在环境下的性能（例如，任务调度，定时，功率输出），并且可以验证预期的控制软件与硬件上运行的能力。

3、软硬件级的外部和内部接口的一致性和正确性应采用表 7 给出可行的测试方法来证明。

软硬件级的外部和内部接口的一致性和正确性验证方法

**Table 7 — Consistent and correct implementation of external and internal interfaces at the hardware-software level**

	Methods	ASIL			
		A	B	C	D
1a	Test of external interfaces <sup>a</sup> 外部接口测试	+	++	++	++
1b	Test of internal interfaces <sup>a</sup> 内部接口测试	+	++	++	++
1c	Interface consistency check <sup>a</sup> 接口一致性检查	+	++	++	++

<sup>a</sup> Interface tests of the test object include tests of analogue and digital inputs and outputs, boundary tests and equivalence-class tests to completely test the specified interfaces, compatibility, timings and other specified ratings for the test object. Internal interfaces of an ECU can be tested by static tests for the compatibility of software and hardware as well as dynamic tests of Serial Peripheral Interface- (SPI) or Integrated Circuit- (IC) communications or any other interface between elements of an ECU.

a 测试对象的接口测试，包括模拟和数字输入和输出测试，边界测试和等价类测试，以完全测试测试对象的具体接口，兼容性，定时和其它指定的测试项。ECU 的内部接口可以通过软硬件兼容性的静态测试和串行外设接口-（SPI）或集成电路 - （集成电路）通信或 ECU 的元件之间的任何其他接口的动态测试实现。

4、对于故障模式，软硬件级的安全机制的诊断覆盖率的有效性，应使用表 8 给出可行的测试方法证明。

#### 软硬件级的安全机制的诊断覆盖率有效性验证

**Table 8 — Effectiveness of a safety mechanism's diagnostic coverage at the hardware-software level**

Methods	ASIL			
	A	B	C	D
1a Fault injection test <sup>a</sup> 故障注入测试	+	+	++	++
1b Error guessing test <sup>b</sup> 错误猜测测试	+	+	++	++

<sup>a</sup> A fault injection test uses special means to introduce faults into the test object during runtime. This can be done within the software via a special test interface or specially prepared hardware. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

<sup>b</sup> An error guessing test uses expert knowledge and data collected through lessons learned to anticipate errors in the test object. Then a set of tests along with adequate test facilities is designed to check for these errors. Error guessing is an effective method given a tester who has previous experience with similar test objects.

a 故障注入测试使用特殊的手段在运行时引入故障到测试对象。这可以通过在软件中设置一个特殊的测试接口或专门准备的硬件来完成。该方法经常被用于提高安全性的测试覆盖率的要求，因为在正常运行期间的安全机制不会被调用。

b 错误猜测测试采用专业的经验教训积累的知识和数据来预测在测试对象中错误，然后使用足够的测试设备设计一组测试用例，以检查这些错误。错误推测对于专业测试人员是一种有效的方法。

5、软硬件级的鲁棒性依据表 9 来实现

#### 软硬件级的鲁棒性

**Table 9 — Level of robustness at the hardware-software level**

Methods	ASIL			
	A	B	C	D
1a Resource usage test <sup>a</sup> 资源使用率测试	+	+	+	++
1b Stress test <sup>b</sup> 负荷测试	+	+	+	++

<sup>a</sup> A resources usage test can be done statically (e.g. by checking for code sizes or analyzing the code regarding interrupt usage, in order to verify that worst-case scenarios do not run out of resources), or dynamically by runtime monitoring.

<sup>b</sup> A stress test verifies the test object for correct operation under high operational loads or high demands from the environment. Therefore, tests under high loads on the test object, or with exceptional interface loads, or values (bus loads, electrical shocks, etc.), as well as tests with extreme temperatures, humidity or mechanical shocks, can be applied.

A 资源使用率测试可以静态地进行（例如，通过检查代码的大小或分析有关中断使用的代码，以验证最坏的情况不会耗尽资源）也可以通过运行时动态地监控。

b 负荷测试验证测试对象高负荷情况，或从高要求环境下的正确运行情况。因此，高负荷情况下测试测试对象，或用特殊接口的负载，或变量（总线负载，电击等），以及在极端温度，湿度或机械冲击测试中，都可以应用。

## 系统集成和测试

### 系统集成

根据系统的设计，在系统中包含的各个元素应被集成，按照 ISO26262-5 和 ISO26262-6

中指定的系统集成测试。

### 系统测试的测试目标和方法

为检测系统集成过程中的系统故障，依据下面表格中的测试目标和测试方法。

#### 1、 系统级功能安全和技术安全需求验证，表 10

**Table 10 — Correct implementation of functional safety and technical safety requirements at the system level**

Methods	ASIL			
	A	B	C	D
1a Requirement-based test <sup>a</sup>	++	++	++	++
1b Fault injection test <sup>b</sup>	+	+	++	++
1c Back-to-back test <sup>c</sup>	o	+	+	++

<sup>a</sup> A requirements-based test denotes a test against functional and non-functional requirements.

<sup>b</sup> A fault injection test uses special means to introduce faults into the system. This can be done within the system via a special test interface or specially prepared elements or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

<sup>c</sup> A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.

#### 2、 系统级的功能性能，精度和安全机制的时序的正确性，表 11

**Table 11 — Correct functional performance, accuracy and timing of safety mechanisms at the system level**

Methods	ASIL			
	A	B	C	D
1a Back-to-back test <sup>a</sup>	o	+	+	++
1b Performance test <sup>b</sup>	o	+	+	++

<sup>a</sup> A back-to-back test compares the responses of the test object with the responses of a simulation model to the same stimuli, to detect differences between the behaviour of the model and its implementation.

<sup>b</sup> A performance test can verify the performance (e.g. actuator speed or strength, whole system response times) of the safety mechanisms concerning the system.

#### 3、 系统级的外部和内部接口的一致性和正确性，表 12

**Table 12 — Consistent and correct implementation of external and internal interfaces at the system level**

Methods	ASIL			
	A	B	C	D
1a Test of external interfaces <sup>a</sup>	+	++	++	++
1b Test of internal interfaces <sup>a</sup>	+	++	++	++
1c Interface consistency check <sup>a</sup>	o	+	++	++
1d Test of interaction/communication <sup>b</sup> 交互/通信测试	++	++	++	++

<sup>a</sup> An interface test of the system includes tests of analogue and digital inputs and outputs, boundary tests, and equivalence-class tests, to completely test the specified interfaces, compatibility, timings, and other specified characteristics of the system. Internal interfaces of the system can be tested by static tests (e.g. match of plug connectors) as well as by dynamic tests concerning bus communications or any other interface between system elements.

<sup>b</sup> A communication and interaction test includes tests of the communication between the system elements, as well as between the system under test and other vehicle systems during runtime, against the functional and non-functional requirements.

<sup>b</sup> 通信和交互测试包括系统元素之间的测试，以及系统和其他车辆系统之间在运行过程中的测试，针对功能性和非功能性需求测试

#### 4、 系统级的安全机制的诊断覆盖率的有效性，表 13

**Table 13 — Effectiveness of a safety mechanism's failure coverage at the system level**

Methods	ASIL			
	A	B	C	D
1a Fault injection test <sup>a</sup>	+	+	++	++
1b Error guessing test <sup>b</sup>	+	+	++	++
1c Test derived from field experience <sup>c</sup> 根据现场经验测试	o	+	++	++

<sup>a</sup> A fault injection test uses special means to introduce faults into the system. This can be done within the system via a special test interface, specially prepared elements, or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety measures are not invoked.

<sup>b</sup> An error guessing test uses expert knowledge and data collected through lessons learned and field experience to anticipate errors in the system. Then a set of tests along with adequate test facilities is designed to check for these errors. Error guessing is an effective method given a tester who has previous experience with similar systems.

## 5、系统级的鲁棒性，表 14

**Table 14 — Level of robustness at the system level**

Methods	ASIL			
	A	B	C	D
1a Resource usage test <sup>a</sup>	o	+	++	++
1b Stress test <sup>b</sup>	o	+	++	++
1c Test for interference resistance and robustness under certain environmental conditions <sup>c</sup> 在一定条件下的抗干扰性和鲁棒性测试	++	++	++	++

<sup>a</sup> At the system level resource usage testing is usually performed in dynamic environments (e.g. lab cars or prototypes). Issues to test include power consumption and bus load.

<sup>b</sup> A stress test verifies the correct operation of the system under high operational loads or high demands from the environment. Therefore, tests under high loads on the system, or with extreme user inputs or requests from other systems, as well as tests with extreme temperatures, humidity or mechanical shocks, can be applied.

<sup>c</sup> A test for interference resistance and robustness, under certain environmental conditions, is a special case of stress testing. This includes EMC and ESD tests (e.g. see [2], [3]).

C 干扰性和鲁棒性测试，在一定的环境条件下，是压力测试的一种特殊情况。这包括 EMC 和 ESD 测试 整车集成和测试 整车集成 应进行车内通信网络的接口规范和车内电源网络的验证。

## 测试目标和测试方法

在整车集成过程中为了检测系统故障，从需求中产生的测试目标，应通过适当的测试方法来解决，下面的表格来阐述。

## 1、整车级功能安全验证，表 15

**Table 15 — Correct implementation of the functional safety requirements at the vehicle level**

Methods	ASIL			
	A	B	C	D
1a Requirement-based test <sup>a</sup>	++	++	++	++
1b Fault injection test <sup>b</sup>	++	++	++	++
1c Long-term test <sup>c</sup> 长期测试	++	++	++	++
1d User test under real-life conditions <sup>c</sup> 用户真实测试	++	++	++	++

<sup>a</sup> A requirements-based test denotes a test against functional and non-functional requirements.

<sup>b</sup> A fault injection test uses special means to introduce faults into the item. This can be done within the item via a special test interface or specially prepared elements or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety mechanisms are not invoked.

<sup>c</sup> A long-term test and a user test under real-life conditions are similar to tests derived from field experience but use a larger sample size, normal users as testers, and are not bound to prior specified test scenarios, but performed under real-life conditions during everyday life. These tests can have limitations if necessary to ensure the safety of the testers, e.g. with additional safety measures or disabled actuators.

长期测试和用户真实环境是相似的，都是野外体验，但使用更大的样本量，普通人员作为测试人员来测试，并没有指定固定的测试场景，但是在现实生活中日常场景。如果需要的话这些测试可以有限制，以确保测试人员的安全性，如额外的安全措施或禁用驱动器。

## 2、整车级的功能性能，精度和安全机制的时序的正确性，表 16

**Table 16 — Correct functional performance, accuracy and timing of safety mechanisms at the vehicle level**

Methods	ASIL			
	A	B	C	D
1a Performance test <sup>a</sup>	+	+	++	++
1b Long-term test <sup>b</sup>	+	+	++	++
1c User test under real-life conditions <sup>b</sup>	+	+	++	++

<sup>a</sup> A performance test can verify the performance (e.g. fault tolerant time intervals and vehicle controllability in the presence of faults) of the safety mechanisms concerning the item.  
<sup>b</sup> A long-term test and a user test under real-life conditions are similar to tests derived from field experience but use a larger sample size, normal users as testers, and are not bound to prior specified test scenarios, but performed under real-life conditions during everyday life. These tests can have limitations if necessary to ensure the safety of the testers, e.g. with additional safety measures or disabled actuators.

同表 15c, d 项

3、整车级的外部和内部接口的一致性和正确性，表 17

**Table 17 — Consistent and correct implementation of internal and external interfaces at the vehicle level**

Methods	ASIL			
	A	B	C	D
1a Test of external interfaces <sup>a</sup>	o	+	++	++
1b Test of interaction/communication <sup>b</sup>	o	+	++	++

<sup>a</sup> An interface test at the vehicle level tests the interfaces of the vehicle systems for compatibility. This can be done statically by validating value ranges, ratings or geometries as well as dynamically during operation of the whole vehicle.  
<sup>b</sup> A communication and interaction test includes tests of the communication between the systems of the vehicle during runtime against functional and non-functional requirements.

4、整车级的安全机制的诊断覆盖率的有效性，表 18

**Table 18 — Effectiveness of a safety mechanism's failure coverage at the vehicle level**

Methods	ASIL			
	A	B	C	D
1a Fault injection test <sup>a</sup>	o	+	++	++
1b Error guessing test <sup>b</sup>	o	+	++	++
1c Test derived from field experience <sup>c</sup>	o	+	++	++

<sup>a</sup> A fault injection test uses special means to introduce faults into the vehicle. This can be done within the vehicle via a special test interface, specially prepared hardware or communication devices. The method is often used to improve the test coverage of the safety requirements, because during normal operation safety measures are not invoked.  
<sup>b</sup> An error guessing test uses expert knowledge and data collected through lessons learned to anticipate errors in the vehicle. Then a set of tests along with adequate test facilities is designed to check for these errors. Error guessing is an effective method given a tester who has previous experience with similar vehicle applications.  
<sup>c</sup> A test derived from field experience uses the experience and data gathered from the field. Erroneous vehicle behaviour or newly discovered operational situations are analysed and a set of tests is designed to check the vehicle with respect to the new findings.

5、整车级的鲁棒性，表 19

**Table 19 — Level of robustness at the vehicle level**

Methods	ASIL			
	A	B	C	D
1a Resource usage test <sup>a</sup>	o	+	++	++
1b Stress test <sup>b</sup>	o	+	++	++
1c Test for interference resistance and robustness under certain environmental conditions <sup>c</sup>	o	+	++	++
1d Long-term test <sup>d</sup>	o	+	++	++

<sup>a</sup> At the item level, resource usage testing is usually performed in dynamic environments (e.g. lab cars or prototypes). Issues to test include item internal resources, power consumption or limited resources of other vehicle systems.  
<sup>b</sup> A stress test verifies the correct operation of the vehicle under high operational loads or high demands from the environment. Therefore tests under high loads on the vehicle or with extreme user inputs or requests from other systems as well as tests with extreme temperatures, humidity or mechanical shocks can be applied.  
<sup>c</sup> A test for interference resistance and robustness, under certain environmental conditions, is a special case of stress testing. This includes EMC and ESD tests (e.g. see [2], [3]).  
<sup>d</sup> A long-term test and a user test under real-life conditions are similar to tests derived from field experience but use a larger sample size, normal users as testers, and are not bound to prior specified test scenarios, but performed under real-life conditions during everyday life.

## 9、安全确认

第一个目标是提供符合安全目标和适用于该项目的功能安全概念的功能安全性证据。第二个目标是提供证据证明的安全目标是正确的，完整的，在车辆级别可以实现的。

之前的验证活动（如设计验证，安全分析，硬件，软件和项目集成和测试）的目的是为了提供证据，证明每个特定活动的结果符合规定要求。在代表性的汽车集成项目的验证的目的是对预期用途提供恰当的证据和确认安全措施的充足性。

这个集成的项目包括：系统，软件，硬件，其他技术元素，外部措施。验证计划应加以完善，包括如下内容：

- a) 项目配置经过包括校正数据的验证。（？？）
- b) 验证程序，测试用例，驾驶策略等验收标准规范；
- c) 装备和必备的环境条件

项目的安全目标应评估以下目标：

- a) 可控性
- b) 控制随机和系统故障的安全措施的有效性
- c) 外部措施的有效性
- d) 其他技术中元素的有效性

在项目级，随机硬件故障的验证应在项目级应进行为：

- a) 由于 ISO26262-5:2011 第 9 条款所定义的随机硬件故障，安全目标违反评估，根据由 7.4.4.3 规定的目标值
- b) 由于 ISO26262-5:2011 第 8 条款所定义的硬件体系故障，安全目标违反评估，根据由 7.4.4.2 规定的目标值

在车辆级，基于安全目标，功能安全要求和预期用途的确认，应作为执行计划使用

- a) 对每个安全目标，验证程序和测试用例都有详细的通过/失败的标准；
- b) 适用范围。这可能包括诸如配置，环境状况，驾驶情况，操作使用情况等。

下列一组方法应使用：

- a) 使用指定的测试程序，测试案例和通过/失败的标准进行重复的测试；例如：功能和安全要求测试，黑箱测试，模拟，边界条件下测试，故障注入，耐久性测试，压力测试，高加速寿命测试（HALT），模拟外部影响。
- b) 分析，例如 FMEA, FTA, ETA, 仿真
- c) 长期测试，如车辆行驶时间和测试车队；
- d) 在现实生活条件的用户测试，盲目测试，专家测试；
- e) 复审

## 10、功能安全评估

本条款的要求目的是评估已通过实现的项目功能安全。与功能安全责任的组织（如车辆

制造商或供应商，如果后者是负责功能安全）启动功能安全的评估。

## 11、产品发布

本条款的目的是规定项目开发完成后产品标准发布，产品发布确认该项目在车辆级符合功能安全的要求。产品发布确认项目准备后后续生产和经营遵守先决条件批量生产的证据是通过下面提供

- 1) 在开发过程中硬件，软件，系统，物品和车辆级别的验证和确认完成。
- 2) 功能安全的成功整体评估 发布文档作为发布产品的基础，由负责发布的人签署。

产品功能安全发布文档应包含下列信息：

- 1) 负责发布的人的名称和签名；
- 2) 项目发布的版本
- 3) 项目发布的配置
- 4) 相关的参考文档
- 5) 发布日期

## ISO26262-5 硬件级产品开发

### 5、硬件级产品开发初始化

在硬件产品开发的启动阶段的目的是确定和规划在硬件开发的各个子阶段功能安全活动。规定的硬件安全活动计划包含在项目的安全计划中。

在硬件层面必要的活动和产品开发过程包括：---技术安全概念的硬件实现 ---潜在的硬件故障及影响分析 ---与软件开发的协调

与软件开发子阶段相比，这部分的 ISO26262 包含两个条款描述项目的总体硬件结构定量评估。第 8 条款介绍了两个指标来评估该项目的硬件架构和实施安全机制的有效性来面向随机硬件故障。作为第 8 条的补充，，第 9 条描述了两种备选方案，以评估违反安全目标行为的残余风险是否足够低，或者通过使用一个全局性的概率方法或使用割集分析，研究确定违反安全目标的每个硬件元件故障的影响。

根据 ISO26262-2 的安全计划详细说明应包括，确定适当的方法和措施，硬件级别的产品开发活动是一致于在 ISO26262-6 中策划的活动。

项目硬件的开发过程包括方法和工具，与整个硬件开发的各个子阶段相一致，并与系统和软件子阶段相一致，使有关规定保持其在硬件开发过程中的准确性和一致性。

硬件开发的安全生命周期应符合 ISO26262 的规定。硬件单元的复用，或合格硬件单元的使用应在安全活动中进行说明和确认。

### 6、硬件安全需求规范拟定

该条款的第一个目标是规定硬件安全需求，参考技术安全概念和系统安全规范。第二个目标是验证硬件安全需求与技术安全概念和系统安全规范一致。更进一步的目标是详细描述软硬件接口规范 HSI。

技术安全需求分配到软件和硬件，硬件安全需求进一步详细，考虑设计约束，这些设计约束在硬件上的影响。

硬件安全需求规范应该是硬件上的技术安全要求，应该包含如下内容：

- 1) 硬件安全需求和相关安全机制的属性来控制硬件单元的内部失效,这包括内部安全机制覆盖瞬态故障,例如,使用的技术。相关属性可以包括定时器和看门狗检测
- 2) 硬件安全需求和相关安全机制的属性能够承受外部单元的失效。例如在 ECU 外部 失效时,对 ECU 输入开路
- 3) 硬件安全需求和相关安全机制的属性能够匹配别的单元的安全需求
- 4) 硬件安全需求和相关安全机制的属性能够检测和指示内部和外部故障
- 5) 硬件安全需求不指定安全机制 产品硬件的设计验证标准,包括环境条件(温度,振动,电磁干扰,等),具体的操作环境(电源电压,任务历程等)和特定组件的要求。

硬件安全要求应按照 ISO 26262-8:2011 条款 6 和 9 验证,具有以下属性:

- 1) 与技术安全概念、系统设计规范、硬件设计规范一致
- 2) 技术安全需求分配给硬件单元的完整性
- 3) 与相关软件安全需求的一致性
- 4) 正确性和精确性

## 7、硬件设计

这一条款的第一个目标是根据系统设计规范和硬件安全需求设计硬件,第二个目标是验证设计。硬件设计包括硬件架构设计和硬件详细设计,硬件架构设计应表示出所有硬件单元及彼此间的关系,硬件详细设计是指在电路原理图上的设计。

### 7.4.1 硬件架构设计

硬件架构应实现硬件的安全要求,每个硬件单元应根据硬件安全要求实现最高的 ASIL。硬件安全要求和实现之间的可追溯性应保存到的硬件单元的最低层,但不需要到硬件详细设计,ASIL 不会分配到硬件元件。

为了避免高复杂性产生的故障,硬件体系架构设计应通过使用表 1 列出原则来具有以下特征:

- a) 模块化
- b) 粒度适当
- c) 简易性

模块化硬件设计属性

Table 1 — Properties of modular hardware design

Properties	ASIL			
	A	B	C	D
1 Hierarchical design 分层设计 精确定于与安全相关的硬件组件	+	+	+	+
2 Precisely defined interfaces of safety-related hardware components	++	++	++	++
3 Avoidance of unnecessary complexity of interfaces 避免不必要的复杂接口	+	+	+	+
4 Avoidance of unnecessary complexity of hardware components 避免不必要的复杂硬件组件	+	+	+	+
5 Maintainability (service) 可维护性	+	+	++	++
6 Testability <sup>a</sup> 可测试性	+	+	++	++

<sup>a</sup> Testability includes testability during development and operation.

