Introduction to Automotive Functional Safety

ISO 26262

Safety Critical Systems

- Safety Critical Systems
  - Systems whose failure or malfunction may result in one (or more) of the following
    - Loss of life or serious injury to people
    - Damage to properties/equipments
    - Damage to the environment
- Examples of safety critical systems
  - Aircraft control systems
  - Robotic surgery machines
  - Railway signal control systems
  - Braking system in vehicles
- Safety-critical systems are increasingly computer-based

Development of Safety Critical Systems

- Creating systems that operate safely by minimizing, controlling and reducing hazards
  - And even if they fail, they are still capable of entering in a controlled safe operation mode
- Different Types of Reliability in Safety Systems
  - **Fail-operational** systems: continue to operate even if their control systems fail
    - Remaining part of the system can complete the operation; E.g. a remote keyless system
  - **Fail-safe** systems: They become safe if they fail
    - They switch to a safe mode and usually inform an operator; E.g. Windows, Insulin pumps and etc.
  - **Fail-secure** systems: They become secure when they fail
    - Usually by locking up to minimize harm; E.g. Electronic doors, lock during power failures
  - **Fail-passive** systems: They continue to operate in the event of a system failure
    - By becoming passive and handing controls over to an operator; E.g. An aircraft autopilot
  - **Fault-tolerant** systems: They continue to operate in the event of failure
    - Usually by detecting at risk components and replace them before they result in any risk
    - E.g. cooling system in a nuclear reactor

Functional Safety

- Absence of **unreasonable risk** of harming people and damaging the environment due to hazards caused by malfunctioning behaviour of systems
- Dependability vs. Safety
  - Dependability is a general term
    - Reliability: Continuity of correct service.
    - Availability: Readiness for correct service.
    - Safety: Absence of harming people and damaging the environment.
- Safety vs. Security (or cyber security)
  - Safety: Absence of **internal** failures which can cause harm or damage
  - Security: Protection against **external** threats which can cause or and damage
  - You can have Security without Safety. But, in most cases, not Safety without Security.
  - We focus on Safety

Functional Safety

- Risk can never be completely eliminated. Probabilities can be reduced.
- Development of safety-critical systems shall be performed according to the state of the art in all aspects of safety.
  - Reducing risks as far as ”reasonably practicable” - good enough.
  - The burden of proof is (will end up) with the manufacturer.
  - Documented proof must be produced.
- International standards are required
- There are different standards for different industries. For example:
  - IEC 62304 for medical software and software within medical devices
  - ISO 26262 for road vehicles functional safety

Automotive Functional Safety

- An example of safety incidents
  - Unintended Acceleration in Toyota products ~ year 2010
  - Toyota recalled 9 millions affected vehicles
  - Toyota payed $1.2 billion to avoid law-suite
  - The issue had been reported within the company during product development
- Potential ”Unsafe” Vehicle Functions
  - Steering system
  - Braking/acceleration systems
  - Air-bag system.
  - Driver information (indirect)
  - And etc.

Automotive Functional Safety - ISO 26262

- ISO 26262 is an adaptation of IEC 61508 for the automotive industry.
  - IEC 61508 is a basic functional safety standard applicable to all industries.
    - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
    - Safety validation can be performed after installation (i.e. process control systems & lines)
    - Safety functions are separated from the system under safety control.
  - In ISO 26262
    - Safety validation is performed before series production.
    - Safety functions are integrated or part of the “normal function”.
- Some vocabularies
  - SIL: Safety Integrity Level
  - ASIL: Automotive Safety Integrity Level
  - Hazard: Potential source of harm caused by malfunctioning behaviour of a function

