Automotive Software Safety: Current Practice and Future Challenges & Opportunities

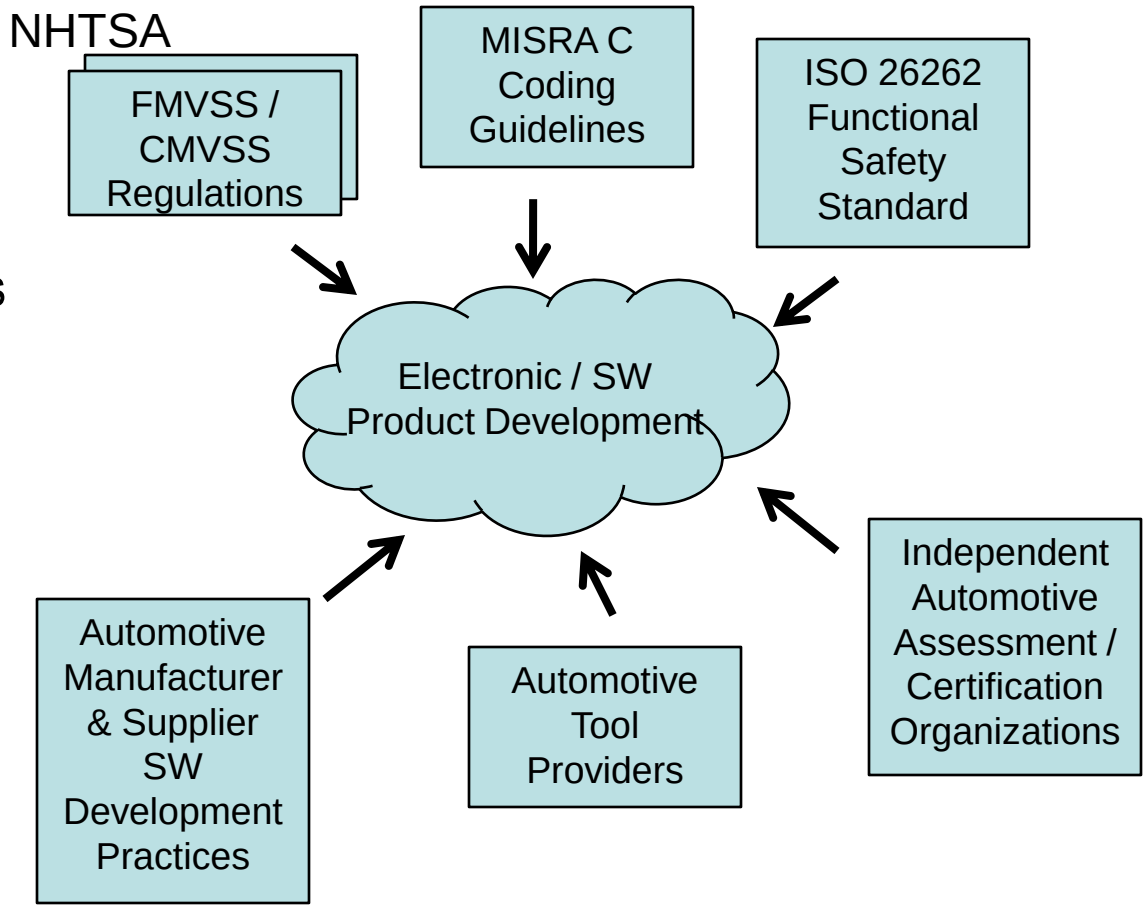
Joseph D'Ambrosio

Lab Group Manager

GM Research Laboratories

Current Automotive "SW Certification" Landscape

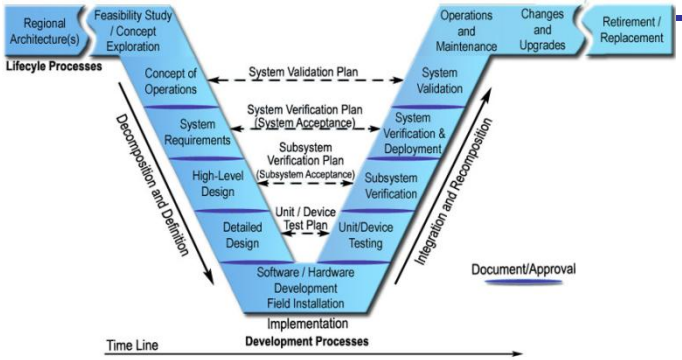
- Automotive companies follow internal best practice SW development methods
- Existing government regulations have very limited influence on SW integrity
- Limited use of "external" independent organizations to assess SW integrity
- Internal independent review / assessment is common
- "Certification" is not practiced
  - Some tools suppliers are starting to certifying their products