对于安全相关的硬件组件故障,在硬件设计过程中的非功能性条款应考虑以下的影响:温度,振动,湿度,灰尘,电磁干扰,无论是从硬件结构的硬件组件或从其他它的环境的串扰源。

#### 7.4.2 硬件详细设计

- 1、为避免设计缺陷，相关的经验教训应该遵循组织的安全文化 2-5.4.7
- 2、与安全相关的硬件部分失效时应考虑硬件详细设计过程中的非功能性原因，包括以下几方面的影响，如：温度，振动，湿度，灰尘，电磁干扰，噪声系数，无论是从硬件组件的其他部件或它的环境的串扰源。
- 3、硬件部分的操作条件应满足他们的环境和操作限制的规范。
- 4、应该考虑稳健设计原则，稳健设计原理，可以利用基于 QM 方法清单。例如保守的组件规范

#### 7.4.3 安全分析

在硬件设计上找出故障原因和故障影响的安全性分析依据表 2 和 ISO 26262-9:2011 条款 8。安全分析的最初目的是支持的硬件设计规范。随后，安全分析可用于硬件设计验证（见 7.4.4）。

硬件设计安全分析  
Table 2 — Hardware design safety analysis

Methods		ASIL			
		A	B	C	D
1	Deductive analysis <sup>a</sup> 推理分析	o	+	++	++
2	Inductive analysis <sup>b</sup> 归纳分析	++	++	++	++
NOTE The level of detail of the analysis is commensurate with the level of detail of the design. Both methods can, in certain cases, be carried out at different levels of detail.					
<sup>a</sup> A typical deductive analysis method is FTA. 典型的分析方法是FTA					
<sup>b</sup> A typical inductive analysis method is FMEA. 典型的分析方法是FMEA					

本要求适用于安全目标 ASIL (B), C, D。每一个安全相关的硬件部件或零件，在确定的安全目标下，安全分析应考虑以下因素：

- A) 安全故障；
- B) 单点故障或残留故障；
- C) 多点故障（或感知，检测或潜在的）。

在大多数情况下，可以将分析限于双点故障。但多点故障比双点故障点故障两个以上的可以显示更高的技术安全概念（例如当实现冗余安全机制）。

双点故障的识别目的是不需要对每一个可能的两个硬件组合的故障进行系统分析，但是，至少，从安全技术概念考虑到组合，（例如两个故障达到或维持一个安全的状态，一个故障影响安全相关的元素，另一故障影响相应的安全机制）。

#### 单点故障

单点故障，是在一个单元中，未被安全机制覆盖且直接会导致违反安全目标的硬件故障。这项规定应用于安全目标 ASIL (B), C 和 D 的，避免单点故障的有效性安全机制证据，应当提供：

- A) 应提供保持安全状态的安全机制，或安全地切换到安全状态的能力，(特别是恰当的缓解故障的容错时间间隔内的能力)；
- B) 应评估关于残余故障的诊断覆盖率

注意 1：如果诊断测试间隔，加上相关安全机制的故障响应时间，超过有关容错的时间

间隔，可以发生在任何时候（例如，不仅在上电时）的故障不能被认为是有效地覆盖。

**注意 2:** 如果故障是可以描述为仅发生在上电时刻，车辆行驶过程中，然后接通电源后，对故障执行测试。

**注意 3:** 采用诸如 FMEA 或 FTA 的分析方法用来组成基本原理。

**注意 4:** 依据硬件组件和他们相关层的失效模式的知识，评价可以是硬件组件的任一个全局范围的诊断，或更详细的故障模式覆盖的评价。

### 潜在故障

潜在故障，是在多点故障检测时间间隔内不能被安全机制检测出来的也不能被驾驶员识别的多点故障。这项规定应用于安全目标 ASIL (B), C 和 D 的，避免潜在故障的有效性安全机制证据，应当提供：

A) 故障检测，并通知到驾驶员的能力，对潜在故障可接受的多点故障检测的时间间隔内，应以确定哪些故障潜伏，哪些故障是不能潜伏的。

B) 对潜在故障的诊断覆盖率进行评价。

**注意 1** 故障不能被认为是覆盖，如果它的诊断测试间隔，加上相关的安全机构的故障响应时间，比潜在故障相关的多点故障检测时间间隔长，则认为故障不能被覆盖。

**注意 2** 采用诸如 FMEA 或 FTA 的分析方法用来组成基本原理。

### 7.4.4 硬件设计验证

硬件设计应按照 ISO26262-8, 第 9 条，针对硬件安全要求合理性和完整性进行验证。为了实现这一目标，在表 3 中列出的方法，应考虑。

硬件设计验证

Table 3 — Hardware design verification

Methods	ASIL			
	A	B	C	D
1a Hardware design walk-through <sup>a</sup> 硬件设计走读	++	++	○	○
1b Hardware design inspection <sup>a</sup> 硬件设计检查	+	+	++	++
2 Safety analyses 安全分析	In accordance with 7.4.3			
3a Simulation <sup>b</sup> 仿真	○	+	+	+
3b Development by hardware prototyping <sup>b</sup> 硬件原型开发	○	+	+	+

NOTE The scope of this verification review is technical correctness of the hardware design.

<sup>a</sup> Methods 1a and 1b serve as a check of the complete and correct implementation of the hardware safety requirements in the hardware design.

<sup>b</sup> Methods 3a and 3b serve as a check of particular points of the hardware design (e.g. as a fault injection technique) for which analytical methods 1 and 2 are not considered to be sufficient.

注意 此验证审查的范围是硬件设计的技术正确性。

a 方法 1a 和 1b 作为在硬件设计过程中硬件安全要求的完整和正确实施的检查 b 方法 3a 和 3b 作为硬件设计特别点的检查（例如，引入一个故障注入技术），因为分析方法 1 和 2 被认为是不够的。

在硬件设计中，如果发现任何硬件安全要求的实施是不可行的，应当按照 ISO26262-8 的变更管理流程发出变更请求。

### 7.4.5 生产、运行、服务和关闭

如果安全分析已经表明，它们是与安全有关的特殊特性相关的，那么这些特殊特性应被指定。特殊特性的属性应包括：

- a) 生产运行的核查措施
- b) 这些措施的验收标准

## 8、硬件体系指标评估

这一条款的目的是评估由故障处理的指标要求来设计的项目硬件架构。

这一条款描述了两个硬件体系结构的评价指标的有效性，项目的架构来应对随机硬件故障。这些标准和相关的目标值适用于整个项目的硬件。这些指标涉及的仅限于一些项目的安全相关的电气和电子硬件部分随机硬件故障，即那些能够显著有助于违规或实现的安全目标，对于单点，剩余的和这些部件的潜在故障。对于电气硬件单元，只考虑电气故障模式和故障率。硬件架构指标可以反复的在硬件架构设计和硬件详细设计过程中应用。硬件体系结构指标依赖于产品的整个硬件。符合规定的硬件架构度量的目标值满足项目所涉及的每个安全目标。

定义这些硬件架构指标来实现以下目标：

- 客观上应是可评价的：指标是可验证的，精确区别不同的架构;
- 支持最终设计评估 --使 ASIL 依赖于通过/失败标准
- 揭示安全机制的覆盖面，防止硬件架构单点或残留故障风险；
- 揭示安全机制的覆盖面，防止硬件架构潜在故障风险；
- 提出单点故障、残余故障和潜在故障 --确保硬件故障率不确定性的鲁棒性；
- 限于安全相关的元素
- 支持不同元素的语法，例如目标值能分配给供应商元素

### 8.4 要求和建议

这项规定安全机制的安全相关的诊断覆盖率应就剩余的故障和有关潜在故障评估 适用于安全目标 ASIL (B), C 和 D。分析硬件部分的故障率应由以下几个方法确定：

a、使用从工业界认可的工业数据源中得到的硬件故障率数据，例如，公认的业内人士以确定硬件部分的故障率和故障模式分布，包括 IEC/ TR62380, IEC61709, MIL HDBK217 F 通知 2, RIAC HDBK217 此外, UTE C80-811, NPROD95, EN50129:2003, 附件 C, IEC62061:2005, 附件 D, RIAC FMD97 和 MIL HDBK338。

b、使用基于现场返回或测试数据。在这种情况下，失效率必须有足够的置信水平，

c、使用工程上基于定量和定性参数的专家判断方法。专家判断应按照结构化的标准来行使这一判断的依据。这些标准应设置故障率的估算 适用于安全目标 ASIL (B), C 和 D。如果不能提供一个单点故障或潜在故障的计算故障率的充分证据，应提出替代手段（例如添加安全机制来检测和控制这个故障）。

对于每一个安全目标，量化指标值根据 ISO26262-4:2011, 7.4.4.2 规定的“单点故障指标”一样，应根据以下参考目标值来源之一：

A、对硬件架构的指标计算

B、依据表 4

#### 单点故障指标值

Table 4 — Possible source for the derivation of the target “single-point fault metric” value

	ASIL B	ASIL C	ASIL D
Single-point fault metric	≥90 %	≥97 %	≥99 %

对于每一个安全目标，量化指标值根据 ISO26262-4:2011，7.4.4.2 规定的“潜在故障指标”一样，应根据以下参考目标值来源之一：

A、对硬件架构的指标计算

B、依据表 5

Table 5 — Possible source for the derivation of the target “latent-fault metric” value

	ASIL B	ASIL C	ASIL D
Latent-fault metric	≥60 %	≥80 %	≥90 %

## 9、随机硬件故障对安全目标影响评价

对于随机硬件失效率指标的规定：

Table 6 — Possible source for the derivation of the random hardware failure target values

ASIL	Random hardware failure target values
D	<10 <sup>-8</sup> h <sup>-1</sup>
C	<10 <sup>-7</sup> h <sup>-1</sup>
B	<10 <sup>-7</sup> h <sup>-1</sup>

NOTE The quantitative target values described in this table can be tailored as specified in 4.1 to fit specific uses of the item (e.g. if the item is able to violate the safety goal for durations longer than the typical use of a passenger car).

## 10、硬件集成和测试

这一条款的目的是通过测试确保开发的硬件满足硬件安全要求。本节所述活动的目的是集成硬件单元和测试硬件设计以验证其符合适当的 ASIL 硬件安全要求。

硬件集成和测试不同于 ISO26262-8:2011，第 13 条的硬件组件活动的限制，它给出了中级层硬件组件符合 ISO26262 的证据。

10.4.1 硬件集成和测试活动，应当按照 ISO26262-8:2011，第 9 条进行。

10.4.2 硬件集成和测试活动，应符合 ISO26262-4:2011，5.5.5 中给出的项目集成和测试计划。

10.4.3 测试设备应该属于质量监控系统

10.4.4 为了使测试规范适合特定的硬件集成测试，测试用例应使用表 10 中所列的方法进行适当组合得到。

硬件集成测试测试用例生成方法  
Table 10 — Methods for deriving test cases for hardware integration testing

Methods	ASIL			
	A	B	C	D
1a Analysis of requirements 需求分析	++	++	++	++
1b Analysis of internal and external interfaces 内部和外部接口分析	+	++	++	++
1c Generation and analysis of equivalence classes <sup>a</sup> 等价类产生和分析	+	+	++	++
1d Analysis of boundary values <sup>b</sup> 边界值分析	+	+	++	++
1e Knowledge or experience based error guessing <sup>c</sup> 基于知识和经验的错误预测	++	++	++	++
1f Analysis of functional dependencies 功能依赖分析	+	+	++	++
1g Analysis of common limit conditions, sequences and sources of dependent failures 分析常见的限制条件、序列和独立失效源	+	+	++	++
1h Analysis of environmental conditions and operational use cases 分析环境条件和运行用例	+	++	++	++
1i Standards if existing <sup>d</sup> 存在的标准	+	+	+	+
1j Analysis of significant variants <sup>e</sup> 分析重要变量	++	++	++	++

<sup>a</sup> In order to derive the necessary test cases efficiently, analysis of similarities can be conducted.  
<sup>b</sup> For example, values approaching and crossing the boundaries between specified values, and out of range values.  
<sup>c</sup> "Error guessing tests" can be based on data collected through a lessons learned process, or expert judgment, or both. It can be supported by FMEA.  
<sup>d</sup> Existing standards include ISO 16750 and ISO 11452.  
<sup>e</sup> The analysis of significant variants includes worst-case analysis.

- A 为了有效地获得必要的测试用例,可以进行相似性分析
- B 例如, 接近值和指定范围的边界值, 超出值
- C “错误预测测试是基于通过经验教训的过程,或者专家判断,或两者兼而有之, 收集的数据。它可以由 FMEA 来支持
- D 现在的标准包括 ISO16750 和 ISO11452 E 重要变量分析包括最差情况分析

10.4.5 硬件集成和测试活动应验证对硬件的安全要求实现的安全机制的完整性和正确性。测试方法如表 11 所示。

Table 11 — Hardware integration tests to verify the completeness and correctness of the safety mechanisms implementation with respect to the hardware safety requirements

Methods	ASIL			
	A	B	C	D
1 Functional testing <sup>a</sup> 功能测试	++	++	++	++
2 Fault injection testing <sup>b</sup> 故障注入测试	+	+	++	++
3 Electrical testing <sup>c</sup> 电气测试	++	++	++	++

<sup>a</sup> Functional testing aims at verifying that the specified characteristics of the item have been achieved. The item is given input data, which adequately characterises the expected normal operation. The outputs are observed and their response is compared with that given by the specification. Anomalies with respect to the specification and indications of an incomplete specification are analysed.  
<sup>b</sup> Fault injection testing aims at introducing faults in the hardware product and analysing the response. This testing is appropriate whenever a safety mechanism is defined. Model-based fault injection (e.g. fault injection done at the gate-level netlist level) is also applicable, especially when fault injection testing is very difficult to do at the hardware product level. For example, showing the response of safety mechanisms to transient faults inside hardware parts, such as a microcontroller, is very difficult to do with fault insertion at the hardware product level since it would require irradiation tests.  
<sup>c</sup> Electrical testing aims at verifying compliance with hardware safety requirements within the specified (static and dynamic) voltage range.

A 功能测试的目的在于保证项目规定的特征已经达到。为项目提供输入数据,使其表现为预期正常运行状态。观察输出并将响应与规定所给相比较,应对与规范不符合的异常和指示规范不完整的情况进行分析;

B 故障注入测试旨在硬件产品中引入故障并分析其响应。无论安全机制是否定义，本测试方法都是合适的。

C 电气测试旨在规定电压范围内（静态和动态）验证与硬件安全要求的一致性。

#### 10.4.6 硬件集成和测试活动，应当验证硬件抵抗外部应力的健壮性。如表 12 所示

##### 抵抗外部应力下的验证鲁棒性的硬件集成测试

Table 12 — Hardware integration tests to verify robustness and operation under external stresses

Methods	ASIL			
	A	B	C	D
1a Environmental testing with basic functional verification <sup>a</sup> 各种环境下的功能测试	++	++	++	++
1b Expanded functional test <sup>b</sup> 扩展功能测试	o	+	+	++
1c Statistical test <sup>c</sup> 统计测试	o	o	+	++
1d Worst case test <sup>d</sup> 最坏情况测试	o	o	o	+
1e Over limit test <sup>e</sup> 过极限测试	+	+	+	+
1f Mechanical test <sup>f</sup> 机械试验	++	++	++	++
1g Accelerated life test <sup>g</sup> 加速寿命试验	+	+	++	++
1h Mechanical Endurance test <sup>h</sup> 机械耐久性试验	++	++	++	++
1i EMC and ESD test <sup>i</sup> EMC试验	++	++	++	++
1j Chemical test <sup>j</sup> 化学试验	++	++	++	++

# ISO26262-6 软件级产品开发

## 5、软件级产品开发启动

这个子阶段的目标是计划和启动软件开发的功能安全活动。软件开发的启动是计划活动，其中软件开发子阶段及其支持过程（见 ISO26262-8 和 ISO26262-9）是根据项目发展的程度和复杂性决定和计划。软件开发子阶段和支持流程是通过确定适当的方法启动，以符合有关规定和各自的 ASIL。方法是指通过指南和工具，对于每个子阶段支持。

### 5.4 要求和建议

5.4.1 应当规划确定产品开发在软件级别的活动和适当的方法。

5.4.2 产品开发的生命周期在软件层面的制定应当按照 ISO26262-2:2011, 6.4.3.4 进行，并根据图 2 中给出的参考模型。

5.4.3 如果开发可配置的软件，应参考附件 C。

5.4.4 对于一个项目软件的软件开发过程，包括生命周期，方法，语言和工具，应当是与整个软件生命周期的所有子阶段一致的，并与系统和硬件开发阶段兼容，使得所需的数据可以被正确地转换。

5.4.5 对于软件开发的每个子阶段，下面的选择，包括为他们的指南，应进行：

A 方法

B 工具

5.4.6 选择一个合适的模型或编程语言时必须考虑的标准是：

- a) 一个明确的定义；
- b) 对嵌入式实时软件和运行时错误处理的支持；
- c) 对模块化，抽象化和结构化结构的支持

5.4.7 为了支持设计和执行的正确性，设计和建模语言，或编程语言，应符合表 1 中所列的主题。例如 MISRA C 和 MISRA AC AGC 作为编程语言 C 的编码规范

Table 1 — Topics to be covered by modelling and coding guidelines

Topics	ASIL			
	A	B	C	D
1a Enforcement of low complexity <sup>a</sup> 低复杂度执行	++	++	++	++
1b Use of language subsets <sup>b</sup> 语言子集的使用	++	++	++	++
1c Enforcement of strong typing <sup>c</sup> 类型的执行	++	++	++	++
1d Use of defensive implementation techniques 使用防守实现技术	o	+	++	++
1e Use of established design principles 使用既定的设计原则	+	+	+	++
1f Use of unambiguous graphical representation	+	++	++	++
1g Use of style guides 使用设计指南	+	++	++	++
1h Use of naming conventions 使用约定的命名	++	++	++	++

<sup>a</sup> An appropriate compromise of this topic with other methods in this part of ISO 26262 may be required.

<sup>b</sup> The objectives of method 1b are

- Exclusion of ambiguously defined language constructs which may be interpreted differently by different modellers, programmers, code generators or compilers.
- Exclusion of language constructs which from experience easily lead to mistakes, for example assignments in conditions or identical naming of local and global variables.
- Exclusion of language constructs which could result in unhandled run-time errors.

