Vector 软件
白 皮 书

使用VectorCAST来满足ISO 26262软件验证和确认标准

目的

本文的目的是为了展示来自于Vector软件的VectorCAST产品是如何用来满足ISO26262标准规定的验证和确认要求的，而ISO 26262标准本身就是从IEC 61508派生出来的。本文不打算对标准作详尽的回顾，而仅仅是提供了一个高层次的视角来看待使用VectorCAST能实现的不同的需求。如需更多信息，请联系Vector软件。

介绍

汽车工业的竞争非常激烈。成功的企业必须通过引入新的功能不断地创新，这其中包含了大量的软件。汽车已经从主要是机械装置转化为了在所有主要的系统中都带有嵌入式软件的集成设备了，这些主要的系统包括：发动机控制系统、动力传动系统、悬挂系统、制动系统和娱乐系统。

汽车嵌入式系统的成本控制对汽车工业的供应商而言非常重要，因为汽车工业中使用的软件数量远高于其他安全关键行业，比如航空电子设备和铁路行业。

软件测试历来是非常费时费力的，但现在发现软件缺陷的成本相对于召回造成的直接成本损失及产品品牌损失要少得多，这使得在汽车工业中进行彻底的测试是很有必要的。

汽车软件验证和确认标准

目前有 **MISRA** 和 **ISO 26262** 两个软件标准应用于车辆软件的验证和确认。

MISRA

自从汽车工业软件可靠性协会（MISRA）C标准问世以来，静态分析一直是汽车应用开发过程中很大的一部分。“***车辆软件C语言使用指南***”是1998年首次出版的，用于促进汽车产业安全使用C语言的文献。它包含了一些规则，这些规则定义了一个现已被普遍认可的，作为一个良好编程实践典范的C语言子集。“***MISRA C++：2008关键系统C++语言使用指南***”发表于2008年，用于为C++语言定义类似的规则。

ISO 26262

ISO 26262是一个目前还在制定中的功能安全标准，被称为“道路车辆——功能安全”标准。该标准是对汽车电气/电子系统功能安全标准IEC 61508的改进。ISO 26262标准的第6部分对作为软件开发标准一部分的动态软件测试和验证提出了规范建议。

建议的活动同时包括了**单元级**和**系统级**测试，比如**功能测试**（基于需求的测试和分区测试）和**结构覆盖测试**。

VectorCAST 如何支持符合 ISO 26262 标准

针对 ISO 26262 的 VectorCAST 嵌入式测试工具满足软件开发标准第 6 部分提出的软件测试和验证规范建议，通过支持测试用例的创建和管理来证明底层软件需求已经被测试过。

VectorCAST 也可用于各种健壮性测试活动，比如值域和边界溢出测试。

此外，VectorCAST工具支持捕捉和报告ISO 26262规定的全部汽车安全完整性等级（ASIL）的结构代码覆盖率。

ASIL 是汽车行业确定产品风险等级的明确风险值。风险级别被定义为 *A* 级到 *D* 级，*ASIL D* 级代表的风险最高。

用于C和C++单元和集成测试的VectorCAST/C++测试工具，与用于系统级测试验证的VectorCAST/Cover相结合，为主机、模拟器和目标级测试提供了一个完整的动态测试套件。

请注意，*ISO 26262*目前处于验收的最后阶段，个别规范在确认之前可能还会有变动。

本文使用以下的文字说明：

R 推荐的活动
HR 强烈推荐的活动

ISO 26262 提出了一个瀑布式的测试方法，明确划分了单元测试活动和系统测试活动。因此，本文将这两个级别的测试分开进行说明。

软件单元测试期间 VectorCAST 的使用

软件单元测试方法

第*9*节定义了单元测试的目标是证明该软件单元满足软件单元规格说明书且不包含不需要的功能。

为了实现这一目标，该标准建议执行下面的单元测试方法（见 *ISO 26262* 表 *12* - 软件单元测试方法）。

方法	ASIL				VectorCAST能否支持
	A	B	C	D	
1a.基于需求的测试	HR	HR	HR	HR	✔
1b.接口测试	HR	HR	HR	HR	✔
1c.错误注入测试	R	R	R	HR	✔
1d.资源使用测试	R	R	R	HR	✔ [1]
1e.模型和代码之间的背靠背测试（如适用）	R	R	HR	HR	✔ [2]

