jama software

ISO 26262 and Automotive Electronics Development

How New Standards Impact Traceability, Risk Management, Validation & Verification

Compliance standards, especially those that involve relatively new functional safety elements, will likely add additional requirements to the development process. But ISO 26262, in particular, will add more than new requirements to the product life cycle for automotive hardware-software systems. This Functional Safety standard will act as a framework impacting integrated requirements traceability, risk management, validation, verification, documentation and collaboration throughout the systems engineering "V" model life cycle process (see Figure). ISO 26262 will also require the qualification of tools used to create automotive systems. This paper examines the impact of the standard on the development process and support tool chains for automotive electronics.

Design for Safety (DFS) Is Not Enough

The increasing complexity and abundance of automotive electronic systems led to the creation of a new functional safety standard called ISO 26262. It is an adaption of IEC 61508, a more generic industrial functional safety standard. Similar regulations abound, such as CENELEC EN 501128 for railway standards, DO-178B/C for aerospace standards and IEC 60601 for medical standards.

Common to all of these safety standards is a risk-based approach to determine the criticality and potential hazards associated with key system functions. The identification and assignment of risk happens in the early architectural phase of the design, before functionality has been partitioned into hardware and software subsystems. The primary goal of these standards is to prevent the failure of a system or device that could cause injury, harm or death. If a failure is unavoidable, then the system should fail gracefully.

ISO 26262 complements good systems engineering practices by requiring that hardware and software safety concerns be addressed and documented in a systematic way throughout the product life cycle. This includes safety requirements traceability, validation and verification, integration and testing, as well as production, maintenance and even end-of-life cycle considerations.