<sup>c</sup> The objective of method 1c is to impose principles of strong typing where these are not inherent in the language.

## 6、软件安全需求规范拟定

这个子阶段的第一个目标是拟定软件安全需求，他们是来自技术安全概念和系统设计规范。第二个目标是细化软硬件接口要求，依据 ISO26262-4:2011，第 7 条。第三个目的是验证该软件的安全要求和硬件的软件接口要求与技术安全概念和系统设计规范一致。

### 6.4 要求和建议

6.4.1 该软件的安全要求应满足每个基于软件的功能，其故障可能违反相应的软件技术安全要求。

例如，功能故障可能导致违反安全规定可以是：

- 使系统达到或保持安全状态的功能
- 相关的检测，显示和处理的安全相关的硬件元件故障的功能；
- 相关的检测，通知和缓解在软件本身的故障功能；

注 1：这些包括在操作系统和应用程序特定的自我监测的软件来检测，表示和处理系统故障的应用程序。

----与车载和非车载测试相关的功能；

注 2：车载的测试可以由系统本身或所述车辆的运行前和运行后阶段的车载网络内的其它系统进行。

注 3：非车载测试指在生产或服务中与安全有关的功能或性能测试。

- 软件生产和服务过程中进行修改的功能；
- 有关性能或时间要求严格的操作功能。

6.4.2 软件安全要求规范应来源于技术安全概念和系统设计，符合 ISO26262-4:2011，7.4.1 和 7.4.5，应考虑以下因素：

- a) 安全要求符合 ISO 26262-8:2011，第 6 条的规定和管理；
- b) 指定的系统和硬件配置；  
配置参数可以包括增益控制，带通频率和时钟分频器。
- c) 有关软硬件接口规范；
- d) 硬件设计规范的有关要求；
- e) 时序约束
- f) 外部接口；
- g) 车辆、系统或硬件的运行模式对软件有影响

6.4.3 如果 ASIL 分解到软件安全技术需求，应遵守 ISO26262-9:2011，第 5 章

6.4.4 在 ISO26262-4:2011，第 7 条发起的软硬件接口规范，详细说明会降至允许正确的控制和硬件的情况，并应说明硬件和软件之间的每一个与安全有关的依赖关系。

6.4.5 如果其它的安全性的要求，除了在 6.4.1 中指定的通过嵌入式软件实施的那些功能，这些功能将被指定，否则引用他们的规范。

6.4.6 软件安全要求的验证，硬件软件接口的规范细化，应按照 ISO26262-8:2011，第 9 条计划。

6.4.7 细化的软硬件接口规范应由负责本系统的硬件和软件开发人员共同进行验

证

6.4.8 软件安全要求和细化的软硬件接口要求应按照 ISO26262-8:2011 第 6 和第 9 章, 进行验证:

- a) 与技术安全要求的合规和一致性;
- b) 符合系统设计;
- c) 与软硬件接口一致。

## 7、软件体系设计

这个阶段的第一个目标是设计软件体系结构以实现软件安全需求, 第二个目标是校验软件体系结构。

### 7.4 要求和建议

7.4.1 为了确保软件体系设计获取正确和有效地必需的信息来进行后续的开发活动, 软件体系结构设计使用的符号应具有适当的抽象水平, 如表 2 中所列的软件体系结构设计说明。

软件架构设计符号

Table 2 — Notations for software architectural design

Methods		ASIL			
		A	B	C	D
1a	Informal notations 非正式符号	++	++	+	+
1b	Semi-formal notations 半正式符号	+	++	++	++
1c	Formal notations 正式符号	+	+	+	+

7.4.2 在软件体系开发的过程中, 应该考虑下列因素:

- a) 软件架构设计的可验证性;
- b) 配置软件的适用性;
- c) 软件单元的设计和实施的可行性;
- d) 软件集成测试中的软件体系结构的可测试性;
- e) 软件体系结构设计的可维护性。

7.4.3 为了避免高复杂性造成的故障, 软件体系结构设计应使用表 3 中列出的原则具有以下性质:

- a) 模块化;
- b) 封装性
- c) 简单化

Table 3 — Principles for software architectural design

Methods	ASIL			
	A	B	C	D
1a Hierarchical structure of software components 软件组件的分层结构	++	++	++	++
1b Restricted size of software components <sup>a</sup> 软件组件的大小限制	++	++	++	++
1c Restricted size of interfaces <sup>a</sup> 接口的大小限制	+	+	+	+
1d High cohesion within each software component <sup>b</sup> 软件组件的高内聚性	+	++	++	++
1e Restricted coupling between software components <sup>a, b, c</sup> 软件组件之间的耦合限制	+	++	++	++
1f Appropriate scheduling properties 合适的调度属性	++	++	++	++
1g Restricted use of interrupts <sup>a, d</sup> 中断使用限制	+	+	+	++

<sup>a</sup> In methods 1b, 1c, 1e and 1g "restricted" means to minimize in balance with other design considerations.  
<sup>b</sup> Methods 1d and 1e can, for example, be achieved by separation of concerns which refers to the ability to identify, encapsulate, and manipulate those parts of software that are relevant to a particular concept, goal, task, or purpose.  
<sup>c</sup> Method 1e addresses the limitation of the external coupling of software components.  
<sup>d</sup> Any interrupts used have to be priority-based.

新版  
反馈

- a 在方法 1B, 1C, 1E 和 1G “限制”是指，设计时考虑尽量减少。
- B 例如，方法 1D 和 1E 可以通过分离识别，封装和操作软件相关的能力实现，指特定概念，目标，任务或目的的那些部分。
- c 方法 1E 解决软件组件的外部耦合的限制
- d 所使用的任何中断都必须基于优先级的

7.4.4 软件体系结构设计应开发到所有软件单元都被识别的水平。

7.4.5 软件体系结构设计应说明：

a) 软件组件的静态设计方面

注 1 静态设计方面包括：

- 软件结构，包括它的等级层次；
- 数据处理的逻辑顺序；
- 数据类型及其特点；
- 软件组件的外部接口；
- 软件的外部接口；
- 约束条件包括架构的范围和外部依赖

注 2：在基于模型的开发的情况下，模型结构是整体模型活动的一个固有部分

b) 软件组件的动态设计方面

注 1 动态设计方面包括：

- 功能和行为；
- 控制流和流程并发；
- 软件组件之间的数据流；
- 在外部接口的数据流；
- 时间限制。

注 2 为了确定动态行为（例如任务，时间片和中断），不同的操作状态（例如，电，关断，正常运行，校准和诊断）应该被考虑。

注 3：为了描述动态行为（例如，任务，时间片和中断），通信关系和他们的分配的

系统硬件（如 CPU 和沟通渠道）应该被指定。

7.4.6 每一个与安全相关的软件组件应被归类为以下之一：

- a) 新开发的
- b) 修改重复利用
- c) 无修改重复利用

7.4.7 新开发的，或经过修改重复使用的安全相关的软件组件，应该符合 ISO26262

7.4.8 那些没有修改重复使用的安全相关的软件组件，则应该符合 ISO26262-8:2011，第 12 条。

7.4.9 该软件的安全要求应分配给软件组件。因此，每个软件组件，应当制定符合任何分配给它的最高 ASIL 要求。

7.4.10 如果嵌入式软件必须实现不同的 ASILs，或安全相关和非安全相关的软件组件，那么所有的嵌入式软件的应按照最高 ASIL，除非软件组件符合的标准与 ISO26262-9:2011，第 6 条共存。

7.4.11 如果软件分区（见附件 D）用于软件组件之间交流，它应确保以下几点：

- a) 共享资源被使用时避免软件分区干扰；
- b) 软件分区是由专用的硬件特性或等效方法来支持（按照 4.3，此规定适用于 ASIL D）；
- c) 实现软件分区的一部分软件开发符合相同或更高的 ASIL 比分配到的软件分区的要求最高 ASIL 更高；
- d) 在软件集成和测试过程中执行软件分区（依据第 10 条）的验证。

7.4.12 按照 ISO26262-9:2011 第 7 条，相关故障的分析应进行，如果软件安全要求的实现依赖于免受干扰或软件组件之间有足够的独立性。

7.4.13 依据 ISO26262-9:2011，第 8 条，在软件架构层面安全分析应进行，以便：

- 识别或确认软件的安全相关部分；
- 支持规范和验证的安全机制的效率。

7.4.14 为了详细说明软件体系结构层次的安全机制，按照 7.4.13 安全分析的结果，如表 4 中所列的错误检测机制应应用。

**Table 4 — Mechanisms for error detection at the software architectural level**

Methods		ASIL			
		A	B	C	D
1a	Range checks of input and output data 输入输出数据范围检测	++	++	++	++
1b	Plausibility check <sup>a</sup> 真实性检测	+	+	+	++
1c	Detection of data errors <sup>b</sup> 数据错误检测	+	+	+	+
1d	External monitoring facility <sup>c</sup> 外部检测设施	o	+	+	++
1e	Control flow monitoring 控制流监控	o	+	++	++
1f	Diverse software design 多样化软件设计	o	o	+	++

<sup>a</sup> Plausibility checks can include using a reference model of the desired behaviour, assertion checks, or comparing signals from different sources.

<sup>b</sup> Types of methods that may be used to detect data errors include error detecting codes and multiple data storage.

<sup>c</sup> An external monitoring facility can be, for example, an ASIC or another software element performing a watchdog function.

A 真实性检查可以包括使用所需的行为的参考模型, assertion 检查, 或比较不同来源的信号。

b 这些方法可用于检测数据中的错误类型, 包括错误检测码和多个数据存储。

c 外部监视设备可以是, 例如, 专用集成电路或其它软件元件执行一个看门狗功能。

7.4.15 本节适用于 ASIL (A), (B), C 和 D, 按照 4.3: 在软件架构层面指定必要的软件的安全机制,, 在安全分析的基础上根据 7.4.13, 如表 5 所列的错误处理机制也应适用。

软件架构层面的错误处理机制

**Table 5 — Mechanisms for error handling at the software architectural level**

	Methods	ASIL			
		A	B	C	D
1a	Static recovery mechanism <sup>a</sup> 静态恢复机制	+	+	+	+
1b	Graceful degradation <sup>b</sup> 故障软化	+	+	++	++
1c	Independent parallel redundancy <sup>c</sup> 独立并联冗余	○	○	+	++
1d	Correcting codes for data 数据纠错码	+	+	+	+

<sup>a</sup> Static recovery mechanisms can include the use of recovery blocks, backward recovery, forward recovery and recovery through repetition. 静态恢复机制包括使用恢复块, 前向恢复, 后向恢复和重复恢复

<sup>b</sup> Graceful degradation at the software level refers to prioritizing functions to minimize the adverse effects of potential failures on functional safety. 软件层的故障软化是指优先减少潜在故障对功能安全的不利影响 独立的并联冗余, 可实现在每个并行路径不同的软件。

<sup>c</sup> Independent parallel redundancy can be realized as dissimilar software in each parallel path

7.4.16 在安全目标下, 软件体系设计未覆盖的新识别的危险, 应写入危险分析和风险评估报告, 按照 ISO26262-8:2011, 第 8 条的变更管理要求来进行。

7.4.17 嵌入式软件的所需的资源, 其中包括:

- a) 执行时间;
- b) 存储空间; 例如, RAM, 栈堆, ROM, 非易失性数据。
- c) 通讯资源。

7.4.18 依据 ISO 26262-8:2011, 第 9 章, 软件体系架构应该被校验, 使用下面表 6 列出的方法, 应具有以下属性:

- a) 遵守软件安全需求;
- b) 兼容目标硬件;
- c) 遵守设计指南。

软件体系设计校验方法

**Table 6 — Methods for the verification of the software architectural design**

	Methods	ASIL			
		A	B	C	D
1a	Walk-through of the design <sup>a</sup> 设计走读	++	+	○	○
1b	Inspection of the design <sup>a</sup> 设计检查	+	++	++	++
1c	Simulation of dynamic parts of the design <sup>b</sup> 设计动态仿真	+	+	+	++
1d	Prototype generation 原型产生	○	○	+	++
1e	Formal verification 正式校验	○	○	+	+
1f	Control flow analysis <sup>c</sup> 控制流分析	+	+	++	++
1g	Data flow analysis <sup>c</sup> 数据流分析	+	+	++	++

<sup>a</sup> In the case of model-based development these methods can be applied to the model.

<sup>b</sup> Method 1c requires the usage of executable models for the dynamic parts of the software architecture.

<sup>c</sup> Control and data flow analysis may be limited to safety-related components and their interfaces.

- a 在基于模型的开发的情况下，这些方法可以应用到该模型
- b 方法 1C 需要软件体系结构可执行模型的动态用法
- c 控制和数据流分析可以限定为与安全有关的部件和接口

## 8、 软件单元设计和实现

这个子阶段的第一个目标是规定软件单元按照软件体系设计及相关的软件安全要求。第二个目的是实现所指定的软件单元。第三个目标是静态验证软件单元设计和实现。

根据软件体系结构设计，开发软件单元的详细设计，详细设计将被实现为一个模型或直接为源代码，根据建模或编码准则。在进行软件单元测试阶段之前，详细设计和开发是静态验证。如果代码是手工开发，在源代码级别实施相关的属性是可以实现。如果基于模型的开发与自动代码生成时，这些属性应用到模型，无需应用到源代码。

为了开发一个软件单元设计，实现软件的安全要求，以及所有非安全的要求。因此，在安全相关和非安全相关的要求都在这个子阶段的过程中处理。

软件单元的实现包括源代码的生成和编译成目标代码。

### 8.4 要求和建议

- 8.4.1 本阶段应符合软件单元安全相关的要求。
- 8.4.2 为确保该软件单元设计允许后续的开发活动获取正确和有效地进行所必需的信息，该软件单元的设计应采用表 7 所列的符号说明。

软件单元设计符号

Table 7 — Notations for software unit design

Methods		ASIL			
		A	B	C	D
1a	Natural language 自然语言	++	++	++	++
1b	Informal notations 非正式符号	++	++	+	+
1c	Semi-formal notations 半正式符号	+	++	++	++
1d	Formal notations 正式符号	+	+	+	+

在基于模型的开发与自动生成代码的情况下，该方法为代表的软件单元设计作为基础的代码生成的模型。

8.4.3 软件单元的规范应描述功能行为和内部设计，以达到必要的细节实施水平。例如，内部的设计可以包括使用寄存器和数据存储的限制。

8.4.4 软件单元源代码级的设计与实现应采用如表 8 所列的的设计原理，达到以下属性：

- a) 软件单元的子程序和函数的正确执行顺序基于软件架构设计
- b) 软件单元之间的接口的一致性
- c) 软件单元内部的数据流和控制流的正确性；
- d) 简约化
- e) 可读性和可理解性；
- f) 鲁棒性
- g) 软件修改的适用性
- h) 可测试性

**Table 8 — Design principles for software unit design and implementation**

Methods 在子程序和函数中仅有一个入口和一个出口点	ASIL			
	A	B	C	D
1a One entry and one exit point in subprograms and functions <sup>a</sup>	++	++	++	++
1b No dynamic objects or variables, or else online test during their creation <sup>ab</sup> 创作过程中没有动态对象或者变量，或动态测试	+	++	++	++
1c Initialization of variables 变量初始化	++	++	++	++
1d No multiple use of variable names <sup>a</sup> 没有多次使用的变量名	+	++	++	++
1e Avoid global variables or else justify their usage <sup>a</sup> 避免全局变量或其他用法	+	+	++	++
1f Limited use of pointers <sup>a</sup> 限制使用指针	○	+	+	++
1g No implicit type conversions <sup>ab</sup> 没有隐式类型转换	+	++	++	++
1h No hidden data flow or control flow <sup>c</sup> 没有任何隐藏的数据流或控制流	+	++	++	++
1i No unconditional jumps <sup>abc</sup> 没有无条件跳转	++	++	++	++
1j No recursions 没有递归	+	+	++	++

a Methods 1a, 1b, 1d, 1e, 1f, 1g and 1i may not be applicable for graphical modelling notations used in model-based development 方法1a, 1b, 1d, 1e, 1f, 1g和1i未必适用于基于模型的开发中使用的图形化建模符号。

b Methods 1g and 1i are not applicable in assembler programming. 方法1g和1i并不适用于汇编语言程序设计

c Methods 1h and 1i reduce the potential for modelling data flow and control flow through jumps or global variables.

方法1h和1i减少潜在的通过跳转或全局变量建模数据流和控制流

新版  
反馈



对于 C 语言, MISRA C 涵盖了许多表 8 所列的方法。

8.4.5 软件单元的设计与实施应按照 ISO26262-8:2011 第 9 条进行验证, 并通过使用表 9 列出的验证方法来证明:

- a) 遵守软硬件接口规范 (根据 ISO26262-5:2011, 6.4.10) ;
- b) 软件安全要求 (按照 7.4.9) 分配给软件单元的实施的可追溯性;
- c) 源代码和设计规范的一致性
- d) 源代码与编码指南一致性;
- e) 软件单元实现与目标硬件的兼容性。

软件单元设计实现的验证方法

**Table 9 — Methods for the verification of software unit design and implementation**

Methods	ASIL			
	A	B	C	D
1a Walk-through <sup>a</sup> 走读	++	+	○	○
1b Inspection <sup>a</sup> 检查	+	++	++	++
1c Semi-formal verification 半正式验证	+	+	++	++
1d Formal verification 正式验证	○	○	+	+
1e Control flow analysis <sup>bc</sup> 控制流分析	+	+	++	++
1f Data flow analysis <sup>bc</sup> 数据流分析	+	+	++	++
1g Static code analysis 静态代码分析	+	++	++	++
1h Semantic code analysis <sup>d</sup> 代码分析	+	+	+	+

a In the case of model-based software development the software unit specification design and implementation can be verified at the model level. 在基于模型的软件开发的情况下, 软件单元规格设计和实现可以在模型级进行验证

b Methods 1e and 1f can be applied at the source code level. These methods are applicable both to manual code development and to model-based development. 方法1e和1f可以在源代码级应用。这些方法既适用手动代码开发又适合于模型的开发

c Methods 1e and 1f can be part of methods 1d, 1g or 1h. 方法1e和1f可以是方法1d, 1g和1h的一部分。

方法1h通过使用变量的可能值的抽象表示用于源代码的数学分析。为此, 没有必要来翻译和执行源代码

d Method 1h is used for mathematical analysis of source code by use of an abstract representation of possible values for the variables. For this it is not necessary to translate and execute the source code.

表 9 列出只有静态验证技术。动态验证技术 (如测试技术) 是包括在表 10, 11 和 12。

## 9、软件单元测试

这个子阶段的目标是证明软件单元实现软件单元的设计规范和不含有不需要的功能。依据软件设计规范建立软件单元设计的测试流程，并依照流程来执行。

9.4.1 该条款要求应符合如果软件单元是与安全相关的。

9.4.2 软件单元测试必须按照 ISO26262-8:2011 第 9 条计划，规定和执行。

9.4.3 在表 10 中列出的软件单元测试方法应适用于验证软件单元的实现：

- a) 遵守软件单元设计规范
- b) 遵守软硬件接口规范
- c) 指定的功能
- d) 不存在非计划的功能
- e) 鲁棒性
- f) 足够的资源支持功能

Table 10 — Methods for software unit testing

Methods	ASIL			
	A	B	C	D
1a Requirements-based test <sup>a</sup> 基于需求的测试	++	++	++	++
1b Interface test 接口测试	++	++	++	++
1c Fault injection test <sup>b</sup> 故障注入测试	+	+	+	++
1d Resource usage test <sup>c</sup> 资源利用测试	+	+	+	++
1e Back-to-back comparison test between model and code, if applicable <sup>d</sup> 如果可能，模型和代码之间的背靠背测试	+	+	++	++

<sup>a</sup> The software requirements at the unit level are the basis for this requirements-based test.

