CHALLENGES OF AN INTEGRATED FUNCTIONAL SAFETY IMPLEMENTATION

Authors – Dr. Harwinder Jutla, Christopher Giordano CpE., Scott A. Vander Weide MSCpE.

VDI Conference – Automotive HMI & Connectivity / September 26 - 27, 2018 Frankfurt, Germany

ABSTRACT

With the acceleration of autonomy, requirements for functional safety are increasing. Further, the need for cross-domain functionality is also growing (e.g. between IVI, Clusters and HUDs). External drivers of regulations, safer vehicles, and insurance companies are driving such needs.

With challenges presented by software, hardware, driver interaction through non-distracting user interface, how does the industry address the functional safety needs? How does one integrate separate stacks into a functionally safe system? This paper examines the options & challenges for implementing functionally safe systems, and how to integrate safety critical and non-safety critical software across multiple displays on a single System on a Chip (SoC).

INTRODUCTION

Many items in our everyday lives were developed with an eye toward keeping us safe: the spill-proof lid on your coffee cup; the microwave in your kitchen with the lockout mechanism on the door; the blood pressure machine at your local pharmacy; or the instrumentation and controls for the airplane you may have flown on recently. Much of these safety considerations are regulated by local governments. Much of this safety is developed through well-known processes. But, where do the control and information systems of your car stand?

Some road statistics, courtesy of ASIRT [(1)] paint a worrying picture:

1. Nearly 1.3 million people die in road crashes each year, on average 3,287 deaths a day
2. An additional 20-50 million are injured or disabled
3. More than half of all road traffic deaths occur among young adults ages 15-44
4. Road traffic crashes rank as the 9th leading cause of death and account for 2.2% of all deaths globally
5. Road crashes are the leading cause of death among young people ages 15-29, and the second leading cause of death worldwide among young people ages 5-14
6. Each year nearly 400,000 people under 25 die on the world's roads, on average over 1,000 a day
7. Over 90% of all road fatalities occur in low and middle-income countries, which have less than half of the world's vehicles
8. Road crashes cost USD $518 billion globally, costing individual countries from 1-2% of their annual GDP. In US alone, road crashes cost them $230.6 billion per year

9. Road crashes cost low and middle-income countries USD $65 billion annually, exceeding the total amount received in developmental assistance
10. Unless action is taken, road traffic injuries are predicted to become the fifth leading cause of death by 2030

Of course, a large portion of these fatalities and accidents are a result of driver distraction and human error with a wide range of complex reasons. This of course could be as simple as distraction caused by inadequate user interface design in the vehicle itself.

According to the Motor vehicle safety data, by the BTS (Bureau of Transportation Statistics), more than 6 million crashes involving motor vehicles are reported every year on an average. [(2)]

As per the U.S. Transportation Department data, United States automakers had to make a record safety recall of 53.2 million vehicles in 2016. This increase in auto safety recalls was caused by the rise in road traffic deaths/road traffic fatalities in US.

An auto recall, according to National Highway Traffic Safety Administration (NHTSA, US) [(8)], is said to be issued when a manufacturer or NHTSA determines that a vehicle, equipment, car seat, or tire can create an unreasonable safety risk or fails to meet minimum safety standards".

These statistics clearly lead us to one common conclusion – how even after technical advancements along the breadths and depths of the industry, an automobile is still a leading contributor to road accidents.

Hence, as we move forward into the complex autonomous vehicle era, safety becomes an increasingly important requirement. Improved software robustness and quality could help lead to a reduction in human error by minimizing distraction.

ISO has provided the industry with the 26262 standard in an attempt to increase road safety. So, how is it applied to HMI? Implementation of functionally safe Automotive HMI systems is still in its early stage, lacking clear regulation and guidance.

This paper discusses several main considerations in the development of functionally safe automotive HMI systems. This paper focuses on:

- Debunking four myths encountered with developing functionally safe HMIs
- Enumerating the multiple challenges faced in the implementation of functionally safe HMIs
- Explaining how other industries, including medical and avionics, handle the development of functionally safe HMIs
- Offering the automotive industry guidance and recommendations implementing functionally safe HMIs based on proven success in other industries requiring functionally safe HMIs

Myths of Developing Functionally Safe Solutions

When working with clients on HMI projects, there are several misconceptions, or myths, on what functional safety is and how it can be implemented arise. Quite often the myths come up in response to attempts to streamline the development process or save money on development tools or components for the HMI.

