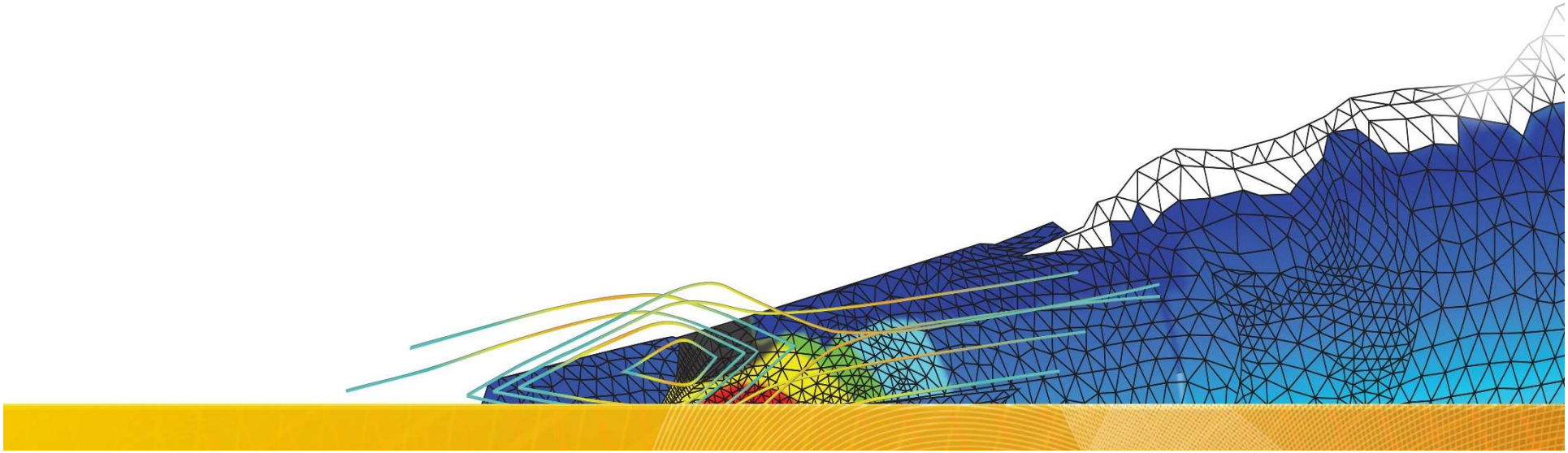
Realize Your Product Promise®

满足ISO 26262的功能安全与嵌入式软件开发

Critical Systems & Software

Development Solutions



Why Safety Become More And More Important

■ 新技术带来的新的安全性要求

- ❖ 新的动力技术
- ❖ 自动驾驶技术
- ❖ 车联网技术

■ 越来越多的电子电器系统将成为强安全相关的系统

- ○ 整车控制器（新能源）
- ○ 电池管理系统（新能源）
- ○ 电机控制器（新能源）
- ○ ESP/ESC（车身稳定控制系统）
- ○ LDWS（车道偏离预警系统）
- ○ CCAS（汽车防撞雷达系统）
- ○ SRS（安全气囊）
- ○ EMS/CATS（自适应悬架控制）
- ○ EBS （电子制动系统）
- ○ ASR（牵引力控制系统）
- ○ BAS（制动辅助系统）
- ○ EBD（电子制动力分配系统）
- ○ EBA（紧急制动辅助系统）
- ○ ACC（自动巡航系统）
- ○ BCM（车身控制系统）

More and more.....

Safety Is a Big Challenge for Non-Traditional Vehicle Industry

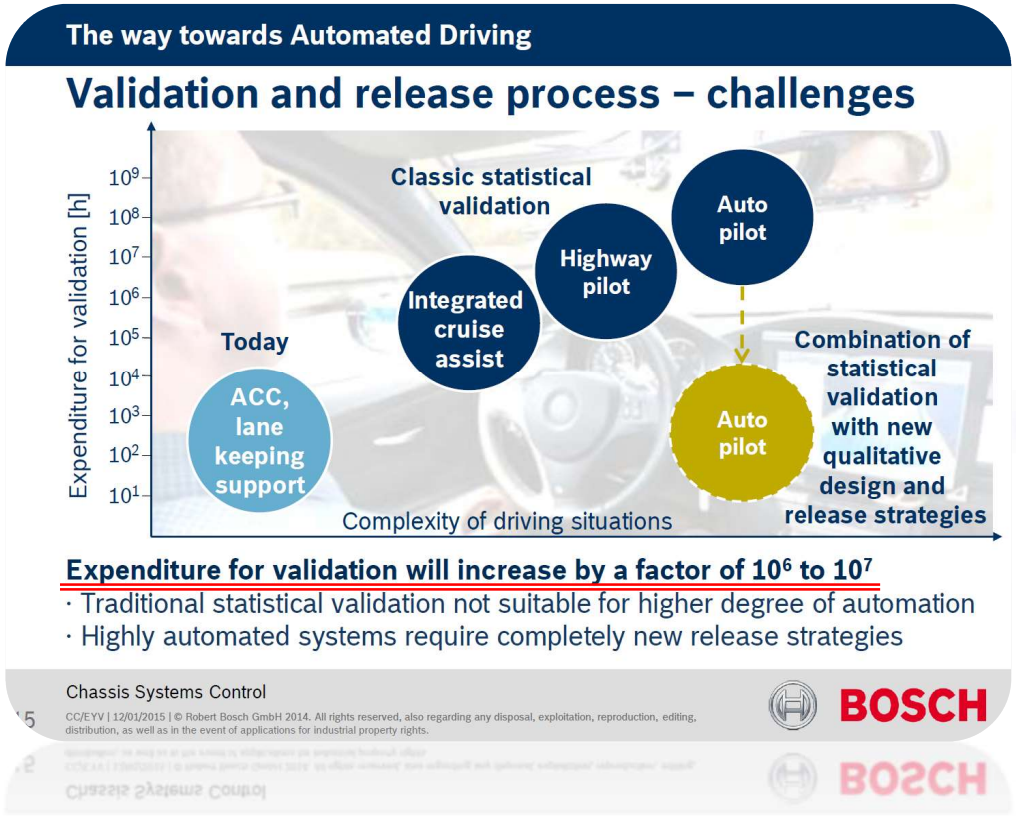
丰田刹车门事件

特斯拉ADAS失效和电池起火

美国亚利桑那州 Uber行人致死事件

Safety Technology

■ BOSCH关于传统验证手段对于自动驾驶技术有效性的预测



safety architecture

functional safety concept

hazard analysis

controllability analysis

failure mode and effects analysis

hardware diagnostic coverage metrics

change management

risk assessment

Configuration management

fault tree analysis

safety validation

safety goals

safety requirements

driving situation analysis

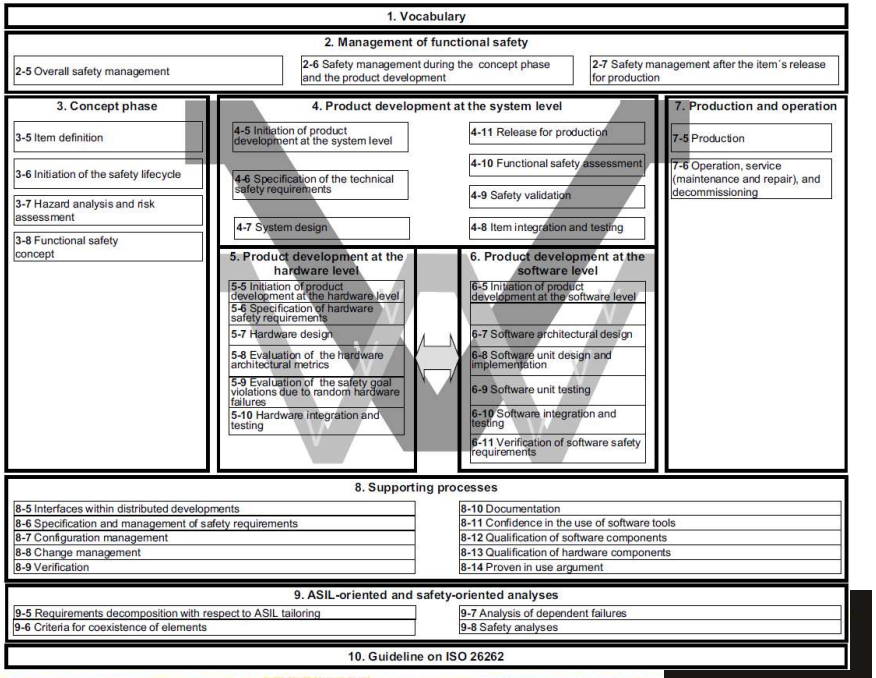
Safety standards like ISO 26262 require to perform multiple analysis methods in a consistent, thorough manner

ANSYS

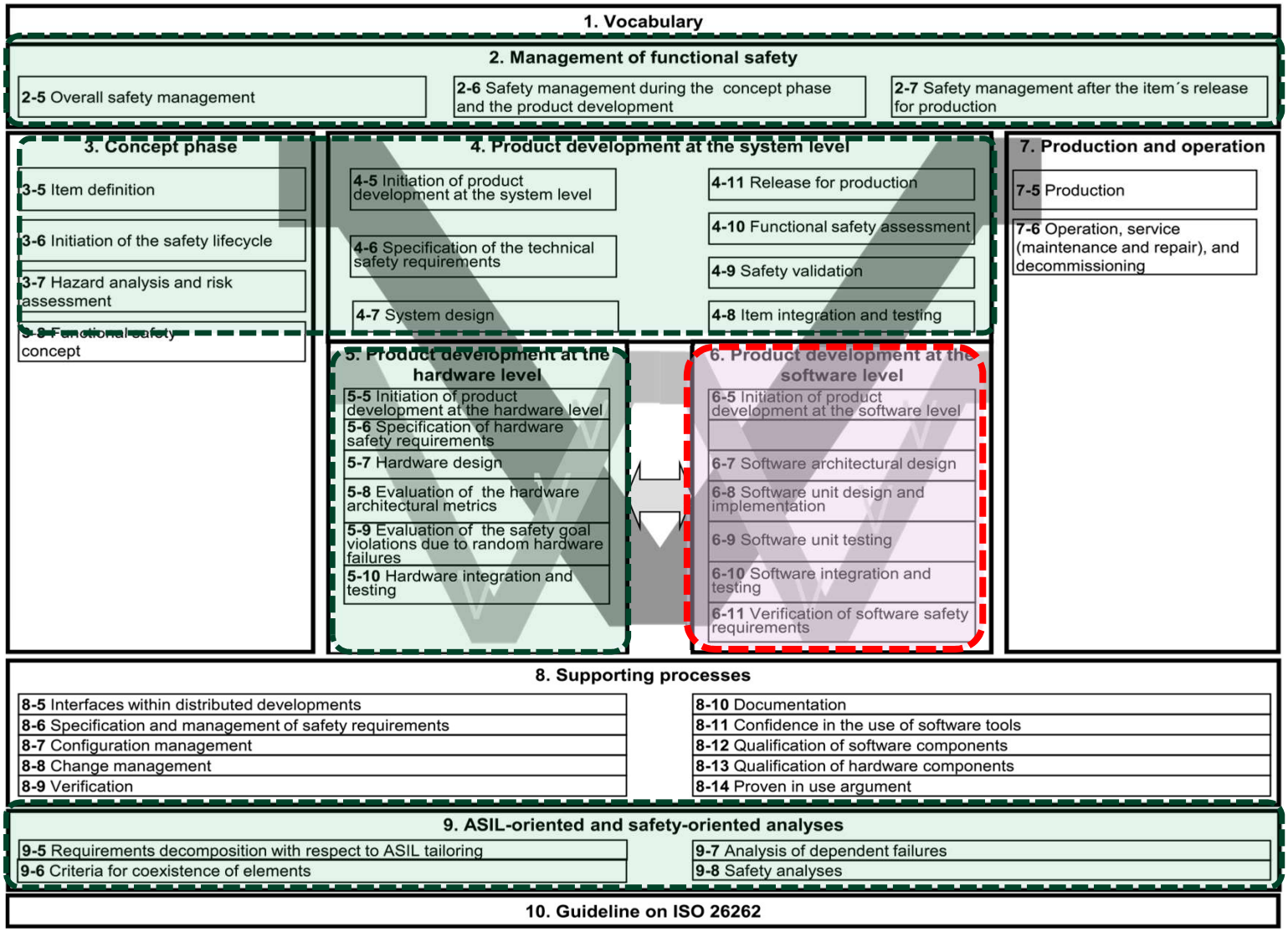
ISO 26262 Standard

■ ISO 26262是汽车的一个安全性国际标准

- ISO26262是从电子、电气及可编程器件功能安全基本标准IEC61508派生出来的。标准主要定位在汽车行业中特定的**电气器件、电子设备、可编程电子器件**等专门用于**汽车控制领域**的**部件和系统**。
- 标准的核心价值在于：通过系统的**功能安全研发管理流程**，以及针对汽车电子控制系统**硬件和软件**的**系统化验证和确认方法**，**保证电子系统的安全功能在面对各种严酷条件时不失效**，从而保证驾乘人员以及路人的安全。

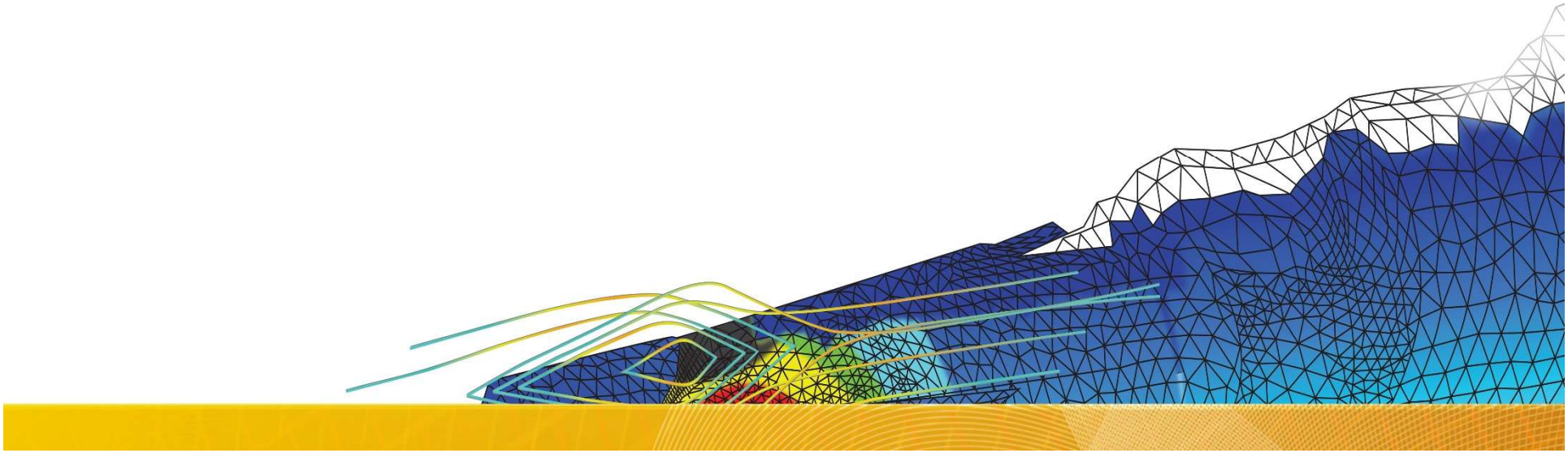


ANSYS Position In ISO 26262



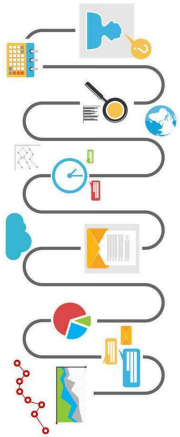
Realize Your Product Promise®

ANSYS MBD for Functional Safety



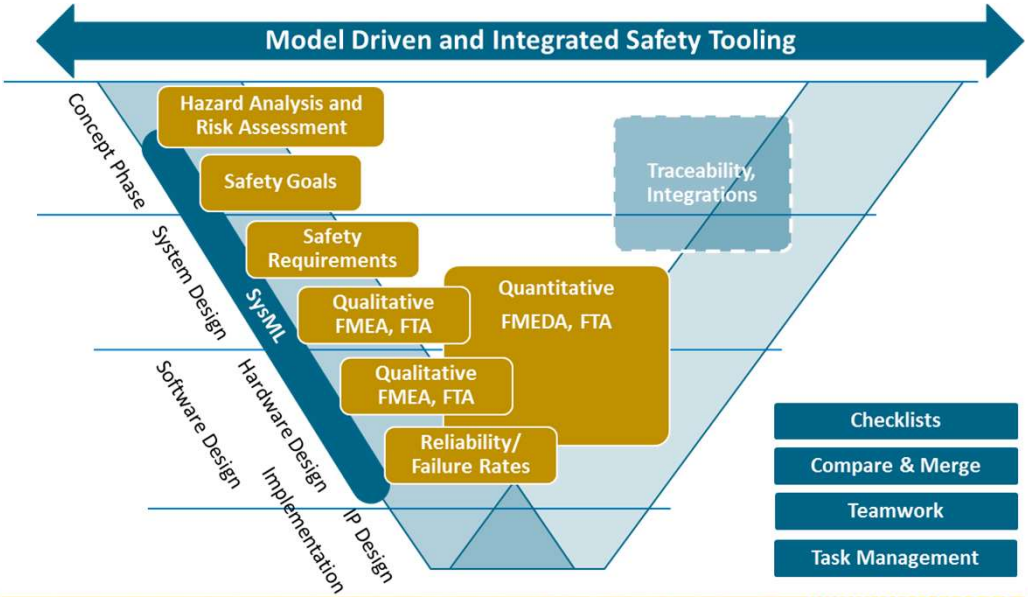
Tell You WHY From Safety Perspective

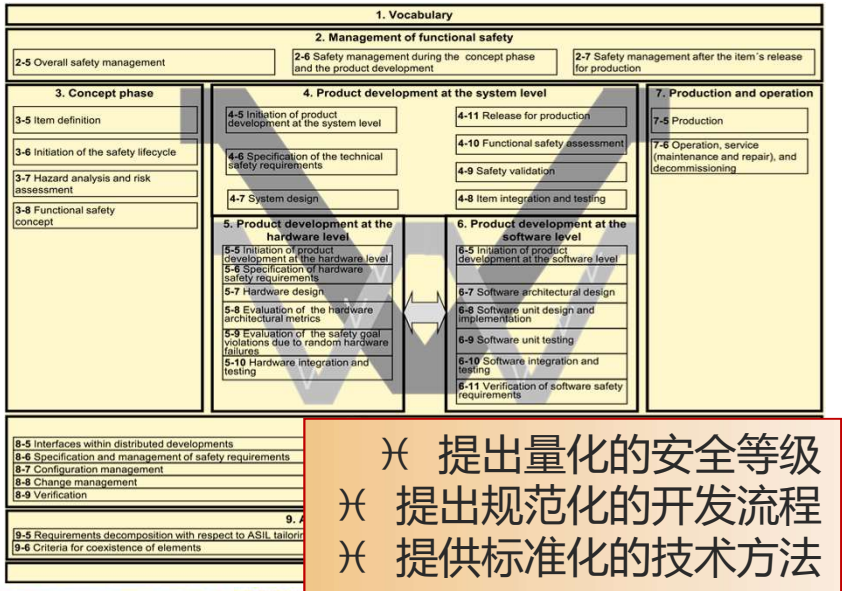
■What activities are executed in FuSa process