ISO26262表12 - 软件单元测试方法

[1] 如果在调试器的控制下或者与其它工具一起使用*VectorCAST/C++*

[2] *VectorCAST*能使用来自模型的数据来生成测试用例。

*1a.*基于需求的测试

基于需求的测试是在有软件需求的地方（比如功能性、健壮性等），首先建立起来的一种方法。执行代码，测试用例与具体的需求相关联来证明代码满足了这些需求。所有可测试的需求都应该被测试。基于需求的测试对ASIL所有等级都是被强烈建议的，它在关键行业中（包括航空电子设备）是大部分软件标准的基础。

用于单元测试的VectorCAST/C++使基于需求的测试尽可能的高效。测试环境由被测试的代码和相关的测试代码组成，这些代码通常表现为驱动程序和桩函数。这种环境是由VectorCAST自动生成的。

在VectorCAST/C++中，通过图形用户界面（GUI）的使用，使创建基于需求的测试变得非常的简单。该图形用户界面可以指定参数的输入和期望值，返回值，全局变量甚至桩函数（取消关联关系的一段代码，以确保更大程度的隔离）。测试用例的创建也可以通过从一个逗号分隔值（CSV）文件导入数据来完成。

使用VectorCAST/Requirements Gateway（VectorCAST/C++的一个模块），每个测试用例也可以链接到一个从需求管理工具（如DOORS）上面下载的具体的软件需求。一旦建立链接，也可以将一些信息和测试状态信息一起，上传到需求数据库，从而更轻松地监控哪些需求得到满足，哪些没有。

1b.接口测试

ISO 26262 同样强烈建议全部 ASIL 等级都要测试所有单元的接口。这可以通过各种方式来完成，例如，通过使用接口的边界值，非法值和中间值。这些值也可以通过不同的方式组合在一起。这里的目标是确保该接口是健壮的，以防发生错误。

所有这些活动使用VectorCAST/ C + +都可以很轻松地执行。极限值可以根据函数的值域或类型来指定，非法值可以很容易地指定。在目标环境中，VectorCAST/ C + + 充分测试了每个数据类型的准确值域，这样就得到了含有精度的边界值，并且可用它来自动生成边界测试用例。这些测试也可以通过一个组合模式来完成，完全自动地增加了来源于一个单一数据集的有效测试用例的数目。

1c. 错误注入测试

单元测试可用来自发地引入错误，包括损坏指定变量的值，以测试嵌入在单元中的安全机制。这种测试尤其是在ASIL D级中被建议执行。VectorCAST/C++提供了这种测试类型的功能。扩展到库调用自己的桩函数功能，可以用于在函数中间引入故意的错误。如果需要更大的灵活性，测试用例也可以在底层调试器的控制下运行，使变量值在不同的点适时改变。

1d. 资源使用测试

使用针对C和C + +的VectorCAST单元测试工具，在单元级的资源使用测试可以被部分自动化。利用VectorCAST/RSP模块，测试可以从目标板（或模拟器）运行。此外还可以使用附加的工具来测量资源消耗。

1e.模型和代码之间的背靠背测试

VectorCAST 工具已被终端客户用于从 Simulink®和 Rhapsody®的模型中自动生成测试代码。Vector 与这两个产品的发行商联系，使单元测试自动生成代码的过程更加无缝。

获得软件单元测试用例的方法

除了这几种不同的单元测试方法以外，ISO 26262 还建议了在单元测试期间生成测试用例的四种不同的策略。它们在下面列出（见 ISO 26262 表 13 – 获得软件单元测试用例的方法）。

方法	ASIL				VectorCAST 能否支持
	A	B	C	D	
1a.需求分析	HR	HR	HR	HR	✔
1b.等价类产生和分析	R	HR	HR	HR	✔
1c.边界值分析	R	HR	HR	HR	✔
1d.错误推测	R	R	R	R	✔