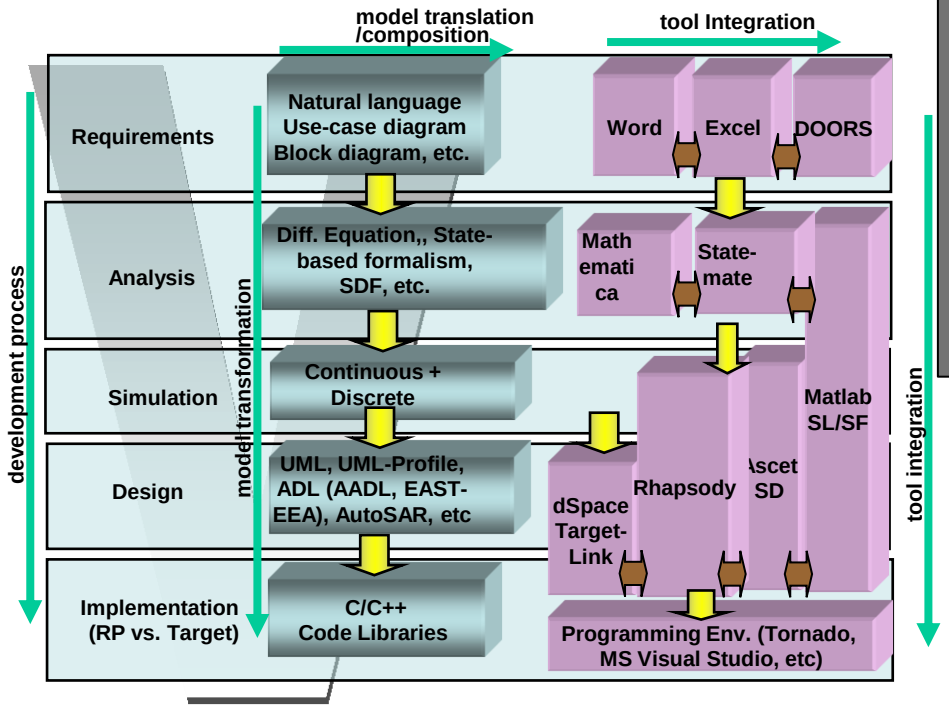


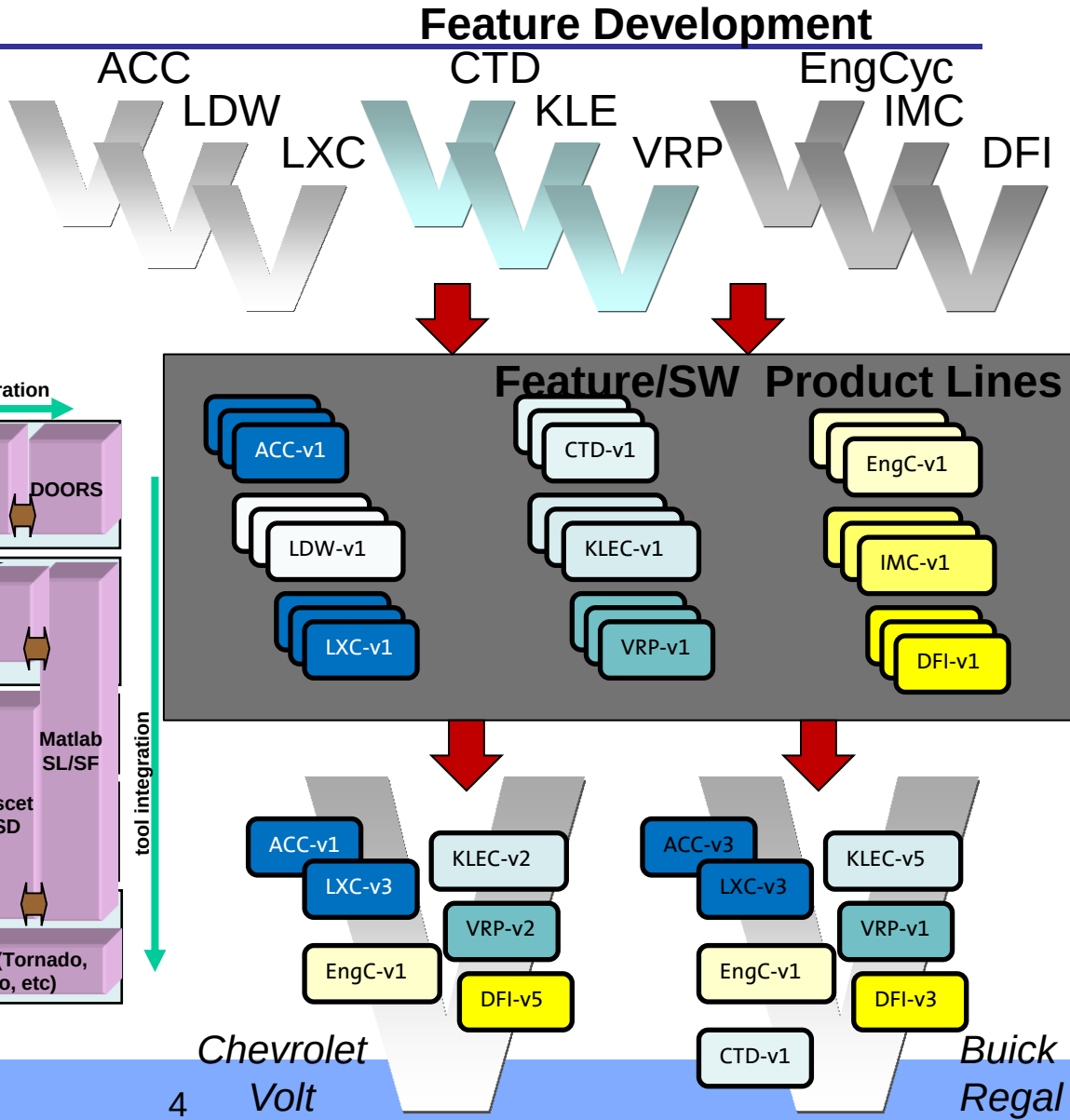
Overview

- Automotive SW Development Practices
- Automotive Software Safety Best Practice & ISO 26262
- Future Developments and Potential Impact of Unintended Acceleration Issues
- Summary & Conclusions