- 安全管理计划
- 系统定义（边界，架构和功能）
- 危害和事故识别（原因和后果）
- 安全需求开发及其验证
- 降低风险至可接受水平并关闭危险
- 根据证据确定系统安全的逻辑论据
- 系统安全的限制和最终结论

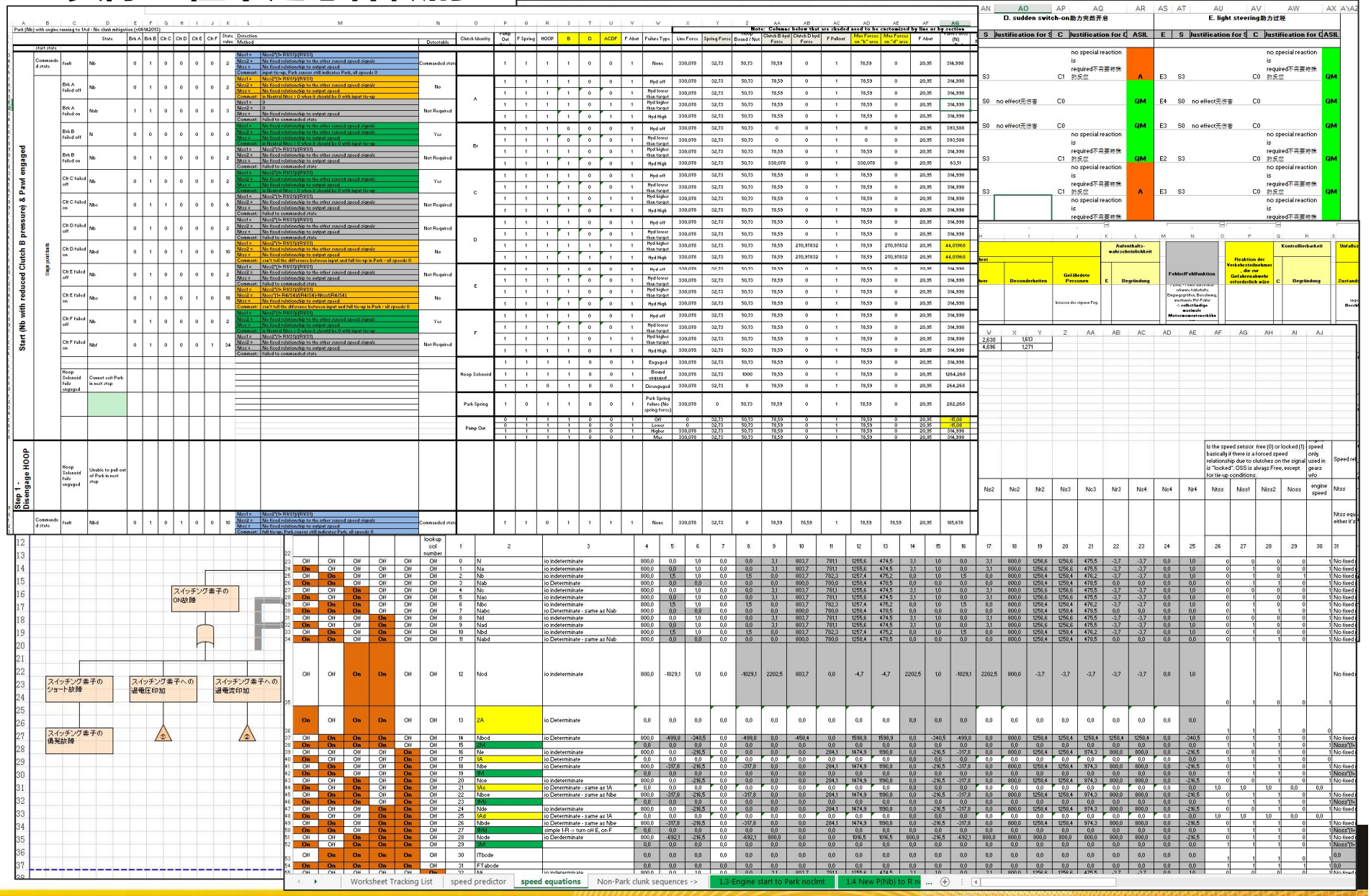
- 危害和可操作性分析（HAZOP）
- 危害分析和风险评估（HARA/PHA/FHA）
- 安全完整性等级确定（SIL/ASIL）
- 故障模式和影响分析（FMEA/FMECA）
- 故障模式，影响和诊断分析（FMEDA）
- 故障树分析（定性&定量FTA ）
- 可靠性和概率计算（硬件指标计算）





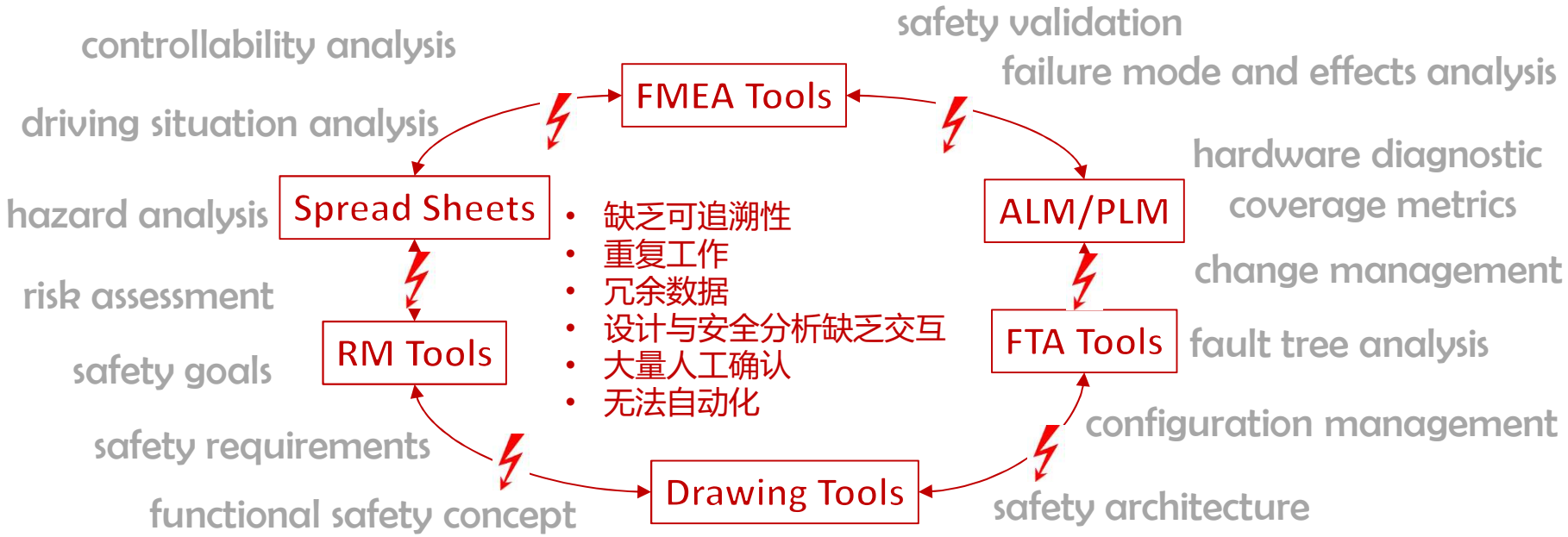
- 提出量化的安全等级
- 提出规范化的开发流程
- 提供标准化的技术方法

… 实际工程中是怎样做的?


スイッチング素子の
ON故障
スイッチング素子の
ショート故障
スイッチング素子への
過電圧印加
スイッチング素子への
過電流印加
スイッチング素子の
開発故障
①
②

功能安全的传统方法：单个任务的点工具

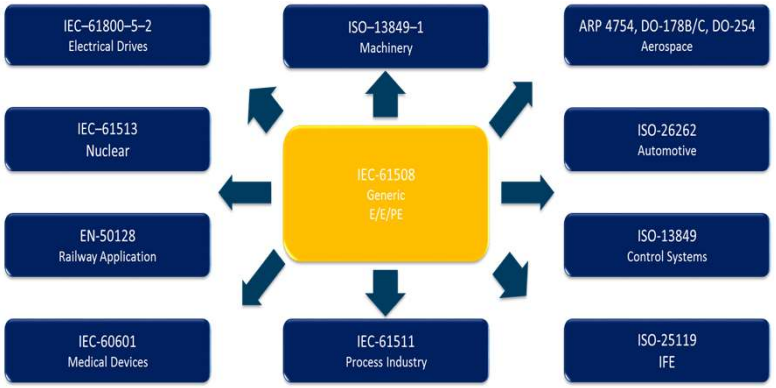
工程与安全分析之间没有整合

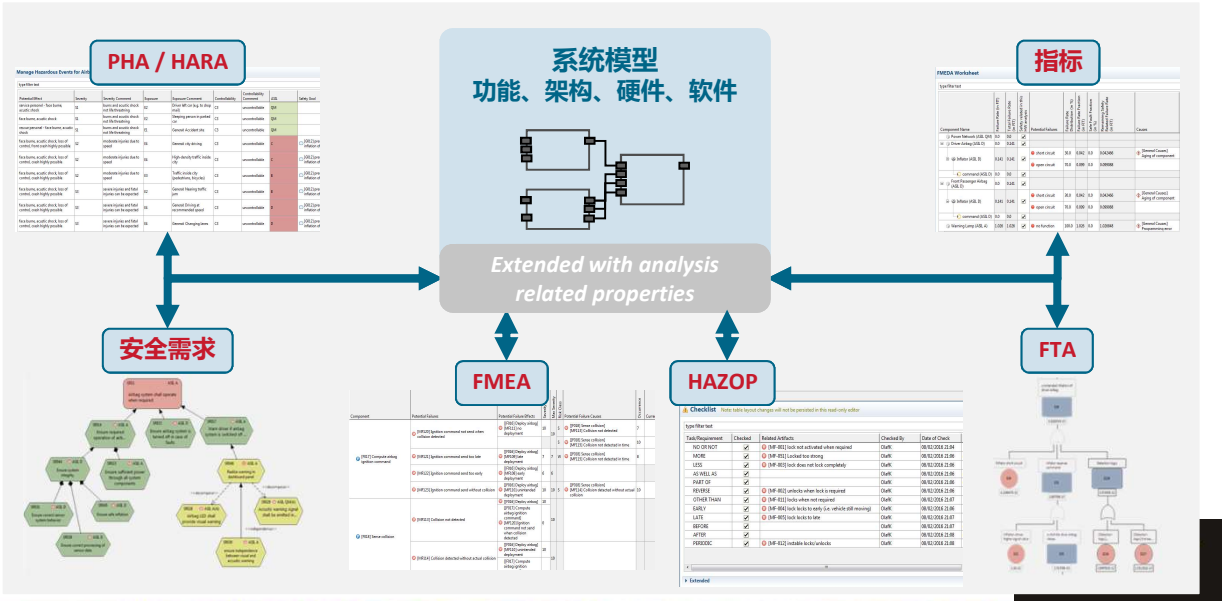


使用点工具的传统方法容易出错，耗时且浪费人力

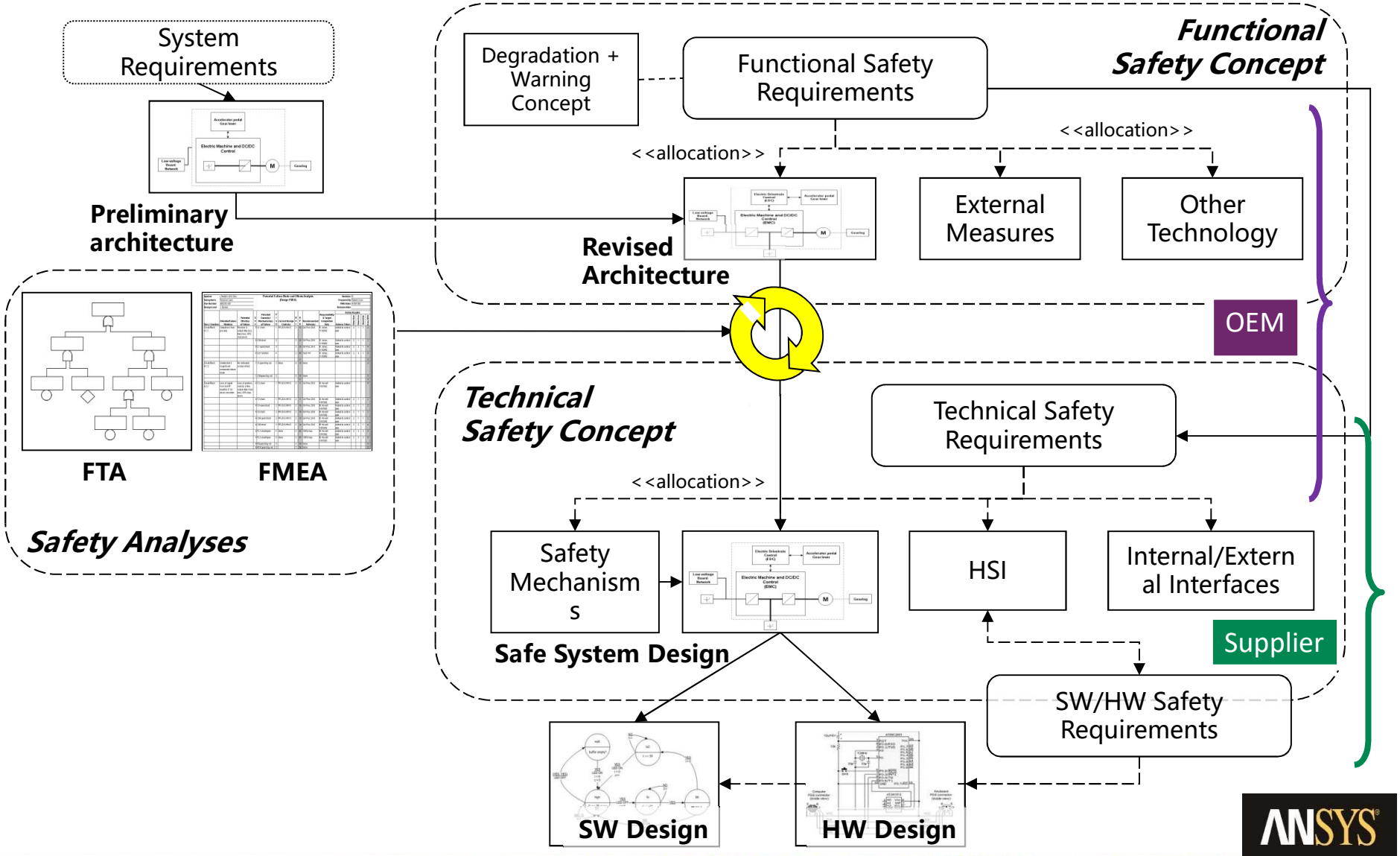
ANSYS medini for Model-based Safety Analysis

- **ANSYS medini 基于模型的功能安全分析工具**
  - 功能安全与可靠性工程的综合解决方案
  - 基于模型的方法，覆盖全生命周期，支持在概念、系统、软件、PCB、芯片级进行安全分析，确保追踪性和一致性
  - 符合功能安全标准的最佳实践： IEC 61508，EN50126/50128/50129 , ISO 26262，ARP4761/ARP4754A
  - 内置大量高效的工程模板、检查单和手册，SN 29500，IEC 62380，MIL HDBK 217F，FIDES Guide 2009，支持复用和自动化，大大减少成本和上市时间





Safety Activities in Whole Development Process



ISO 26262 FuSa In Conception Stage

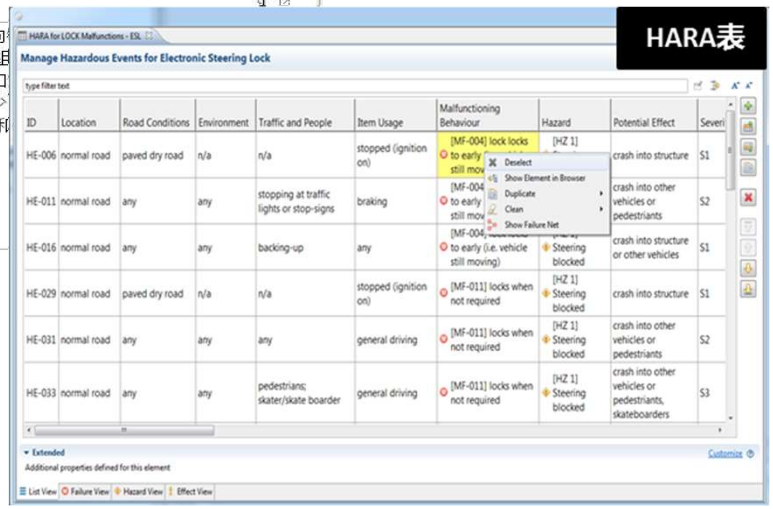
■ 概念阶段（Concept Phase）

- © 项目定义
- © 危害分析
- © 确定安全目标

❖ ANSYS 提供的功能方法：

- ✓ 驾驶状况和危险事件管理
- ✓ 危害和可操作性分析（HAZOP）
- ✓ 危害分析和风险评估（HARA）
- ✓ ASIL测定与风险图，自动计算ASIL等级