ISO26262表13 – 获得软件单元测试用例的方法

单元级的基于需求的测试是符合 ISO 26262 标准的核心，因为它是被所有的 ASIL 等级都强烈的。

该标准还根据代码本身的结构对测试作出了规定（等价类产生和分析及边界值分析）。不管是测试一个系统到 ASIL B 级，C 级还是 D 级，该活动都是被强烈推荐的。

1a. 需求分析

正如在上一节关于基于需求的测试中所讲，单元测试用例可以从底层软件需求中获取。在VectorCAST中，这些测试用例可以链接到指定的需求，并且如果存在一个需求管理系统，那么可以把它们的状态输出到该系统。

1b.等价类产生和分析

这一策略的目标是通过在输入域上确定使用该软件需要的分区来充分地测试软件。这些测试用例旨在充分地测试程序。它们要么根据软件规范，要么根据程序内部结构（或两者一起）来建立。

这些类型的活动在VectorCAST/C++中也被高度自动化了，使用户能够根据值的范围和列表快速地创建测试用例。这些输入能以一个非组合、线性的模式来执行，或者工具也能使用所有可能的输入组合来运行测试。无论是在一台主机、模拟器还是目标环境中，这些复杂的测试用例都是自动执行完成的。

VectorCAST 还设有一个分区测试用例生成器，用来在提供的域上自动地创建额外的测试用例。

*1c.*边界值分析

这些测试的目标是消除在参数极限或边界处潜在的软件错误，这些地方是最容易出错的。根据ISO 26262表13，应被测试的值包括等于、接近和超过接口边界的值。

无论是变量类型的值域还是函数的值域，在VectorCAST/ c ++中这都可以很容易地做到。它还可以通过自动地生成MIN-MID-MAX测试用例来进一步自动化，分别设置所有的值为它们的最小值、中间值和最大值。最小值和最大值是通过在目标板或模拟器上测试程序中所存在的每一种数据类型的值域来确定的。因此，无论在目标板上还是在模拟器上使用该工具都将保证在系统中通过自动生成的MIN-MID-MAX测试用例来测试的边界值的范围是有效的，无论它是8位、16位还是32位。

这两个工具还可以测试近似值，非法值，甚至特殊值， 比如非数字（NaN）、浮点变量的正负无穷大。

1d. 错误推测

根据 *ISO 26262* 表 *13*，这些测试可根据凭“积累的经验”和专业的判断所收集到的数据来建立。对于 *ASIL A* 级，*B* 级和 *C* 级它们是被推荐的，且如果要测试系统符合 *ASIL D* 级，它们则是被强烈推荐的。它们的目标是确保软件的组件尽可能的防错。

使用 VectorCAST，创建一个测试用例是一件简单的事情，不需要写脚本。可以抛出异常信号，可以自动的保留不被初始化的指针来看看代码是否会防止这种类型的错误的发生。执行“假定推测”的场景很容易产生，而采用手工测试或基于脚本的工具将需要开发大量的测试代码。

所有的测试用例都存放在测试套件之外直到需要用的时候，这意味着测试用例可以被建立和删除而不需要重编译代码，与其它“类似的”工具相比，最大限度地提高了生产率。

软件单元级的结构覆盖度量

ISO 26262 强烈建议使用代码覆盖率作为度量来衡量在一个给定的单元上是否执行了足够的测试。由于将系统作为一个整体测试时（使用语句覆盖），不可能达到 100%的代码覆盖率，ISO 26262 进一步建议在单元测试期间，至少每一行代码都要被充分的测试过。

这样做的目的显然是想按照选定的覆盖标准达到 *100%*的覆盖率。根据 *ISO 26262* 表 *14*，注解 *4*，对于达到的覆盖水平将会给出一个基本原因。（比如：因为根据不同的软件配置，可接受的死代码或代码段），或者其它没有覆盖的代码可以使用互补方法来验证（比如：检查）。