Vehicle Control System Development

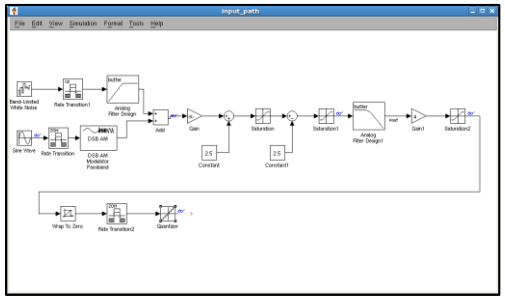




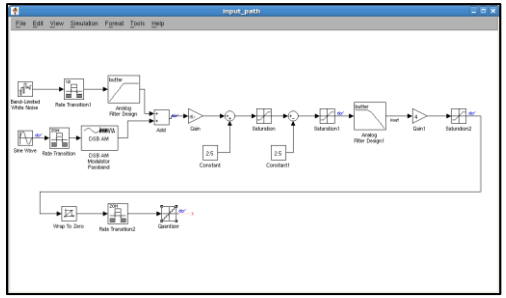


Typical Algorithm Development

Matlab Simulink Model



Matlab Simulink Model



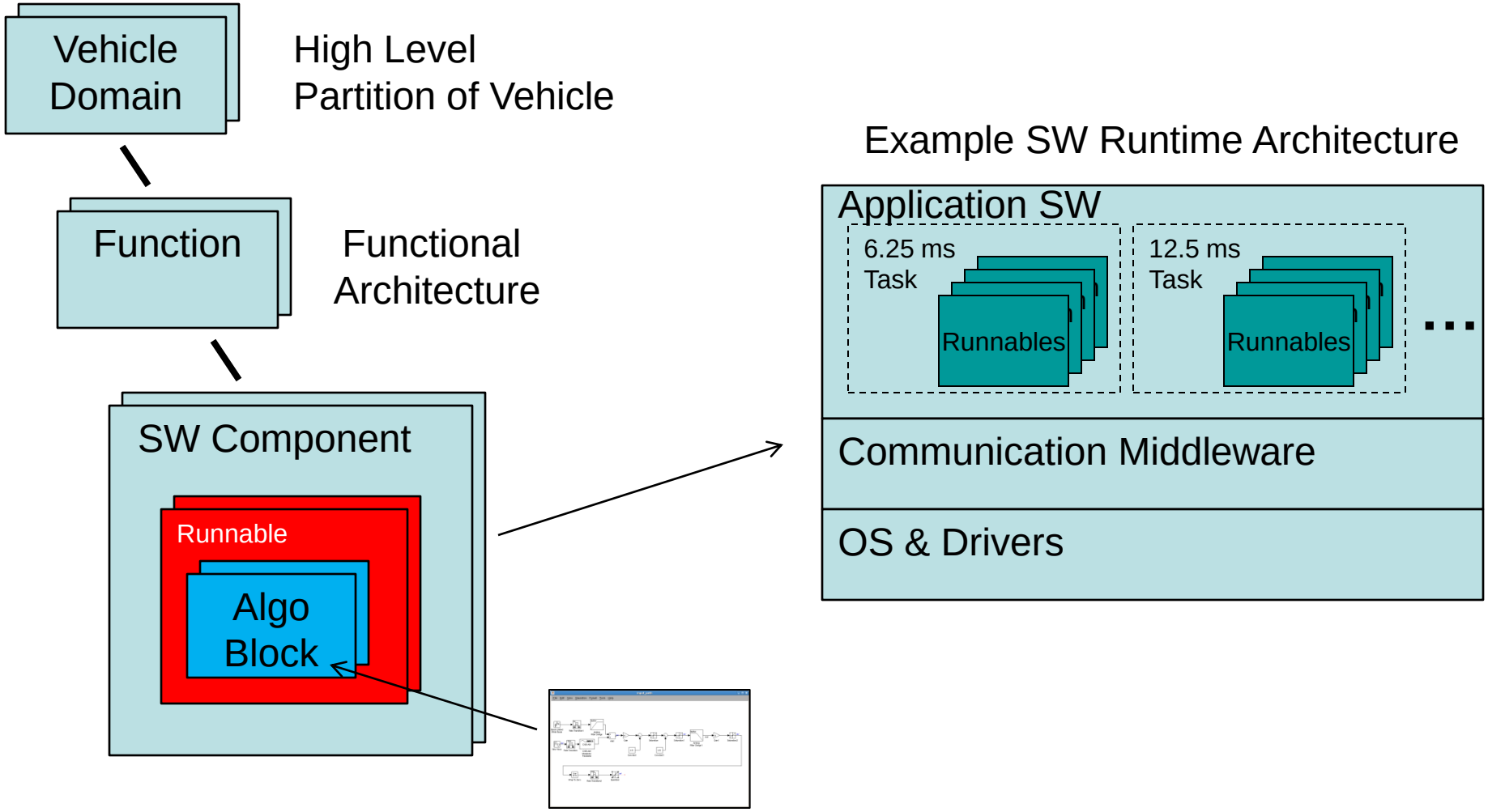
Host PC

CARSim Plant Model

Rapid Prototyping Controller

Surrogate Vehicle

SW Development

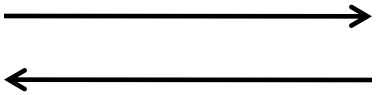


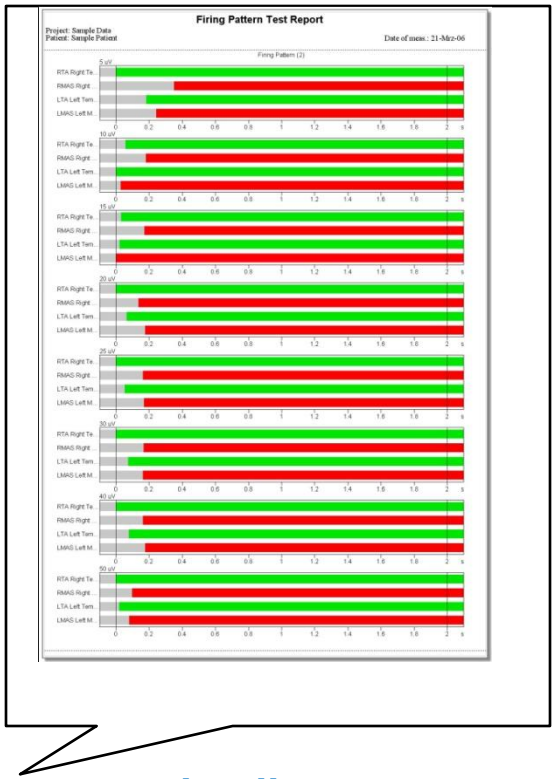
SW-In-The-Loop for Unit Testing

Software Runnable – Win32 Build

```
int *A = malloc(n);
int *B = malloc(n);
int *C = malloc(n);

A_desc = chi_alloc_surface(A, X3000_INPUT, n, 1);
B_desc = chi_alloc_surface(B, X3000_INPUT, n, 1);
C_desc = chi_alloc_surface(C, X3000_OUTPUT, n, 1);
#pragma omp parallel target(x3000) shared(A,B,C)
    descriptor(A_desc,B_desc,C_desc) private(i)
{
  for (i=0; i<n/8; i++)
    __asm
    {
       shl.1.w    vr1 = i, 3
       ld.8.dw    [vr2..vr9] = (A, vr1, 0)
       ld.8.dw    [vr10..vr17] = (B, vr1, 0)
       add.8.dw   [vr18..r25] = [vr2..vr9], [vr10..vr17]
       st.8.dw    (C, vr1, 0) = [vr18..vr25]
    }
}
#pragma omp parallel for shared(D,E,F) private(i)