MYTH 1: 'YOU CAN CERTIFY PARTS OF SYSTEMS'

Can an HMI be broken down into its components and certified separately? Can certification be performed on one component, or a few? If we only develop certain components in a functionally safe way is the whole HMI safe? According to the ISO 26262 [(4)], the answer is "No". When we talk about component, we expect an automotive HMI system to have at least the following components: SoC and related interconnects – the hardware, a computer operating system, a set of drivers to interface between the operating system and the hardware, and the user interface software.

The ISO 26262 standard is very clear that safety applies to a full system. From the introduction of part 1 of the standard, it states:

> Safety is one of the key issues of future automobile development. New functionalities not only in areas such as driver assistance, propulsion, in vehicle dynamics control and active and passive safety ***systems*** increasingly touch the domain of system safety engineering. Development and integration of these functionalities will strengthen the need for ***safe system development processes*** and the need to provide evidence that all reasonable ***system*** safety objectives are satisfied.

Note that this paragraph talks about "system" (appropriate text highlighted) and not component.

It is especially well illustrated in the "V" model (Figure 1) from the standards committee that defines the development process, where product development starts at the system level and is then broken down into hardware and software.

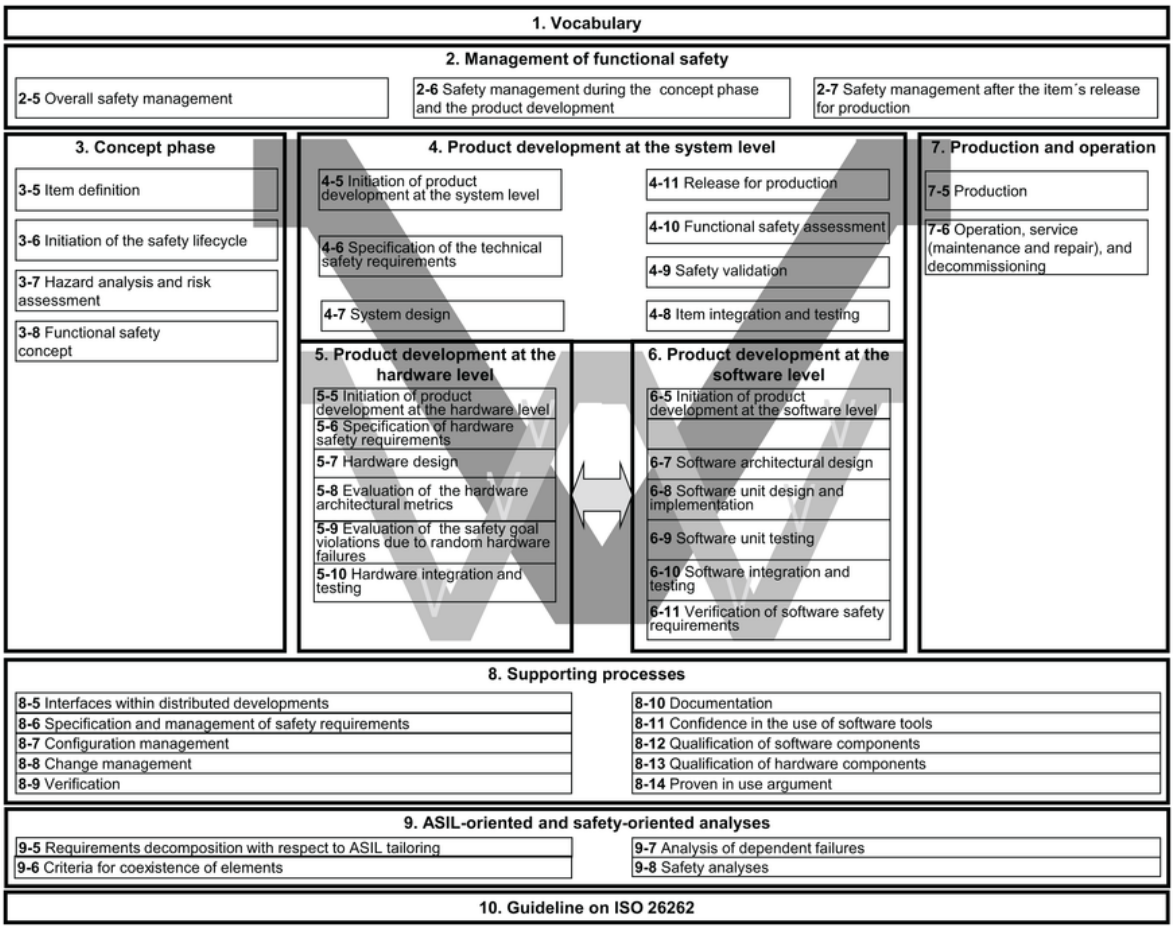

FIGURE 1 -ISO 26262 'V' MODEL

Source of the diagram: https://www.iso.org/obp/ui/#iso:std:iso:26262:-9:ed-1:v1:en [4]

The ISO 26262 standard defines a system as a "set of elements that relates at least a sensor, a controller, and an actuator with one another," allowing the sensor and actuator to be external to the system and allowing the system to contain sub-systems. The standard goes on to describe a system as containing one or more hardware and software components.

There are supplier companies in the industry who bring significant safety critical development expertise. Some have arrived at the automotive industry from Aerospace & Defense and medical where they have been operating in a functional safety environment where performance and safety are paramount. Such players, with 15-20 years of experience in functional safety are set to support the automotive industry in this relatively new and challenging venture.

Myth 2: 'You can achieve Functional Safety with Linux'

The Linux operating system (or GNU/Linux) is widely used for desktop, mainframe, mobile, and embedded computing, among many other application spaces. It is attractive for use widely because it is highly adaptable, the source code is open to anyone to use and modify, and it is free of cost.