项目定义



HARA表

操作场景　失效模式　危害及影响　危害的严重度、暴露概率和可控性　ASIL分级

Location	Road Conditions	Environment	Traffic and People	Item Usage	Malfunctioning Behaviour	Hazard	Potential Effect	Severity	Severity Comment	Exposure	Exposure Comment	Controllability	Controllability Comment	ASIL
normal road	any	any	pedestrians; skater/skate boarder	general driving	[MF-011] locks when not required	[HZ 1] Steering blocked	crash into other vehicles or pedestriants, skateboarders	S3	medium speed	E3	Traffic inside city (pedestrians, bicycles)	C2	distance to other vehicles or pedestriants	B
normal road	any	any	stopping at traffic lights or stop-signs	braking	[MF-011] locks when not required	[HZ 1] Steering blocked	crash into other vehicles or pedestriants	S2	medium speed	E4	Stopping at traffic lights etc in cities	C2	distance to other vehicles or pedestriants	B
normal road	any	any	turning right	general driving	[MF-011] locks when not required	[HZ 1] Steering blocked	crash into other vehicles or pedestriants	S2	medium speed	E3	Turning into side-road; crossing bike-road	C2	distance to other vehicles or pedestriants	A
normal road	any	any	changing lanes	general driving	[MF-011] locks when not required	[HZ 1] Steering blocked	crash into other vehicles	S2						

支持符合ISO 26262标准的ASIL测定

ISO 26262 FuSa In System Development

■ 功能安全概念阶段（FSC）

© 从功能安全目标推导功能安全需求

© 把功能安全需求分配给子系统

❖ **ANSYS 提供的功能方法：**

✓ 失效模式影响分析（FMEA）

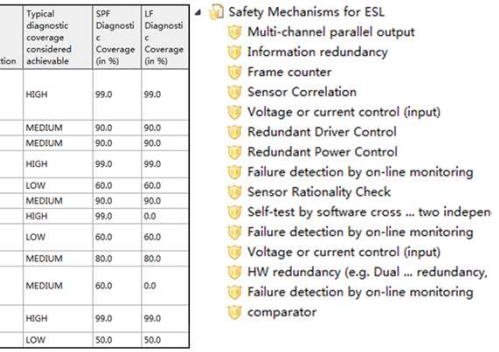
✓ 安全需求定义和分配

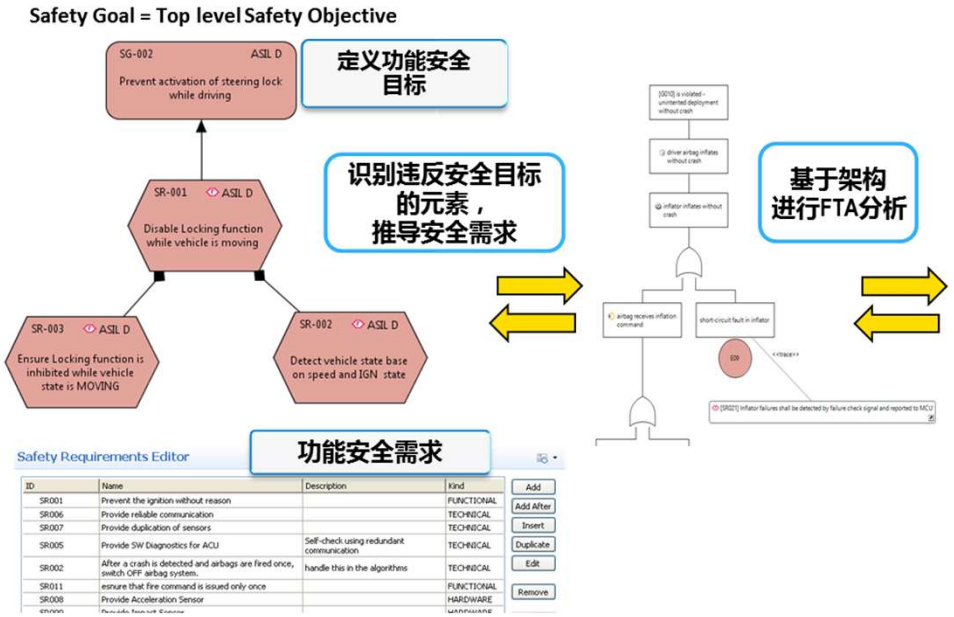
✓ 定义架构组件的安全等级

✓ 诊断覆盖度（DC）自动分析

✓ 支持创建安全措施库

Safety mechanism/measure	ISO section key or identifier	Description	Detection	Typical diagnostic coverage considered achievable	SPF Diagnostic Coverage (in %)	LF Diagnostic Coverage (in %)
Multi-channel parallel output	D.2.6.3	Additional separate communication of ESL_enable	0	HIGH	99.0	99.0
Information redundancy	D.2.7.6	E2E Protection	0	MEDIUM	90.0	90.0
Frame counter	D.2.7.7	E2E Protection	0	MEDIUM	90.0	90.0
Sensor Correlation	D.2.10.2	Sensors for both positions	0	HIGH	99.0	99.0
Voltage or current control (input)	D.2.8.1		0	LOW	60.0	60.0
Redundant Driver Control			0	MEDIUM	90.0	90.0
Redundant Power Control			0	HIGH	99.0	0.0
Failure detection by on-line monitoring	D.2.1.1		0	LOW	60.0	60.0
Sensor Rationality Check	D.2.10.3		0	MEDIUM	80.0	80.0
Self-test by software cross exchange between two independent units	D.2.3.3		0	MEDIUM	60.0	0.0
Failure detection by on-line monitoring	D.2.1.1		0	HIGH	99.0	99.0
Voltage or current control (input)	D.2.8.1		0	LOW	50.0	50.0





ISO 26262 FuSa In Detail Design Stage

■ 技术安全概念阶段（TSC）

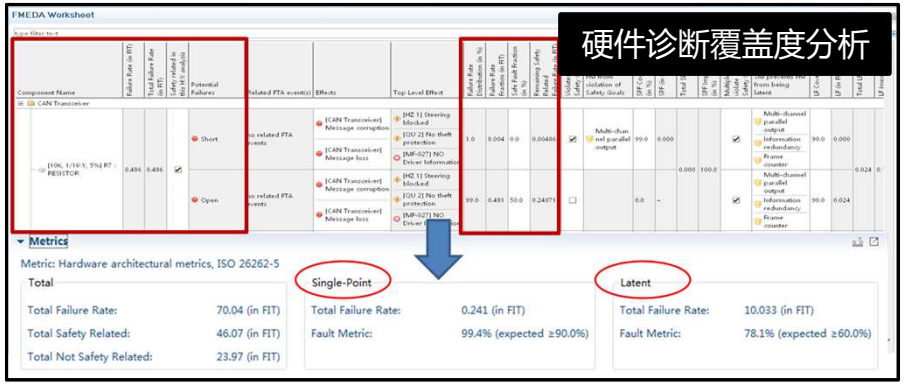
© 按照功能安全概念规定技术安全需求

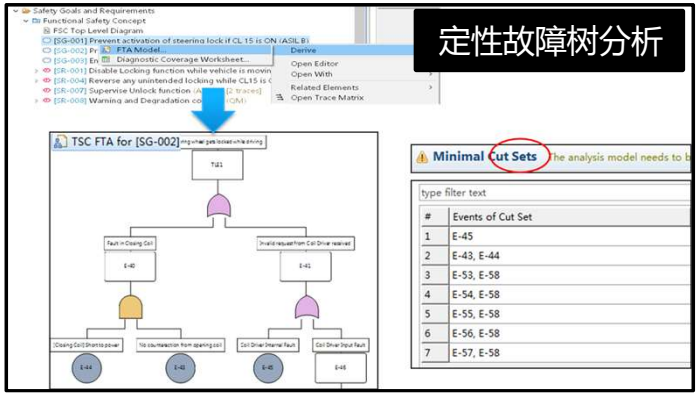
© 对技术安全需求规定必要的安全机制

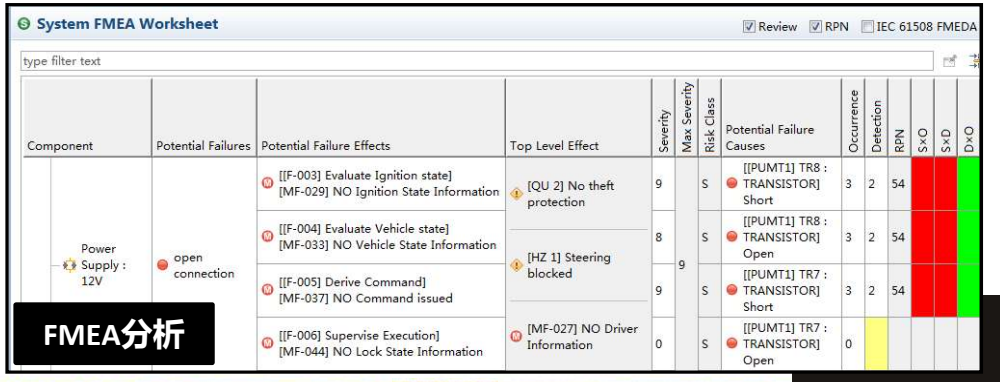
© 技术安全需求应分配给系统设计元素（软硬件）

❖ ANSYS 提供的功能方法：

- ✓ 故障模式和影响分析（FMEA）
- ✓ 定性/定量故障树分析（FTA）
- ✓ 故障模式，影响和诊断分析（FMEDA）
- ✓ 硬件安全性分析的诊断覆盖度指标
- ✓ 支持单点故障分析
- ✓ 支持潜在故障硬件度量
- ✓ 支持依据故障率手册的可靠性预测
- ✓ 内置SN29500, IEC62380等失效率手册
- ✓ 支持导入硬件BOM和生成硬件 LIB
- ✓ 自动计算硬件指标



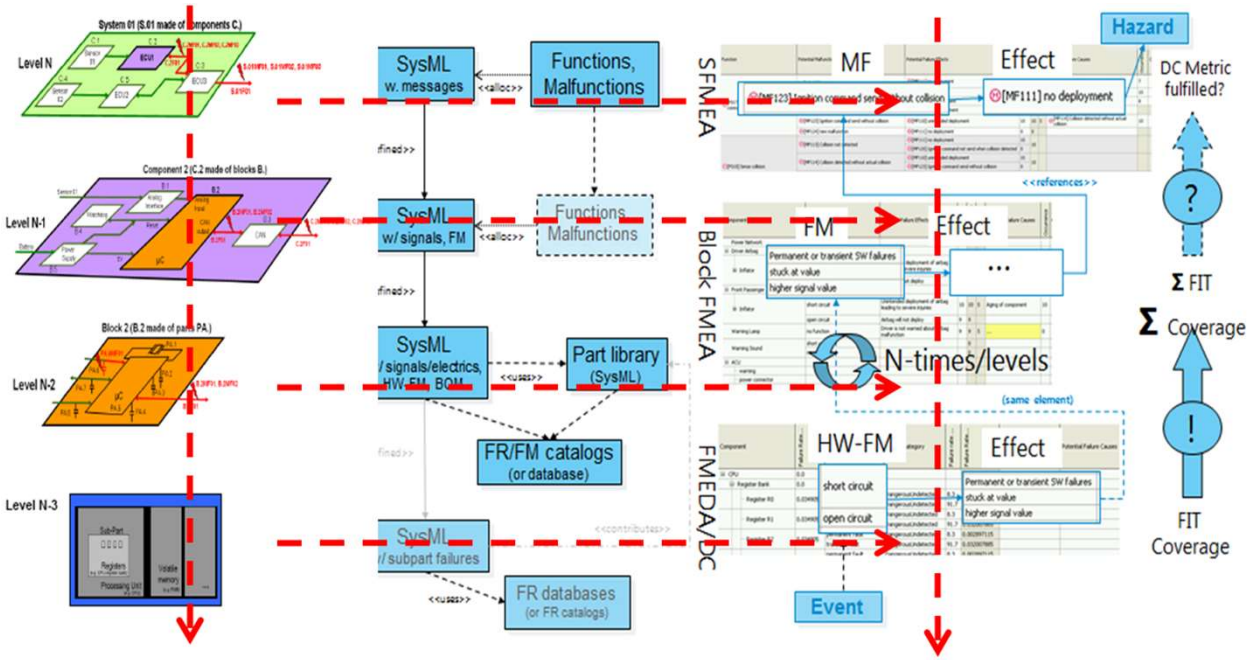




项目管理

■ 功能安全数据与研发数据的追踪管理

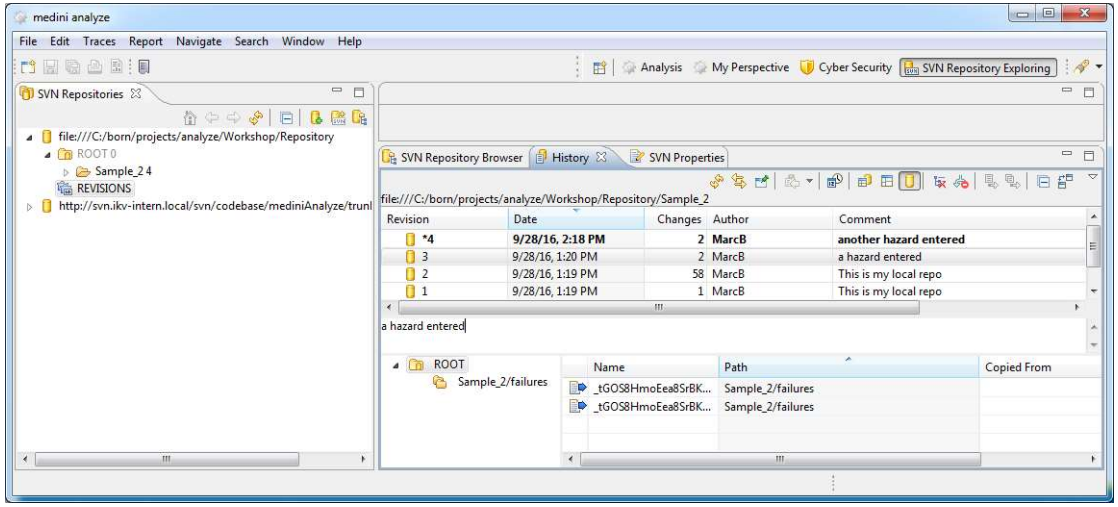
- ✓ 需求和架构之间的跟踪（Allocation of SRs）
- ✓ 失效模式、效应之间的跟踪（Failure Net）
- ✓ 架构（系统/硬件/软件）之间的跟踪、变更管理（结构化、分层设计）
- ✓ 不同层级之间证据链的跟踪（一致性）

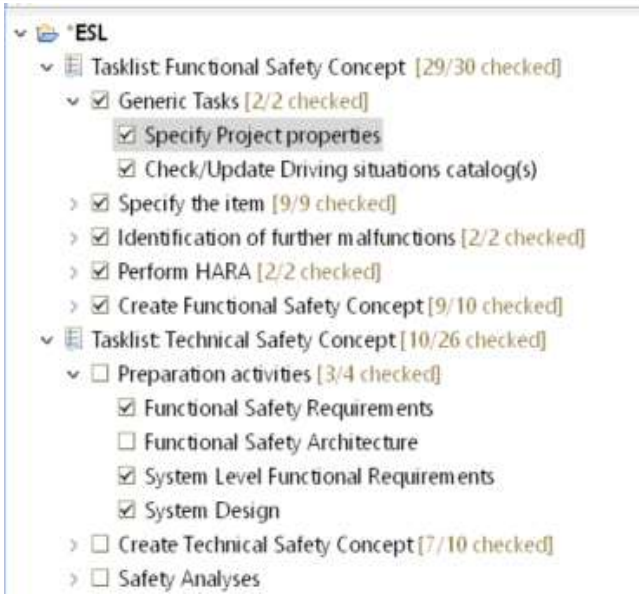


项目管理

- **任务检查表（Checklist）**：可用于验证和确认活动、安全计划和进度跟踪

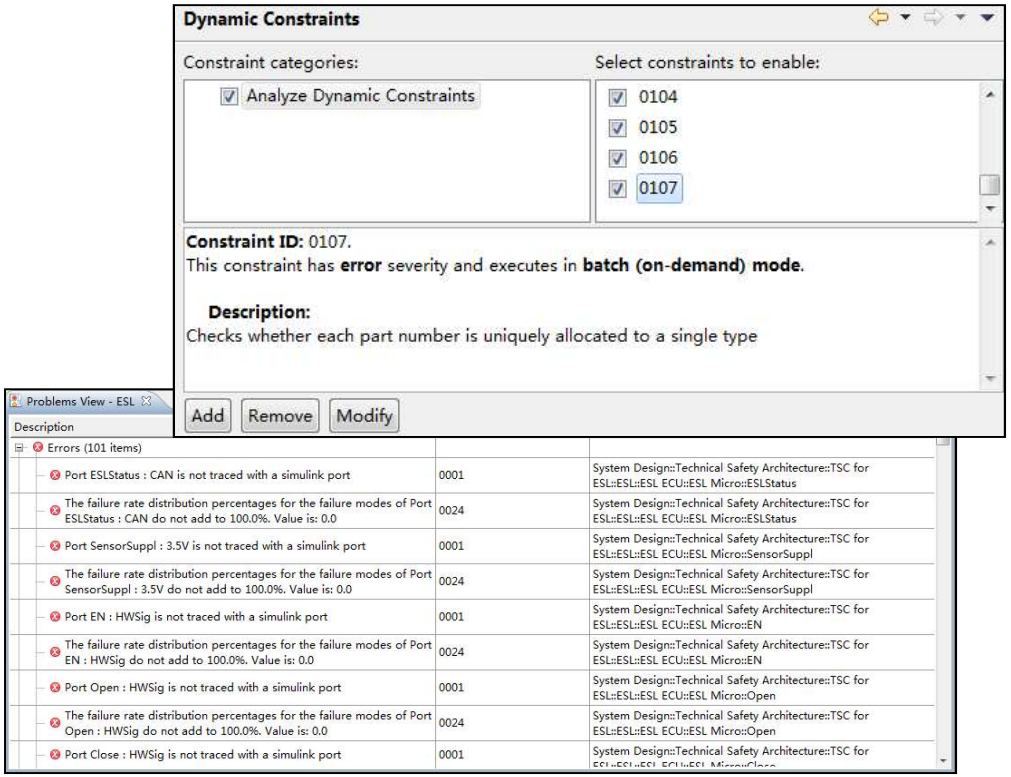
- **支持团队协作：**项目级别比较和合并功能
- **与配置管理系统集成（SVN, ClearCase, PTC Integrity等）**