{
   for (i=0; i<n; i++)
      F[i] = D[i] + E[i];
}
```



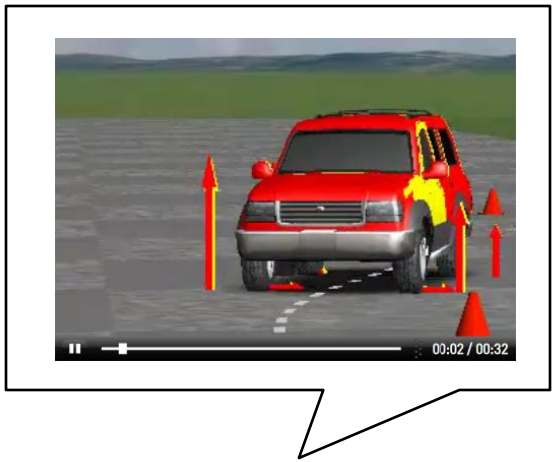


Stimuli

Host PC

HIL for Integration Testing

Plant Model



Software – Target Build

```
int *A = malloc(n);
int *B = malloc(n);
int *C = malloc(n);

A_desc = chi_alloc_surface(A, X3000_INPUT, n, 1);
B_desc = chi_alloc_surface(B, X3000_INPUT, n, 1);
C_desc = chi_alloc_surface(C, X3000_OUTPUT, n, 1);
#pragma omp parallel target(x3000) shared(A,B,C)
    descriptor(A_desc,B_desc,C_desc) private(i)
{
  for (i=0; i<n/8; i++)
    __asm
    {
       shl.1.w     vr1 = i, 3
       ld.8.dw     [vr2..vr9] = (A, vr1, 0)
       ld.8.dw     [vr10..vr17] = (B, vr1, 0)
       add.8.dw    [vr18..r25] = [vr2..vr9], [vr10..vr17]
       st.8.dw     (C, vr1, 0) = [vr18..vr25]
    }
}
#pragma omp parallel for shared(D,E,F) private(i)
{
   for (i=0; i<n; i++)
      F[i] = D[i] + E[i];
}
```

ECU

HIL Simulator

In Vehicle Validation

- Test Track Evaluation
- Pre-Production Vehicle Public Road Evaluation
  - Vehicle qualified for operation on public roads
- Production Vehicle Public Road Captured Test Fleet Evaluation