Linux was never developed for use in functionally safe systems. There is no unified process for requirements analysis, design, implementation, and verification, as required by the ISO 26262 standard and other functional safety standards. There is no guarantee the developers contributing to the Linux project have any understanding of functional safety. There have been over 13,000 developers contributing to just the Linux kernel over the past 13 years. [5]

There are no doubts that there are initiatives in the automotive industry to try and take Linux towards a functional safety environment. However, these are still far and in -between.

Even if the Linux project were developed in a manner that it could be certified, the cost would be significant. The Linux kernel alone has over 11 million lines of code [5]. For comparison, "The process required to create the software and compile the necessary certification evidence can take months or years and cost on the order of $100 per line of code…" [6] This would put the cost of a certified Linux kernel at over $1 billion dollars. This does not include the supporting GNU projects necessary to make a complete operating system or the code developed for automotive HMI.

Myth 3: 'Crashing gracefully is functionally safe'

It is a widely held view in the industry and implemented as such in most vehicles, that rapidly rebooting an automotive HMI or switching to a failsafe display mode when a software error occurs is an acceptable interpretation of the safety standard. Rapidly rebooting is usually defined as being able to display the first set of data to the HMI screen within one second. Recall that a vehicle travelling at 60 miles per hour (96.6 kilometers per hour), travels 88 feet per second (26.8 meters per second), thus in a 1 second boot time, the vehicle has travelled 88 feet. A failsafe display mode would usually be implemented on a secondary, bare metal system executing on an MCU, but with drastically different appearance and a significantly reduced amount of information. Any safety indicator that might be missed in that time probably would not be detrimental to safety, but the distraction that may be caused to the driver could have disastrous results.

On January 1, 2007, Adam Air flight 574 crashed into the Indian Ocean due to errors in the navigation systems. "The pilots of the Boeing 737 became so engrossed in troubleshooting the problem that they

inadvertently disconnected the autopilot… they also forgot to check the airplane's altitude and position on other cockpit instruments." [7]

The crash of Adam Air flight 574 is analogous to the automotive safety problem of distracted driving. "Distracted driving is any activity that diverts attention from driving, including talking or texting on your phone, eating and drinking, talking to people in your vehicle, fiddling with the stereo, entertainment or navigation system—anything that takes your attention away from the task of safe driving." [8] In 2015, over 391,000 people were injured in the United States due to distracted driving.