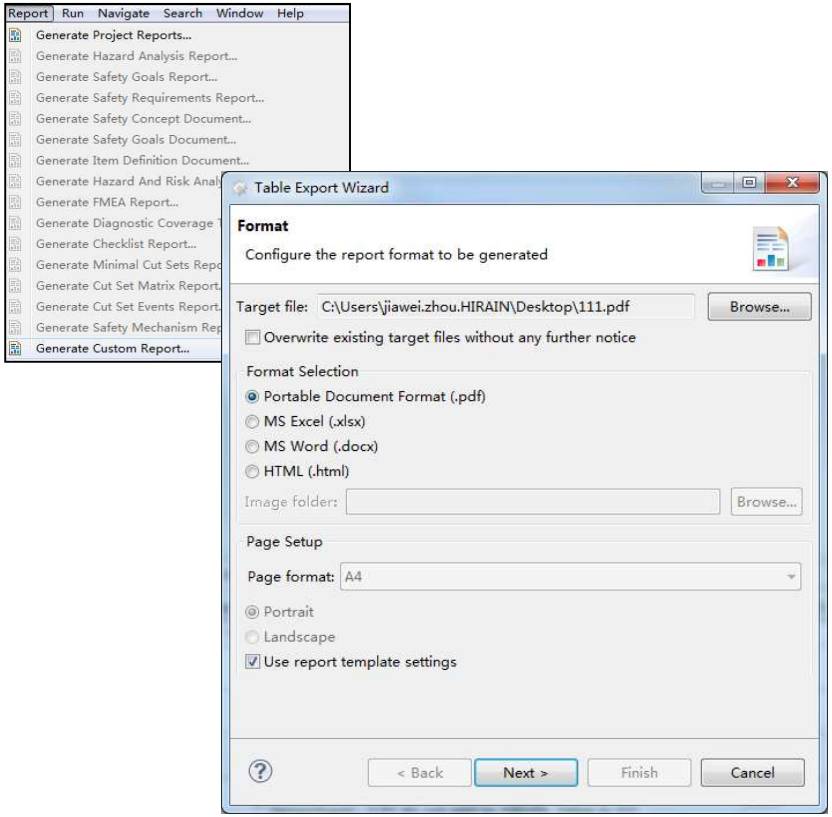




其它辅助功能

- **安全验证（Validation）**
  - ✓ 包含107条常见错误，且支持扩展
  - ✓ 精确定位错误位置

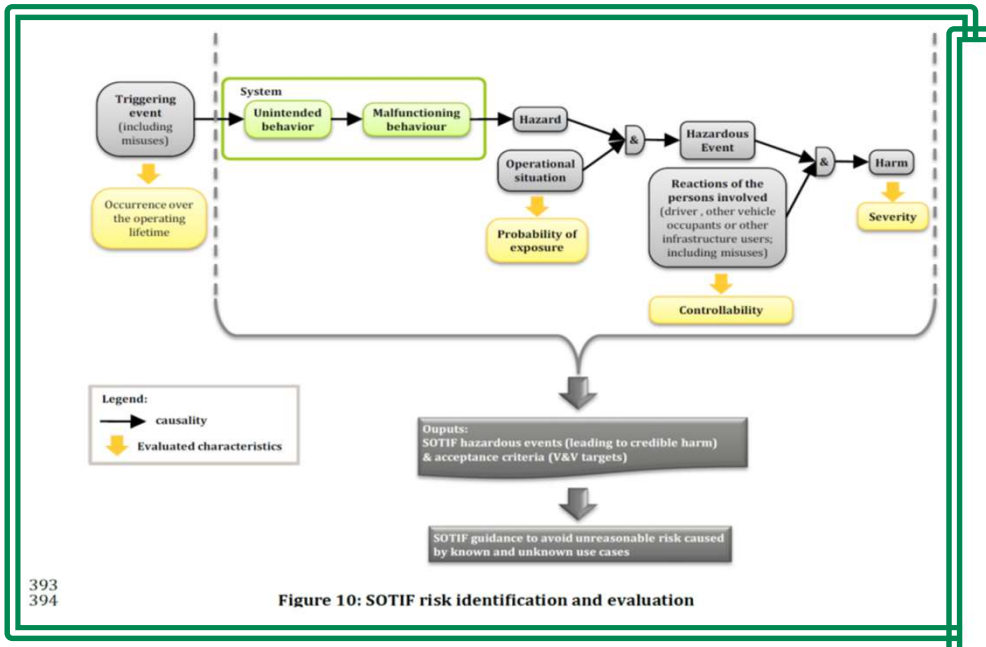




- **生成报告（Generate Report）**
  - ✓ 支持PDF/Word/Excel/HTML格式文档输出
  - ✓ 支持自定义开发报告模板（二次开发）
  - ✓ 支持各阶段产物输出

PAS 21448（SOTIF）

- **ISO 26262着眼于车辆电子电器系统失效引起的安全风险**
- **PAS 21448（SOTIF）着眼于由外部非预期因素引起的安全风险，包括：**
  - 功能具备正确理解环境和行为安全的能力；
  - 功能具备一定的健壮性容忍外部环境的干扰。



393
394 **Figure 10: SOTIF risk identification and evaluation**

455 **Triggering events related to algorithm:**

456 An analysis of triggering events related to algorithms is used to determine:

457 – SOTIF risk improvement methods and measures according to Clause 8.3;

458 – decision algorithm verification according to Clause 10.3; and

459 – validation of functionality according to Clause 11.

460 The analysis considers categories such as:

461 – environment and location;

462 – driving Situation;

463 – driver behaviour (including driver misuse);

464 – (Expected) behaviour of other drivers/road users;

465 – unusual traffic situations; and

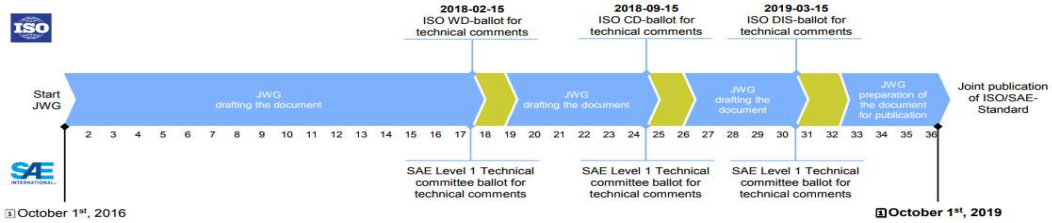
466 – algorithm limitation.

467
468 NOTE: The identified limitations are included in the list mentioned in Clause 5.

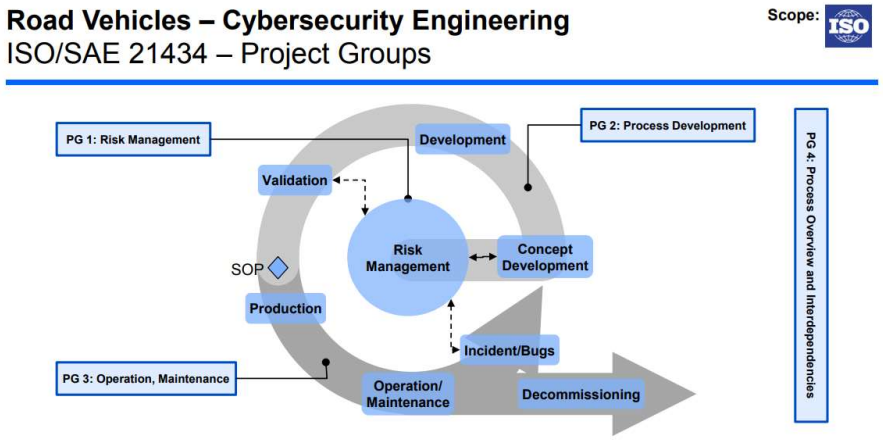
469 EXAMPLE: a construction site, an accident, a traffic jam with emergency corridor, driving th
470 can all be considered as unusual traffic situations.

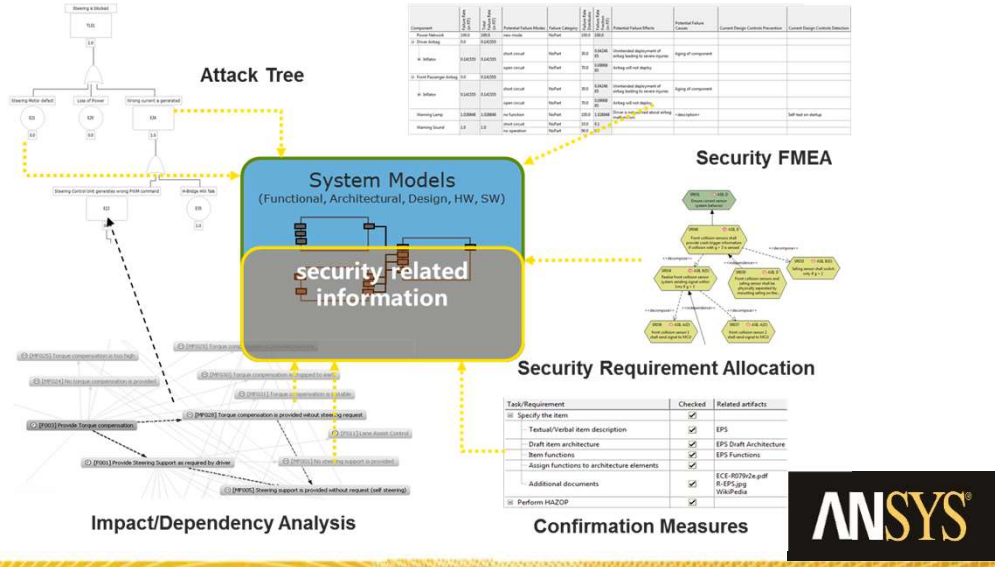
信息安全标准

- **汽车行业相关标准正在开发中**
  - ISO21434
  - SAE
- **兼容原有V流程和安全过程**
- **复用原有功能安全方法和工具**
  - HARA – TARA
  - Fault Trees – Attack Trees
  - FMEA – Security FMEA/FMVEA
- **支持功能安全和信息安全的接口分析**
  - Redundancy (safety mechanism) increases attack surface
  - Encryption (security mechanism) increases WCET
  - Security updates conflict with stability of safe systems



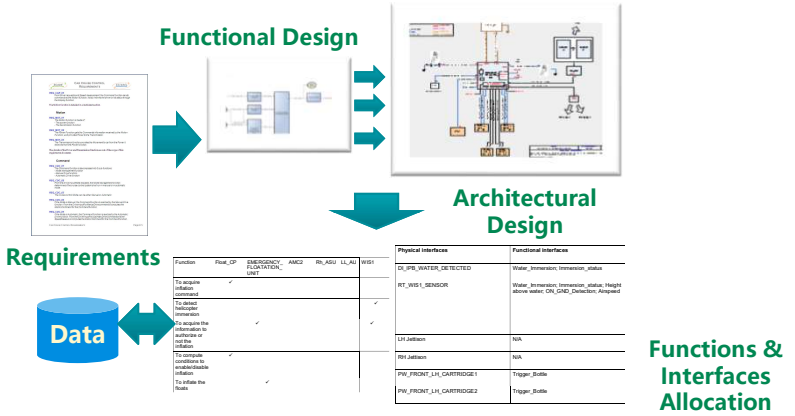
CD: Committee Draft; DIS: Draft International Standard; FDIS: Final Draft International Standard

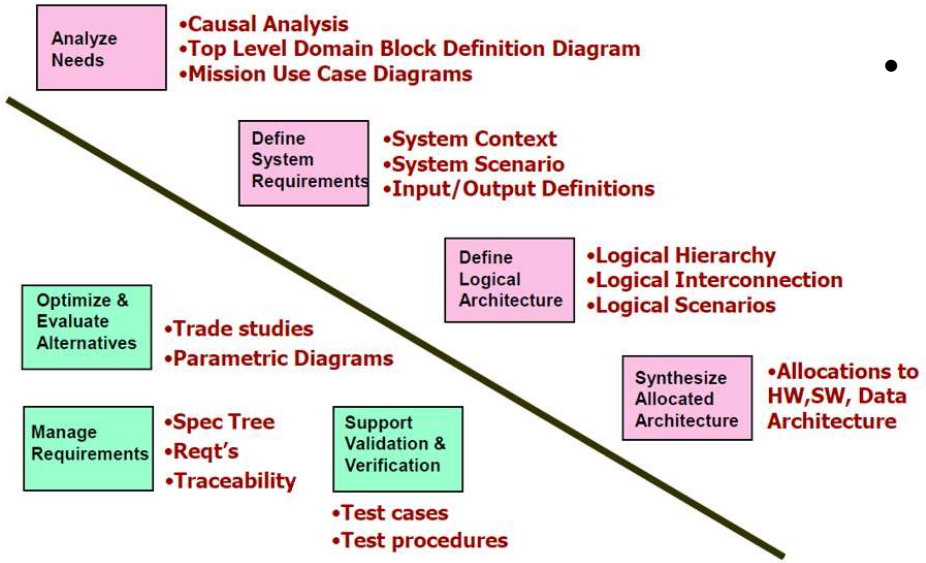




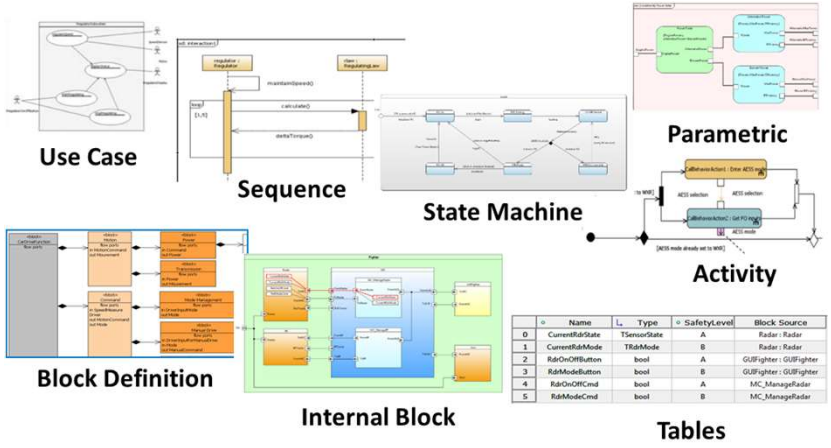
System Design (1/2)

■ 系统工程协会定义的主要系统设计活动

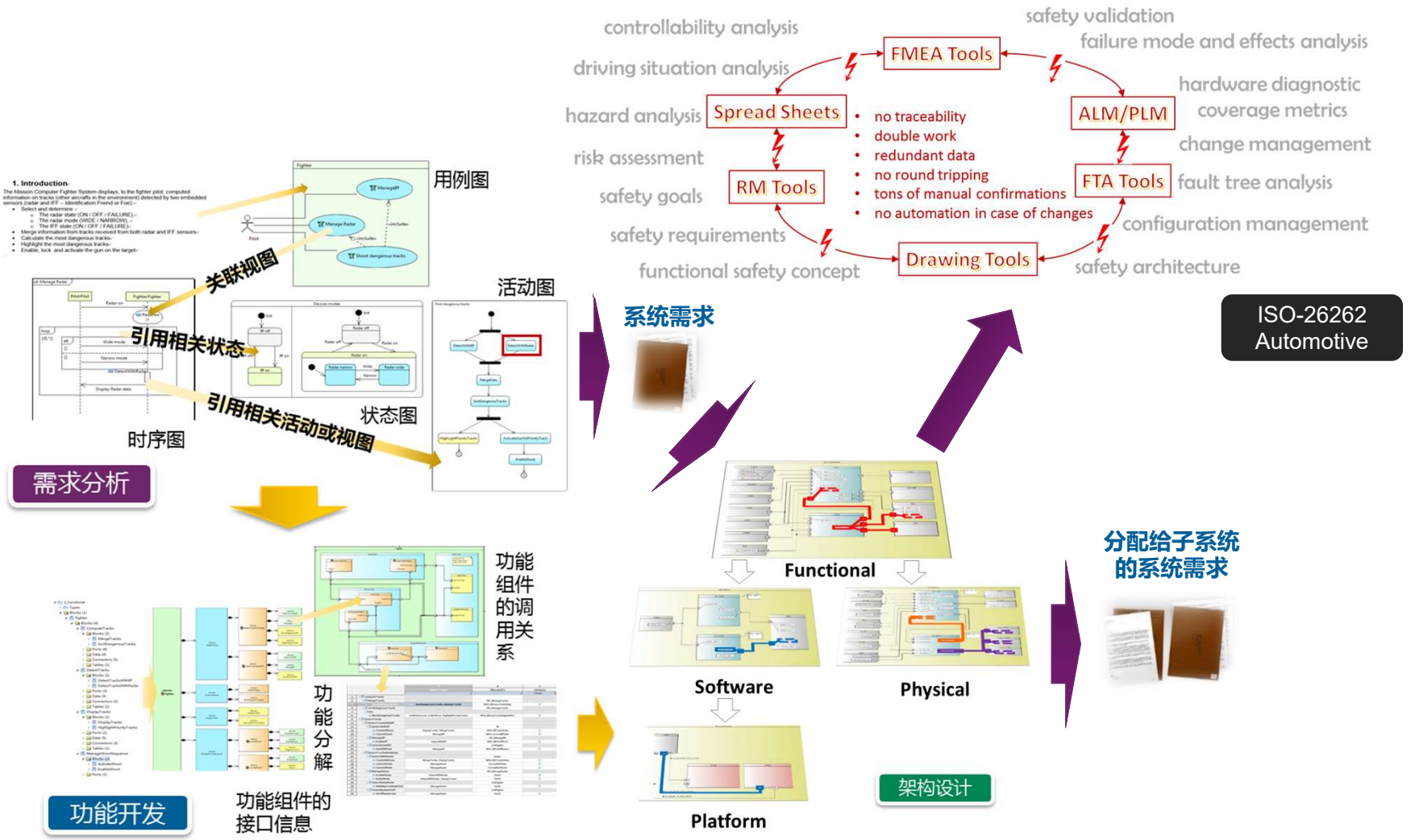




- 基于SysML模型的图形化设计



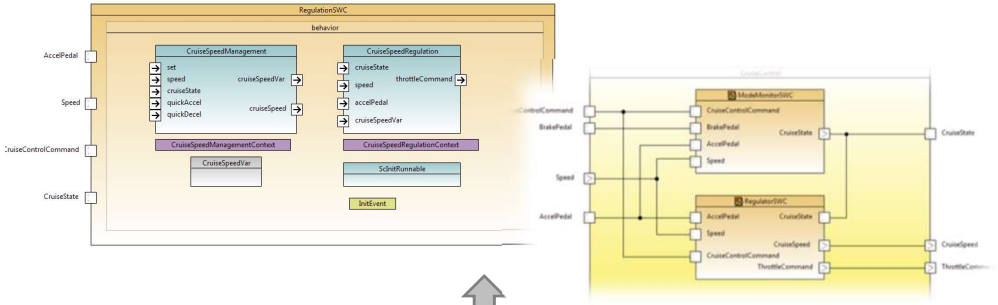
System Design (2/2)