Target ECU

Target Vehicle

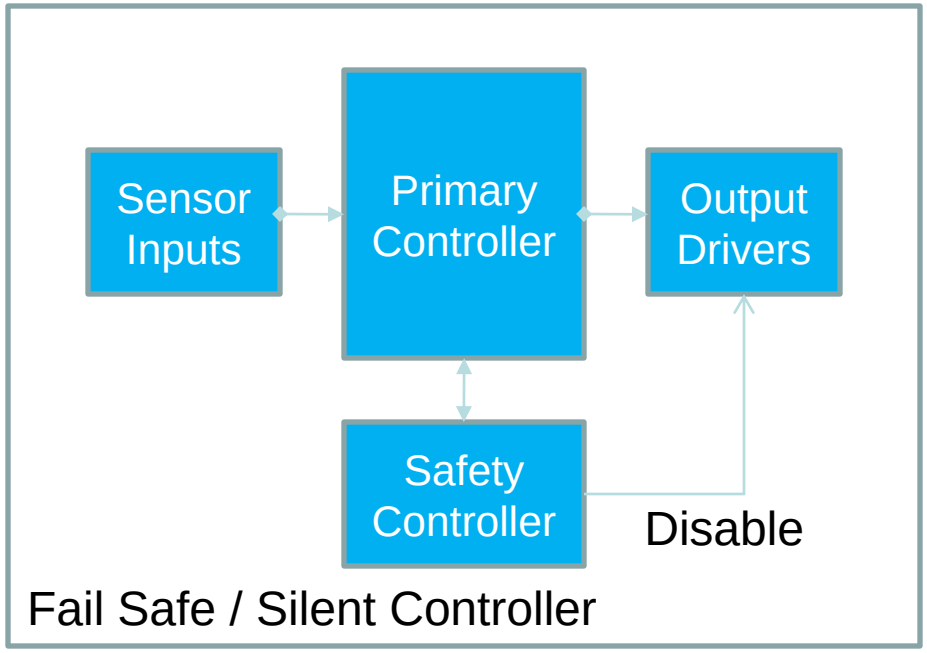
Overview

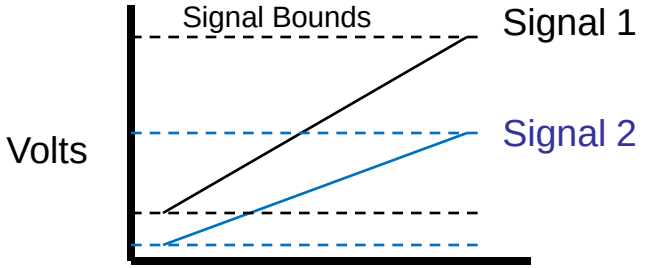
- Automotive SW Development Practices
- ISO 26262 & Automotive Software Safety Best Practice
- Future Developments and Potential Impact of Unintended Acceleration Issues
- Summary & Conclusions

Example Automotive Potential Hazards

- Unintended Acceleration
- Unintended Deceleration
- Unintended Lateral Acceleration
- Loss of Lateral Control / Steering Effort Too High
- Loss of Vehicle Park
- …

Typical Safety-Critical Controller Concept

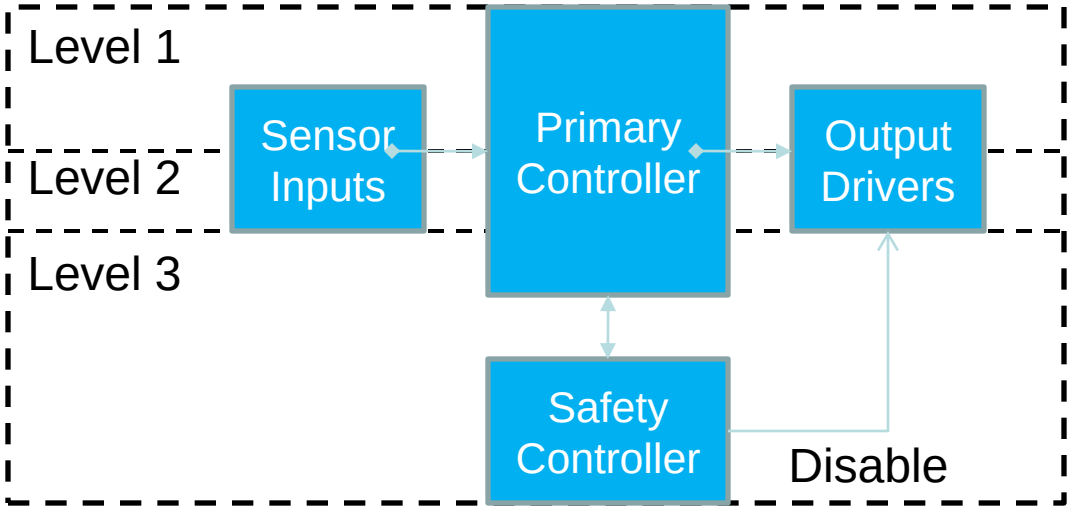




Redundant Sensor Strategy

Typical Safety-Critical SW Concept

(a) Software Safety Architecture



(b) Allocation to Hardware Components

What is ISO 26262?

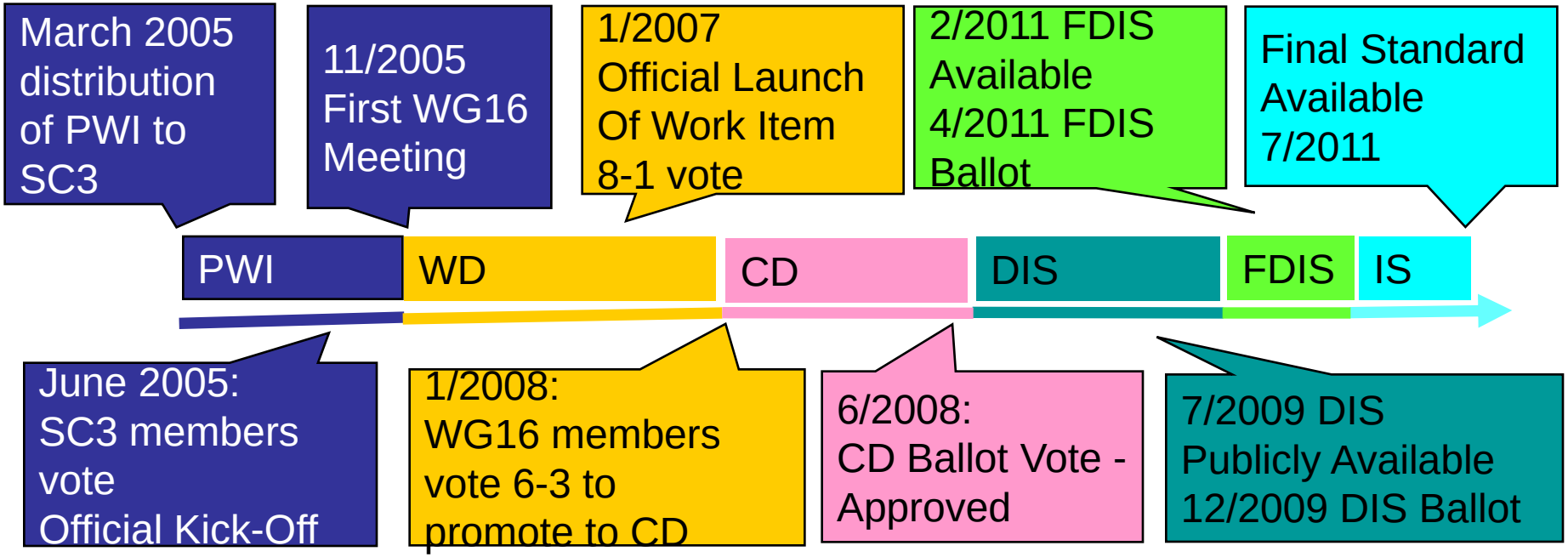
- Adaptation of IEC 61508 to comply with the specific needs of E/E systems within road vehicles
  - Specifies a functional safety life-cycle for automotive products
- Applies to all activities during the safety lifecycle of safety-related systems comprised of electrical, electronic, and software components
- Is a standard, not a regulation
  - Broad industry participation in its development
  - Likely to represent automotive state of the art
- Key concept: Automotive Safety Integrity Level (ASIL)
  - Specify risk associated with a potential hazard
  - Dictate development requirements to achieve required integrity with respect to systematic and random hardware failures