<sup>b</sup> This includes injection of arbitrary faults (e.g. by corrupting values of variables, by introducing code mutations, or by corrupting values of CPU registers).

<sup>c</sup> Some aspects of the resource usage test can only be evaluated properly when the software unit tests are executed on the target hardware or if the emulator for the target processor supports resource usage tests.

<sup>d</sup> This method requires a model that can simulate the functionality of the software units. Here, the model and code are stimulated in the same way and results compared with each other.

- a 在单元级别的软件测试是基于需求的测试
- b 这包括注射任意故障（通过变量破坏，例如，通过引入代码突变，或通过损坏 CPU 寄存器的值）。
- c 当软件单元测试是执行在目标硬件或模拟目标处理器，测试的资源使用情况的某些方面才能正确评估
- d 此方法需要一个模型，它可以模拟软件单元的功能。这里，模型和代码被运行以同样的方式和结果进行相互比较。

9.4.4 为了使适当的测试用例按照 9.4.3 软件单元测试规范，测试用例应采用表 11 中列出的方法得出。

**Table 11 — Methods for deriving test cases for software unit testing**

Methods		ASIL			
		A	B	C	D
1a	Analysis of requirements 需求分析	++	++	++	++
1b	Generation and analysis of equivalence classes <sup>a</sup> 等价类生成和分析	+	++	++	++
1c	Analysis of boundary values <sup>b</sup> 边界值分析	+	++	++	++
1d	Error guessing <sup>c</sup> 错误预测	+	+	+	+

<sup>a</sup> Equivalence classes can be identified based on the division of inputs and outputs, such that a representative test value can be selected for each class.

<sup>b</sup> This method applies to interfaces, values approaching and crossing the boundaries and out of range values.

<sup>c</sup> Error guessing tests can be based on data collected through a "lessons learned" process and expert judgment.

a 等价类可以基于对输入和输出，选择一个代表性的试验值，实验值识别每个类别的边界。

b 此方法适用于接口，接近值，范围值和超过界限值。

c 错误预测测试是基于通过“经验教训”的过程和专家判断收集到的数据

9.4.5 为了评估测试用例的完整性，并证明没有额外功能，在软件单元级别要求的覆盖范围应确定，结构范围应按照表 12 中列出的指标进行测定。如果实现的结构范围被视为是不够的，那么额外的测试用例应指定或提供理由。

**Table 12 — Structural coverage metrics at the software unit level**

Methods		ASIL			
		A	B	C	D
1a	Statement coverage 语句覆盖	++	++	+	+
1b	Branch coverage 分支覆盖	+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage) 修正条件/判定覆盖	+	+	+	++

注 1：所述的结构覆盖可以通过使用适当的软件工具来确定。

注 2：在基于模型的开发的情况下，可以在模型层使用的模型类似结构覆盖度量进行结构覆盖分析。

注 3：如果检测的代码用于确定覆盖度的等级，必须表明该仪器具有不受试验结果影响。这可以通过没有仪表代码的重复测试来完成。

9.4.6 软件单元测试的测试环境应尽可能与目标环境密切对应。如果软件单元测试不在目标环境中进行，源代码和目标代码中的差异，以及测试环境和目标环境之间的差异，应在指定目标环境中额外的随后的测试阶段加以分析。

注 1：在测试环境和目标环境之间的差异可以发生在源代码或目标代码，例如，由于不同的位宽度的数据字与该处理器的地址字。

注 2：根据测试的范围内，应使用适当的软件执行单元的测试环境（如目标处理器，处理器仿真器或开发系统）。

注 3：软件单元测试可在不同的环境中被执行，例如：

----模型在环测试

----软件在环测试

----处理器在环测试

## ----硬件在环测试

注 4: 对于基于模型的开发, 软件单元测试, 可以在模型级进行, 在模型和对象代码之间的背对背比较测试之后。背对背的对比测试用于确保该模型的行为对于测试目标等同于自动生成的代码。

## 10、软件集成和测试

这部分的第一个目的是集成软件元素, 第二个目的是要证明, 软件体系结构设计是由嵌入式软件实现。

### 10.4 需求和建议

10.4.1 软件集成的计划应说明整合各个软件分层单元到软件组件的步骤, 直到嵌入式软件完全集成, 并应考虑:

- a) 相关的软件集成的功能依赖关系;
- b) 软件集成和软硬件整合之间的依赖关系。

注: 对于基于模型的开发, 该软件集成, 可以在模型层和随后的自动代码生成集成的模型替换为集成。

10.4.2 软件集成测试应根据 ISO26262-8:2011, 第 9 条计划, 规定并执行。

10.4.3 在表 13 中列出的软件集成测试方法应应用来证明软件组件和嵌入式软件的实现

- a) 遵守第 7 条的软件架构设计;
- b) 符合 ISO26262-4:2011, 第 7 条的软硬件接口规范
- c) 规定的功能
- d) 鲁棒性
- e) 充足的资源来支持功能

Table 13 — Methods for software integration testing

Methods	ASIL			
	A	B	C	D
1a Requirements-based test <sup>a</sup> 基于需求的测试	++	++	++	++
1b Interface test 接口测试	++	++	++	++
1c Fault injection test <sup>b</sup> 故障注入测试	+	+	++	++
1d Resource usage test <sup>cd</sup> 资源使用测试 模型和代码间的背对背比较测试	+	+	+	++
1e Back-to-back comparison test between model and code, if applicable <sup>e</sup>	+	+	++	++

<sup>a</sup> The software requirements at the architectural level are the basis for this requirements-based test.

<sup>b</sup> This includes injection of arbitrary faults in order to test safety mechanisms (e.g. by corrupting software or hardware components).

<sup>c</sup> To ensure the fulfilment of requirements influenced by the hardware architectural design with sufficient tolerance, properties such as average and maximum processor performance, minimum or maximum execution times, storage usage (e.g. RAM for stack and heap, ROM for program and data) and the bandwidth of communication links (e.g. data buses) have to be determined.

<sup>d</sup> Some aspects of the resource usage test can only be evaluated properly when the software integration tests are executed on the target hardware or if the emulator for the target processor supports resource usage tests.

<sup>e</sup> This method requires a model that can simulate the functionality of the software components. Here, the model and code stimulated in the same way and results compared with each other.

- a 在单元级别的软件测试是基于需求的测试
- b 这包括注入任意故障 (通过变量破坏, 例如, 通过引入代码突变, 或通过损坏 CPU 寄存器的值)。

c 为确保硬件体系结构设计的足够的容量满足需求，如平均和最大处理器性能，最小或最大执行时间，存储使用情况（例如 RAM 用于堆栈和堆，ROM 中的程序和数据）和带宽性能通信链路（如数据总线）必须确定。

d 当软件单元测试是执行在目标硬件或模拟目标处理器，测试的资源使用情况的某些方面才能正确评估

e 此方法需要一个模型，它可以模拟软件单元的功能。这里，模型和代码被运行以同样的方式和结果进行相互比较。

10.4.4 为了使适当的测试用例按照 10.4.3 软件集成测试规范，测试用例应采用表 14 中列出的方法得出。

软件集成测试测试用例生成方法

**Table 14 — Methods for deriving test cases for software integration testing**

Methods	ASIL			
	A	B	C	D
1a Analysis of requirements 需求分析	++	++	++	++
1b Generation and analysis of equivalence classes <sup>a</sup> 等价类产生和分析	+	++	++	++
1c Analysis of boundary values <sup>b</sup> 边界值分析	+	++	++	++
1d Error guessing <sup>c</sup> 错误预测	+	+	+	+

<sup>a</sup> Equivalence classes can be identified based on the division of inputs and outputs, such that a representative test value can be selected for each class.

<sup>b</sup> This method applies to parameters or variables, values approaching and crossing the boundaries and out of range values.

<sup>c</sup> Error guessing tests can be based on data collected through a "lessons learned" process and expert judgment.

a 等价类可以基于对输入和输出，选择一个代表性的试验值，实验值识别每个类别的边界。

b 此方法适用于变量参数，接近值，范围值和超过界限值。

c 错误预测测试是基于通过“经验教训”的过程和专家判断收集到的数据

10.4.5 为了评估测试用例的完整性，并证明没有额外功能，在软件集成级别要求的覆盖范围应确定。如果实现的结构范围被视为是不够的，那么额外的测试用例应指定或提供理由。

10.4.6 为了评估测试用例的完整性，并证明没有额外功能，在软件集成级别要求的覆盖范围应确定，结构范围应按照表 15 中列出的指标进行测定。如果实现的结构范围被视为是不够的，那么额外的测试用例应指定或提供理由。

软件体系架构覆盖矩阵

**Table 15 — Structural coverage metrics at the software architectural level**

Methods	ASIL			
	A	B	C	D
1a Function coverage <sup>a</sup> 功能覆盖	+	+	++	++
1b Call coverage <sup>b</sup> 调用覆盖	+	+	++	++

<sup>a</sup> Method 1a refers to the percentage of executed software functions. This evidence can be achieved by an appropriate software integration strategy. 方法1a中是指所执行的软件功能的百分比。这方面的证据可以通过适当的软件整合战略来实现

<sup>b</sup> Method 1b refers to the percentage of executed software function calls. 方法1B是指所执行的软件功能的回调的百分比。

注 1：所述的结构覆盖可以通过使用适当的软件工具来确定。

注 2：在基于模型的开发的情况下，可以在模型层使用的模型类似结构覆盖度量进行结构覆盖分析。

10.4.7 它应按照 ISO26262-4:2011, 第 11 条, 验证嵌入式软件作为产品发布的一部分, 包含的所有特定功能, 并且不包含其他未指定的功能, 如果这些功能不损害遵守软件安全要求。

10.4.8 软件集成测试的测试环境应尽可能与目标环境密切对应。如果软件集成测试不在目标环境中进行, 源代码和目标代码中的差异, 以及测试环境和目标环境之间的差异, 应在指定目标环境中额外的随后的测试阶段加以分析。

注 1: 在测试环境和目标环境之间的差异可以发生在源代码或目标代码, 例如, 由于不同的位宽度的数据字与该处理器的地址字。

注 2: 根据测试的范围内, 应使用适当的软件执行单元的测试环境 (如目标处理器, 处理器仿真器或开发系统)。

注 3: 软件集成测试可在不同的环境中被执行, 例如:

----模型在环测试

----软件在环测试

----处理器在环测试

----硬件在环测试

## 11、软件安全需求验证

本章节的目的是验证嵌入式软件完成软件安全需求。

### 11.4 需求和建议

11.4.1 软件安全要求的验证应按照 ISO26262-8:2011, 第 9 条计划, 指定和执行。

11.4.2 为了验证嵌入式软件满足软件安全要求, 测试应进行在表 16 中列出的测试环境。

软件安全需求验证测试环境

Table 16 — Test environments for conducting the software safety requirements verification

	Methods	ASIL			
		A	B	C	D
1a	Hardware-in-the-loop 硬件在环测试	+	+	++	++
1b	Electronic control unit network environments <sup>a</sup> ECU网络环境	++	++	++	++
1c	Vehicles 车辆	++	++	++	++

<sup>a</sup> Examples include test benches partially or fully integrating the electrical systems of a vehicle, "lab-cars" or "mule" vehicles, and "rest of the bus" simulations.

a 例如包括测试台部分或完全集成的车辆的电气系统, 实验室汽车或“mule” 的车辆, 并且模拟“总线休眠”

11.4.3 软件安全需求实施的试验应在目标硬件系统上执行。

11.4.4 软件安全需求验证结果应就进行评估:

- 符合预期的结果;
- 覆盖的软件安全需求
- 通过或失败的标准。

## ISO26262-7 生产运行

### 5、生产

## **5.1 目标**

本条款的第一个目标是开发和维护生产过程中的被安装在道路车辆上的与安全相关的元素或项目。第二个目的是在生产过程中由相关制造商或人员负责该进程来实现功能安全(车辆制造商, 供应商, 二级供应商, 组织等)。

## **5.2 概述**

### **5.4 建议和要求**

#### **5.4.1 生产计划**

5.4.1.1 生产流程应通过项目评估来计划，并通过考虑以下因素：

- a) 生产需求
- b) 存储、运输和硬件单元的处理条件，如，硬件元素的允许存储时间
- c) 经产品发布文档认可的配置，
- d) 以前产品生产计划中积累的经验
- e) 生产过程中的适用性，生产资料，工具和测试设备，与安全有关的特殊特性
- f) 人员能力。

5.4.1.2 生产计划应说明生产步骤，顺序和实现该项目，系统或元件的功能安全要求的方法。它应包括：

- a) 生产工艺流程和说明
- b) 生产工具和方法
- c) 可追溯性实现，如标签
- d) 若可以，适用于硬件开发的硬件单元专用措施的实施应符合 ISO26262-5:2011，  
9.4.2.4 中规定。

5.4.1.3 作为生产过程的一部分，流程应被定义，以确保正确的嵌入式软件和相关的校准数据被写入的 ECU。

例 1，使用校验和，从而使装载的可执行文件和配置数据的校验和与用于这个特定的模型和车辆配置的正确的校验和进行比较。

例 2，从写入的 ECU 软件读回的零件号与从材料规定的零件号比较，以及从特定车辆回读的已加载的校准数据和从材料规定的特定车辆的校准数据比较。

5.4.1.4 当开发生产控制计划，该控件的描述和项目的标准，系统或元件以及与安全相关的特殊特性应予以考虑。

5.4.1.5 顺序和控制步骤方法应在生产控制计划描述，再加上必要的测试设备，工具和测试标准。

5.4.1.6 合理的可预见的过程故障及其对功能安全的影响应该被识别，并实施适当的措施来解决相关过程失败。

5.4.1.7 系统硬件或软件开发层的安全要求的可生产性在生产计划中规定，并指定专门的人负责开发(见 ISO 26262 - 4, ISO 26262 - 5 和 ISO 26262 - 6)。

5.4.1.8 如果系统或元素在生产过程中需要改变，管理过程中所描述的条款 ISO 26262 - 8:2011, 应当遵守。

#### **5.4.2 试制批量产品**

5.4.2.1 试制过程和控制措施应与目标生产工艺对应。

5.4.2.2 试制过程和目标生产工艺之间的差异应进行分析以确定生产过程的哪个部分可以在试制过程哪个部分在目标生产工艺中，并评估两个过程中重要的部分。

注：如果预生产过程中等于目标的生产过程中，评估的结果根据  
ISO26262-2:2011 6.4.9.4 可以在功能安全评估时使用（如生产过程能力的证明）。

例如

偏差涉及生产速度，生产顺序和方法和或控制步骤，以及生产的必要手段，测试设备和工具。

#### 5.4.3 生产

5.4.3.1 生产过程及其控制措施应按计划实施和保持。

注：生产人员适当的培训是该实现的一部分。

5.4.3.2 生产过程中的失败（包括其授权范围内的与安全相关的特殊特性的偏差）和对功能安全潜在影响，应分析，应采取相应的措施，保持功能安全能力应当得到验证。例如，这些措施可以包括执行进一步的控制措施，分类，处理和交换元素。

5.4.3.3 以下关于功能安全的能力应当评估和维护

- a) 生产流程
- b) 生产方法
- c) 工具和测试设备

注 1：过程能力可以由周期性过程审核或执行流程步骤的每个人周期性资格认证来证明。

注 2：过程能力覆盖保持与安全相关的特殊特性的能力。

5.4.3.4 测试设备应当受到监控。

5.4.3.5 执行的控制应按照相关的生产控制计划，控制报告应当包括下列信息：控制日期，控制对象的识别和控制结果。

注 1 对于手动控制，控制对象和控制结果的识别就足够了。

注 2 控制对象的识别可以是车辆识别号码或车辆级的控制措施生产号或零件号或控制组件序列号

注 3 控制结果可以由一个状态（如通过或失败）或一组数据边界条件的评价。

5.4.3.6 在生产中发布的版本，只有经过批准的配置才可以被生产，除非发布的产品文档的偏差是由相应负责人的授权。该生产文件版本过后应该更新授权的偏差。

5.4.3.7 在生产阶段出现的生产过程变更应当遵守条款 5。

## 6、运行、服务（保养和维护）和关闭

### 6.1 目标

这一条款的目的是规定项目，系统或元素的客户信息、维修说明以及拆卸说明，以维护汽车生命周期的功能安全。

### 6.2 概述

这一条款为开发维修说明和用户信息，包括用户需求手册和规划，执行和监控的维护工

作，考虑到该项目的相关安全特点。

#### 6.4 要求和建议

##### 6.4.1 计划

6.4.1.1 操作，维修和保养过程的计划中应项目的评估，并考虑以下情况：

- 1) 维护和修理的要求；
- 2) 要求应提供给用户的信息，确保车辆的安全运行；
- 3) 警告和降级概念
- 4) 现场数据收集和分析的措施；
- 5) 储存，运输和处理硬件元素的条件；
- 6) 经批准的生产文档版本配置；
- 7) 涉及到的人员能力

6.4.1.2 维护计划应当描述维护步骤或活动顺序和方法、维护间隔和维护的必要手段和工具。

6.4.1.3 维修计划和维修指令应当描述如下：

- 1) 工作步骤、规程、诊断程序和方法；
- 2) 维护工具和手段；
- 3) 控制步骤的顺序和方法，用于验证安全控制标准的特殊特性；
- 4) 项目，系统或元素配置，包括追溯性措施；
- 5) 项目、系统或元素允许的停止，和必要的修改；
- 6) 允许停止和修改的驾驶信息
- 7) 替换零件条款

6.4.1.4 用户信息，包括用户手册，应当提供相关说明和关于物品的正确使用使用警告，以及以下信息：

- 1) 相关的功能描述（即预期使用、状态信息或用户交互）和操作模式；
- 2) 客户行为的描述确保所需的可控制性的失效显示，表示警告和降级的信号；
- 3) 从客户中的警告和降级概念所表示的故障情况下预期的维修活动的描述；
- 4) 关于已知的危害作用的警告结果与第三方产品；
- 5) 关于安全的创新功能项警告，可能导致司机的误解或误用。

6.4.1.5 分解说明书应说明活动之前被应用措施拆装，并且需要防止拆卸，处理过程的违反的安全目标的分解车辆，物品或其元素。

6.4.1.6 系统硬件或软件级安全需求在运行、维护和分解计划中规定，并指定相关负责人负责开发（见 ISO 26262 – 4, ISO 26262 – 5 和 ISO 26262 – 6）。

#### 6.4.2 运行、维护（维修）和分解

6.4.2.1 现场监控过程，涉及到该项目功能安全事故应按计划实施符合 IS026262-2:2011, 7.4.2.4, 以便：

- 1) 提供现场数据，应当分析来检测任何存在的功能性安全问题，如果发现，触发行动，解决这些问题，
- 2) 提供所需的证据证明在使用参数，根据 ISO 26262 – 8:2011 条款 14。

6.4.2.2 项目系统或其元素的维护、维修和拆除,应该被管理和记录按照维修计划,和保养、维修指令。

6.4.2.3 零部件的供应和他们的存储和运输根据 6.4.1.3 必须按计划执行。

6.4.2.4 如果后续生产变更是由操作,现场监测, 维护、修理或分解, 按照 ISO 26262 – 8:2011 第八条款变更管理过程, 应当遵守。

## ISO26262-8 支持过程

### 5、 分布式开发接口

#### 5.1 目标

这一条款的目的是描述程序, 并在分布式开发的项目和元素中分配相关的责任。

#### 5.4 要求和建议

##### 5.4.1 要求应用

5.4.1.1 第 5 条的规定应当按照适用于 ISO26262 开发每一个项目和元素,除了现成的硬件部件,下列情况适用:

1) 没有特定的硬件安全要求分配给硬件部分;