Well integrated with ANSYS FuSa Products

SCADE ARCHITECT

System Architecture



Iterative approach

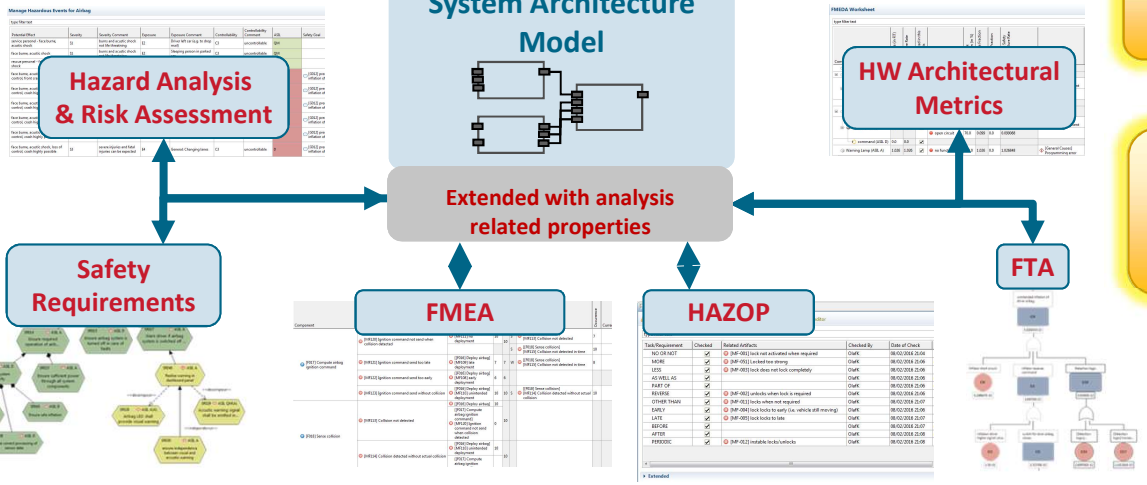
Safety process seamlessly integrated with system development

Safety analysis results always consistent

Safety requirements discovered and considered early in the design process

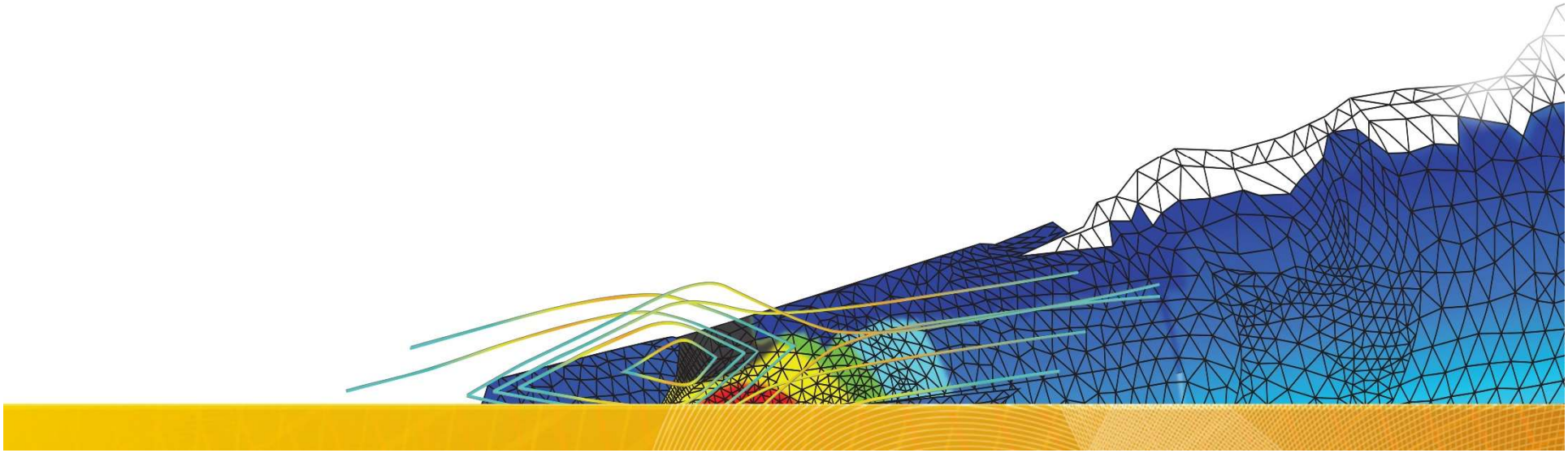
medini™ analyze

Functional Safety Analysis and Design



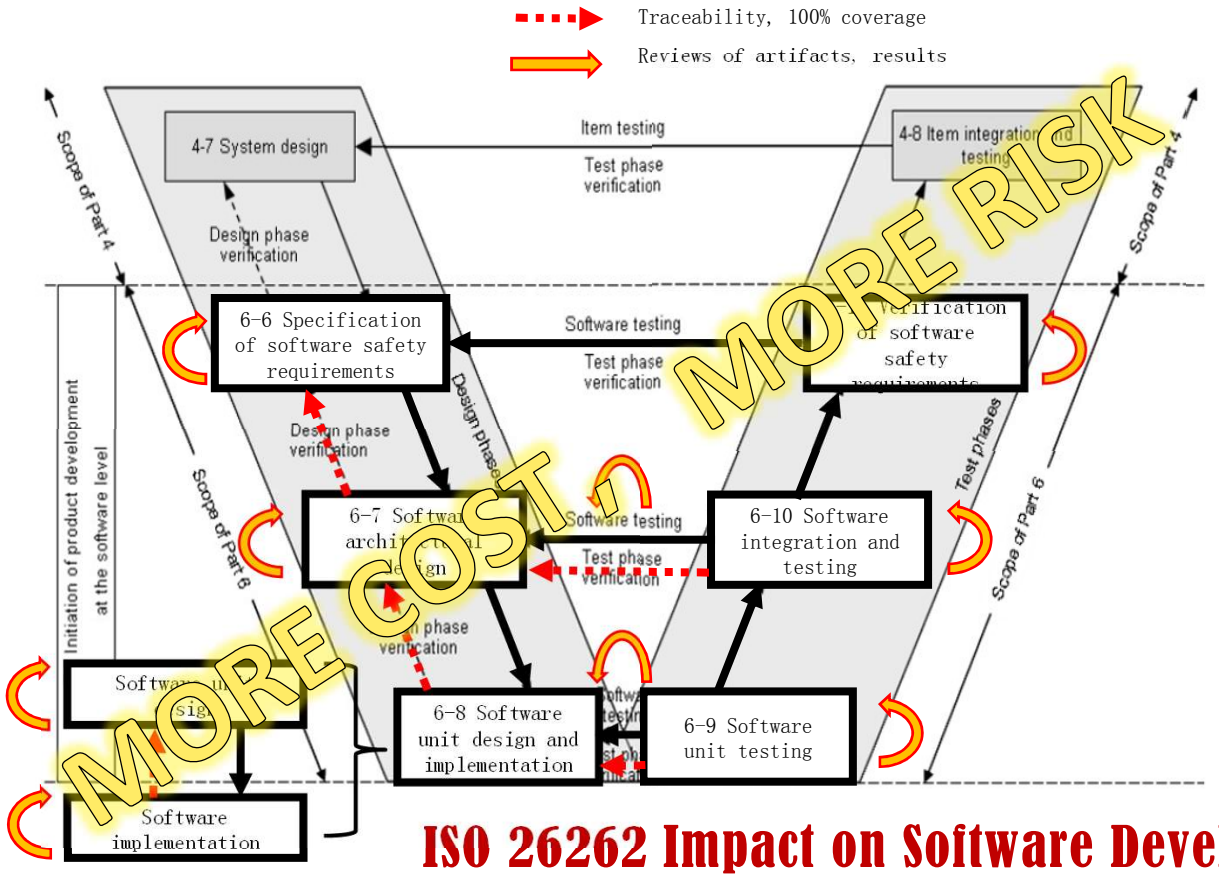
Realize Your Product Promise®

ANSYS MBD for Safety & Critical Embeded Software



ANSYS MBD Process for Safety & Critical Software (1/2)

■ **Strong focus on software engineering compliance with industry SW safety standard**



■ *满足安全需求的实现和验证*

■ *鲁棒性的设计和验证*

■ *过程和数据的可追溯*

■ *符合性和一致性验证*

■ *正确性和完备性的验证*

■ *过程和数据的可信和完备*

■ *漫长的取证过程*

■ ……

ISO 26262 Impact on Software Development Activities

ISO 26262: Software Level of Product Development

■ ISO26262中的软件开发要求

Table 3 — Informal verification of software safety requirements

Methods and Measures		According to req.	ASIL A	B	C	D
1a	Inspection of software safety requirements[a]	5.4.11	++	++	++	++
1b	Walkthrough of software safety requirements	5.4.11	++	+	O	o
1c	Model inspection	5.4.11	++	++	++	++
1d	Model Walkthrough	5.4.11	++	+	O	o

[a] The software safety requirements need to be informally verified by an appropriate combination of methods 1x). This combination typically depends on the applied methods and measures for specifying software safety requirements (see ISO 26262-8, Table°1).

Table 12 — Design principles for software unit design and implementation

Methods and Measures		According to req.	ASIL A	B	C	D
1	One entry and one exit point in subprograms and functions [a]	7.4.3	+	+	++	++
2a	No dynamic objects or variables[a, b]	7.4.3	+	++	++	++
2b	Online test during creation of dynamic variables[, b]	7.4.3	+	++	++	++
3	Initialisation of variables[a]	7.4.3	++	++	++	++
4	No multiple use of variables[a]	7.4.3	+	++	++	++
5	Avoid global variables or justify their usage[a]	7.4.3	+	+	++	++
6	Limited use of pointers[a]	7.4.3	O	+	+	++
7	No implicit type conversions[a, c]	7.4.3	+	++	++	++
8	No hidden data flow or control flow	7.4.3	+	++	++	++
9	No unconditional jumps[a, c, d]	7.4.3	++	++	++	++
10	No recursions[d]	7.4.3	+	++	++	++

Table 17 — Structural coverage

Methods and Measures		According to req.	ASIL A	B	C	D
1	Statement coverage[a, b, c, e]	8.4.3	++	++	+	+
2	Decision coverage [b, c]	8.4.3	+	+	++	++
3	MC/DC (Modified Condition Decision Coverage), conditions affecting the decision [b, c, d]	8.4.3	+	+	+	++
4	Model coverage [c]	8.4.3	++	++	++	++

Table 6 — Error detection at software architecture level

Methods and Measures		According to req.	ASIL A	B	C	D
1	Plausibility check[a]	6.4.17	++	++	++	++
2	Detection of data errors[b]	6.4.17	+	+	++	++
3a	External monitoring facility	6.4.17	o	+	+	++
3b	Control flow monitoring	6.4.17	o	+	++	++
3c	Diverse software design[c]	6.4.17	o	o	+	++
3d	Series inhibits[d]	6.4.17	o	o	+	++

Table 16 — Functional software unit testing

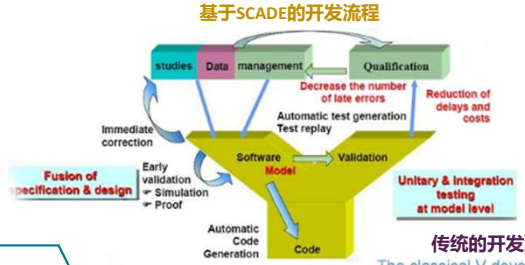
Methods and Measures		According to req.	ASIL A	B	C	D
1	Requirement-based test[a]	8.4.3	++	++	++	++
2a	Software unit interface test[b]	8.4.3	++	++	++	++
2b	Model-based testing[b]	8.4.3	++	++	++	++
3	Fault injection test	8.4.3	+	+	+	+
4	Error guessing test[c]	8.4.3	+	+	+	+
5	Equivalence class test based on input domain partitions	8.4.3	+	+	++	++

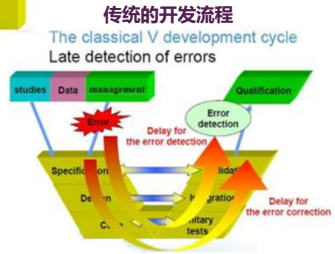
Table 21 — Software safety acceptance testing

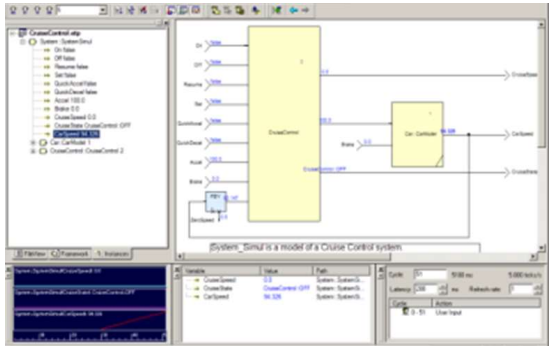
Methods and Measures		According to req.	ASIL A	B	C	D
1	Tests of software safety requirements[a]	10.4.4	++	++	++	++
2	Test interface[b]	10.4.4	+	+	+	+
3a	Hardware-in-the-loop tests[c]	10.4.4	+	+	++	++
3b	Tests within the electronic control unit network[c, d]	10.4.4	++	++	++	++
3c	Tests in the test vehicle[c]	10.4.4	++	++	++	++

ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

■ ANSYS MBD







The specification of a target in a state machine is valid if the identifier refers to the name of a state accessible from the current state machine.

$$\text{(Valid Fork Decision)}\ \frac{\begin{array}{l}\forall i \in [1,n],\ prog;\ gscope;\ lscope \overset{vpi}{\vdash} e_i \ \wedge\ prog;\ gscope;\ lscope \overset{vpi}{\vdash} scope_i \\ \forall i \in [1,n],\ prog;\ gscope;\ lscope;\ state\_names \overset{vpi}{\vdash} fork_i\end{array}}{prog;\ gscope;\ lscope;\ state\_names \overset{vpi}{\vdash} \begin{array}{l}\textbf{if } e_1 \textbf{ do } scope_1\ fork_1 \\ \textbf{elsif } e_2 \textbf{ do } scope_2\ fork_2\ \ldots \\ \textbf{else do } scope_n\ fork_n\end{array}}$$

A fork is valid if all the guards, scopes and subforks are valid.

$$\text{(Valid Transition)}\ \frac{prog;\ gscope;\ lscope \overset{vpi}{\vdash} e \qquad prog;\ gscope;\ lscope \overset{vpi}{\vdash} scope}{prog;\ gscope;\ lscope;\ state\_names \overset{vpi}{\vdash} fork}$$
$$prog;\ gscope;\ lscope;\ state\_names \overset{vpi}{\vdash} \textbf{if } e \textbf{ do } scope\ fork$$

A transition is valid if the guard, the scope and the fork part are valid.

$$\text{(Valid State)}\ \frac{\begin{array}{l}prog;\ gscope;\ lscope;\ state\_names \overset{vpi}{\vdash} tr\_strong \\ prog;\ gscope;\ lscope \overset{vpi}{\vdash} scope \\ prog;\ gscope;\ lscope \cup \overleftarrow{sig\_vars(scope)};\ state\_names \overset{vpi}{\vdash} tr\_weak \\ prog;\ gscope;\ lscope \cup \overleftarrow{sig\_vars(scope)};\ state\_names \overset{vpi}{\vdash} tr\_synchro\end{array}}{prog;\ gscope;\ lscope;\ state\_names \overset{vpi}{\vdash} \begin{array}{l}\textbf{state } id \textbf{ unless } tr\_strong \\ scope \textbf{ until } tr\_weak \textbf{ synchro } tr\_synchro\end{array}}$$

8.3 State machines

State machines offer the most elaborate com
state machines extends the conditional bloc
orizing extra expressions. In this case, the
whose states contain the needed information
between states, weak or strong, that influen

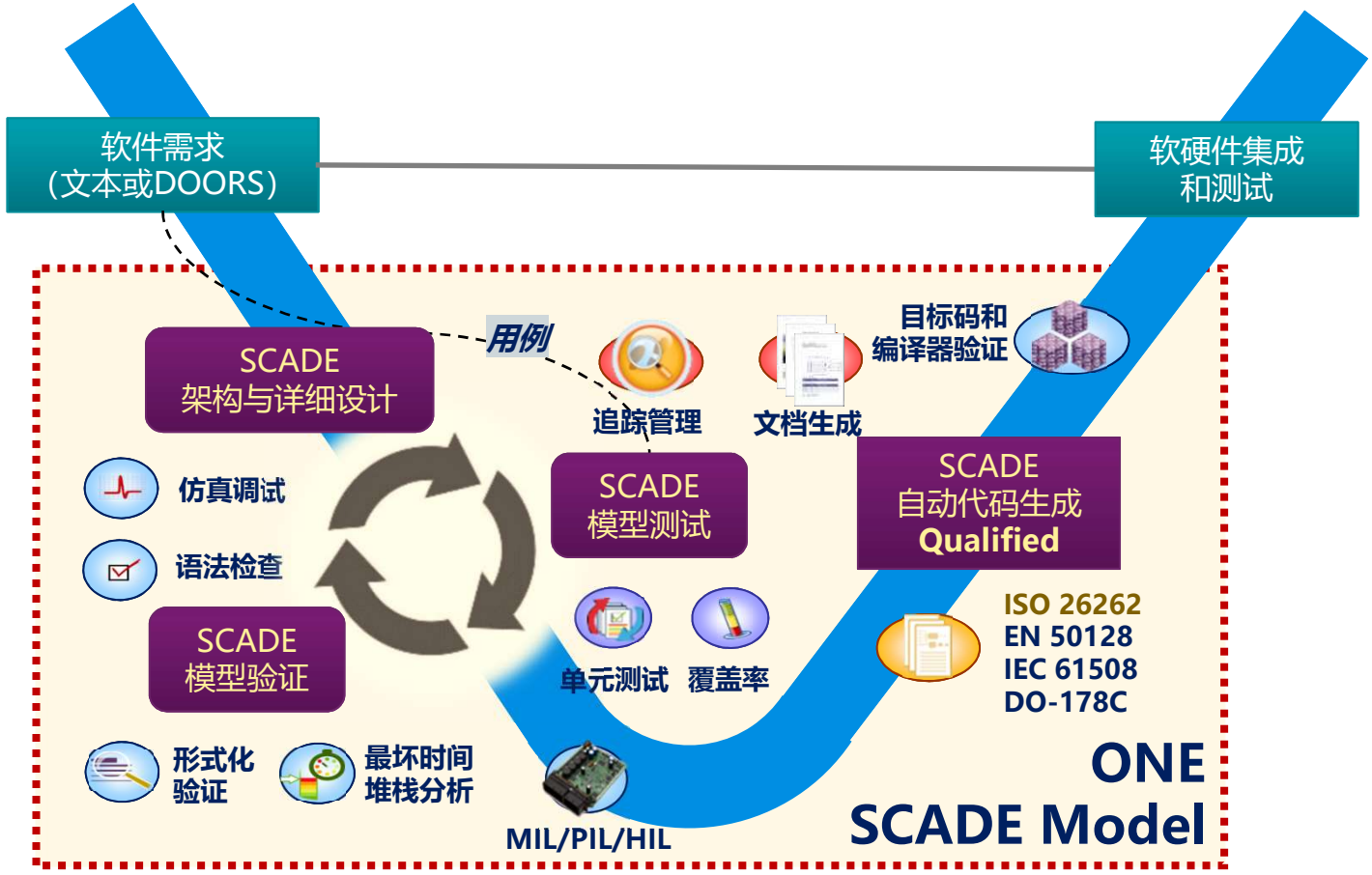
8.3.1 State machines

$$\begin{array}{lcl} state\_machine & ::= & \textbf{automaton } [\ \text{ID}\ ]\ \{\!\{\ state\_decl\ \}\!\} \\ state\_decl & ::= & [\![\ \textbf{initial}\ ]\!]\ [\ \textbf{final}\ ]\ \textbf{state } \text{ID} \\ & & [\ \textbf{unless } \{\!\{\ transition\ ;\ \}\!\}^+\ ] \\ & & data\_def \\ & & [\ \textbf{until } \{\!\{\ transition\ ;\ \}\!\} \\ & & [\ \textbf{synchro } [\ actions\ ]\ fork\ ;\ ]\ ] \end{array}$$