在单元级，对于代码覆盖率，有三种不同的标准可供选择，如下表所示（见 ISO26262 表 14 - 软件单元测试的结构覆盖度量）

方法	ASIL				VectorCAST 能否支持
	A	B	C	D	
1a.语句覆盖	HR	HR	HR	HR	✔
1b.分支覆盖	R	HR	HR	HR	✔
1c. MC/DC (修正条件/判定覆盖)	R	R	R	HR	✔

ISO26262表14 - 软件单元测试的结构覆盖度量

应当指出的是，这些覆盖水平越来越难以实现。当使用 VectorCAST 这样的工具时（它使得用户能够独立地测试 MC / DC，分支和语句覆盖），为了实现兼容了 ISO26262 字面文字和内在精神的结构性覆盖水平 ，应使用下面的标准组合。

- 仅对*ASIL A*级的语句覆盖
- 对*ASIL B*级和*C*级的语句和分支覆盖
- 对*ASIL D*级的语句，分支和*MC/DC*覆盖

VectorCAST/C++ (以及 VectorCAST/Cover)有一个简单实用的代码覆盖率查看器。覆盖率查看器通过符号和颜色码来表示代码（a）是否被完全覆盖（用绿色），（b）是否被部分覆盖（用橘色），或(c)是否未被覆盖（用红色）。将光标留在所有被覆盖的行的任何地方，使用户能够明白测试用例具体覆盖了哪些行。

在目标环境具有足够内存和宽松的时间约束的情况下，代码覆盖率甚至可以是动态的——逐步地“回放”如何在执行过程中获取代码覆盖率。对于更受限制的环境，代码覆盖率也可以以节省内存空间和/或受时间影响较小的模式来运行。VectorCAST代码覆盖率工具和数据收集有几个选项，允许用户为特殊应用自定义最大资源利用效率。

*1a.*语句覆盖

语句覆盖，试图在一个给定的单元中覆盖个别的代码行。例如，覆盖下面的行需要一个单独的测试用例。

```
if(i < 10 && j == 0 || k > 12)
```

应当注意一个单独的测试用例使条件判定为false将不会覆盖包含在‘IF’语句中的其它代码行。同样，如果这个‘IF’语句有一个附属于它的‘ELSE’语句，一个判定为true的测试用例也不会覆盖‘ELSE’语句的内容。然而，无论是true还是false的测试用例都会覆盖‘IF’语句本身。

VectorCAST/C++ (以及 VectorCAST/Cover)支持语句覆盖，无论是单独的还是与分支和/或与 MC/DC 覆盖组合在一起的。此外，VectorCAST/C++中的自动生成测试用例功能可以帮助工程师设计达到最大化语句覆盖的测试用例。例如，根据基本路径（代码中的独立路径）自动地生成测试用例通常会提供一个高度的语句覆盖。

*1b.*分支覆盖

分支覆盖集中在判定点的状态上，它可以是 true 或 false。例如，覆盖下面这行代码将需要两个测试用例—— 一个'IF'语句返回真，另一个返回假。

```
if(i < 10 && j == 0 || k > 12)
```

VectorCAST/C++ (以及 VectorCAST/Cover)完全支持分支覆盖，无论是单独的，还是与其它代码覆盖标准组合在一起。为了符合 ISO 26262，VectorCAST 能在单个测试执行期间同时产生语句和分支覆盖水平。