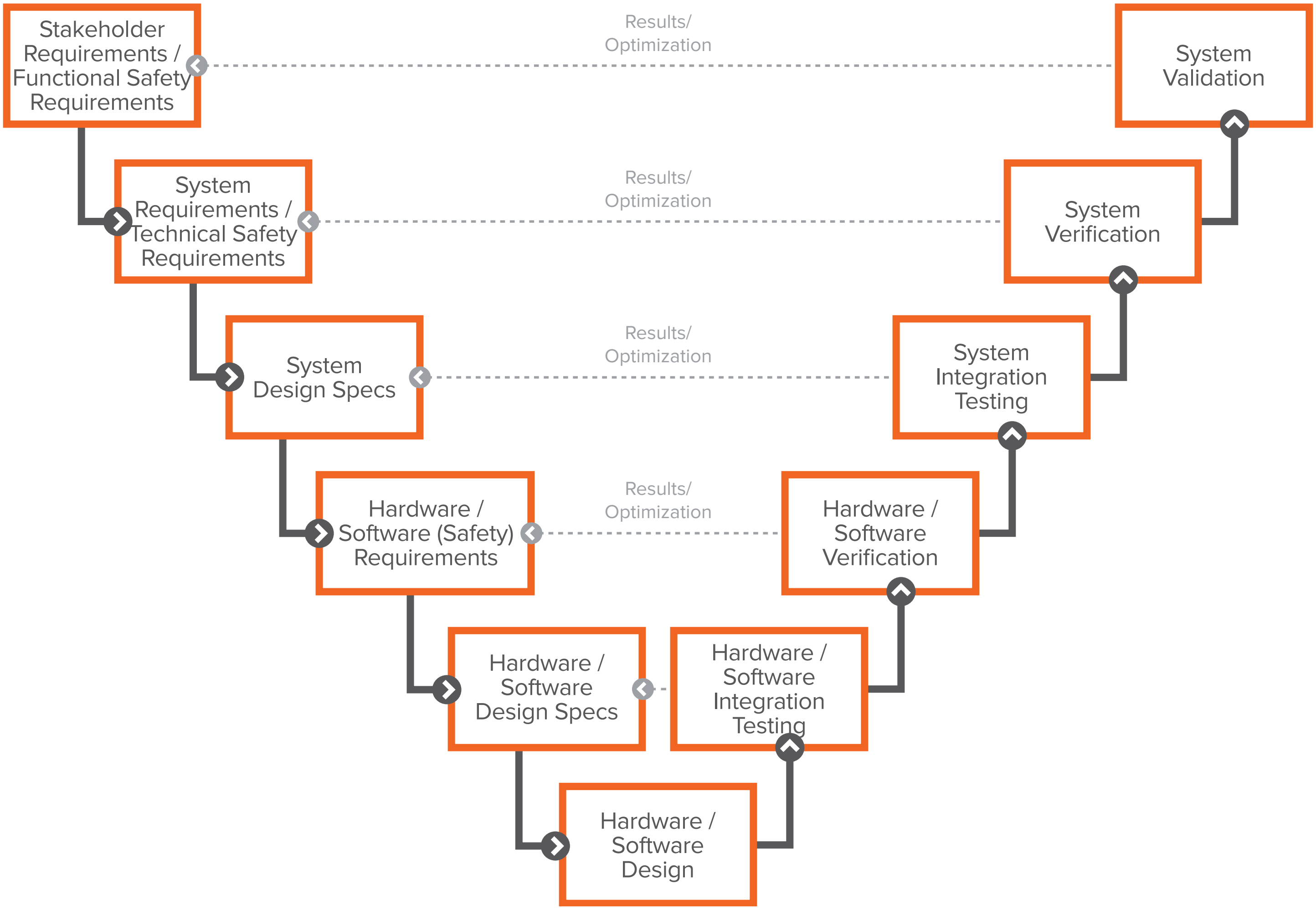

Stakeholder Requirements / Functional Safety Requirements
Results/ Optimization
System Validation
System Requirements / Technical Safety Requirements
Results/ Optimization
System Verification
System Design Specs
Results/ Optimization
System Integration Testing
Hardware / Software (Safety) Requirements
Results/ Optimization
Hardware / Software Verification
Hardware / Software Design Specs
Hardware / Software Integration Testing
Hardware / Software Design

Figure: Life cycle "V" diagram for automotive electronic systems. (Courtesy SAE: Software-Hardware Integration in Automotive Product Development)

In the past, safety design was considered part of a general requirements activity. But merely identifying and tracing requirements in the software and hardware teams are not enough. The common practice of implementing a design in isolation (i.e., hardware and software teams working in silos) will not guarantee the kind of safety coverage required by ISO 26262. Even with the best of tools for requirements management, modeling, IDEs and testing there can still be a gap between the high-level design phase and the lower level component creation, integration and testing portions of the life cycle. How can the problem be resolved?

One of the key things missing from the general approach to requirements are the traceability links between phases. Many tools do a great job of requirements management and traceability within a particular phase but provide a poor auditable trail for traceability between phases. The activities of comprehensive and complete life cycle traceability become an auditing afterthought to be finished after the project is completed. This is the result that ISO 26262 tries to avoid through documented attention to the development process, decision making and selection of supporting tools.

How the tool is used within a tool chain will determine the probability that an error introduced by the tool will be detected. A confidence level is assigned to a given tool, or a flow within a tool, based upon the probability that it will insert or cause an error, combined with the likelihood that the error will be detected during the development process. There has been some confusion around who assigns these confidence levels. Tools or vendors will often provide their own confidence levels but it's ultimately the responsibility of the company using the tool define their own TCL based on their intended use.

The standard recognizes the value of tools (and components) that have been used without incidents in other applications. This means that legacy systems predating ISO 26262 certification yet developed in accordance with past best practices might qualify, depending upon the similarities of the past and current usage model. The importance of the tool confidence level is that it will determine the cost an organization must invest in tool qualification.

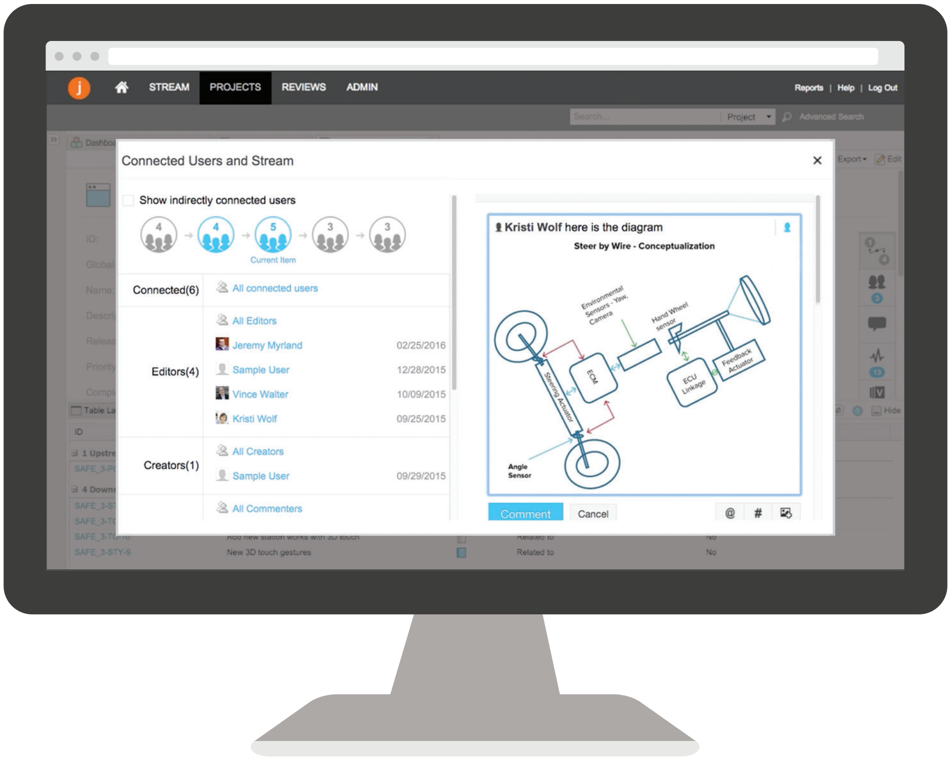
A good example of the broader holistic approach offered by ISO 26262 is shown in its distinction between failure mode and hazards when conducting risk analysis. The standard requires that hazards be considered from a more top-down approach that considers the system context and environment in which the human user interacts with the product or component. As Mike Bucala, Lead Engineer for Vehicle Systems Quality at Daimler Trucks NA and member of the ISO 26262 committee, explains: "The ISO standard is different from other risk standards because it focuses on hazards to persons that result from the malfunctioning behavior of EE systems–as opposed to the risk of failure of a product. For purposes of liability and due care, reducing that risk implies a certain rigor in documentation that has never been there before."