2) 现成的硬件部分具有基于全球质量标准(例如电子元器件 AEC 设计标准)完善的过程 和对目标应用程序覆盖范围的参数。

##### 5.4.1.2 客户-供应商关系的要求应适用于所有客户-供应商

注 1 这包括分包了顶级供应商,分包的分包商等

注 2 内部供应商可以相同的方式管理外部供应商

##### 5.4.2 供应商选择标准

5.4.2.1 供应商选择标准应根据 ISO 26262 包括评估供应商的开发和生产项目的复杂性和 ASIL 等级的能力。

注 供应商选择标准包括:

◆ 供应商的质量管理体系的证据

◆ 供应商过去的性能和质量;

◆ 确认供应商的能力的功能安全作为供应商的投标的一部分

◆ 根据 ISO 26262 – 2:2011 6.4.9 以前的安全评估的结果

◆ 考虑影响的功能安全的从开发、生产、质量和物流部门的汽车制造商部门

##### 5.4.2.2 从客户到供应商候选人申请的 RFQ 应当包括:

1) 符合 ISO 26262 的正式请求,

2) 项目或功能规范定义的元素,

3) 安全目标、功能安全要求或技术安全要求,包括他各自己经可用的, ASIL 这取决于供应商报价

##### 5.4.3 分布式开发的启动和规划

5.4.3.1 客户和供应商应指定一个 DIA (开放源码的流程图软件) 包括下列:

1) 任命客户和供应商的安全管理人员

2) 依照 ISO 26262 – 2:2011 6.4.5 联合定制的安全生命周期

- 3) 定义由供应商执行的活动和过程
  - 4) 交付的信息和工作产品,
  - 5) 负责该活动的所有负责人
- 6) 目标值的沟通, 源自于系统级目标, 各相关方确认满足单点故障指标的目标值和潜伏性故障指标的目标值, 根据硬件架构的评估指标和评估安全的目标避免硬件随机故障(见 ISO 26262 – 5),
- 7) 支持过程和工具, 包括接口, 保证客户和供应商之间的兼容性。

5.4.3.2 如果供应商进行危险分析和风险评估, 危险分析和风险评估应提供给客户确认。

5.4.3.3 负责项目开发方应根据 ISO26262-3 建立功能安全概念。功能安全要求应在客户供应商之间一致。

#### 5.4.4 分布式开发的执行

5.4.4.1 供应商应向客户报告每个不符合的风险问题, 比如项目计划、安全计划, 集成和测试计划根据 ISO26262-4 或软件验证计划按照 ISO26262-6, DIA 或其他规定。

5.4.4.2 供应商应向客户报告每个异常发生在开发活动区域或其分包商的责任。

5.4.4.3 供应商应确定是否符合每个安全要求。如果没有, 安全概念应当重新审查, 如果有必要, 修改满足安全要求。

5.4.4.4 每个可能影响项目或计划的活动符合 ISO 26262 的安全性的变更根据条款 8 应向另一方获得支持。

5.4.4.5 双方应该考虑根据 ISO26262-2:2011, 5.4.2.7 在类似的项目中以前的经验发展来派生安全要求满足当前的开发。

5.4.4.6 供应商应向客户的安全经理对安全计划中定义的任务和里程碑取得的进展进行报告。报告的格式和交付日期应在供应商和客户之间的达成一致。

例如, 定期间隔, 或当里程碑指定的计划达到, 客户检查发布供应商编制的质量管理报告。

5.4.4.7 双方(供应商或客户)应达成一致履行符合 ISO26262 - 4 的安全验证。

注: 如果供应商执行集成和验证, 由供应商提供所需的能力和资源非常重要, 因为安全验证需要集成的车辆(见 ISO26262-4)。

5.4.4.8 这个需求适用于 ASIL D 按照 4.3。客户应该执行其他功能安全审计在供应商的允许任何适当的时候。

#### 5.4.5 供应商要求下的功能安全评估

5.4.5.1 按照 4.3 本要求适用于 ASILs (B), C, D。在到达里程碑定义时间点一个或多个功能安全评估应当执行, 这些评估应包括项目开发的每个阶段。对于项目的复杂性和 ASILs 的安全目标, 功能安全评估应在适当的细节层。功能安全评估应依照 ISO26262-2:2011 6.4.9 执行。

5.4.5.2 按照 4.3 本要求适用于 ASIL B。功能安全评估应该进行。

注: 这个可以通过客户, 另一个组织或由供应商本身。

5.4.5.3 按照 4.3 本要求适用于 ASILs C 和 D。功能安全评估根据 ISO26262-2:2011

6.4.9, 应当在供应商的要求前提下由客户, 或由客户指定的一个组织或个人。

注: 这可以通过供应商自身来做。

5.4.5.4 按照 4.3 本要求适用于 ASILs(B), C 和 D。功能安全评估报告应当在客户和供应商的前提要求下。

5.4.5.5 按照 4.3 本要求适用于 ASILs(B), C 和 D。每个潜在影响交付供应商的异常识别, 应分析和寻求根源解决这些冲突。双方应达成协议对谁执行所需的操作。

#### 5.4.6 生产发布

5.4.6.1 供应商应当提供证据给客户依照 ISO26262-2:2011, 和, ISO26262-7:2011, 条款 5 中满足和维护的过程能力。

5.4.6.2 按照 ISO26262-2:2011 7.4.2.1 客户和供应商之间的供应协议应解决功能安全的责任并定义双方的安全活动。

5.4.6.3 供应协议应描述双方安全相关的交流、生产监控记录特点。

5.4.6.4 任何一方开始意识到安全的事件时应当根据供货协议及时报告。如果一个安全事件发生时, 事件应当进行分析。这种分析应包括类似的项目和类似的事件有关各方的潜在影响。

## 6、安全需求规范和管理

### 6.1 目标

第一个目标是针对它们的属性和特点确保正确的安全要求规范。第二个目标是确保的安全需求在整个安全管理生命周期的一致性。

### 6.2 概述

安全要求构成所有的需求, 旨在实现和确保所需的 ASILs 等级。安全生命周期期间, 安全要求详细规定在一个层次结构。在 ISO 26262 中安全要求结构和依赖关系如图 2 所示。安全需求分配各单元之中。

安全需求管理包括管理需求, 取得协议的要求, 获得承诺实施要求, 并保持可追溯性。为了支持安全需求管理, 应该使用合适的安全需求管理工具。这一条款包括对安全要求的规范和管理要求(见图 3)。

有关的安全要求在不同层级的具体要求内容在 ISO26262-3, ISO26262-4, ISO26262-5 和 ISO26262-6 中列出。

### 6.4 要求和建议

#### 6.4.1 安全要求规范

6.4.1.1 为了获取表 6.4.2.4 的安全要求特性, 安全要求应该通过一个合适的结合方式来详细说明:

(1) 自然语言

(2) 表 1 中列出的方式

注: 对于更高级别的安全要求(如功能性和技术性安全要求)用自然语言比较合适的, 而对于较低级别的安全要求(如软件和硬件安全要求)用符号在表 1 中列出的是比较合适的。

安全需求说明  
Table 1 — Specifying safety requirements

Methods	ASIL			
	A	B	C	D
1a Informal notations for requirements specification 非正式符号的需求说明	++	++	+	+
1b Semi-formal notations for requirements specification 半正式符号需求说明	+	+	++	++
1c Formal notations for requirements specification 正式符号需求说明	+	+	+	+

#### 6.4.2 安全要求的属性和特性

##### 6.4.2.1 安全要求应明确标识为安全要求。

注：为了符合这个要求，安全要求可以列在一个单独的文件。如果安全要求和其它要求被施用在同一文档中，安全性要求可以明确地通过使用一个特殊的属性识别，如 6.4.2.5 中所述。

6.4.2.2 安全要求应当从 ASIL 等级中派生出来，除非 ASIL 分解是按照 ISO26262-9 应用。

注：由于安全目标是顶级的安全要求，ASILs 开始在安全目标水平（见 ISO26262-1:2011，定义 1.108）。

6.4.2.3 安全要求应被分配到每一个项目或元素。

6.4.2.4 安全要求应具有以下特点：

一) 清楚和易于理解，

注 1：要求是清楚的，对现存的有关规定的含义达成共识。

注 2：要求是可以理解的，读者在相邻的抽象级别（即无论是利益相关者或该要求的消费者）理解其含义。

二) 分单元，

注：在一个分层等级的安全要求是分配在每个单元的，在这样的方式它们不能考虑的单元被分成一个以上的安全要求。

三) 内部一致性，

注：与外部的一致性不同，多重安全规定并不互相矛盾，内部一致性是指每一个的安全需求包含在本身没有任何矛盾。

四) 可行性

注：一个要求是可行的，它可以在产品开发的限制内被实现（资源、状态灯）。

五) 可验证。

6.4.2.5 安全要求应具有以下属性：

一) 安全生命周期中一个独特的识别保持不变，

例如，一个需求的独特识别可以通过各种各样的方式，如加下标，每个实例用“应当”一词，例如“系统应该检查…”，每个句子或连续编号包含“应当”一词，如“在……的情况下，系统应检查…”。

二) 状态， 例如，实施安全要求的状态可以是“建议”，“假设”，“接受”或“评论”。

三) ASIL。

### 6.4.3 安全要求管理

#### 6.4.3.1 安全要求有以下属性：

一) 层次结构，

注：层次结构意味着安全要求在几个连续的水平结构，如图 2 所示。这些水平总是一致遵守相应的设计阶段。

二) 依据适当的分组方案构成的组织结构

注：安全要求的组织结构意味着安全要求在每个级别分组，通常对应于体系结构。

三) 完整性、

注：完整性意味着安全要求在某种程度上完全实现的所有安全要求水平。

四) 外部一致性，

注：与内部一致性，个人安全需求本身并不矛盾，外部一致性意味着多个安全要求并不相互矛盾。

五) 在任何级别的分层结构没有重复的信息，

注：没有重复的信息意味着安全要求的内容不重复，安全要求在一个级别的层次结构是真正的在每个层次的水平。

六) 可维护性。

注：可维护性意味着可以修改或扩展的需求，例如，引入新版本的需求或添加/移除需求的要求。

#### 6.4.3.2 安全要求应可追踪的参考

一) 每个安全要求来源上层级别，

二) 每个派生的安全要求在较低层次级别，实现的设计，

三) 规范的验证按照 9.4.2。

注：另外，可追溯性支持：针对特殊要求的变更影响分析，功能安全评估。

6.4.3.3 结合表 2 中列出的验证方法来验证安全要求符合要求需求条款，他们遵守特定的要求，验证各自的 ISO26262 的部分内的安全要求。

安全需求验证方法

Table 2 — Methods for the verification of safety requirements

	Methods	ASIL			
		A	B	C	D
1a	Verification by walk-through 走读验证	++	+	○	○
1b	Verification by inspection 查读验证	+	++	++	++
1c	Semi-formal verification <sup>a</sup> 半正式验证	+	+	++	++
1d	Formal verification 正式验证	○	+	+	+

<sup>a</sup> Method 1c can be supported by executable models. 1c 由执行模式来支持

#### 6.4.3.4 安全需求应归属配置管理，依据第 7 节

## 7、 配置管理

### 7.1 目标

第一个目标是确保工作产品创造的原则和基本条件，可以被唯一地标识和在任何时间是

受控的。第二个目标是确保早期版本和当前版本可以之间的联系与区别是可追溯的。

## 7.2 概述

配置管理是汽车行业内公认的做法,根据 ISO/TS16949, ISO10007 和 ISO/IEC12207。

ISO26262, 每个工作产品都适用配置管理。

## 7.4 要求和建议

7.4.1 配置管理应计划。

7.4.2 配置管理过程应符合:

一) 质量管理体系的有关要求 (如 ISO/ TS 16949 或 ISO9001)

二) 根据 ISO / IEC12207 有关配置管理的条款进行软件开发的具体要求。

7.4.3 符合 ISO26262-2 要求的安全计划中的工作产品应置于配置管理, 并根据配置的管理策略建立基线。

7.4.4 配置管理下的工作产品应记录在配置管理计划。

7.4.5 配置管理应保持在整个安全生命周期。

## 8、 变更管理

### 8.1 目标

变更管理的目的是在整个安全生命周期分析和控制变更对工作产品相关的影响。

### 8.2 概述

变更管理保证了系统的规划, 控制, 监测, 执行和文档的变更, 同时保持每个工作产品的一致性。功能安全潜在影响评估在变更之前。为此, 变更决策流程被引入、建立, 职责分配给有关各方。

注意, 这里的变更理解为修改原因是: 异常, 删除, 增加, 增强部件或零件等的过时

### 8.4 要求和建议

8.4.1 规划和启动变更管理

8.4.1.1 变更管理过程应策划和发起, 在工作产品变更之前。

注意配置管理和变更管理在同一时间启动。两个过程的接口定义和维护以保持变更的可追溯性。

8.4.1.2 工作产品应在变更管理中确定, 并应包括 ISO 26262 要求的配置管理之下。

8.4.1.3 申请变更管理流程应定义在每个工作产品中。

8.4.1.4 变更管理过程应包括:

- 一) 按照 8.4.2 的变更要求,
- 二) 按照 8.4.3 的变更要求分析,
- 三) 按照 8.4.4 关于变更请求的决定和理由,
- 四) 按照 8.4.5 可接受的变更实现,
- 五) 按照 8.4.5 的文档。

8.4.2 变更请求

8.4.2.1 唯一标识应分配给每个变更请求。

8.4.2.2 作为最低要求, 每一个变更请求应包括以下信息:

- a) 日期

- b) 请求变更的原因,
- c) 请求变更的准确描述,
- d) 任何变更要求是基于配置管理。

#### 8.4.3 变更请求分析

8.4.3.1 每个变更请求涉及到的项目、接口和相关项目的影响分析应进行。以下情况应予以解决:

- a) 变更请求的类型,  
注意可能的变更类型包括: 解决错误, 适应, 增强, 预防。
- b) 工作产品变更标识和工作产品的影响,
- c) 在分布式开发的情况下, 有关方的识别和介入的影响,
- d) 变更对功能安全潜在的影响,
- e) 变更实现和验证的时间。

8.4.3.2 每次工作产品的变更完成将返回到合适的安全生命周期阶段。后续阶段应符合 ISO26262 进行。

#### 8.4.4 变更请求评估

8.4.4.1 变更请求应使用按照 8.4.3.1 影响分析的结果进行评估, 由授权人员决定接受, 拒绝或延迟。

例如: 通常情况下, 获授权人士包括:

- ◆ 项目经理,
- ◆ 安全经理,
- ◆ 负责质量保证的人员,
- ◆ 涉及到的开发人员。

注: 接受的变更请求可以优先与相关的接受的变更请求相结合。

8.4.4.2 对于每个接受的变更请求应当决定在变更到期之前谁执行变更。这一决定应考虑变更请求涉及到的接口。

#### 8.4.5 执行和记录变更

8.4.5.1 变更应按计划执行并验证。

8.4.5.2 如果变更对安全相关功能有影响, 功能安全评估的影响和适用的确认评价, 按照 ISO26262-2:2011, 6.4.7 和 6.4.9, 应更新, 在项目发布之前。

8.4.5.3 变更的文件应包含以下信息:

- a) 变更的工作产品在一个适当的水平, 包括配置和版本列表, 根据第 7 条 (配置管理),
- b) 变更执行的细节
- c) 变更发布的计划日期。

注: 变更请求在被拒绝的情况下, 变更请求和拒绝的理由也记录

## 9、 验证

### 9.1 目标

验证的目的是确保工作产品符合他们的需求。

## 9.2 概述

验证适用于安全生命周期的下列阶段。

a) 在概念阶段, 验证确保概念对项目的边界条件是正确的, 完整的和一致的, 边界条件定义本身是正确的, 完整的和一致的, 所以概念是可以实现的。

b) 在产品开发阶段, 验证以不同的形式进行的, 如下所述。

1) 在设计阶段, 验证是评估工作产品, 如要求规范、体系设计、模型或软件代码, 从而确保他们与先前建立需求的正确性, 完整性和一致性。评估可以由审查, 仿真和分析技术。评价计划, 指定, 执行和记录系统。

注: 设计阶段是 ISO26262-4:2011 条款 7(系统设计)、ISO26262-5:2011 条款 7(硬件设计)、ISO26262-6:2011 条款 7(软件架构设计)和 ISO26262-6:2011 条款 8(软件单元设计和实现)。

2) 在测试阶段, 验证是在一个测试环境下评估工作产品以确保符合他们的需求。测试应以系统的方式计划, 执行, 评估和记录。

c) 在生产经营阶段, 验证确保:

- 1) 安全需求是实现在生产过程、用户手册、修理和维护指令;
- 2) 项目的安全性能满足通过生产过程的控制措施。

## 9.4 需求和建议

### 9.4.1 验证计划

9.4.1.1 验证计划进行安全生命周期的每个阶段和子阶段执行, 具体内容如下:

a) 工作产品的内容进行验证,

b) 用于验证的方法,

注: 验证方法包括审查, 走读, 查读, 模型检查, 仿真, 工程分析, 演示和测试。通常适用于这些和其他方法的组合进行验证。

c) 通过和失败的验证标准,

d) 合适的验证环境,

注: 验证环境可以是测试或仿真环境。

e) 合适的验证工具,

f) 检测到异常的操作,

g) 回归策略。

注: 回归策略指变更项目元素后如何进行重复验证。验证可以完全或部分重复进行, 可以包括可能影响其他项目或元素的验证结果。

9.4.1.2 验证的计划应该考虑以下几点:

a) 应用的充足的验证方法,

b) 验证工作产品的复杂性,

c) 之前的验证相关主题的经验,

注: 这包括服务历史以及使用证明论点的程度, 已经达到的成就。

d) 所使用的技术的成熟程度，或使用这些技术相关联的风险。

#### 9.4.2 验证规范

9.4.2.1 验证规范应当选择和指定用于验证的方法，应包括：

- a) 检查或分析清单，
- b) 仿真场景，
- c) 测试用例、测试数据和测试对象。

9.4.2.2 对于测试，测试用例的规范应包括以下几点：

- a) 独特的标识，
- b) 相关的验证工作产品的参考版本，
- c) 前提条件和配置，

注：如果一个完整的工作产品验证可能的配置(如系统的变量)不是可行的，选择一个合理的子集(例如最小或最大功能系统的配置)。

- d) 在适当的情况下环境条件，

注：环境条件相关的物理环境(如温度)是进行测试或模拟测试的一部分。

- e) 输入数据、时间序列和它们的值，

f) 预期的行为包括输出数据，可接受范围的输出值，时间行为和兼容的行为。

注 1：当指定预期的行为，它可能需要指定初始输出数据来检测变化。

注 2：为了避免冗余的规范和存储的先决条件，用于各种条件的测试用例配置和环境，建议使用一个明确的参考数据。

9.4.2.3 对于测试，测试用例应当按照适用的测试方法分组。对于每个测试方法，除了测试用例，以下应该规定：

- a) 测试环境，
- b) 逻辑、时间依赖性，
- c) 资源。

#### 9.4.3 验证执行和评估

9.4.3.1 验证按计划执行，依照 9.4.1 和 9.4.2 中指定。

9.4.3.2 验证结果的评价应包含下列信息：

- a) 验证工作产品的唯一标识，
- b) 验证计划和验证规范参考，
- c) 验证环境的配置和验证工具的使用，以及在评估期间使用的校准数据，
- d) 验证结果与预期结果的一致性水平，
- e) 明确的声明验证是否通过还是失败，如果验证失败声明中应当包括失败的理由和建议对验证工作产品的变更，

注：验证评估依据标准的完成和验证的终止(见 9.4.1.1 c)) 和预期的验证结果。

- f) 没有执行任何验证步骤的原因。

## 10、 文档

### 10.1 目标

主要目标是开发整个安全生命周期的管理策略文档，以便文档管理过程是有效的，可重

复的。

## 10.2 概述

ISO 26262 要求的文档重点是信息，而不是布局和外观。实体文档不需要提供信息，除非是 ISO 26262 特别指定的。文档可以采取多种形式和结构，可以使用来生成文档。