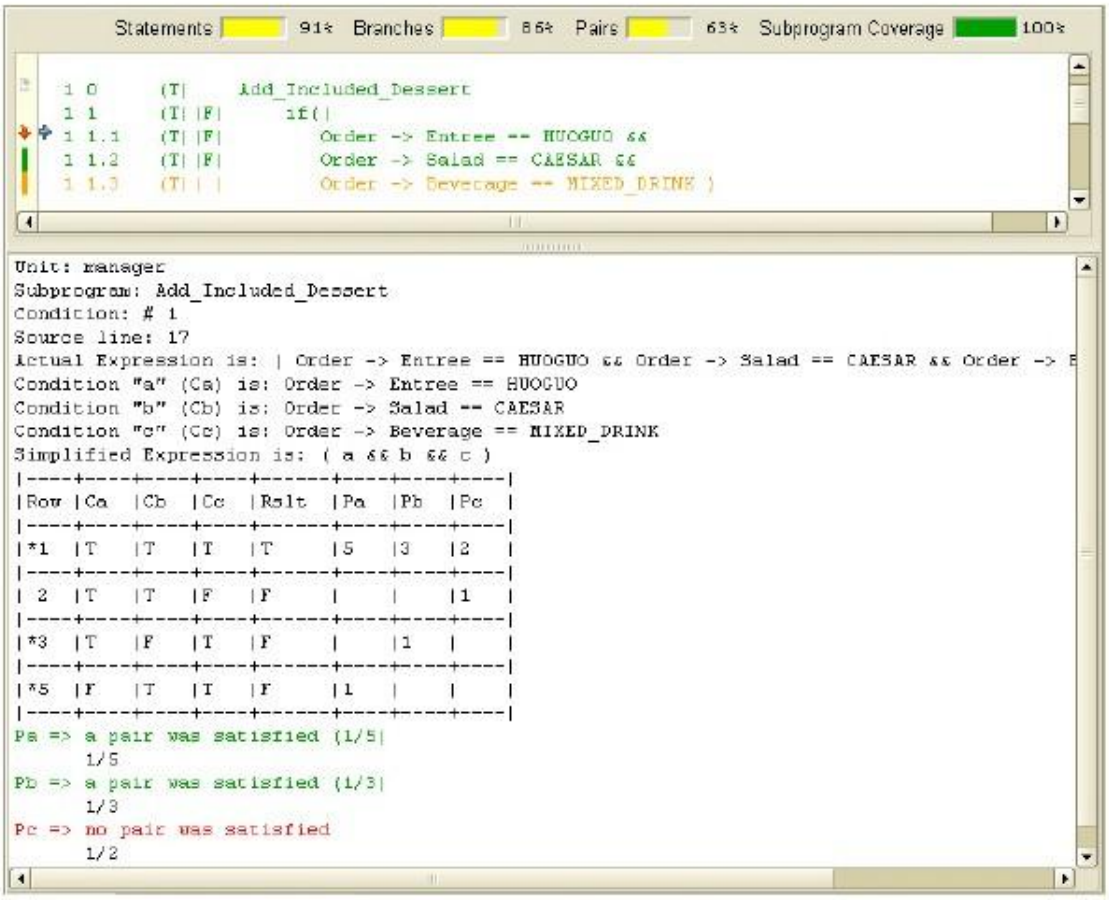
1c. MC/DC (修正判定/条件覆盖)

MC/DC 需要最大数量的测试来达到覆盖要求。这种覆盖标准表明所有作为条件语句一部分的子条件都能独立影响条件语句本身的结果。例如，下面语句的情况：

```
if(i < 10 && j == 0 || k > 12)
```

有一点可以证明，通过改变'i'的值，同时保持其它子条件的值不变，最终的值将会改变。

即使是最有经验的工程师，要完成这个任务也是非常艰巨的。然而，VectorCAST 通过它的VectorCAST/MCDC模块提供了一个非常有效的方式来完成这种类型的测试。自动生成的真值表清楚的显示了需要哪些测试用例组合来完成 MC/DC 覆盖，然后标记出提供了哪些测用例和测试用例组合。



在目标板或者模拟器上执行

ISO 26262 强烈偏向于在尽可能与程序要运行设备的真实环境相接近的环境中执行单元测试用例。正如在 9.4.5 节中提到的，软件单元测试的测试环境应尽可能与目标环境一致。如果软件单元测试不是在目标环境中执行，源代码和目标代码之间的差异，以及测试环境和目标环境之间的差异，应该被分析出来以便在后面的测试阶段在目标环境中指定额外的测试。