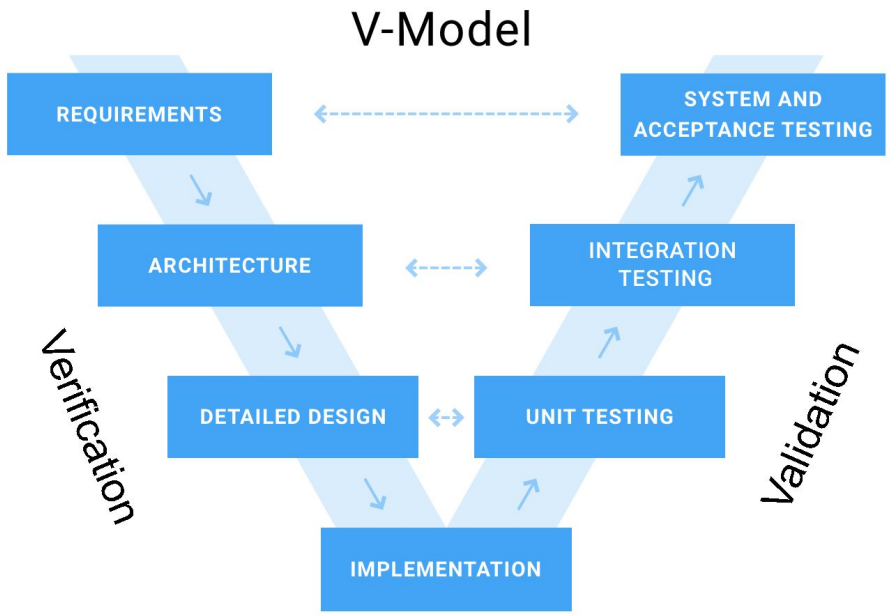
Automotive Functional Safety - ISO 26262

- ISO 26262: Functional Safety for Road Vehicles
  - Part 1: Vocabulary
  - Part 2: Management of functional safety
  - Part 3: Concept phase
  - Part 4: Product development at the system level
  - Part 5: Product development at the hardware level
  - ***Part 6: Product development at the software level***
  - Part 7: Production and operation
  - Part 8: Supporting processes
  - *Part 9: Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses*
  - ***Part 10: Guideline on ISO 26262***

ISO 26262 - Product Development: V-Model

- Requirements document
  - It specifies exactly what the system attempts to accomplish
  - Analyzing the system to identify risks and potential hazards
  - It outlines what the system must do or not do for the sake of safety
  - It must specify how the system will completely fulfill the requirements
- Architecture design
  - E.g. The software architecture based on AUTOSAR
- Detailed design
  - The modules and components
- Implementation
- Verification and validation

ISO 2626 uses a classic V-Model framework at **system**, **hardware** and **software** levels