例如，形式可能有：纸质、电子媒体、数据库。

什么是被认为足够的信息取决于多种因素，包括复杂性的程度、安全相关系统/子系统和需求相关的特殊应用程序。

在一个文档或文档之间重复的信息应该避免，这有助于可维护性。

注：在一个文档中使用交叉引用另一个复制的信息应该给读者源文档的信息。

## 10.4 需求和建议

10.4.1 文档过程应当计划为了使文档：

- a) 在整个安全生命周期的每个阶段的有效完成阶段和验证活动，
- b) 管理功能安全，
- c) 作为输入的功能安全评估。

10.4.2 ISO 26262 的工作产品的识别应解释为文档包含需求有关结果要求。

注：文档的形式可以是一个文档，其中包含完整的信息、工作产品或一组文件，包含工作产品的完整信息。

10.4.3 文件应该是：

- a) 准确、简洁，
- b) 以明确的方式结构化，
- c) 面向的用户容易理解的
- d) 易于维护。

10.4.4 整个文档的结构应该考虑内部程序和工作实践。应当组织有利于搜索相关信息。

例如，文档树。

10.4.5 每个工作产品或正式文档应当具有以下元素：

- a) 标题，指的是内容的范围，
- b) 作者和批准者，
- c) 每个文档不同版本(版本) 的唯一标识，
- d) 变更历史，

注意变更历史包含，变化，作者的名字，日期和一个简短的描述。

e) 状态。

例如“草案”、“发布”。

10.4.6 按照条款 7，应当可以识别当前适用的版本(版本) 文档或项目信息。

# 11、可信的软件工具

## **11.1 目标**

这一条款的第一个目标是提供标准来确定当前的软件工具所需的信任水平。这一条款的第二个目的是提供对应用的软件工具的认证，为了创建证据表明该软件工具适用于用于定制 ISO 26262 所需的活动或任务（即，用户可以依靠软件工具的正确的功能通过 ISO 26262 需要的活动或任务）。

## **11.2 概述**

在开发一个系统或软件或硬件的元素时使用的软件工具，可以支持或定制安全生命周期，定制 ISO 26262 要求的活动和任务。在这种情况下，软件工具需要有效地达到以下目标：

- a) 开发产品的系统故障，由于故障导致的风险，软件工具将输出错误最小化，
- b) 开发过程完全符合 ISO 26262，如果 ISO 26262 要求的活动或任务依靠使用的软件工具的正确功能。

注：“软件工具”的理解可以从不同使用的一套独立软件工具到一个集成的工具链。

例：这些软件工具可以是商业工具，开源工具，免费软件工具，共享软件工具或内部开发用户的工具。

为了确定所需的使用的软件工具的信心水平，以下标准进行评估：

- ◆ 该软件工具及其相应的错误的输出可以引入故障的可能性或无法检测到的与安全有关的产品或正在开发的元件的误差。
- ◆ 防止或检测其相应的输出错误的信心。

为了评估防止或检测措施信任，测量内部的软件工具（如监控），以及测量外部的软件工具（如准则，测试，评价）实施的与安全相关的项目或元素的开发过程考虑并进行评估。

如果按所确定的工具置信水平表示，那么相应资质的方法应用到符合这两个工具的信心都分配给安全要求级别和最大 ASIL 该项目或元件即使用软件工具被开发。否则没有必要应用这样资格的方法。

## **11.4 需求和建议**

### **11.4.1 基本要求**

11.4.1.1 如果安全生命周期包含开发一个系统或它的硬件或软件元素时使用的软件工具，这样 ISO 26262 所要求的活动或任务依靠正确的软件工具的功能，工具适用的处理步骤的相关输出不检查或验证，这样的软件工具应当遵守本条款的要求。

### **11.4.2 预定工具置信水平或资格的有效性**

11.4.2.1 如果软件工具的置信度评估或资格在特定的安全项目或元素的开发中独立执行，这个预定的工具置信水平或资格的有效性确认后，按照 ISO 26262-2:2011，表 1，软件工具被用于开发一个特定的安全项目或元素。

注：收集软件工具信息可以是一个跨组织活动，从而有利于分类或认证。

### **11.4.3 软件工具符合评价标准或其资质**

11.4.3.1 使用软件工具时，应当确保其用法、环境、功能限制、一般操作条件符合其评估标准或其资格。

例：使用的版本和配置、用例与实施预防措施或故障检测相应的错误输出应记录在这个

软件工具的合格报告中。

#### 11.4.4 规划软件工具的使用

11.4.4.1 一个软件工具的使用应当有计划，包括确定：

- a) 确定软件工具的版本号，
- b) 软件工具的配置，

例：编译器的配置是通过设置编译器开关和定义 C 源文件的 “#pragma” 声明。

- b) 有关软件工具的使用案例，

注 1：用例可以描述用户使用工具时软件工具或软件的应用的功能。

注 2：用例可以包括用于软件工具执行时该工具的配置和环境的要求。

- d) 软件工具执行环境，

e) 所有安全要求中分配给项目或元素中可以避免的最大的 ASIL，如果软件工具出现错误并产生相应的错误输出

注：最大的 ASIL 可对于一个特定的开发，还可以是软件工具的通用用法被确定的假设。

在这种情况下一个假设的预先确定的 ASIL 用假设来验证。

- f) 软件工具的认定方法，需要在置信等级的基础上来确定

11.4.4.2 可以的话，确保软件工具的评价或使用，下列信息可用：

- a) 软件工具的描述特性，功能和技术性能
- b) 适用的用户手册或其他使用指南
- c) 操作所需的环境的描述，
- d) 若可以的话，描述软件工具在异常的操作条件下的行为，

例 1 异常操作条件可以通过禁止编译器的组合开关，环境不遵守用户手册或不正确的安装。

例 2 异常操作条件下的预期行为可以抑制输出产生，用户指示或用户报告。

e) 若可以的话，描述已知的软件工具故障和适当的保护措施，避免或者解决方案的措施，

例 1 使用指南或解决已知的故障，通过编译器或者使用一组有限的模型构建块限制代码的优化。

例 2 保障包括预防通过使用约束，已知故障和问题的检测报告，提供安全的替代技术来执行相应的活动。

f) 软件工具所需的置信程度的确定过程中鉴别软件工具检测故障的措施和相应的错误输出。

注：检测错误和相应的输出的措施可以解决已知的和潜在的软件工具的输出错误。

例：冗余软件工具的输出比较，执行的测试，静态分析或者评论，分析软件工具的日志文件。

#### 11.4.5 软件工具的分析评估

11.4.5.1 软件工具的使用描述应包含下列信息：

- a) 目的

例：仿真功能，源代码生成，或嵌入式软件的测试，安全生命周期的定制或 ISO 26262

要求的活动和任务的自动简化。

b) 输入和预期输出,

例: 后续开发活动所需的输入数据, 源代码, 仿真结果, 测试的结果, , 或 ISO 26262 的其他工作产品。

c) 若可以的话, 环境和功能的限制。

例: 嵌入软件工具的开发过程, 不同软件工具的共享数据和其他使用条件, 防止或检测软件工具故障的措施。

11.4.5.2 软件工具分析和评估的目的, 以确定:

a) 一种可能性, 特定软件工具的故障在被开发的安全相关的项目或元素中被引入或无法检测错误。这被分类表示为工具影响 (TI)。

1) TI1 时应是没有这样的可能性;

2) TI2 应在所有其他情况下;

b) 防止软件工具产生故障和输出相应的错误措施, 或检测到软件工具发生故障和产生了相应的错误输出措施。这通过错误检测类的工具 (TD) 来表示:

1) TD1 应当选择如果对故障及其相应错误的输出将阻止或发现有高度的信心;

2) TD2 应当选择如果对故障及其相应错误的输出将阻止或发现有一定程度的信心; 3) TD3 在其他所有情况下应选择。

注 1: 预防或检测可以通过过程步骤, 任务冗余或软件工具本身的合理性检查。

注 2: TD3 通常适用于在开发过程中没有系统性的措施可用的情况, 因此只能随机的检测软件工具的故障和其相应的错误输出。

注 3: 如果一个软件工具, 用于验证另一软件工具的输出, 评估后续软件工具时和后续软件工具的 TD 选择时考虑那些软件工具之间的相互依赖性。

注 4: 使用情况分析细节水平需要允许适当地确定双方的 TI 和 TD。

例 1 , 可以选择 TD1 的情况, 一个代码生成器生成的源代码是根据 ISO 26262 验证。

例 2 使用指南可以防止发生故障, 如一个编译器产生的不正确或模棱两可的代码解释。

11.4.5.3 如果 TI 或 TD 的正确选择尚不清楚或不能确定, TI 和 TD 应该保守估计。

11.4.5.4 如果软件工具定制的开发过程的 ISO 26262 所要求的活动或任务是省略的, TD2 不得选择。

11.4.5.5 基于价值观决定 TI 和 TD 的类(依照 11.4.5.2 11.4.5.3 或 11.4.5.4), 所需的软件工具的信心水平取决于表 3。

**Table 3 — Determination of the tool confidence level (TCL)**

		Tool error detection		
		TD1	TD2	TD3
Tool impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

11.4.6 软件工具认证

11.4.6.1 TCL3 软件工具认证分类见表 4 中列出的方法。TCL2 软件工具认证分类见表 5 中列出的方法。TCL1 软件工具分类不需要认证方法。

TCL3软件工具认证分类

Table 4 — Qualification of software tools classified TCL3

Methods	ASIL			
	A	B	C	D
1a Increased confidence from use in accordance with 11.4.7 从使用中增加信任	++	++	+	+
1b Evaluation of the tool development process in accordance with 11.4.8 开发过程中评估	++	++	+	+
1c Validation of the software tool in accordance with 11.4.9 软件工具检验	+	+	++	++
1d Development in accordance with a safety standard <sup>a</sup> 依据安全标准开发	+	+	++	++

<sup>a</sup> No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected. 没有安全标准完全适用于软件工具的开发。相反，相关的安全要求的子集可以选择标准。

EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178.

按照ISO 26262、IEC 61508或RTCA - 178开发的软件工具

Table 5 — Qualification of software tools classified TCL2

Methods	ASIL			
	A	B	C	D
1a Increased confidence from use in accordance with 11.4.7	++	++	++	+
1b Evaluation of the tool development process in accordance with 11.4.8	++	++	++	+
1c Validation of the software tool in accordance with 11.4.9	+	+	+	++
1d Development in accordance with a safety standard <sup>a</sup>	+	+	+	++

<sup>a</sup> No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.

EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178.

11.4.6.2 软件工具认证应当记录以下内容:

- a) 软件工具的标识和版本号,
- b) 软件工具分类的最大工具置信水平与评价分析参考
- c) 预先设定的最大 ASIL 或特定 ASIL, 任何可能违反安全要求的相应软件工具故障和产生错误的输出,
- d) 软件工具的配置和环境是合格的,
- e) 个人或组织进行认证,
- f) 申请方法依照 11.4.6.1 认证
- g) 措施的结果应用于软件工具的认证, 和
- h) 如果适用的话, 使用约束和故障识别进行认证。

11.4.7 在使用中增加信任度

11.4.7.1 使用中增加信任度的方法按照表 4、表 5 软件工具的认证需求条款。

11.4.7.2 增加软件工具信任度, 提供以下证据:

注: 在条款 14 中证明的要求, 在这部分不适用。

- a) 以前使用的用于相同目的的软件工具有可比较的用例、确定的操作环境和具有类似功能的限制,
- b) 在使用中增加信任度的理由基于充分和足够的数据,  
注: 数据可以通过大量的积累使用(如持续时间或频率)。
- c) 软件工具的规范是不变,

d)发生故障和相应的错误输出需要在软件工具以前的使用中以系统化的方式积累。

11.4.7.3 在开发活动中以前使用的软件工具的经验应分析和评估通过考虑以下信息：

- a) 软件工具的标识和版本号，
- b) 软件工具的配置，
- c) 使用的详细信息和相关数据，

例如，软件工具的使用特性和相关软件工具的用例的使用频率。

- d) 文档，描述软件工具的故障和相应的错误输出以及导致他们的条件，
- e) 以前版本的监控列表，在相关的版本的每个故障清单，和
- f) 保障措施，已知故障的规避措施或对策，或错误输出的检测措施，如果适用的话。

例如，使用报告的来源可以日志；软件工具的供应商提供的版本历史发表勘误表。

11.4.7.4 信心增加须考虑软件工具的有效版本。 11.4.8 工具开发过程评估

11.4.8.1 申请认证的软件工具开发过程评估的方法按照表 4、表 5，分条款应符合要求。

11.4.8.2 开发过程中软件工具的应用开发应符合一个适当的标准。

注：对开放源码的工具，一些社区的标准也可以适当。

11.4.8.3 应用于软件工具开发的开发过程的评价由合适的国家或国际标准和合适的应用评估开发过程来演示评价。

注：这个评估覆盖足够的开发和相关的软件工具子集的特性。

例：使用基于汽车的 SPICE, CMMI, ISO 15504 评估方法。

#### 11.4.9 验证的软件工具

11.4.9.1 “如果法”验证的软件工具”根据表 4 和表 5 的应用资格的软件工具，这分条款应符合的要求。

11.4.9.2 软件工具验证应当满足下列条件：

- a) 验证措施应当证明该软件工具符合规定要求，

注：测试评估功能性和非功能性的质量方面的软件工具应进行验证。

例：编程语言的标准有助于定义验证有关编译器的要求。

- b) 软件工具的故障及其相应的错误输出应当验证并根据信息分析其可能的后果和避免的措施。

例：软件工具对异常操作的反应应当检查；

例：工具的可预见的误用，输入数据不完整，不完整的更新软件，禁止使用的组合配置设置。

#### 11.4.10 确认评审软件工具认证

按照 4.3，本条款适用于 ASILs (B), C, D。使用的软件工具的置信度应按照 ISO 26262 – 2:2011 表 1 评估以确保：

- a) 正确评估所需的软件工具的置信度，和
- b) 软件工具的认证符合其需要的置信度。

## 12、 软件组件证明

### 12.1 目标

软件组件认证的目的是提供在项目开发时符合 ISO 26262 重用的证据。

### 12.2 概述

合格的软件组件的重用可以避免软件组件之间相同功能的重复开发。

注：软件组件的理解包括源代码，模型，预编译代码，或者编译和链接软件。

例：软件组件包括：

- ◆ 第三方供应商提供的（商用软件） 软件库；
- ◆ 在电子控制单元已经使用的内部组件。

### 12.4 需求和建议

#### 12.4.1 概述

一个软件组件能够作为合格的，应提供如下内容：

- a) 软件组件的规范按照 12.4.3.1，
- b) 软件组件按照 12.4.3.2, 12.4.3.3, 12.4.3.4 符合其要求的证据，
- c) 依照 12.4.4, 软件组件适用于其预期使用的证明，和
- d) 软件组件的开发过程是基于合适的国家或国际标准的证据。

注：在以前开发的软件组件的情况下，一些再造活动可以执行以遵守这个分条款。

#### 12.4.2 软件组件认证计划

12.4.2.1 软件组件的认证计划应确定：

- a) 软件组件的唯一标识，
- b) 如果软件组件以前执行不正确，任何可能违反安全要求的最大 ASIL 目标，
- c) 进行软件组件认证的活动。

#### 12.4.3 软件组件的认证

12.4.3.1 软件组件的规范应包括：

- a) 软件组件的需求，

例如，下面的条件：

功能需求，

算法或数值精度，算法精度考虑程序错误，只提供近似解和数值准确性考虑舍入误差，造成计算不精确，截断误差造成的许多函数的近似表示

在失败的情况下的行为，

- ◆ 响应时间，

- ◆ 资源使用情况，

- ◆ 运行环境的要求，

- ◆ 在过载情况下的行为（鲁棒性）。

- c) 配置的描述，

注：对于软件组件包含多个软件单元，配置的描述包括标识和每个软件单元的配置。

- c) 接口描述，

- d) 应用手册，在适当的地方，

e) 软件组件集成描述,

注: 描述可能包括所需要的开发工具集成和使用的软件组件。

f) 异常操作条件下的功能反应,

例: 非重调用软件组件下的重调用。

g) 软件组件与其他的依赖关系, 和

h) 异常情况下相应的变通措施描述。

12.4.3.2 为了提供一个软件组件符合验证需求的证据, 软件组件应:

a) 依照 ISO 26262 – 6:2011 条款 9 显示需求覆盖度。

注: 验证主要是基于需求的测试。基于软件组件的需求的测试结果执行在其开发或在以前可以使用集成测试中。

例: 在软件组件的实现和集成中, 专用的认证测试套件应用程序, 分析所有已经执行的测试。

b) 包括在失败的情况下正常操作条件和行为, 和

c) 导致违反安全要求的未知的错误结果。

12.4.3.3 按照 4.3 本条款适用于 ASIL D。

架构覆盖率应按照 ISO 26262 – 6:2011 条款 9 衡量, 以评价测试用例的完整性。如果有必要, 额外的测试用例或理由应当有明确规定。

12.4.3.4 依照 12.4.3.2 认证是有效的, 对于没有变更的软件组件。

12.4.3.5 软件组件的认证应记录包括以下信息:

a) 软件组件的唯一标识,

b) 独特的软件组件配置,

c) 实施认证的个人或组织,

d) 用于认证的环境,

e) 应用于验证符合软件组件的措施, 和

f) 任何安全需求的最大 ASIL 目标。

12.4.4 验证软件组件认证

12.4.4.1 软件组件认证的结果和这些结果的有效性, 关于软件组件的预期用途应当验证。如果有必要, 应用额外的措施。

注: 认证的有效性受到影响, 当认证在另一个工业或汽车领域的环境中执行时。

例: 发动机控制、车身控制和底盘控制是汽车不同的领域。铁路和公路、航空电子设备是不同的工业领域。

12.4.4.2 软件组件的规范应符合使用软件组件预定的要求。

## 13、硬件组件证明

### 13.1 目标

硬件组件认证的第一个目标是提供中间水平硬件组件和作为项目的一部分使用的部件, 符合 ISO 26262 开发的系统或元素的合适的证据, 考虑他们安全概念目标的功能行为和操

作限制。

硬件组件认证的第二个目标是提供相关信息：

- ◆ 失效模式，
- ◆ 失效模式分布，
- ◆ 对项目安全概念的诊断能力。

### 13.2 概述

在 ISO 26262 范围内每一个安全相关的硬件组件和部分使用从属于标准认证，来解决一般功能性能、生产一致性、环境耐力和鲁棒性。

例 1 资格按照 ISO 16750, 或与 AEC-Q100 AEC-Q200 标准的认证，部分或等同电子公司标准。对于基础部分(无源元件，离散半导体)，标准认证就足够了。这些基本可以用在按照 ISO 26262 - 5 的硬件设计部分。

这一条款的要求适用于中级水平的硬件组件或部件，提供专用的功能系统。

例 2 传感器、制动器、asic 和专用功能(如协议适配器)。

如果中级水平的硬件组件或部分是安全的，根据其水平，按照 ISO 26262 - 4 或 ISO 26262 - 5 集成和测试或根据这一条款来认证。

通常这一条款中描述的认证可以应用于组件或部件的失效模式或已知的经过充分测试认为可能失效的故障。

例 3 在燃油压力传感器的开发过程中，传感器的正确功能被批准，它的边界操作 200bar 燃油压力和 140° C 的温度。这种燃油压力传感器的认证允许使用这种传感器实现一个特定的安全项目功能考虑传感器的性能和它的故障，提供相同或更低的边界的运行应用。在这种情况下，设计分析和传感器的基本硬件的集成和测试根据 ISO 26262 - 5 可以省略，集成活动可以直接进行依据 ISO 26262 - 4 考虑分配给传感器的技术安全要求。

基本部分，硬件零部件的认证和集成总结如表 6 所示。

**Table 6 — Qualification, integration and test activities to be conducted depending on the level of hardware part or component**

Activity	Hardware part or component			
	安全相关的基本硬件单元	安全相关的中间件硬件单元	安全相关的中间级硬件组件	安全相关的复杂硬件组件
如, 电阻, 三极管	如, 解码器	如, 燃油压力传感器	如, ECU 单元	
标准认证	Applicable	Applicable	—	—
依据 13 条款的认证	—	Applicable	Applicable	—
依据 ISO26262-5 集成测试	—			Applicable
依据 ISO26262-4 集成测试	—	Applicable <sup>a</sup>	Applicable <sup>a</sup>	Applicable

<sup>a</sup> The hardware part or component will be integrated in accordance with ISO 26262-4, or ISO 26262-5, or both ISO 26262-4 and ISO 26262-5, depending on its level.