Myth 4: Testing Directly Improves Quality

There is a misconception that testing is done to improve safety and quality. This is not the case. We carry out testing to assess the quality of our implementation. Otherwise, we could implement poor coding and architecture and get the testing to make it perfect; lots of testing would thus improve some poor implementation. We know from experience in User Interface development for Aerospace, Avionics, Medical and Automotive that this is not the case.

We should carry out testing to assess the quality, with appropriate feedback loops in the development process to improve requirements, traceability, coverage and robustness. [9] All this has to be proportionate to the ASIL standard being implemented.

Current Challenges

We will now look at the current challenges in the automotive industry in planning and implementing functionally safe automotive HMI systems. This includes determining the need for functional safety and defining what parts of the system are critical, identifying what is needed to start the development of functionally safe systems, and addressing industry trends for HMI system development and how that fits with functional safety.

What is Critical?

The ISO 26262 Part 3, Section 1 of the standard states "ISO 26262 addresses possible hazards caused by malfunctioning behavior of E/E safety-related systems, including interaction of these systems." The ISO 26262 standard has three criteria for determining the criticality of a possible hazard: severity; probability of exposure; and controllability as stated in the standard in Part 3, Section 7.2. Severity has to do with the type and amount of injuries to be expected if the hazard occurs. Probability of exposure has to do with how likely the hazard is to occur. And, controllability has to do with how likely it will be for control of the vehicle to be maintained or regained if the hazard occurs. These criteria are then used to assign an ASIL level to the hazard.

The standards writers do not give explicit guidance on particular hazards or classes of hazards. It is up to the individual implementers to determine all the possible hazards a system may encounter, assign ASIL levels to each, and implement the system in such a way as to mitigate the hazard. The following illustration (Figure 2) from SafeAssufe by NXP shows the various forms of hazards and their associated ASIL levels that could be addressed. [10]

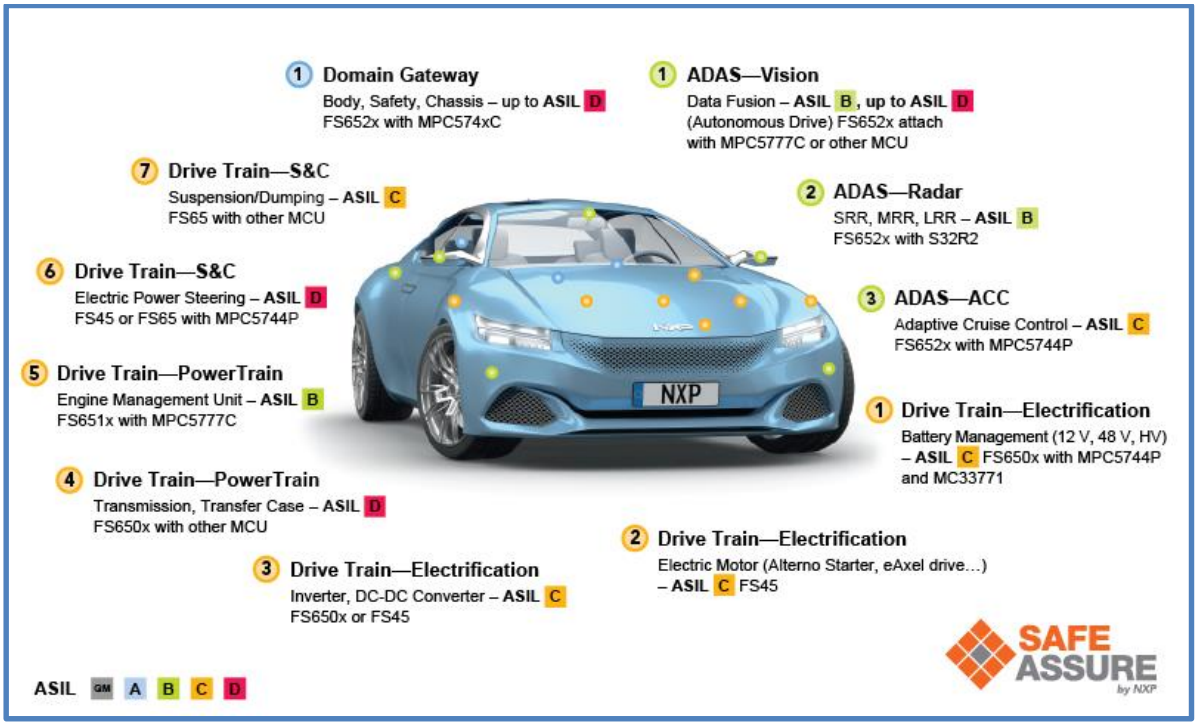

FIGURE 2 - AUTOMOBILE SAFETY ISSUE TYPES

LACK OF...