ISO 26262 Working Group 16

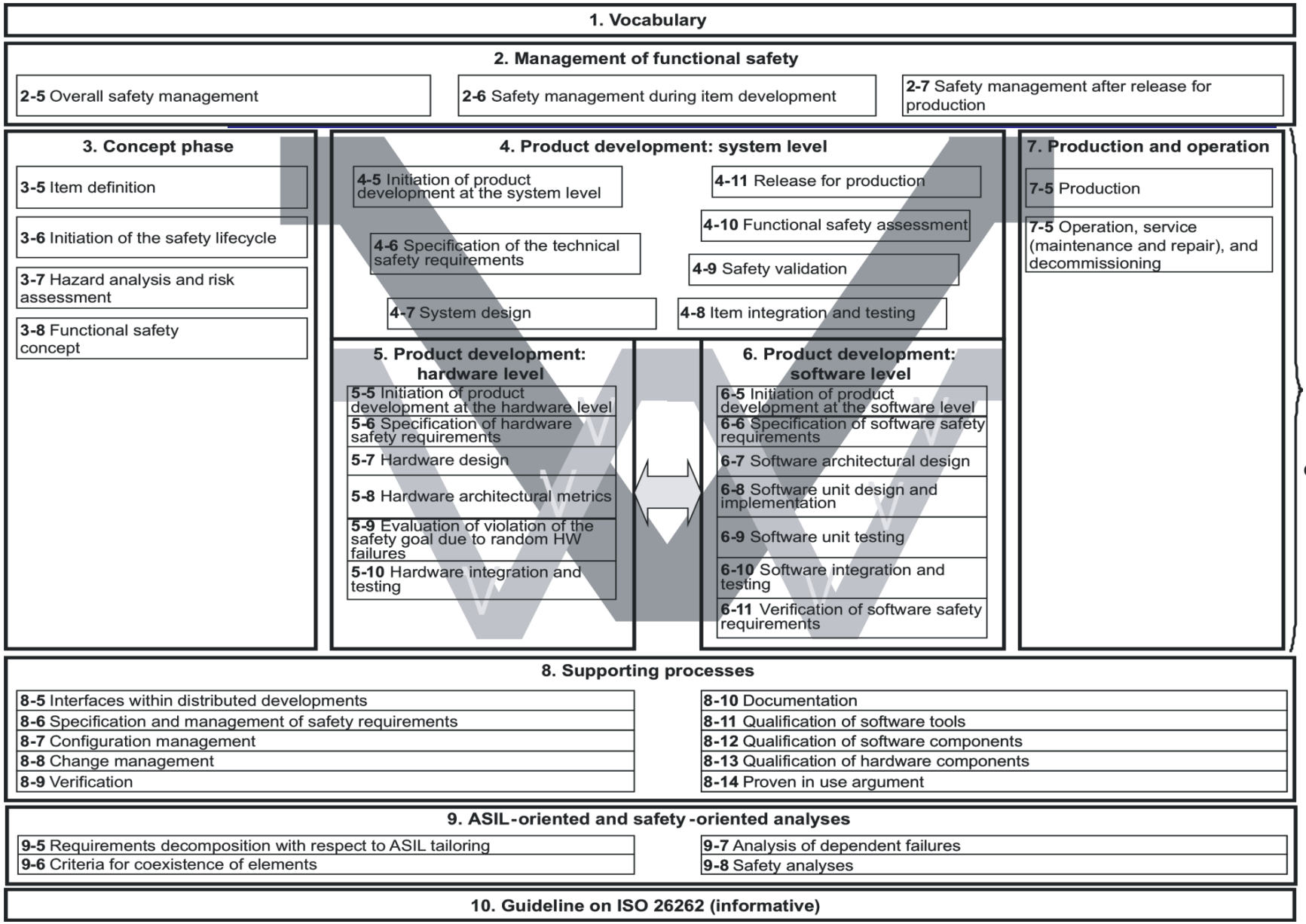
Convenor	Ch. Jung, Independent Consultant
Secretary	E. Fritzsche, VDA
Germany	BMW, Daimler , VW, Bosch, Continental
France	PSA, Renault, Continental, Valeo
UK	Landrover, MIRA, Renesas
Sweden	Delphi, Volvo Cars, AB Volvo, Mecel
Italy	Centro Ricerche Fiat, Fiat Auto, TRW
Japan	Denso, Hitachi, Honda, Nissan, Toyota
USA	GM, IBM, TRW,
Belgium	Nissan, Toyota Motor Europe