VectorCAST/C++加上 VectorCAST/RSP (运行时支持包)是符合 ISO 26262 本节要求的理想工具。通过使用 RSP 模块，VectorCAST 能在终端目标上（或者模拟器在资源贫乏的环境中）自动地执行测试用例。VectorCAST/RSP 对接口专门设计了一套独特的编译器，目标板，调试器和 RTOS 实时操作系统（如果有）的组合，这使得它在几乎没有用户输入的真实目标上运行测试用例。在命令行重运行测试用例（回归测试）也是 100%自动的。

软件集成和测试期间*VectorCAST*的使用

第 10 节定义了对应测试软件架构设计的特殊*集成层级*。这些集成层级建立在软件分层结构之上——从最底层（测试个体单元）到最高层（将软件作为一个整体测试）。

虽然 ISO26262 没有作区分，但将测试模块（由代表一个功能过程，而不是整个软件的单元组合而成）看作是模块测试或者集成测试（www.vectorcast.com上使用的术语）是有利的。在整个应用程序上进行的测试通常被称为系统测试。

在 ISO26262 中，这个过程应该是根据特定的软件层次将不同的单元逐步的集成在一起，直到嵌入的软件完全集成到系统级（10.4.1 节）。因此，建议模块测试和系统测试都要进行。

软件集成和测试以及获得测试用例的方法

跟在单元测试阶段一样，也推荐了一些软件集成测试的方法。它们被列在下面（见 ISO26262 表 15 - 软件集成测试方法）。

方法	ASIL				VectorCAST 能否支持
	A	B	C	D	
1a.基于需求的测试	HR	HR	HR	HR	✔
1b.接口测试	HR	HR	HR	HR	✔
1c.错误注入测试	R	R	HR	HR	✔
1d.资源利用测试	R	R	R	HR	✔ [3]
1e.模型和代码之间的背靠背测试（如适用）	R	R	HR	HR	✔ [4]

ISO26262表15 - 软件集成测试方法

同样地，获得软件集成测试用例的方法在下面列出（见 *ISO26262* 表 *16 –* 获得软件集成测试用例的方法）

方法	ASIL				VectorCAST 能否支持
	A	B	C	D	
1a.需求分析	HR	HR	HR	HR	✔ [5]
1b.等价类产生和分析	R	HR	HR	HR	✔ [5]
1c.边界值分析	R	HR	HR	HR	✔ [5]
1d.错误推测	R	R	R	R	✔ [5]

ISO26262表16 – 获得软件集成测试用例的方法

这些方法跟在单元测试部分中定义的完全相同。在本节中，我们将重点关注VectorCAST专门针对集成测试的有用功能。

[3] 如果从对调试器的控制执行测试用例。

[4] *VectorCAST* 可以使用来自模型的数据生成测试用例。

[5] 只适用于 *VectorCAST/C++*

如何搭建一个集成测试环境

VectorCAST/C++可以用于在模块/集成测试期间运行集成测试用例。它的用法跟在单元测试阶段所描述的相同——唯一的变化是用来构建测试套件的是一些包含了一个功能过程的文件而不是一个单一的文件。

VectorCAST/C++提供真正的集成测试——放在一起的文件本质上成为了一个更大的集成模块。因此，如果在第一个单元的一个函数上执行了测试，如果这个单元本身调用了第二个单元的另一个函数，这两个函数都将被测试且都将生成代码覆盖率。跟单元测试的情况一样，不用编写生成集成测试环境的脚本—— 一切都是自动生成的。

从单元测试中重用测试用例

VectorCAST 先进的回归功能在集成测试过程中也非常有用。基本上，它可以重用单元测试期间设计的部分或全部的测试用例，并将它们无缝地应用于集成测试。代码可能已经改变，但测试用例的运行没有任何问题（所以只要函数接口保持不变，否则测试用例将被丢弃或标记为不再相关）。导入和导出测试用例的功能也是非常节约时间的。

测试来自第三方供应商的软件模块（库）

此外，VectorCAST/C++可以用来测试库，甚至不需要访问代码。当集成的软件组件来自第三方，比如供应商和承包商时，这也同样有用。建立一个围绕一个库文件的测试用例和建立一个围绕真实代码的测试套件一样简单，测试活动可以以一个非常类似的方式来执行。