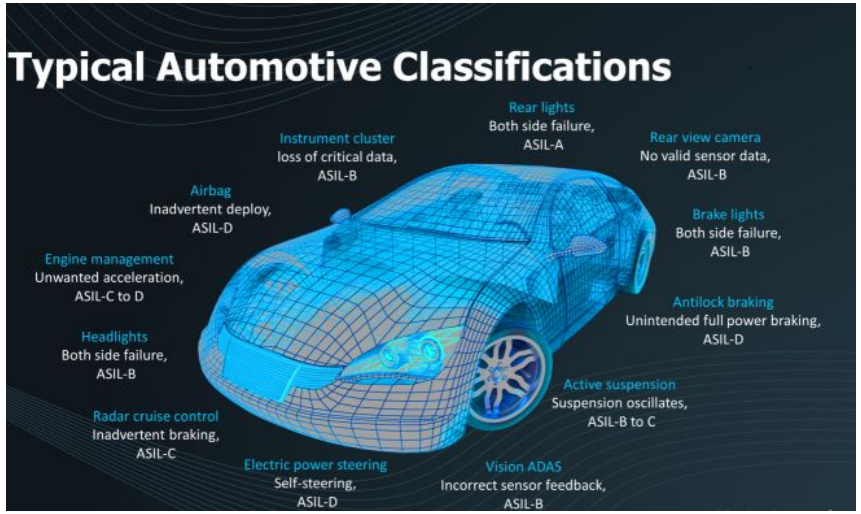


Automotive Functional Safety - ISO 26262

- ASIL (Automotive Safety Integrity Level)
  - Each hazardous event is classified and gets an ASIL (QM, A, B,C or D) depending on
    - Quality Management (**QM**) means that the risk associated with a hazardous event is not unreasonable and therefore it is not subject to ISO 26262 requirements.
    - Severity (**S**) of injuries caused by a hazardous event
      - **S1**: Light to moderate injuries
      - **S2**: Severe to life-threatening injuries
      - **S3**: Life-threatening to fatal injuries
    - Exposure (**E**) - The probability of the operational conditions in which the injury can happen
      - **E1** (Very low), **E2** (Low), **E3** (Medium) and **E4** (High probability)
    - Controllability (**C**) - The relative likelihood that the driver can act to prevent the injury
      - **C1** (Simple), **C2** (Normal) and **C3** (Difficult or Uncontrollable)

Automotive Functional Safety - ISO 26262

- Automotive safety integrity level and a typical classification



Severity	Exposure	Controllability		
		C1 (Simple)	C2 (Normal)	C3 (Difficult, Uncontrollable)
S1 LIGHT AND MODERATE INJURIES	E1 (Very low)	QM	QM	QM
	E2 (Low)	QM	QM	QM
	E3 (Medium)	QM	QM	A
	E4 (High)	QM	A	B
S2 SEVERE AND LIFE THREATENING INJURIES – SURVIVAL PROBABLE	E1 (Very low)	QM	QM	QM
	E2 (Low)	QM	QM	A
	E3 (Medium)	QM	A	B
	E4 (High)	A	B	C
S3 LIFE THREATENING INJURIES, FATAL INJURIES	E1 (Very low)	QM	QM	A
	E2 (Low)	QM	A	B
	E3 (Medium)	A	B	C
	E4 (High)	B	C	D

QM (Quality Management)
Development supported by established Quality Management is sufficient.

A **lowest ASIL**
Low risk reduction necessary

B

C

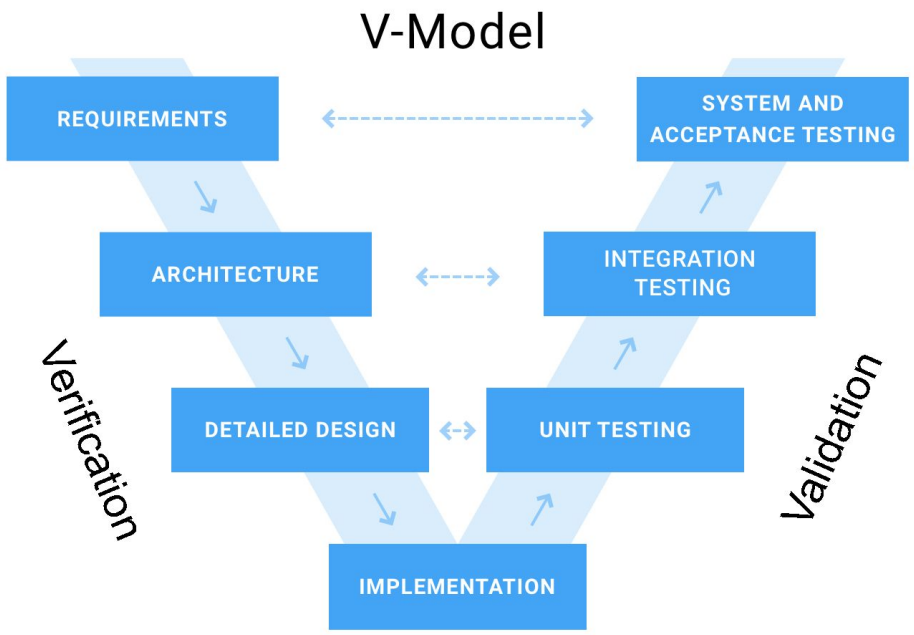
D **highest ASIL**
High risk reduction necessary

Automotive Functional Safety - ISO 26262

- ISO 26262 Safety goals and Safety Concept
  - Hazardous events are identified using hazard analysis and risk assessment.
  - An Automotive Safety Integrity Level (ASIL) is assigned to each hazardous event.
  - A safety goal is determined for each hazardous event, inheriting the ASIL of the hazard.
  - Functional safety concept is a statement of the architecture to achieve the safety goals.
    - Broken down into functional safety requirements.
  - Technical safety concept is a statement of how this functionality is implemented in hardware or software.
    - Broken down into technical safety requirements
- **E.g.** Hazardous event: Parking brake activated at high speed (typically as ASIL D)
  - The safety goal: The Parking brake must not spontaneously or manually be activated at speeds over 5 km/h.