The role of software in automotive systems is increasing at a dramatic rate: 90% of all automotive innovations are driven by electronics and software [(11)]; and, up to 40% of a vehicle's development costs are determined by electronics and software [(12)] and that 50 - 70% of the development costs for an ECU (Electronic Control Unit) are related to software [(13)]. The automotive industry is in need of competent software developers for developing functionally safe systems. With the consumer technological savviness increasing daily due to exposure to cell phones and connected devices, the need to focus on the automotive HMI is even more urgent. A recent search at LinkedIn shows there are over 18,000 open positions for the search criterion "automotive software".

As mentioned in the section "What is Critical?" the ISO 26262 standard does not give guidance on what items of a system may induce hazards that need to be addressed in a functionally safe way, or, for our particular discussion, what elements of the automotive HMI require an ASIL rating. There is no government oversight yet from any nation on this question. The SAE Ground vehicle Technical Committees: Functional Safety Committee focuses on the following areas: Brakes, Trailer Brake and Park Brake; Steering and Suspension; and Propulsion and Driveline. Notably, User Interface is missing! [(14)]

And, though there is a standard, there is no certifying body for the standard. Aviation and medical device functional safety certification is managed by government bodies. In the United States for example, the FAA and the FDA respectively. In Europe that would be the JAA and the MDD respectively Quality processes certification, like ISO's 9001 standard, is managed by a hierarchy of policing organizations. At the moment, ISO 26262 is a self-certifying standard for automotive OEMs.

Industry Expectations

In the interests of saving money and reducing complexity, many automotive HMI producers are considering using one powerful SoC to run multiple aspects of the HMI, including Cluster, HUD, and IVI. The expectation is the SoC, with multiple processing cores, will run multiple applications outputting to multiple displays. But, this reduces the HMI to a single system in the automobile. And, as we discussed earlier regarding certification ISO 26262 standard, full systems must be certified. So, it must be determined if there is a way to compartmentalize software units that do not require function safety. Hardware, operating systems, and drivers have been designed to accommodate this compartmentalization.

At the other end of the implementation spectrum is the use of multiple low-compute-power SoCs. This reduces some of concerns for compartmentalization of software when developing for functional safety. This architecture may however raise other concerns. As stated earlier, complexity of implementation for the complete vehicle increases as wiring harnesses, installation locations, and data passing mechanisms for multiple systems must be taken into account. Considerations for sharing data on multiple displays must be considered. Also, there may at some point in the design of the vehicle, be a need to enforce functional safety on a formerly non-critical system, such as needing a new safety warning on the HUD.

Examples and Discussion

We now discuss examples of certification efforts in several industries, enumerate strategies for applying functionally safe development practices and considerations to automotive HMI systems, and share insights gained and best practices gained from our experiences implementing such systems.

Examples in Aerospace and Medical

Whether it is a Primary Flight Display for an airliner on approach to land or the control panel heart-lung machine, the HMI system cannot fail.

The commercial aviation's DO-178C standard states:

> Every point of entry and exit in the program has been invoked at least once, every condition in a decision has taken all possible outcomes at least once, every decision in the program has taken all possible outcomes at least once, and each condition in a decision has been shown to independently affect that decision's outcome. A condition is shown to independently affect a decision's outcome by: (1) varying just that condition while holding fixed all other possible conditions, or (2) varying just that condition while holding fixed all other possible conditions that could affect the outcome. (DO-178C:"Software Considerations in Airborne Systems and Equipment Certification", RCTA, December 2011)