硬件单元或组件的认证可以使用两种不同的方法：测试或分析。这些方法可以单独使用或组合使用取决于硬件单元或组件。

测试时，硬件组件或部分处于目标环境和运行条件，兼容其功能需求被评估。复制精确的环境条件是困难的，而且任何推断都容易是错误的，因此这类测试的条件局限被认为是测试的结果的解释。

认证分析依赖于所使用的分析方法和假设的理由。在一般情况下，一个硬件组件太复杂而不能仅靠分析取得认证。然而，分析可有效地用于测试数据的推断和来决定较小的在已经测试的硬件组件中的变更影响。

即使使用不同的评定方法，最终结果在认证报告中都是可用的（认证报告可以由一组文档，包括报告，发现，笔记，解释等等），提供硬件组件假设、限定条件、测试用例和结果的证据。如果可能的话，最好是制定综合的方式独立检查；它通常包括性能数据，认证过程，结果和基本原理。

ISO 16750 的内容对定义认证测试的类型和顺序是有用的。

### 13.4 需求和建议

#### 13.4.1 概述

13.4.1.1 应用本条款的标准是：

- a) 认证的组件或单元应该是中间复杂度，不包括复杂的硬件组件和基本的硬件单元，
- b) 要认证的组件或单元的相关失效模式应当认为是由测试、分析或两者一起来验证的。

#### 13.4.2 目标硬件组件或单元的认证

13.4.2.1 硬件组件或单元认证时应当实现以下目标：

- a) 的组件或部件足够的功能性能安全概念的目的，
- b) 失效模式的识别和模型（分布的量化）通过使用适当的测试（如超过极限试验、加速试验…）或分析，
- c) 足够的健壮性、和
- d) 限制使用组件或零件的识别。

13.4.3 硬件组件或单元的认证方法 13.4.3.1 硬件组件或单元的认证选择合适的下列方法：

- a) 分析，
- b) 测试

#### 13.4.4 认证计划

13.4.4.1 认证计划应开发和描述：

- a) 硬件组件或单元的准确标识和版本，
- b) 硬件组件或单元被使用的环境，
- c) 认证策略和基本原理，

注：策略包括：分析、必要的测试和分阶段的描述。

- d) 执行此策略的必要的工具和设备，

- e) 实施这一战略的负责方, 和
- f) 用来评估认证的硬件组件或单元是通过还是失败的标准。

#### 13.4.5 认证论证

13.4.5.1 一个全面的论证, 硬件组件或单元履行的性能规范应当提供。

注: 所需的性能包括它所承受的确信的正常环境条件和预知失败的环境条件时产生的行为。

13.4.5.2 13.4.5.1 的综合性能应当依据以下类型的信息组合: a

- ) 分析方法和假设使用; 或
- b) 从运行经验中得到的数据; 或
- c) 现有的测试结果。

13.4.5.3 给出每个假设的理由, 包括推断。

#### 13.4.6 认证分析

13.4.6.1 分析应当以很容易理解的形式表达, 由具有相关工程或科学学科人员复核。

注: 分析方法可以使用包括推断、数学模型、破坏分析或类似分析的方法。

13.4.6.2 分析应当考虑所有硬件组件或单元暴露的环境条件, 这些条件的限制, 与操作相关的其他额外的压力(如预期开关周期, 充电和放电, 长时间关断)。

#### 13.4.7 资格通过测试

13.4.7.1 应当开发和测试计划应包含以下信息:

- a) 描述硬件组件或单元的功能,
- b) 测试的数量和顺序,
- c) 要求组装和连接的要求,
- d) 加速老化的过程, 考虑硬件组件或单元的操作条件,
- e) 模拟运行和环境条件,
- f) 建立的通过/失败标准,
- g) 环境参数测量,
- h) 测试设备的要求, 包括准确性和
- i) 在测试中允许的维护和更换流程。

13.4.7.2 应当使用一个标准化的测试规范。

注: 该规范可以基于 ISO 16750 系列或相关公司的标准。

13.4.7.3 测试按计划进行并生成可用的测试数据。

#### 13.4.8 认证报告

13.4.8.1 认证报告应描述是否硬件组件或单元已通过或失败相关的工作范围认证。

注: 认证报告可以由一组文件, 包括报告的结果和笔记解释。

13.4.8.2 认证报告应当验证按照条款 9。

## 14、论证证明

### 14.1 目标

这一条款为论证证明过程提供指南。论证证明的另一种意思是符合 ISO 26262 的使用，在现在项目或元素中重用的情况下数据是可用的。

### 14.2 概述

论证证明在使用参数可以应用于任何类型的产品，它的定义和使用条件相同或已经发布或运行的产品有非常高的通用性。它也可以应用于与此产品相关的任何工作产品。

注 1：论证证明不是内部变更：一个产品，替代设计或实现，旨在取代论证证明产品不能被认为可以使用因为它满足原始功能需求，除非该产品符合这一条款中指定的标准。

一个项目或一个元素，如系统，功能，硬件或软件产品，可能是论证证明的备选。

备选也可以参考系统，硬件或软件工作产品等技术安全概念、算法、模型、源代码、目标代码、配置或标定数据。

论证证明的动机包括：

- a) 在商业用途汽车应用程序旨在部分或完全转入到另一个目标；
- b) 运行的 ECU 实现额外的功能；或
- c) 候选人的领域在 ISO 26262 发布之前；或
- d) 候选人被用于其它安全行业；或
- e) 候选人是一个广泛使用的 COTS 产品不一定用于汽车应用。

论证证明被证实，候选人通过适当的文档，配置管理和变更管理记录和有关安全事件的字段数据。

一旦候选人被定义（见 14.4.3）与预期的信用使用证明（见 14.4.2），需要考虑两个重要的标准准备论证证明

候选人在服务期间字段数据的相关性（见 14.4.5），和如果有的话，变更，可能会影响候选人的服务期（见 14.4.4）。

在使用项目或元素不使用证明以下免除这些物品或元素

在项目或元素使用的证明并不能免除以下项目相关的安全管理活动中的项目或元素：

在安全计划中描述的信用使用证明，和论证证明过程中产生的数据和工作产品是安全情况的一部分还需要确认措施。

### 14.4 需求和建议

#### 14.4.1 概述

14.4.1.1 以下小节指 ASILs 适用于将来备选使用。

#### 14.4.2 信用证明

14.4.2.1 信用证明使用只有当候选人符合 14.4.2 – 14.4.5。

14.4.2.2 由论证证明产生的信用证明应根据 ISO26262 -2:2011 6.4.3.5 计划。

14.4.2.3 信用证明应当限于由论证证明候选的安全生命周期阶段和活动。

14.4.2.4 在一个项目或元素中使用的综合证明措施应实施在适当级别依照

ISO 26262 – 4:2011 条款 8。

例：ECU 硬件有一个满意的服务历史，目的是要达到 100% 的应用程序。信用证明可以应用到子阶段和活动发展的硬件元素。类似地，如果软件是一个 100% 的延滞与满意的服务历史信用证明就可以也被应用到软件子阶段和活动。

14.4.2.5 项目的安全验证在使用元素嵌入证明时进行根据 ISO26262-4:2011 条款 9。

14.4.2.6 项目的确认措施应当考虑使用元素证明参数和相关数据按照 ISO26262-2:2011 6.4.7。

14.4.2.7 项目或元素的使用证明的任何改变应当遵守 14.4.4 和对应的信用证明的维护。

注：这一条款适用于任何类型的修改包括启动安全相关事件的结果。

#### 14.4.3 备选的最少信息

14.4.3.1 备选的描述和它以前的使用应该是可用的，包括以下内容：

a) 候选的标识和内部元件或元素的可追溯性。

b) 相应的需要描述的安装、形式和功能要求，若可以，接口和环境，物理和化学，功能和性能特征

c) 备选的安全要求和相应的 ASILs 等级 14.4.4 备选的变更分析

#### 14.4.4.1 备选证明

备选变更和使用环境应依据 14.4.4.2 到 14.4.4.3 来识别。

注 1：备选变更指设计变更和完成变更。设计变更由需求变更、功能增强或性能增强产生，完成变更不影响备选的规格和功能，而仅仅是它的实现特征。完成变更由软件更正，新开发使用和生产工具产生。

注 2：配置数据和备选数据变更被认为是自身为避免安全目标影响行为而产生的。

注 3：备选环境的变更是由于备选使用在新的应用类型，具有不同的安全目标和需求，安装在新的目标环境下（例如，各种车辆，各种环境条件）或者各种元件与它周围有相互作用的更新。

#### 14.4.4.2 项目变更进入新的应用

项目或使用环境变更以进行新的应用应遵守 ISO26262-3:2011 条款

6.4.2. 14.4.4.3 要素变更进入新的应用 要素或使用环境变更以进行新的应用不同于项目变更，应遵守条款 8。

#### 14.4.4.4 备选变更独立于新的应用

备选变更在服务周期之后引入，独立于未来应用，应提供证据证明使用状态是有效的。

#### 14.4.5 现场数据分析 1

##### 4.4.5.1 配置管理和变更管理

应该提供证据证明备选在配置管理和变更管理服务周期之内或之后，以便于备选当前的状态能被确立。

##### 14.4.5.2 使用证明的目标值

注：当任何 ASIL 尚未分配给备选，ASIL D 目标保守选择。

14.4.5.2.1 备选服务期的计算基本原理应可用。

- 14.4.5.2.2 备选的服务期应来源于所有标本的观察期之外的结果，参考  
 14.4.5.2.3。
- 14.4.5.2.3 作为备选的每个相同规格和实现的样本的观察期间和运行在一个车辆超过平均每年车辆操作时间的备选被认为在进行服务周期的分析。
- 14.4.5.2.4 由备选获得的使用状态的证明，其服务演示应遵守每个安全目标，按照表 7，可以违反，的单面低置信度为 70%（使用卡方分布）。

#### 观测事件率限制

**Table 7 — Limits for observable incident rate**

<b>ASIL</b>	<b>Observable incident rate</b>
D	$<10^{-9}/\text{h}$
C	$<10^{-8}/\text{h}$
B	$<10^{-8}/\text{h}$
A	$<10^{-7}/\text{h}$

注 1： 使用参数证明的目的，一个可观测到意味着失败的事件报告给制造商，造成候选人有可能导致违反安全目标。

注 2 特性和事件观测率解释为分析潜在的违反安全目标

注 3 表 8 给出一个示例所需的最低服务周期间没有可观察到的事件率必须达到 70% 的置信度：

#### 最小服务周期内备选目标

**Table 8 — Targets for minimum service period of candidate**

<b>ASIL</b>	<b>Minimum service period without observable incident</b>
D	$1,2 \times 10^9 \text{ h}$
C	$1,2 \times 10^8 \text{ h}$
B	$1,2 \times 10^8 \text{ h}$
A	$1,2 \times 10^7 \text{ h}$

注意 4 如果标本的收集数据中发现可观测到的事件，必要的最低的服务期间，  
 $t_{\text{service}}$  可以调整如下：

$$t_{\text{service}} = t_{\text{MTTF}} \times \frac{(\chi_{\text{CL};2f+2})^2}{2}$$

CL 指信心水平取一个绝对值（例如 0.7, 70%）；

$t_{\text{MTTF}}$  指平均失效时间（1 / 失败率）；

f 指安全相关事件的数量；

$(\chi_{\alpha,\nu})^2$  指卡方分布的误差  $\nu$  和自由度  $\alpha$ 。

$(\chi_{\alpha,\nu})^2$

14.4.5.2.5 使用的信用证明应用可能是预期暂时，证明之前使用状态(按照 14.4.5.2.4)获得。在这种情况下，候选人的服务周期应符合每个安全目标，违反安全目标的依据表 9 的单面的置信水平低于 70%(使用卡方分布)。

#### 可测事件率的限制 (过渡周期)

**Table 9 — Limits for observable incident rate (interim period)**

ASIL	Observable incident rate
D	$<3 \times 10^{-9}/h$
C	$<3 \times 10^{-8}/h$
B	$<3 \times 10^{-8}/h$
A	$<3 \times 10^{-7}/h$

14.4.5.2.6 在任何的情况下观察到的事件在这一领域过渡时期在 14.4.5.2.5 描述的应当遵守如下：

- ◆ 停止使用表 9 的事件率和使用表 7 作为候选人；
- ◆ 提供证据表明，按照 ISO26262 观测事件的根源完全识别和消除或者，保持计算候选人的累积时间，重置特定根源的计算时间计数器和记录在安全情况下这些证据。

14.4.5.2.7 备选项目有一个非连续故障率的情况下，应采用额外的措施来验证证明。例如，由于疲劳的破坏情况。

注：这些措施应用于

#### 14.4.5.3 现场问题

系统报告的问题应该确定任何观测到有潜在安全影响的事件，在备选运行周期内记录和可检索 (ISO 26262-7:2011, 6.4.2.1)。

## ISO26262-9 面向汽车安全完整性等级 (ASIL) 和安全的分析

### 5、考虑 ASIL 裁剪等级分解要求

#### 5.1 目的

提供安全要求分解的规则和指南，得到 ASIL 下一等级的细节裁剪。

#### 5.2 通则

正在开发过程中的以安全为目标的要素的传播贯穿与要素开发的全过程。

从安全目标开始，安全要求在开发过程中会被分解和提炼。ASIL 作为安全目标的一个属性，会被每一个后续的安全要求所继承。功能和技术安全需求被分配到每个组成要素中，从开始的基本结构假设到最后的软硬件要素。

在设计过程中的 ASIL 裁剪方法被称作“ASIL 分解”。在分配阶段，优势来自架构决定，包括存在足够独立的架构要素。这些好处在于：

--应用冗余的安全要求通过独立的架构要素；

--分配一个可能更低的 ASIL 给这些分解后的安全要求;

如果这些架构要素不是足够独立的,那么冗余的要求和架构要素继承初始化的 ASIL。

**NOTE1** ASIL 分解是一种 ASIL 裁剪方法,这种方法可以应用在一个条款或要素的功能、技术、硬件、软件安全要求。

**NOTE2** 作为一个基本规则,ASIL 分解需要分配给充分独立的架构模块的安全要求是冗余的。

**NOTE3** 在一致性冗余的应用中(比如:备份设备和软件)和考虑到软硬件的系统性失效,ASIL 不能降低除非失效分析充分表明足够的独立性存在或者潜在的一般原因不会导致危险。因此,一致性冗余一般来书不是充分条件对于降低 ASIL 等级,这是由于在组件之间缺少足够的独立性。

**NOTE4** 一般情况下,ASIL 分解不适用于那些在多通道架构设计中的通道选择和开关。一般情况下,ASIL 分解允许在多个组件中以一个安全要求为目标的 ASIL 分摊 以确保相同安全要求的兼容。在一个预期功能和他对应的安全机制之间的 ASIL 分解在特定条件下是允许的(见 5.4.7)。

对于由于随机硬件失效造成的包括硬件架构矩阵评估和违反安全目标评估(见 ISO26262-5)保持不变。

### 5.3 本节输入

#### 5.3.1 前提条件

下列信息必须遵守:

--- 在 ASIL 分解的层级的安全要求: 系统、硬件、软件根据

ISO 26262-3:2011, 8.5.1, or ISO26262-4:2011, 6.5.1, or ISO26262-5:2011,  
6.5.1 or ISO26262-6:2011, 6.5.1; and

--- 在 ASIL 分解的架构信息层级: 系统, 硬件或软件依据

ISO26262-4:2011, 7.5.2, 或 ISO26262-5:2011, 7.5.1, 或  
ISO26262-6:2011, 7.5.1.

#### 5.3.2 进一步的支持信息

下列信息应当被考虑:

--- 条款定义(见 ISO 26262-3: 2011, 5.5)

----安全目标(见 ISO26262-3: 2011, 7.5.2)

### 5.4 要求和推荐

5.4.1 如果应用 ASIL 分解,在这一条款中所有的要求应当被遵守。

5.4.2 ASIL 分解在执行过程中应当单独考虑每一个初始安全要求;

**NOTE** 一些安全要求能被分配给相同的独立要素,这会造成不同的初始安全要求的 ASIL 分解。

5.4.3 初始安全要求应当被分解给足够独立的要素的冗余安全要求。

5.4.4 每一个分解的安全要求应当遵守它自己的初始安全要求;

**NOTE** 这些要求由定义提供了冗余。

5.4.5 在由于随机硬件失效评价硬件结构矩阵和违反安全目标的要求在 ASIL 分解过程

中保持不变，这是根据 ISO26262-5 的要求。

5.4.6 如果 ASIL 分解应用在软件层级，在要素之间应用的要求分解的充分独立性应当在系统级进行检查，在软件级、硬件级、系统级要采用适当的措施以实现充分的独立性。

5.4.7 如果一个初始的安全要求的 ASIL 等级分解导致了计划的功能和相关的安全机制的要求分配，那么：

- a ) 相关的安全机制应当被分配到更高一级的 ASIL 分解中；

注：一般，这个安全机制有更低的复杂性和更小体积与计划的功能来比的话。

- b ) 一个安全要求应当被分配到一个计划的功能，通过应用对应的 ASIL 分级进行应用。

注：如果选择 ASIL<sub>x</sub>(x) + QM(x) 的分解机制，那么 QM(x) 意味着质量管理系统是充分的对于开发组件，这些组件实现了分配给这些计划功能的安全要求。QM(x) 同时意味着质量管理系统能支持在计划功能和系统机制之间独立的基本原理。

5.4.8 如果一个违反初始的安全规范不能通过关掉这个组件来阻止的话，那么必须有足够的独立组件应用分解的安全规范应当体现出来。

5.4.9 当应用 ASIL 分解到一个安全规范时，那么：

- a ) ASIL 分解应当符合 5.4.10；

- b ) ASIL 分解可以应用多次；

- c ) 每一个分解的 ASIL 应当被标记出来，通过括号出来安全目标的 ASIL；

例如：如果一个 ASIL D 规范被分解到一个 ASIL C 规范和一个 ASIL A，那么应当标记如下：“ASIL C(D)” and “ASIL A(D)”。如果 ASIL C(D) 规范被进一步分解为一个 ASIL B 和一个 ASIL A，那么安全目标应当标记如下：

“ASIL B(D)” and “ASIL A(D)”