Static Semantics

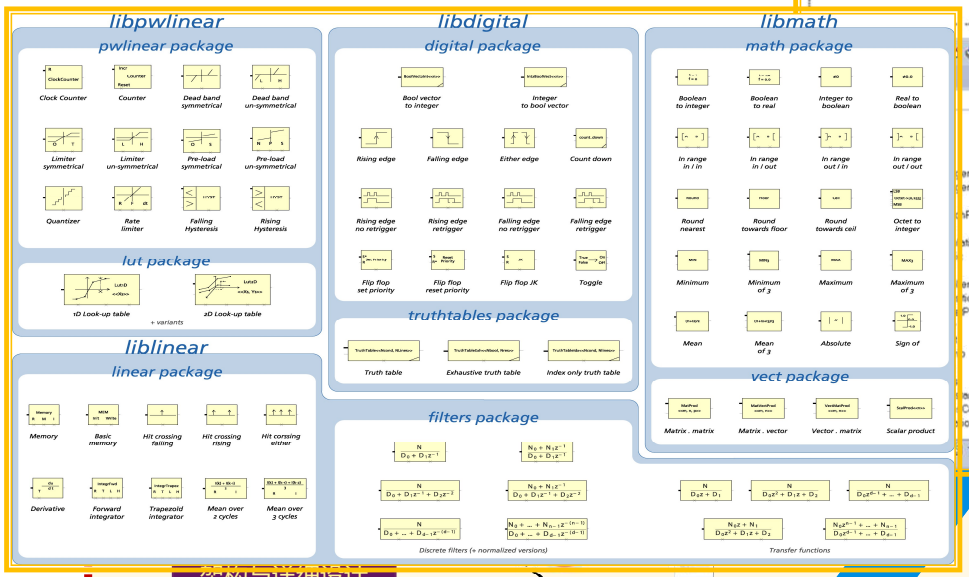
ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

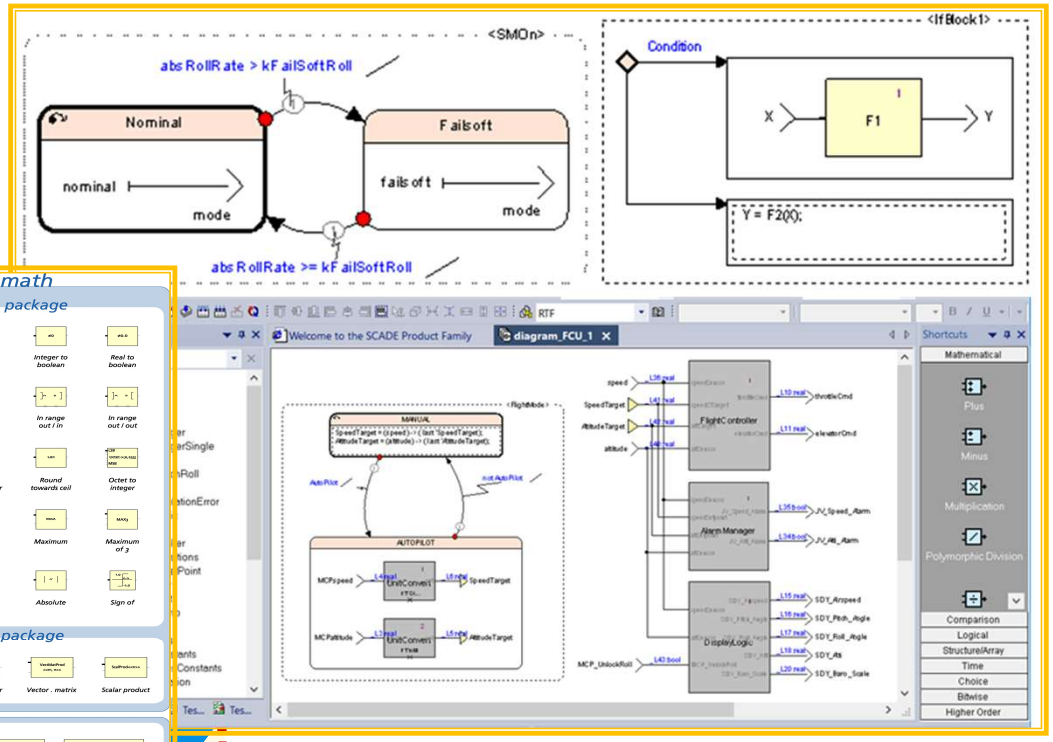
■ ANSYS MBD

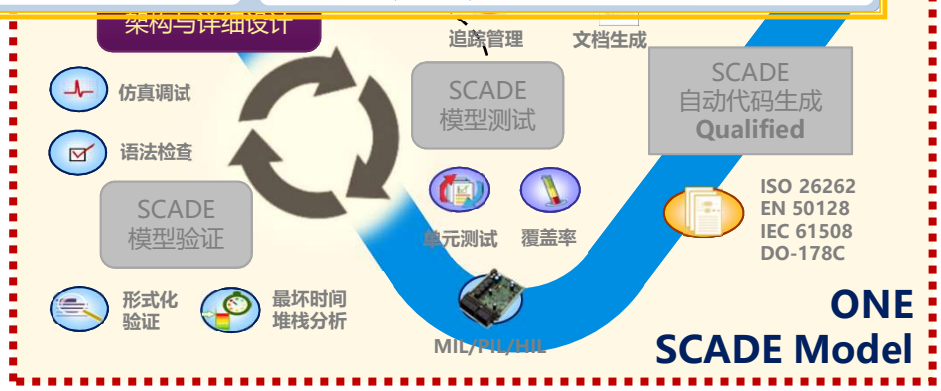


ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

■ ANSYS MBD process







■ 图形化的软件架构和详细设计

- ✓ **数据流对象（内嵌的语义）**：数学运算、逻辑、比较、条件判断、位操作、时序操作、复合类型操作、循环体等；
- ✓ **控制流对象（内嵌的语义）**：状态机、可激活状态操作符；
- ✓ **基础库（自定义模型）**：高阶数学运算库、数字信号操作库、滤波与传递函数库、插值与LUT操作库等。

ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

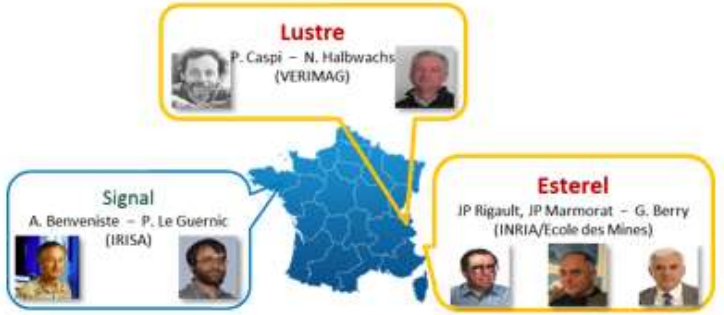
■ ANSYS MBD process

❖ **SCADE语言是同步且形式化的语言（起源于同步语言Lustre）**

- ❑ SCADE语言简单稳定，确定性、强类型、明确的语义
- ❑ SCADE模型的解释不依赖于读者和运行环境，仅由数学逻辑唯一确定

❖ **SCADE语言是包含安全结构和表述的语言子集（Correct/Safety by construction）**

- ❑ 完全符合安全关键软件的设计/编码要求——内嵌建模约束
- ❑ 内嵌的防御性建模机制——减少人因错误
- ❑ 模块化、低耦合的建模机制——便于复用和验证



- ✓ 1. Rule: 不使用递归和跳转
- ✓ 2. Rule: 所有循环必须有上下边界
- ✓ 3. Rule: 初始化后没有动态的内存分配
- ✓ 4. Rule: 函数长度要求.
- ✓ 5. Rule: 代码密度要求.
- ✓ 6. Rule: 数据的声明范围和使用范围保持一致
- ✓ 7. Rule: 函数返回值检验要求
- ✓ 8. Rule: 限制头文件和宏文件预编译
- ✓ 9. Rule: 指针需要被限制使用
- ✓ 10. Rule: 代码编译要求

The Power of Ten –
Rules for Developing Safety Critical Code[1]

Gerard J. Holzmann
NASA/JPL Laboratory for Reliable Software
Pasadena, CA 91109

Most serious software development projects use coding guidelines. These guidelines are meant to state what the ground rules are for the software to be written: how it should be structured and which language features should and should not be used. Curiously, there is little consensus on what a good coding standard is. Among the many that have been written there are remarkable few patterns to discern, except that each new document tends to be longer than the one before it. The result is that most existing guidelines contain well over a hundred rules, sometimes with questionable justification. Some rules, especially those that try to stipulate the use of white-space in programs, may have been introduced by personal preference; others are meant to prevent very specific and unlikely types of error from earlier coding efforts within the same organization. Not surprisingly, the existing coding guidelines tend to have little effect on what developers actually do when they write code. The most dooming aspect of many of the guidelines is that they rarely allow for comprehensive tool-based compliance checks. Tool-based checks are important, since it is often infeasible to manually review the hundreds of thousands of lines of code that are written for larger applications.

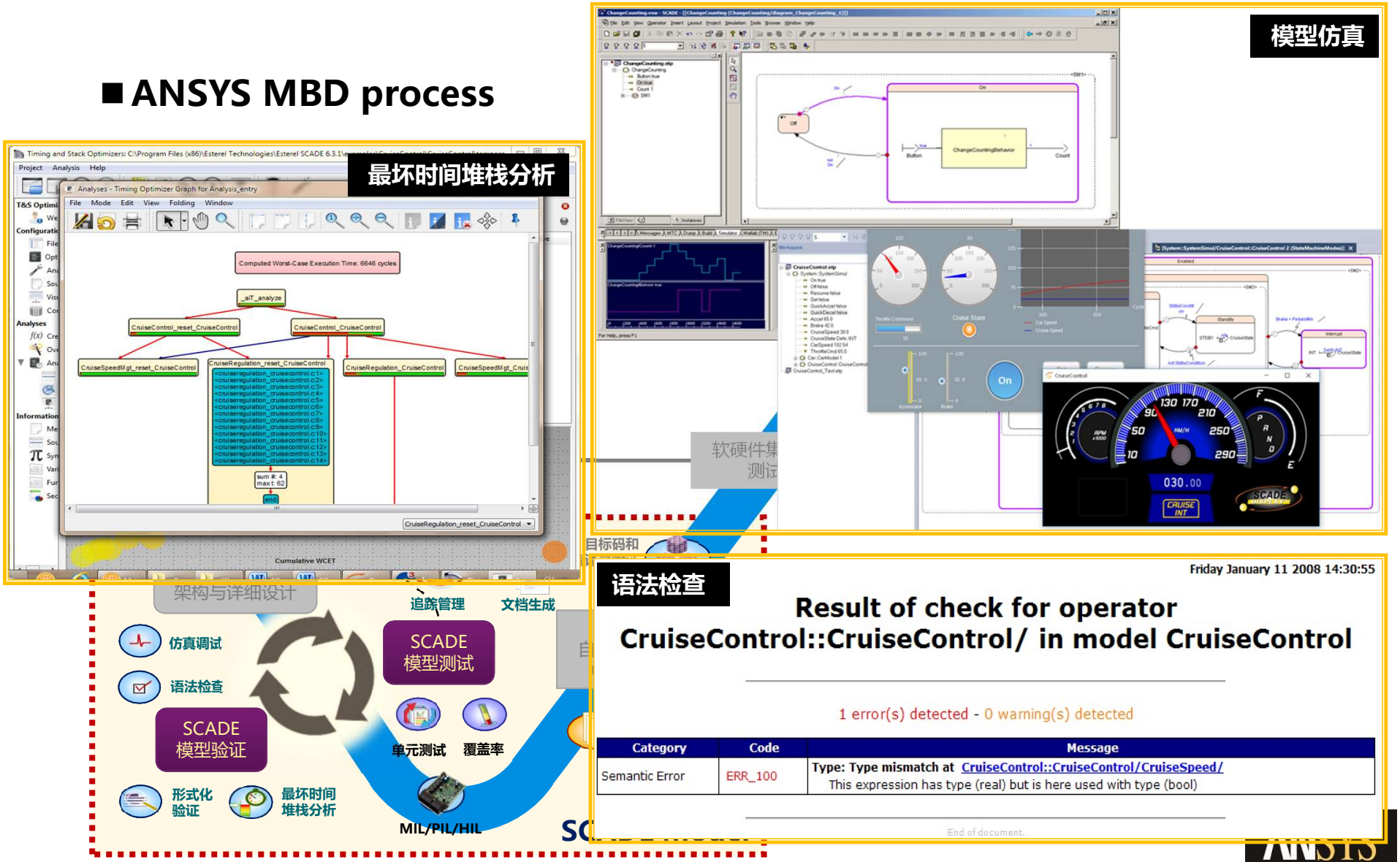
Conclusion of Benefit For ISO 26262 SW Process

ISO 26262	要求	ANSYS SCADE的符合性
Table 1 — Modelling and coding guidelines	★ 软件设计的规范要求： • 明确语言子集、复杂度、强类型、防御性设计、清晰明确的表述、开发工具没有缺陷。	• 包含安全结构和表述的语言子集 • 形式化语言（强类型、清晰明确的表述） • 包含安全状态机、数据结构和操作（部分防御性设计）
Table 4 — Notations for software architectural design Table 11 — Notations for software units design	选取合适的开发方法（非形式化、半形式化、形式化） • 考虑实施的代价和对安全性的保障	• 形式化的设计，具备半形式化和形式化的验证手段
Table 5 — Design principles for software architectural design	★ 架构的设计要求： • 限制规模和复杂度，降低耦合度，限制中断; • 考虑软件安全等级对架构的影响。	• 低耦合的设计（没有全局变量、数据接口的访问机制） • 确定性调用、不支持中断
Table 6 — Error detection at software architecture level Table 7 — Error handling at software architecture level	★ 错误检查和容错设计： • 架构设计中错误的检查和规避; • 架构层面的错误诊断和容错机制——根据安全性需求;	• 提供基础语义支持相关的设计 • 生成的数据易于保存，提供相关接口
Table 12 — Design principles for software unit design and implementation	★ 软件实现的规范要求： • 一个访问接口、没有动态的数据、变量初始化、变量的多次使用、避免全局变量、限制指针、显示的类型转换、限制非条件跳转、无迭代……	• 包含安全状态机、数据结构和操作（ISO26262的要求全为SCADE内嵌的建模约束） • 提供建模规范文档，约束组织、命名和实现相关的特殊要求。