Automotive Functional Safety - ISO 26262

- ISO 26262 Part 6: Product development at the software level
  - Specification of software safety requirements.
  - Software architectural design.
  - Software unit design and implementation.
  - Software unit testing.
  - Software integration and testing.
  - Verification and validation of software safety requirements in each phase.



Automotive Functional Safety - ISO 26262 (Guidelines)

- Modelling and coding guidelines
  - Enforcement of low complexity
  - Use of language subsets (MISRA C)
  - Enforcement of strong typing
  - Use of defensive implementation techniques
  - Use of well-trusted design principles
  - Use of unambiguous graphical representation
    - E.g. Block diagrams, graphs and etc.
  - Use of naming conventions
  - Use of style guides
    - Like Visual Studio Style Guides for C/C++
    - Activate it in Visual Studio Code


```
{
    "C_Cpp.updateChannel": "Insiders",
    "git.autofetch": true,
    "window.zoomLevel": 0,
    "workbench.startupEditor": "none",
    "typescript.updateImportsOnFileMove.enabled": "always",
    "terminal.integrated.rendererType": "dom",
    "editor.fontFamily": "monospace",
    "terminal.integrated.fontFamily": "monospace",
    "editor.formatOnPaste": true,
    "editor.formatOnSave": true,
    "editor.formatOnType": true,
    "editor.fontLigatures": true,
    "editor.autoIndent": "full",
    "workbench.iconTheme": "vscode-icons",
    "vsicons.dontShowNewVersionMessage": true,
    "workbench.colorTheme": "Default Light+",
    "doxdocgen.generic.authorEmail": "faroch.mehri@ya.se",
    "doxdocgen.generic.authorName": "Faroch Mehri"
}
```

Automotive Functional Safety - ISO 26262 (Guidelines)

- Principles for software architectural design
  - Hierarchical structure of software components
  - Restricted size and complexity of software components
  - Restricted size of interfaces
  - Strong cohesion within each software component
  - Loose coupling between software components
  - Appropriate scheduling properties (timing property of the components)
  - Appropriate management of shared resources
  - Appropriate isolation of the software components
  - Restricted use of interrupts
  - Use of design notation, like pseudocode, flow charts and structure charts

Automotive Functional Safety - ISO 26262 (Guidelines)

- Design principles for software unit design and implementation
  - Use of semi-formal notation; like UML, timing diagram and etc.
  - One entry and one exit point in subprograms and functions
  - No dynamic objects or variables, or else online test during their creation
  - Initialization of variables
  - No multiple use of variable names
  - Avoid global variables or else justify their usage
  - Restricted use of pointers
  - No implicit type conversions
  - No hidden data flow or control flow
  - No unconditional jumps
  - No recursions

Automotive Functional Safety - ISO 26262 (Guidelines)

- Methods for software verification
  - Walk-through
  - Pair-programming
  - Semi-formal verification
  - Control flow and data flow analysis
  - Static code analysis; like IAR, CodeSonar, Parasoft C/C++test and etc.
  - Requirement-based test
  - Interface test
  - Fault injection test
  - Resource usage evaluation
  - Analysis of boundary values

Automotive Functional Safety - ISO 26262 (Guidelines)

- Structural coverage metrics
  - Statement coverage
  - Branch coverage
  - Modified condition/decision coverage (MC/DC)
    - Each entry and exit point is invoked
    - Each decision takes every possible outcome
    - Each condition in a decision takes every possible outcome
    - Each condition in a decision is shown to independently affect the outcome of the decision.
    - Independence of a condition is shown by proving that only one condition changes at a time.
  - Function coverage
  - Call coverage

https://www.feabhas.com/sites/default/files/2016-06/A%20quick%20guide%20to%20ISO%2026262[1]_0_0.pdf