Membership as of Nov 2010

ISO 26262 Development Time Line



Overview of ISO/DIS 26262



Source ISO/DIS 26262

Management of Functional Safety

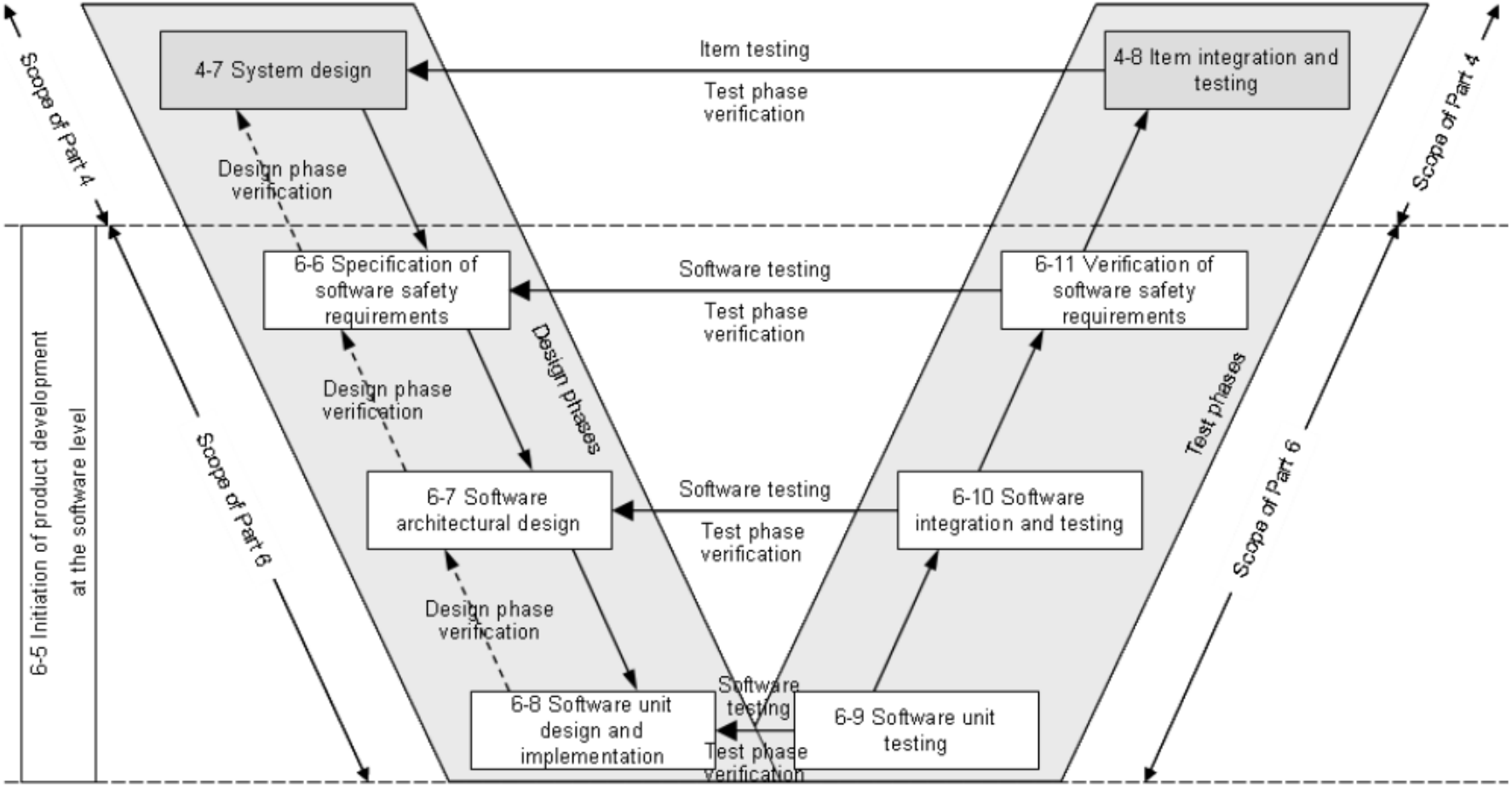
- Reviews / Assessments:
  - ASIL determines level of review independence
- Safety Case Required
  - Compiled set of work products
  - No inconsistencies / Open Items

Table 1 — Required confirmation measures, including the required level of independency

Confirmation measures	Degree of independency[a] applies to ASIL				Scope
	A	B	C	D	
Confirmation review of the hazard analysis and risk assessment of the item (see ISO 26262-3:2011, Clauses 5 and 7, and, if applicable, ISO 26262-8:2011, Clause 5) Independence with regard to the developers of the item, project management and the authors of the work product	I3	I3	I3	I3	The scope of this review shall include the correctness of the determined ASILs and quality management (QM) ratings of the identified hazards for the item, and a review of the safety goals
Confirmation review of the safety plan (see 6.5.1) Independence with regard to the developers of the item, project management and the authors of the work product	—	I1	I2	I3	Applies to the highest ASIL among the safety goals of the item
Confirmation review of the item integration and testing plan (see ISO 26262-4) Independence with regard to the developers of the item, project management and the authors of the work product	I0	I1	I2	I2	Applies to the highest ASIL among the safety goals of the item
Confirmation review of the validation plan (see ISO 26262-4) Independence with regard to the developers of the item, project management and the authors of the work product	I0	I1	I2	I2	Applies to the highest ASIL among the safety goals of the item
Confirmation review of the safety analyses (see ISO 26262-9:2011, Clause 8) Independence with regard to the developers of the item, project management and the authors of the work products	I1	I1	I2	I3	Applies to the highest ASIL among the safety goals of the item
Confirmation review of the software tool qualification report[b] (see ISO 26262-8:2011, Clause 11) Independence with regard to the persons performing the qualification of the software tool	—	I0	I1	I1	Applies to the highest ASIL of the requirements that can be violated by the use of the tool

Source ISO/DIS 26262

Software Development



Reference Phase Model for the Software Development

Source ISO/DIS 26262

Example Software Architecture Design Requirements

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	++	++	++	++
1b	Plausibility check[a]	+	+	+	++
1c	Detection of data errors[b]	+	+	+	+
1d	External monitoring facility[c]	o	+	+	++
1e	Control flow monitoring	o	+	++	++
1f	Diverse software design	o	o	+	++

[a] Plausibility checks can include using a reference model of the desired behaviour, assertion checks, or comparing signals from different sources.