系统测试

一旦集成达到系统级，一般的编译系统通常会接管，测试用例的输入将通过其它工具来完成，比如信号发生器，控制台按钮，或者可能用一个模拟器。在系统测试期间，VectorCAST/Cover 用来捕捉代码覆盖率（见下面两节）。

软件架构级的结构覆盖度量

函数覆盖是指每个函数在测试期间至少被调用一次。调用覆盖要求每个函数调用至少执行一次。10.4.6 节进一步规定未指定的软件组件必须被识别出来并将其删除或者取消激活。

方法	ASIL				VectorCAST 能否支持
	A	B	C	D	
1a.函数覆盖	R	R	HR	HR	✔
1b.调用覆盖	R	R	HR	HR	✔ [6]

*ISO26262表17 -*软件架构级的结构覆盖度量

[6] *VectorCAST 2012* 将支持

VectorCAST/Cover（以及 VectorCAST/C++）捕捉函数覆盖。此外，调用覆盖可以通过局部语句覆盖来间接完成，它会激发集成或系统测试环境中存在的每个函数的所有调用。

执行集成测试

如之前单元测试所述，ISO 26262 强烈偏向于在尽可能与程序要运行设备的真实环境相接近的环境中执行单元测试用例。这是使用 VectorCAST/C++ 和 VectorCAST/Cover 都可以完全实现的。

关于 VectorCAST/Cover 的一个常见的问题是问该工具是如何从不同的目标导出代码覆盖率数据的。事实上，该工具可以通过三大方法导出覆盖率数据：（a）保存到一个文件（对存在一个文件系统或可以建立一个模拟的文件 I/O 的情况特别有用），（b）通过端口发送（如一个串行端口），还有（c）将数据储存在内存缓冲区，然后通过调试器或其它手段的帮助保存下来。

报告

ISO26262 规定了一些文档的创建，比如软件验证说明书和软件验证报告。

VectorCAST产品创造了各种基于单元测试，集成测试，或系统测试的测试产物。这些报告可以生成为Text格式或HTML格式。它们可以保存在VectorCAST之外，并已用于符合各种标准，比如：IEC61508，CENELEC，DO-178B等。

认证

VectorCAST 和 VectorCAST/Cover 都能被认证为符合 ISO 26262 的活动。更多信息请联系 Vector 软件。

其它工具

总之，VectorCAST/C++, VectorCAST/Cover 和 VectorCAST/Requirements Gateway 可以在某种程度上自动化测试和代码覆盖活动，这使符合 ISO_26262 的要求更加有效。所有这些工具都可以在 HTML 或 Text 中导出它们单独的报告，它已在过去成功用于满足许多苛刻的工业标准。

VectorCAST/RSP 是能够在模拟器或目标板上执行一个测试套件的模块。这个过程是完全自动化的，所以测试案例可以被单独的执行或通过简单的点击鼠标作为一个整体来执行，或通过命令行执行。执行本身不需要用户输入。

此外，VectorCAST/Manage 为所有 VectorCAST 生成的测试提供了一个完全自动化的回归测试功能。

在安全关键市场的成功

VectorCAST工具成功地用于企业开发安全与关键任务应用程序已经超过了15年的历史，如：军事，航空，航天，医疗器械，铁路，汽车，工业控制。数百家企业使用VectorCAST以满足即将应用于符合ISO26262的汽车市场的嵌入式软件测试的要求。

关于Vector 软件

Vector 软件股份有限公司是全球领先的自动化软件测试工具的独立供应商，为开发人员提供安全关键性的嵌入式应用程序。Vector 软件的 VectorCAST 产品线，自动化管理有关于单元、集成及系统级测试的复杂任务。VectorCAST 产品支持 C, C++ 和 Ada 编程语言。

Vector Software, Inc.
1351 South County Trail, Suite 310
East Greenwich, RI 02818 USA
P: 401.398.7185
F: 401.398.7186
E: info@vectorcast.com
W: vectorcast.com