内嵌的安全语义语法建模与防御性设计，减少人因错误和人工检查

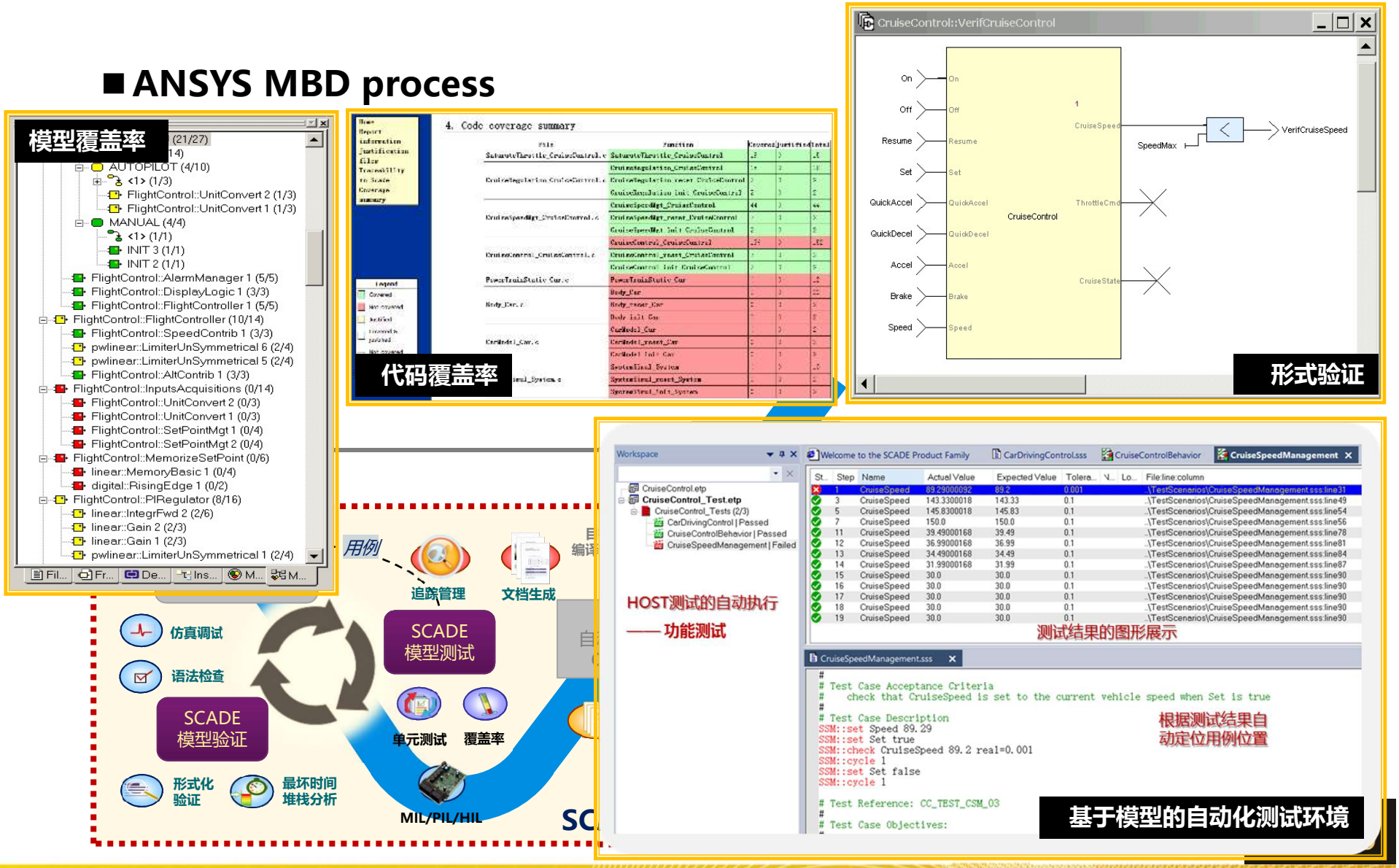
ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

■ ANSYS MBD process



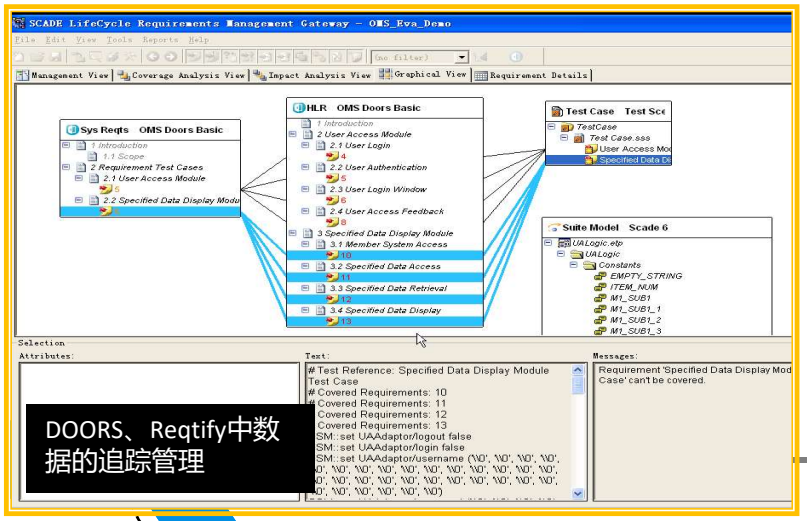
ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

■ ANSYS MBD process

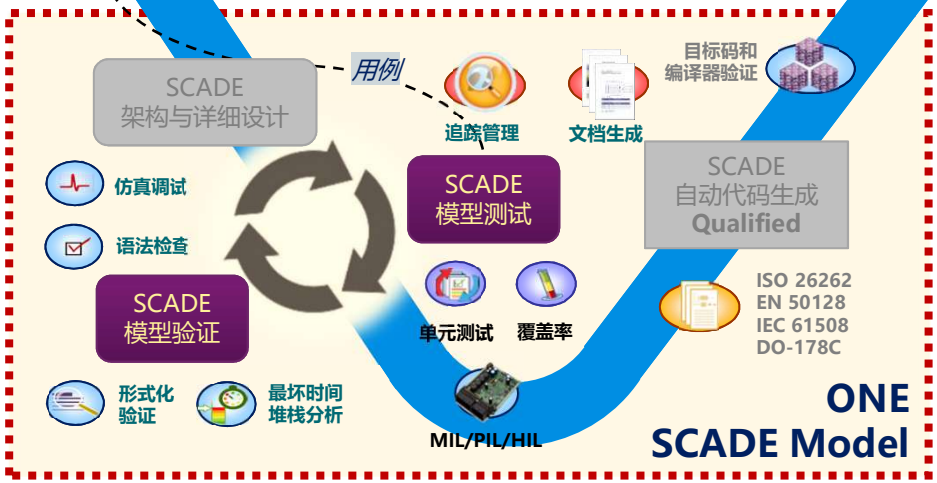


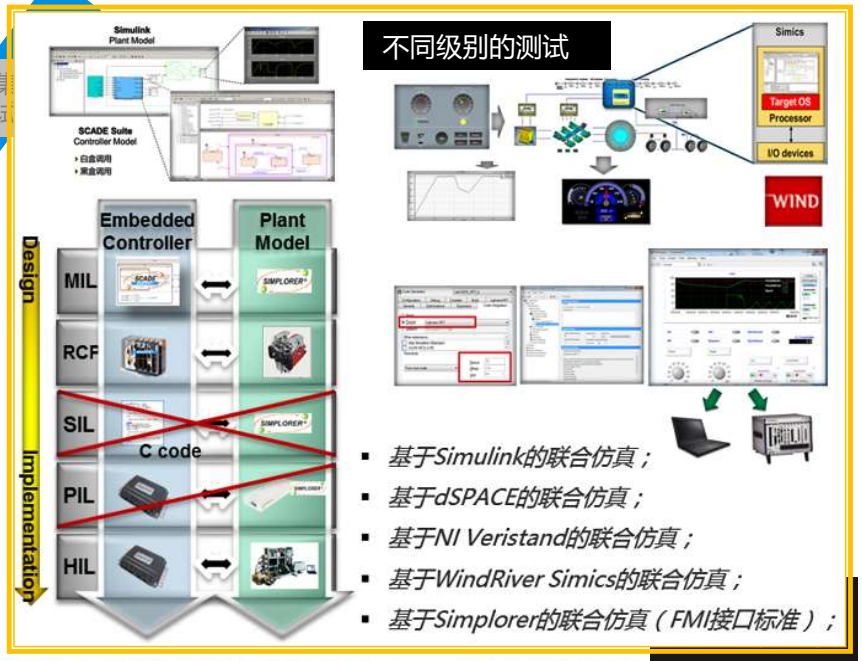
ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

■ ANSYS MBD process

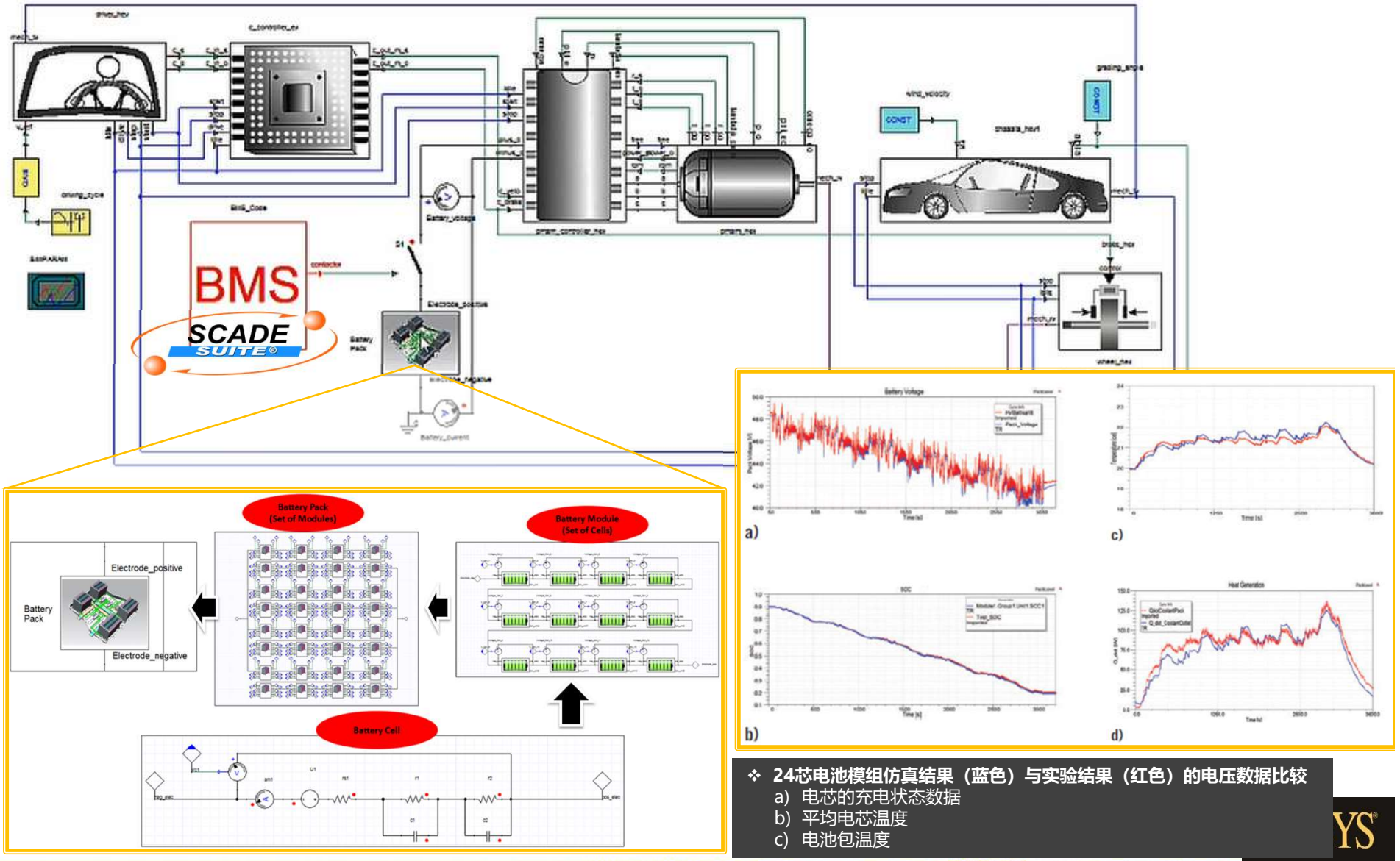


软硬件集成测试





MIL Simulation Example



❖ **24芯电池模组仿真结果（蓝色）与实验结果（红色）的电压数据比较**
- a) 电芯的充电状态数据
- b) 平均电芯温度
- c) 电池包温度

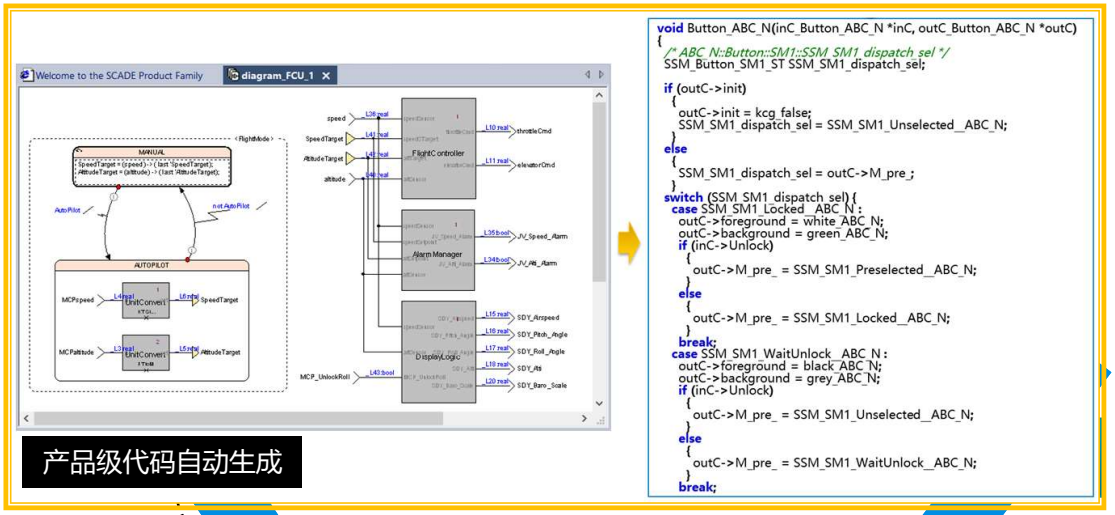
Conclusion of Benefit For ISO 26262 SW Process

ISO 26262	要求	ANSYS SCADE的符合性
Table 2 — Verification of software safety requirements Table 3 — Informal verification of software safety requirements	• 静态分析（工具或人工）	• 走查、评审
Table 8 — Verification of software architectural design Table 9 — Informal verification of software architectural design Table 10 — Semi-formal verification of software architectural design	★ **静态分析（工具或人工）和仿真** • 与需求（安全性需求）/设计的符合性 • 与目标环境的兼容性 • 与需求（安全性需求）（和架构）的追踪性 • 符合设计标准	• 自动语义语法的规则检查 • 自动生成设计报告（人工走查） • 仿真调试 • 时间堆栈分析 • 形式化验证
Table 13 — Verification of software unit design and implementation Table 14 — Informal verification of software unit design and implementation		
Table 15 — Software unit testing Table 16 — Functional software unit testing Table 17 — Structural coverage	★ **模型和代码级的测试** • 基于需求的测试 • 功能和结构测试 • 故障注入和错误假设 • 等价类测试 • 语句、DC、MC/DC覆盖率准则，模型覆盖	• 模型级的测试 • 基于模型的单元、集成测试 • 基于模型的覆盖率
Table 18 — Software integration testing Table 19 — Functional software integration testing Table 20 — Structural software integration testing		
Table 21 — Software safety acceptance testing	基于安全性需求的真机测试	• 形式化验证 • 模型在环的测试

基于模型的验证手段，提前验证减少迭代，形式化的证明手段

ANSYS MBD Process Help for ISO 26262 ASIL C & D Application

■ ANSYS MBD process

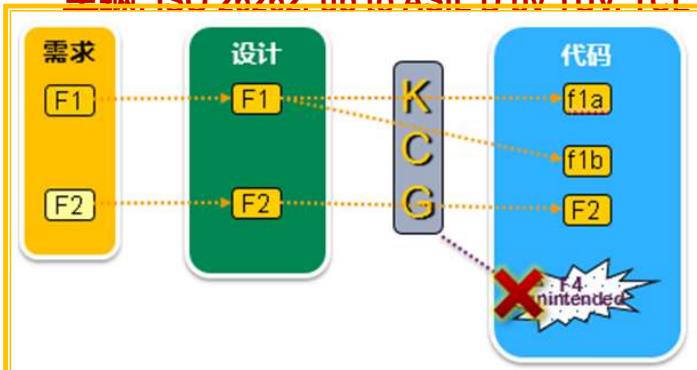



```
void Button_ABC_N(inC_Button_ABC_N *inC, outC_Button_ABC_N *outC)
{
  /* ABC_N::Button::SM1::SSM_SM1_dispatch_sel */
  SSM_Button_SM1_ST SSM_SM1_dispatch_sel;

  if (outC->init)
    {
      outC->init = kcg_false;
      SSM_SM1_dispatch_sel = SSM_SM1_Unselected__ABC_N;
    }
  else
    {
      SSM_SM1_dispatch_sel = outC->M_pre_;
    }
  switch (SSM_SM1_dispatch_sel) {
    case SSM_SM1_Locked__ABC_N :
      outC->foreground = white_ABC_N;
      outC->background = green_ABC_N;
      if (inC->Unlock)
        {
          outC->M_pre_ = SSM_SM1_Preselected__ABC_N;
        }
      else
        {
          outC->M_pre_ = SSM_SM1_Locked__ABC_N;
        }
      break;
    case SSM_SM1_WaitUnlock__ABC_N :
      outC->foreground = black_ABC_N;
      outC->background = grey_ABC_N;
      if (inC->Unlock)
        {
          outC->M_pre_ = SSM_SM1_Unselected__ABC_N;
        }
      else
        {
          outC->M_pre_ = SSM_SM1_WaitUnlock__ABC_N;
        }
      break;
```

产品级代码自动生成

车辆: ISO 26262, up to ASIL D by TÜV, TCL 3



- 确保生成的标准代码与模型的完全一致；
- 确保模型到代码的100%追踪性要求；
- 消除代码的人工走查工作；
- 消除代码级的单元、集成测试工作；
- 消除代码级的覆盖率测试工作。

消除代码
相关活动

Development

符合MISRA-C编码规范

3. MISRA-C:2004 RULES ANALYSIS

This section presents the Analysis of the MISRA-C:2004 rules.

3.1. ENVIRONMENT

1.1 (req) All code shall conform to ISO 9899:1990 "Programming languages – C", amended and corrected by ISO/EIC/9899/COR1:1995, ISO/EIC 9899/AMD1:1995, and ISO/EIC 9899/COR2:1996.

2.3. RESULTS SUMMARY AND SAFETY ANALYSIS

This section presents the summary of Analysis results using the code generated by SCADE Suite KCG 6.4.

Compliance	Number of rules
Full	99
Derivable	16
Configurable	12
Partial	3
External	3
None	6
Not Applicable	2

Table 2: Compliance results

Note that some rules are only violated in imported code of the analyzed test cases: it has been verified that this is only due to the imported code and compliance is considered as full.