Tool Impact

ISO 26262 describes a qualification process to ensure that software tools and even the tool chains are suitable for safety-intense systems. Tool qualification depends upon how the tool is used, which in turn determines what impact the tool could have on safety. For example, depending upon its usage, can the tool introduce a hardware defect or software bug into the system?

Implementation Strategies

Both tools and process qualifications must be initiated to fully implement a safety standard like ISO 26262. One way to qualify tool and tool suites is to provide a common usage model. Of course, the usage model will be different for each application of the tool, e.g., requirements generation versus modeling or configuration management. For example, satisfying the safety requirements of a project requires focusing on the way defects and errors are introduced, detected and mitigated in the tool along with the organizational procedures to raise and communicate such issues to design teams.



Connected Users and Stream

☐ Show indirectly connected users

4 → 4 → 5 → 3 → 3

Current Item

Connected(6)	All connected users	
Editors(4)	All Editors	
	Jeremy Myrland	02/25/2016
	Sample User	12/28/2015
	Vince Walter	10/09/2015
	Kristi Wolf	09/25/2015
Creators(1)	All Creators	
	Sample User	09/29/2015
	All Commenters	

Kristi Wolf here is the diagram

Steer by Wire - Conceptualization

SAFE_3-STY-9 New 3D touch gestures Related to No

Process-wise, many companies striving for ISO 26262 certification will start by implementing the standard on a "proof of process" pilot project. This approach highlights the changes needed in both the development process and tools suites to comply with the standard. Changes may also be needed in the organizational mindset as safety is more than just a checklist of enhanced documentation and audits. As with other standards, implementing the ISO 26262 process requires iteration through a number of basic steps:

1. **Determine the existing process and tools, or, "Where are we now?" Review the current embedded hardware and software development processes and tool chains. Understand the application(s) to be developed and assign levels of confidence in terms of safety.**

2. **Gap analysis or, "Where would we like to be?" Perform a gap or impact analysis to identify current challenges and process efficiency improvements– often done using model-based design techniques.**

3. **Training and Instruction. Provide design-for-safety training and instruction to address the previously identified gaps.**

4. **Hands-on deployment support. Apply the knowledge gained in the previous steps to a specific pilot project. This will require assistance in a wide range of areas including requirements traceability, modeling, simulation, code generation, verification, validation, tool qualification and system integration.**

Alignment with Best Practices

This holistic approach to functional safety exemplifies several key elements of good system engineering processes: collaboration, traceability, validation and verification (V&V), risk analysis and mitigation, and careful integration within the tool chain. Let's consider each of these elements.

Recording formal and informal interactions and decision points in the collaborative development of today's automotive electronic systems is an important documentation aspect of ISO 26262. But such collaboration between team members and global partners in the supply chain must be done with as little intrusion into the normal workflow as possible.

Ensuring functional safety requires clear traceability of requirements, functions, implementations and tests throughout the life cycle process. For tool vendors, traceability offers a way to ensure that new versions of a tool won't break existing customer software or hardware.

Traceability naturally provides the path for verification of requirements. But validation of the system is also important. Taken together, validation and verification (V&V) makes sure engineers are building the right thing and building the thing right, respectively. A focus on functional safety means that V&V throughout the decomposition and integration flows of the V life cycle model might require a level of rigor and consistency that is absent a non-compliant development process.

Functional safety is achieved by determining, analyzing and mitigating risk hazards. ISO 26262 details how to assign an acceptable risk level to a system or component and document the overall mitigation process. The standard attempts to address mitigation of potential hazards through good engineering practices that reduce risk down to a tolerable and statistically acceptable level.

Risk mitigation covers both process and usage of specific tools. This can require that tool vendors certify or prove the safety functionality of their tools suites. In general, vendors need to convince customers that their tools wouldn't introduce problems when used in support of the standard. That is why ISO 26262 describes a qualification process to ensure that software tools and even the tool chains are suitable for safety-intense systems.

Conclusions

Automotive electronic hardware and software designers need to understand how to implement the ISO 26262 process within their overall development effort. Component part and tool vendors must understand the certification effort required by ISO 26262 to support designers in their activities.

Picture this: Traceable communication. Documented decisions and actions. All product and systems info organized and contextualized from concept to launch. With Jama, it's your reality. Try Jama and see how we can help you solve your team's systems engineering challenges.

About Jama Software

Jama Software is the product development platform for companies building complex, smart and connected products. The Jama solution enables enterprises to accelerate development time, mitigate risk, slash complexity and verify regulatory compliance. More than 600 product-centric organizations, including NASA, Thales and Caterpillar, use Jama Software to modernize their process for bringing complex products to market. For more information, visit www.jamasoftware.com.