[b] Types of methods that may be used to detect data errors include error detecting codes and multiple data storage.

[c] An external monitoring facility can be, for example, an ASIC or another software element performing a watchdog function.

Source ISO/DIS 26262

Example Software Unit Design Table

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

	Methods	ASIL			
		A	B	C	D
1a	Walk-through[a]	++	+	o	o
1b	Inspection[a]	+	++	++	++
1c	Semi-formal verification	+	+	++	++
1d	Formal verification	o	o	+	+
1e	Control flow analysis[bc]	+	+	++	++
1f	Data flow analysis[bc]	+	+	++	++
1g	Static code analysis	+	++	++	++
1h	Semantic code analysis[d]	+	+	+	+

[a] In the case of model-based software development the software unit specification design and implementation can be verified at the model level.

[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.

[c] Methods 1e and 1f can be part of methods 1d, 1g or 1h.

[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.

Source ISO/DIS 26262

SW Development Work Products

- Safety plan (refined)
- Software verification plan
- Design and coding guidelines for modelling and programming languages
- Software tool application guidelines
- Software safety requirements specification
- Hardware-software interface specification (refined)
- Software verification plan (refined)
- Software verification report
- Software architectural design specification
- Safety analysis report
- Dependent failures analysis report
- Software unit design specification
- Software unit implementation
- Software verification specification (refined)
- Embedded software

Source ISO/DIS 26262

Overview

- Automotive SW Development Practices
- Automotive Software Safety Best Practice & ISO 26262
- Future Developments and Potential Impact of Unintended Acceleration Issues
- Summary & Conclusions

Next Steps – SAE Functional Safety Committee

- Initiated Feb., 15, 2011
  - 30 members, 16 companies
- Mission: common understanding of ISO 26262
- Focus:
  - Harmonizing ASIL assessment methods and levels
  - Harmonizing hazard metrics
    - How to measure for safety goal violation and what specific value constitutes a violation
- Similar activities in Japan & Europe

http://www.sae.org/servlets/works/committeeHome.do?comtID=TEVEFS

Participating Companies:

GM, Ford, Chrysler, FIAT, TRW, Bosch, ZF, Magna, Continental, Autoliv, BWI, MIRA, MOBIS, Kostal, Lab Telemetric, TI

Active recruiting of other companies (including Japanese & European)

Motor Vehicle Safety Act of 2010

- Proposed legislation introduced in 2010
- Prompted by Unintended Acceleration Concerns
- Has 23 major provisions
- Status:
  - No vote taken in 2010
  - Opposition based on budget constraints
  - Future - ???

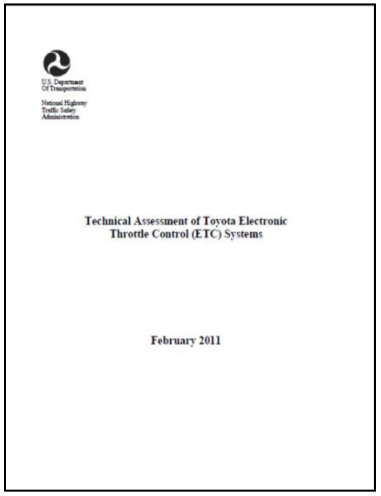
Highlights:

- NHTSA to form Center for Vehicle Electronics, SW, & Emerging Technologies
- Initiate new federal motor vehicle safety standard(s) to:
  - Prevent unintended acceleration through brake override system
  - Prevent pedal obstructions
  - Require electronic systems to meet minimum performance standards
  - Standards fro keyless ignition and gear shift controls
- Increase civil penalties, whistleblower protections, …

Source: http://www.opencongress.org/bill/111-s3302/show

NHTSA Unintended Acceleration Investigation

- March 2010 NHTSA enlisted NASA to support investigation of specific complaints
- NASA did not find an electronic cause
  - Dual-point fault identified, but unlikely the cause
- Future NHTSA actions, include
  - Consider regulations for brake-override, keyless ignitions, & event data recorders
  - Initiate study on reliability / security of electronic control systems
    - Consider NASA recommendation related to controls from other industries, diagnostic trouble codes, SW design & validation methods, protection against dual-fault scenarios
  - Investigate placement of accelerator and brake pedals
  - Continue to enhances its expertise in this area


U.S. Department Of Transportation
National Highway Traffic Safety Administration

Technical Assessment of Toyota Electronic Throttle Control (ETC) Systems

February 2011

- NHTSA has engaged the National Academy of Sciences to study broad issue of electronic control systems in vehicles
  - Recommendations expected in fall of 2011

Source: http://www.nhtsa.gov/UA

Overview

- Automotive SW Development Practices
- Automotive Software Safety Best Practice & ISO 26262
- Future Developments and Potential Impact of Unintended Acceleration Issues
- Summary & Conclusions

Summary & Conclusions

- ISO 26262 represents the overall industry mid-term vision of best practices methods for developing safety-critical software
  - ISO 26262 likely to strongly contribute to automotive state of the art
  - Industry move to ISO 26262 will roll out over the next several years
  - Significant industry effort to make this transition
  - Strong chance that automotive companies will harmonize ASILs and associated metrics
- Not likely that the industry will move towards full certification in the midterm time frame without additional external influences
- Impact of unintended acceleration issues on new potential regulations uncertain