Conclusion of Benefit For ISO 26262 SW Process

ISO 26262要求（7.4.8）	ANSYS SCADE的符合性
Compliance with the hardware-software interface specification	设计报告的走查
Completeness regarding the software safety requirements and the software architecture through traceability	• 安全性需求和架构的追踪性基于ALM Gateway； • 模型到代码的追踪性由认证级的KCG保证；
Compliance of the source code with its design specification	认证级的KCG工具保证
Compliance of the source code with the coding guidelines	认证级的KCG工具保证

消除代码验证工作，确保过程数据一致性

ANSYS MBD Great Help for High Safety & Critical Embedded System

■ Leverage brought by ANSYS MBD

❖ 关注和解决

- 更好的软件工程化与质量保障过程的实践；
- 更好的行业认证对软件开发过程要求的实践；
- 软件模块化、系统分层的设计思想；
- 关键安全软件的建模实现和鲁棒性设计。

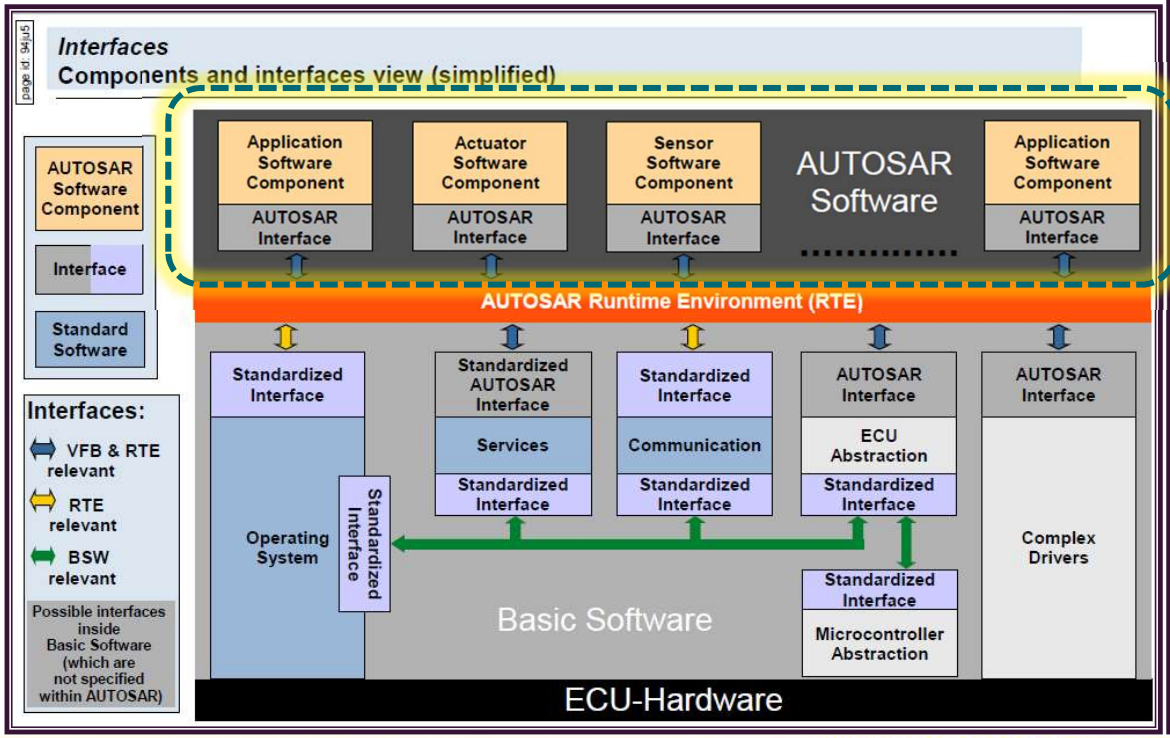
❖ 效益

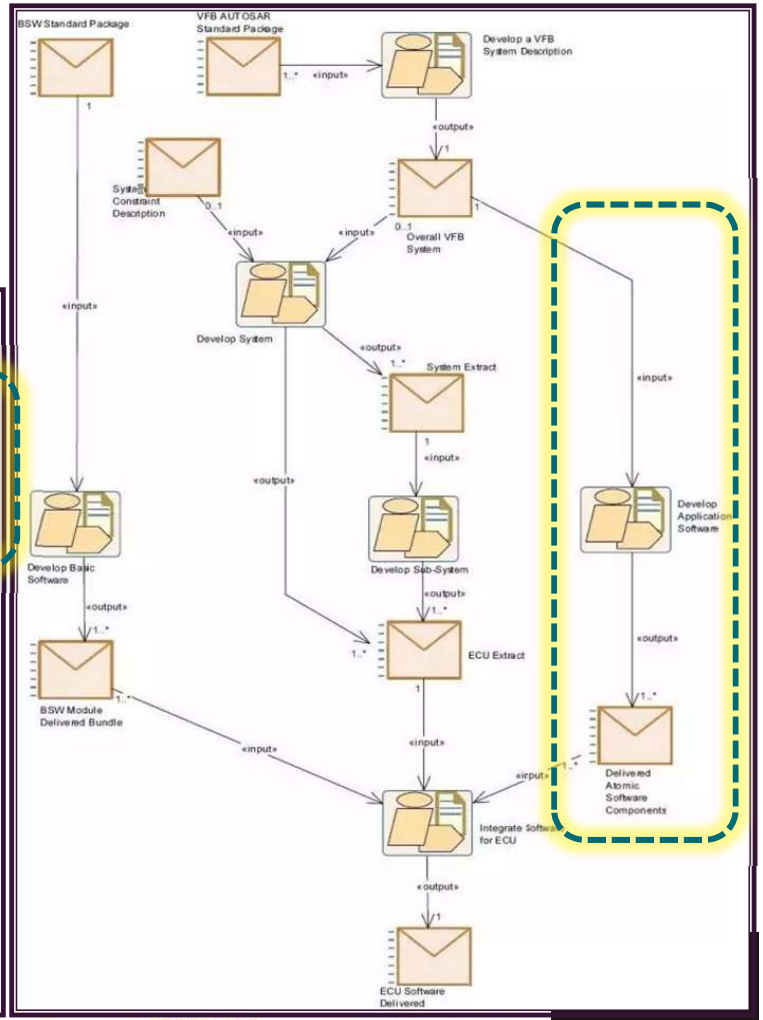
- 认证级项目大于**40%**的时间人力的节省；
- 极高的认证信用，降低认证风险；
- 极大的提高软件质量，规避人因错误；
- 协助推动软件研发流程的不断改进。

Compliance With Automotive Standard (1/3)

■ ANSYS ATUOSAR solution for application layer

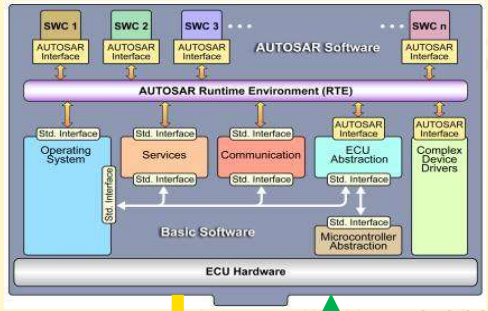
- ✓ Description of SWC on VFB Level
- ✓ Description of SWC on RTE Level





Compliance With Automotive Standard (2/3)

lution for applica

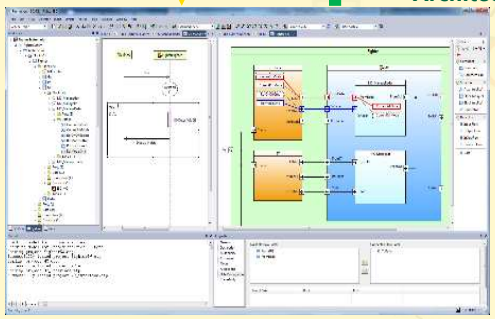


2

AUTOSAR Authoring Tool (AAT)

ARXML

ANSYS SCADE Architect



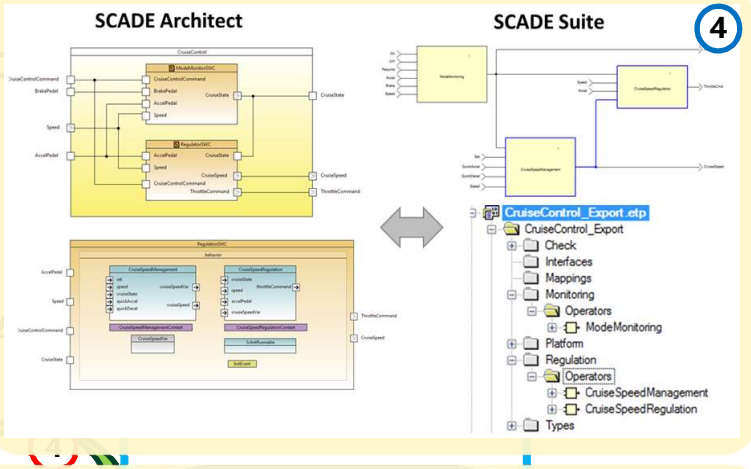
SCADE Architect

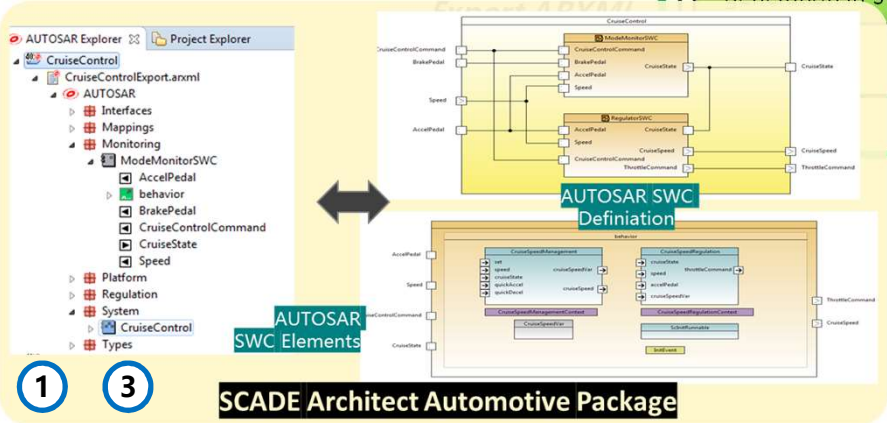
1 Create SCADE AUTOSAR project

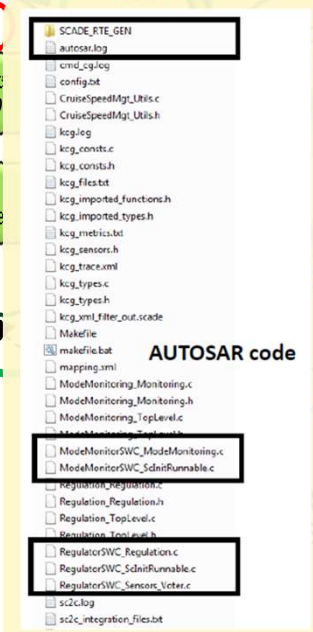
SCADE AUTOSAR model 3

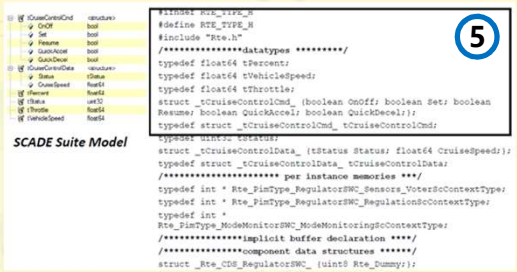
Edit model using concept from AUTOSAR (optional)

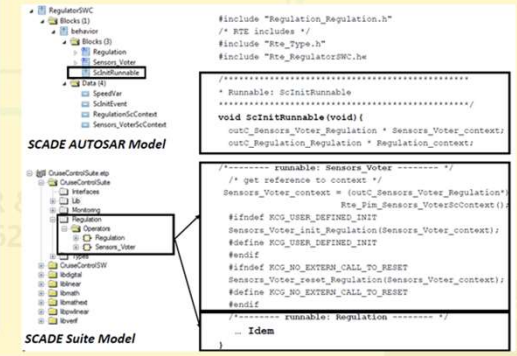
Add data for code generation in SCADE Suite







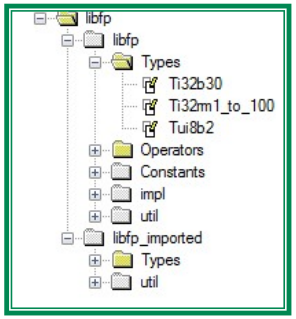


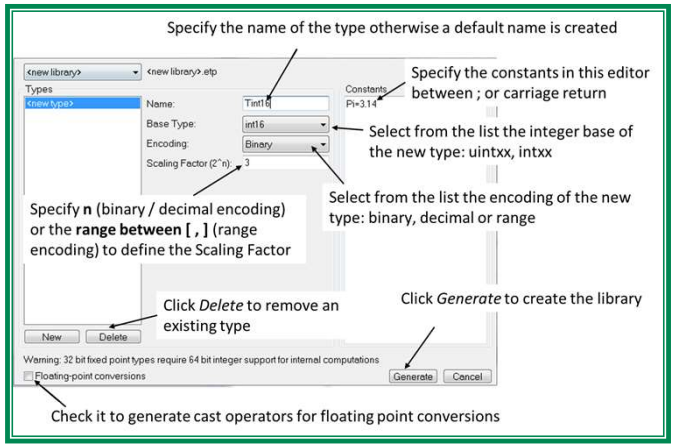


Compliance With Automotive Standard (3/3)

■Support Fixed Point Library

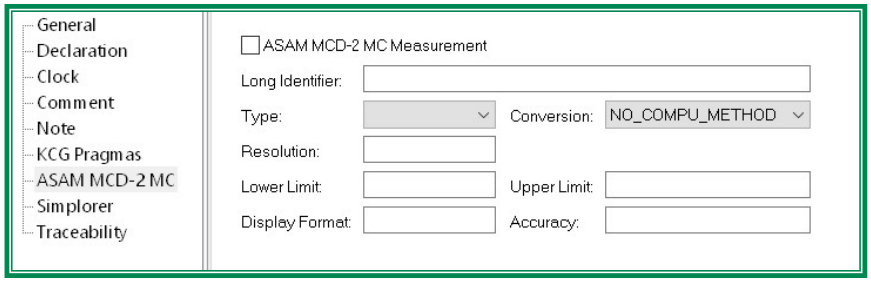
- ✓ Types and constants in base 2, base 10, or range encoding
- ✓ Comparison, arithmetic and cast operators

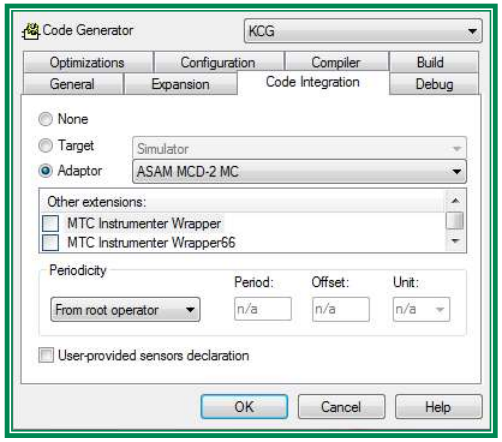




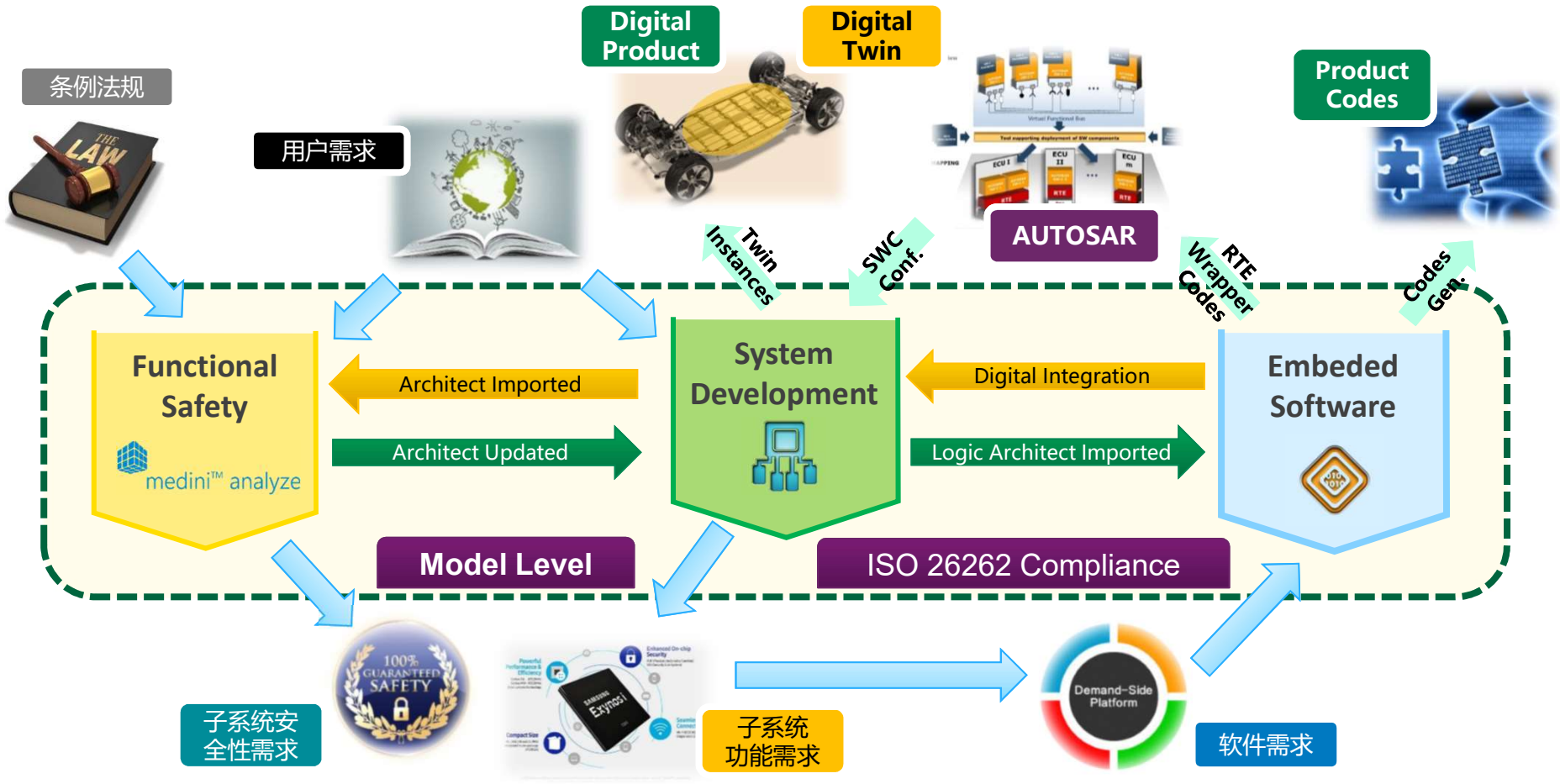
■Support ASAM MCD-2 DC Standard

- ✓ Option for ASAM MCD-2 MC characteristics, calibration vector and measurements
- ✓ Generate ASAM MCD-2 MC file, including A2L file and the declaration of the variables of the calibration vector structures





Completed MBD for Automotor E/E System



THANK YOU