5.4.10 根据之前的 ASIL 分解（图 2 所示）以下的分解策略之一应当被选择，或者更高一级的 ASILs 的策略被应用。

注：从一个选择的分解策略到下一级的步骤定义了一个 ASIL 分解。

a) 一个 ASIL D 要求应当被分解为以下的一种：

- 1) 一个 ASIL C(D) 要求和一个 ASIL A(D) 要求；或者
- 2) 一个 ASIL B(D) 要求和一个 ASIL B(D) 要求；或者
- 3) 一个 ASIL D(D) 要求和一个 QM (D) 要求；

b) 一个 ASIL C 要求应当被分解为以下的一种：

- 1) 一个 ASIL B(C) 要求和一个 ASIL A(C) 要求；或者
- 2) 一个 ASIL C(C) 要求和一个 QM (C) 要求；

c) 一个 ASIL B 要求应当被分解为以下的一种：

- 1) 一个 ASIL A(B) 要求和一个 ASIL A(B) 要求；或者
- 2) 一个 ASIL B(B) 要求和一个 QM (B) 要求；

d) ASIL A 不能被进一步分解，除非必须的话，一个 ASIL A(A) 要求和一个 QM (A) 要求。

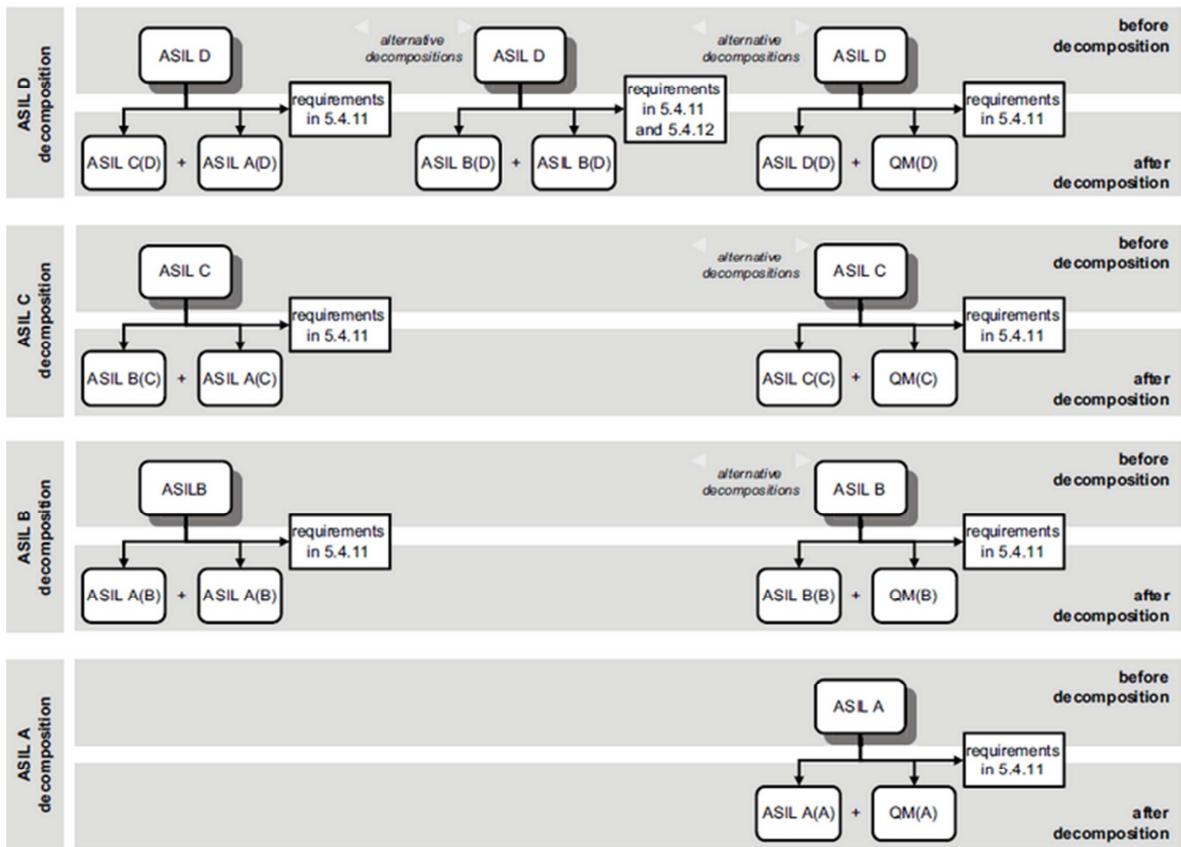


Figure 2 — ASIL decomposition schemes

例如：在 5.4.7 中描述的实例在最右列表示，QM 被分配给一个计划功能，一个 ASIL 等同于一个初始 ASIL 被分配各了一个相关的安全机制，

注意：每一个分解步骤的最上端阴影框代表了在 ASIL 被分解前。

5.4.11 当应用在 5.4.10 中给定的分解机制，那么：

a) 根据 ISO26262-2:2011 6.4.7 中确定的测量应当被应用以符合安全目标的 ASIL。

b) 分解之后的每个组件的足够的独立性证据应当是可行的。

注：如果在分解之前经过相关故障分析没有找到相关故障原因能导致安全要求被违反，部件是充分独立的（参见本部分条款 7），或者每一个相关故障的确定原因根据安全目标的 ASIL 被足够的安全措施所控制

5.4.12 当对在 5.4.10 (2) 中给定的 ASIL D 应用分解方案时，那么：

a) 被分解的安全要求应与 ISO 26262-8:2011, 条款 6 要求的 ASIL C 一致； 注意：对 ASIL C 要求的更形式化的公式与 ASIL B 比较，减少了系统故障和与 ASILB(D) 应用的依赖性。

B) 根据 ISO 26262-8 中的软件工具使用方法，如果同样的软件工具被用来分解组件的开发，那么这些软件工具应当被视为开发 ASIL D 要素或组件的软件工具。

5.4.13 根据 ISO26262-4 和 ISO 26262-6 的 ASIL 要求（分解后），作为最小单元，在系统级和软件级的分解组件开发应当被进行。根据 ISO26262-5 的 ASIL 要求（分解后），作为最小单元，在硬件级的分解组件开发应当被进行，除了硬件架构矩阵评价和由于随机硬件故障违反安全目标的评价。

5.4.14 在每一级的分解设计过程中，根据在分解之前的 ASIL 要求，相应的分解组件的集成和后续活动应当被开展。

## 5.5 工作成果

5.5.1 由 5.4 得到的架构信息更新。

5.5.2 由 5.4 得到的作为安全要求和要素的 ASIL 更新。

# 6、要素共存标准

## 6.1 目标

这一条款提供了在相同要素之间共存的标准：

——安全-子要素相关的没有分配 ASIL；

——安全-相关的子要素有不同的 ASILs；

## 6.2 通则

默认情况下，当一种要素是由几个子要素组成，这些子要素按照相应的最高的 ASIL 措施开发，即最高的 ASIL 安全要求分配到这些要素（见 ISO 26262-4:2011, 7.4.2.3）。

在被分配了不同的 ASILs 子要素共存，或没有安全想过的 ASIL 被分配的子要素共存的情况下，对于他们中的要素等 ASIL 等级，避免升级 ASIL 等级是有益处的。为此，本条款对于这些要素或子要素提供了确定 ASIL 等级的指导。本条款是基于子要素和其他要素的子要素的干扰分析基础上的。

来自于无 ASIL 分配的子要素或者低级 ASIL 对高一级的 ASIL 子要素导致违反要素的安全规范的连锁故障引起的干扰是存在的。

(see ISO 26262-1:2011, definitions 1.13 and 1.49).

当确定一个要素的子要素的 ASIL 等级，面受干扰的原理通过连锁故障的相关故障分析得到支持。(see Clause 7 of this part of ISO 26262).

## 6.3 本条款的输入

### 6.3.1 先决条件

下列信息是可获得的：

——在这个等级的安全要求分析：根据

ISO26262-3:2011, 8.5.1, or ISO 26262-4:2011, 6.5.1, or ISO 26262-5:2011, 6.5.1, or ISO 26262-6:2011, 6.5.1 的系统、硬件、软件。

——在这个等级的架构信息分析：根据

ISO 26262-4:2011, 7.5.2, or ISO 26262-5:2011, 7.5.1, or ISO 26262-6:2011, 7.5.1. 的系统、硬件、软件。

### 6.3.2 其他支持信息 无。

## 6.4 要求与推荐要求

6.4.1 这一条款可能被应用在任何细化步骤在设计过程中，与要素和结构的子要素的安全要求分配平行，特别是在按照

ISO 26262-4, or ISO 26262-5, or ISO 26262-6. ISO 26262-4, 或

ISO 26262-5, 或 ISO 26262-6 系统设计阶段、硬件、软件设计，结构设计。

6.4.2 在应用本条款之前，安全要求应当非配给这些要素的子要素。

注意：对子要素的安全要求分配的结果是安全相关的子要素和无 ASIL 分配的子要素。

6.4.3 在要素分析过程中下列信息应当被考虑 A)分配给要素的每个安全要求和 B)这个要素的每个子要素。

6.4.4 如果一个没有 ASIL 的子要素和安全相关的子要素共存在同一个要素，那么，没有 ASIL 的子要素应当那个被对待为一个 QM 子要素，前提是它不是直接或间接违反任何分配该要素的安全要求。例如，它不能干扰该要素的子要素的任何相关安全性。

注意：这意味着，从这个子要素到安全相关的要素的连锁故障是不存在的。

注意：这可以通过设计注意事项(包括软件的数据流、控制流或 I/O 信号和硬件控制线)来实现。

否则，这个子要素应该被分配给共存的安全相关的子要素的最高一级 ASIL，为此免受干扰的情况是不可能的。

(Otherwise, this sub-element shall be assigned the highest ASIL of the coexisting safety-related sub-elements for which evidence of freedom from interference is not made available.)

6.4.5 如果有不同 ASILS 的，包括 QM(X) 的安全相关子要素共存在同一个要素中，那么，子要素应当作为一个有更低一个 ASIL 的子要素对待，前提是对于分配给要素的每个安全要求与任何最高级的 ASIL 子要素不互相干扰的证据是可获得的。否则，当免受干扰的证据是不存在的，这个子要素应当被分配这个共存的安全相关子要素的最高一级 ASIL.

## 6.5 工作成果

6.5.1 ASIL 作为要素的子要素的属性升级

# 7、 关联故障分析

## 7.1 目的

关联故障分析的目的是确定单一事件或单一原因，这个事件或原因能忽略或无效一个要求的独立或免受给定要素之间的干扰和违反一个安全要或安全目标。

## 7.2 通则

关联故障分析考虑了以下的架构特点：

- 相似和非相似冗余要素；
- 同一个软件或硬件要素的不同功能应用；
- 功能和它们相关的安全机制；
- 功能或软件要素的部分；
- 有或没有分离的硬件要素的物理距离；
- 公用外部资源；

根据 ISO26262-1 中给出的定义，独立性被公共故障原因和串联故障原因影响，而免受干扰只被串联故障影响。

例如 1：高密度的电磁场能引起不同电子设备的故障是一种公共故障原因。车速信息不准将影响一个或多个汽车功能是一个串联故障影响的例子。

关联故障能，同时或者在足够短的时间间隔内，表明对同时故障的影响。

例如 2：在检测功能失效前的一点时间之前，一个检测反常功能行为的监测器能指示功能失效，前提是监测器和被监控的功能是由相同事件或原因引起的。

### 7.3 本条款的输入

#### 7.3.1 前提条件

下列信息是可获得的：

-----在他们被应用的等级定义的独立的规范：根据

ISO 26262-3:2011, 8.5.1, or ISO 26262-4:2011, 6.5.1, or ISO 26262-5:2011, 6.5.1, or ISO 26262-6:2011, 6.5.1 要求的系统、硬件或软件。

-----在他们被应用的等级定义的免于干扰的规范：根据

ISO 26262-3:2011, 8.5.1, or ISO 26262-4:2011, 6.5.1, or ISO 26262-5:2011, 6.5.1, or ISO 26262-6:2011, 6.5.1 定义的系统、硬件或软件，以及

-----在他们被应用的等级定义的独立的或免于干扰的规范：根据

ISO 26262-4:2011, 7.5.2, or ISO 26262-5:2011, 7.5.1, or ISO 26262-6:2011, 7.5.1 定义的系统、硬件或软件。

注意：架构信息被用来确定关联故障分析时的界限。

#### 7.3.2 更进一步支持信息 无。

### 7.4 规范和推荐规范

#### 7.4.1 关联故障的潜在性将根据条款 8 中定义的安全分析结果中确定。

注意 1 系统故障和随机硬件故障都对关联故障存在可能性。

注意 2 关联故障的潜在可能性的认定是基于推论分析：cut sets 的执行或一个 FTA 的重复同一事件能为关联故障指明潜在性。

注意 3 认定可以通过归纳性分析获得支持：在一个 FMEA 中相似部分或组件以相似的故障模式多次出现能够为关联故障的潜在性提供足够的信息。

7.4.2 对关联故障的每个潜在可能性将被评价来确定它的因果关系，例如，一个有理由遇见的原因存在导致了关联故障，并且随后影响了一个要求的独立性或在给定要素之间的免干扰性。

注意：当随机硬件失效需要量化时，同时在评价由于随机硬件失效引起违反安全目标时，（参见 ISO26262-5），公共失效原因的权重评价是基于一个定性的基数，因为没有通用或足够可靠的方法存在对这样的失效进行量化。

#### 7.4.3 评价将考虑运行条件以及被分析要素或条款的不同的运行模式。

#### 7.4.4 当评价时，认为以下主题是适用的：

注意 1 潜在关联失效的可能性评价可以通过一个合适的 checklist 来进行，比如，checklist 是基于现场经验。Checklist 提供了根本原因代表性实例和耦合因素的分析，比如，同一个设计，同一个过程，同一个组件，同一个接口，以及相近项。IEC61508 提供的信息能被用来建立这个 checklist 的基础。

注意 2 评价也能通过过程指导获得支持，这些过程指导是用来阻止导致关联失效的根本原因和耦合因素的引入

- A) 随机硬件失效 例如：公用模块失效，如在大型集成电路（微控制器、ASICs等）中的时钟，测试逻辑和内部电压整流。
- B) 开发失效 例如：规则失效，设计失效，应用失效，由于应用新技术产生的失效和制造过程中引入的失效。
- C) 制造失效 例如：与过程、流程、培训相关的失效；控制计划和监控计划失效；软件刷写和线末端编程失效；
- D) 安装失效 例如：与线束走线相关的失效；与组件的内部互换产生的失效；要素或组件靠近引起的失效；
- E) 维修失效 例如：与过程、流程、培训相关的失效；在查找问题时产生的失效；与产品部分的内部替换产生的失效和由于后续不兼容产生的失效；
- F) 环境因素 例如：温度、振动、压力、湿度/液化、污染、腐蚀、玷污、EMC
- G) 公用外部资源失效 例如：供电、输入数据，系统内部数据总线和通信
- H) 由于特定条件下的压力 例如：磨损、寿命。

#### 7.4.5 关联失效有理性的推理和它们的可能影响

注意：在 7.4.2 中给出的关联失效有理性评价已经表明了可预见的原因。

7.4.6 根据 ISO26262-8 中定义的变更管理，关联失效有理性分解测评应当在开发阶段进行。

7.4.7 关联失效有理性分解测评应当包括阻止本质原因或控制它们影响或减少耦合因素的测评。

### 7.5 工作产出

由 7.4 导出的关联失效分析；

## 8、安全分析

### 8.1 目的

安全分析的目的是验证功能、行为、条款设计和要素相关的故障和失效的后果。安全分析也提供了基于条件和原因的信息，这些原因和信息能够引起违反安全目标或安全规范。

此外，安全分析也有助于新功能或非功能性风险的确认，这些风险在风险分析和危险评价过程中没有确认。

### 8.2 通则

安全分析包括：

- 安全目标和安全概念的确认；
- 安全概念和安全规范的验证；
- 条件和原因的确认，包括故障和失效，这能引起违反安全目标或安全规范； ----检测故障或失效的额外规则的确认；
- 检测故障或失效而要求的响应（行动/测量）的确认；
- 验证安全目标或要求兼容性的额外要求的确认，包括安全相关的车辆测试。 安全分析在概念和产品开发阶段的抽象阶段的合适环节开展。定量分析方法指出了失效频率，而

定性分析方法确认失效但是不能指明失效频次。两种分析方法都依靠相关故障类型和故障模型的认知。

定性分析方法包括：

- 在系统、设计或流程阶段的定性 FMEA
- 定性 FTA;
- HAZOP;
- 定性 ETA.

注意 1：上面列出的定性分析方法可以应用在软件中，当没有更多合适的软件特定分析方法存在时。

定量分析补充了定性安全分析。它们都用来验证硬件设计是否违反了对硬件架构矩阵评价时定义的目标以及由于随机硬件失效（参见 26262-5）引起的违反安全目标的评价。定量安全分析需要额外的知识，即硬件元件的定量失效率。

定量分析方法包括

- 定量 FMEA;
- 定量 FTA;
- 定量 ETA;
- Markov 模型;
- 可靠性框图;

注意 2 定量分析方法只能用于随机硬件失效。在 ISO26262 中，这些分析方法不能用于系统失效分析。

对于安全分析分类的其他标准，通过他们的执行方式给出：

----归纳分析方法是由底-顶的方法，从已知的原因开始，预测未知的影响；  
----推理分析方法是由顶-底的方法，从已知的现象开始，推断未知的原因； 例如：  
系统，FMEAs 的设计和过程，ETA，Markov 模型是归纳分析方法。FTA 和可靠性功能框图是  
推理分析方法。

### 8.3 本条款的输入

#### 8.3.1 必要条件

以下信息是可获得的：

-----在他们被应用的等级定义的安全规范：根据  
ISO 26262-3:2011, 8.5.1, or ISO 26262-4:2011, 6.5.1, or ISO 26262-5:2011, 6.5.1, or ISO 26262-6:2011, 6.5.1 要求的系统、硬件或软件。

-----在他们被应用的安全分析等级定义的要素的架构信息：根据  
ISO 26262-3:2011, 8.5.1, or ISO 26262-4:2011, 6.5.1, or ISO 26262-5:2011, 6.5.1, or ISO 26262-6:2011, 6.5.1 定义的系统、硬件或软件，以  
及

注意：架构信息用来确定安全分析的界限。

-----根据 ISO26262-2: 2011, 6.5.1 制定的安全计划； 注意：安全计划包含了  
安全分析的目标。

### 8.3.2 更进一步支持信息

下列信息要被考虑:

----故障模型（来自与外部源）

## 8.4 要求和推荐要求

8.4.1 安全分析应当根据适当的标准和指导意见执行。

8.4.2 安全分析的结果将指明各自的安全目标和安全要求是否兼容。

8.4.3 如果安全目标和安全规范不兼容，安全分析的结果将被用来预防措施、检测、故障或失效的影响缓解测量。

8.4.4 源自安全分析的测量作为在系统级、硬件级、软件级产品开发的一部分执行，分别根据 ISO 26262-4, or ISO 26262-5, or ISO 26262-6.

8.4.5 在产品研发过程中进行的安全分析时新认定的风险（不在安全目标之内）将根据 ISO26262-8 中规定的变更管理中的风险分析和危险评价中进行引入和评估。

8.4.6 用于安全分析的故障模型与对应的开发阶段是一致的，例如，硬件设计，硬件架构矩阵 评估和由于随机硬件失效（ISO26262-5）导致的违反安全目标的评估。

8.4.7 额外的安全相关的测试实例需求是通过应用故障模型和安全分析结果决定的。

8.4.8 安全分析的姐欧股根据 ISO26262-8 验证

8.4.9 定性安全分析将包括

A) 可能引起违反安全目标和安全要求的故障或失效的系统级认定，起源于：

----条款或要素本身 (the item or element itself;) 或

----条款或要素内部之间的互相作用 或

----条款或要素的应用；

B) 每一个确定的故障的后果对确定违反安全目标或安全要求的潜在性的评价。

C) 每一个认定故障原因的认定 和

D) 潜在安全概念弱点的认定或对认定的支持：包括在处理如潜在的故障、多点故障，公共原因失效和串行失效等异常时安全机制失效。

注意： 在条款或要素之间，条款的内部或外部的相互作用的测试应当要执行，目的是评价独立性或干扰的等级。

8.4.10 如果应用定量安全分析，那么如下内容要包括：

A) 硬件架构矩阵评价和由于随机硬件失效引起的违反安全目标的评价而需要的定量数据（参见 iso26262-5）

B) 可能导致违反安全目标或安全规则的故障或失效的系统级评定；

C) 潜在安全概念弱点的评价和排序，包括安全机制失效，和

D) 诊断测试间隔，紧急操作间隔，在故障检测和维修之间的时间。

8.4.11 如果定性安全分析应用来支持与定量分析要求相兼容，在这些安全分析的细节成度 应到合适选择。

## 8.5 工作成果

8.5.1 安全分析，来自 8.4.

## ISO26262-10 指南