Every flight display in an aircraft cockpit flying over the United States must meet this stringent standard, and many others. Here is one example of the development of safety critical instrumentation produced by Howell Instruments for the U.S. Army's UH-60 Blackhawk helicopters:

> Empowered with GL Studio Safety Critical Embedded Code (SCEC++) to create the upgraded visual display systems of the cockpit, Howell was able to engineer the AEI patented system for the UH-60 Black Hawk. GL Studio's high fidelity output, aesthetically pleasing graphics, and real-time playback performance guided Howell to employ a system for military personnel with the best resources and best possible outcomes. GL Studio was the main component to seamlessly combine the aircraft's indispensable piloting features. The Enhanced Digital Source Collector (EDSC), Multi-Function Central Display Unit (MFCDU), and Multi-Function Pilot Display Unit (MFPDU) work exceptionally together as one cohesive unit. Because of

the GL Studio SCEC++ code generation application, the AEI boasts a compact memory footprint for ultimate portability, increased pilot efficiency and unbeatable reaction performance. [15]

Even with this intense focus on developing the software behind an HMI, usability and aesthetics need not be sacrificed. Datascope Corp's Cardiosave Intra-Aortic Balloon Pump (IABP) has received FDA 510(K) clearance, IEC 62304 certification and the CE Mark (these represent the medical device equivalent of DO-178C or ISO 26262 certification) and is in use around the world helping patients with cardiovascular issues. [16]

It is also raved about by doctors for its easy to use interface. One reviewing physician compared it to an older IABP from Datascope Corp. "The old (interface) was a monochrome orange. The Cardiosave interface is very lively. Everything is color-coded, from the ECG tracing to the point of inflation on the dicrotic notch, pressure waveform and balloon pressure waveform. It's the iPhone of balloon pumps." [17]

Examples in Automotive

Unfortunately, due to the lack of governmental regulation and the inherent protective nature of the automotive industry for development processes, little data is available about the use of the ISO 26262 standard, especially regarding certification of automotive HMI systems. What can be found is the absence of functional safety when reviewing recalls.

Recently, the Wall Street Journal reported two major luxury automotive brands are having problems with instrument clusters. In one, the illumination of the cluster was too dim to be able to read pertinent information. In the other, the instrument cluster would intermittently go blank. In both cases, the software developed for the instrument clusters by suppliers was found at fault and software updates were released. [18]

Certification Efforts

Any effort to certify a functionally safe system needs to start with the standard. Without a thorough understanding of the standard, any development effort for a functionally safe system will quickly become difficult and expensive.

DiSTI performed a certification of the GL Studio Safety Critical runtime library (in a representative system) with TÜV Nord in 2015. [19] Here are some recommendations from our certification experience.

1. The certification authority will want to see all steps of the development process documented in the manner prescribed in the ISO 26262 standard
    a. Take clear, detailed notes from every meeting related to the development effort
    b. Save all emails related to the effort
    c. Extra steps to capture discussions and design and implementation decisions will aid in completed all parts of the required documents
2. As part of all development efforts, use a well-defined software development process
    a. The ISO 26262 standard requires all steps in the process be documented
    b. It is better if the development team is in the habit of following a process before a functional safety initiative reaches your company

    c. The alternative is changing the way a team works while also dealing with the added scrutiny of a certifying authority

3. Implement peer reviews and make sure all reviewers are aware of the ISO 26262 standard
4. Write test cases early
    a. Only ASIL-D requires full Modified Decision / Condition Coverage in testing, but a defined method for determining how tests are executed as well as specific pass / fail criteria must be defined

In the course of discussing functional safety implementations for automotive HMIs, we have only heard discussed the need for ASIL-A and –B levels. This does relieve some burden in the certification process. ASIL-D is the only level that requires outside review and validation of a system. A competent QA and Test organization inside your company is all that is necessary for all other ASIL levels.

Implementing Functional Safety

In the context of automotive functional safety in product development, safety is defined as the absence of unreasonable risk. While some risk will always be present and cannot be eliminated, the industry needs to apply a systematic approach to functional safety throughout product engineering, development & manufacturing to minimize the risk of accidents or other incidents.

The problem is complex as there is an entire eco-system to consider which is necessary to support the development of Functionally Safe HMI. Such considerations are driver support (OpenGL SC, Software GPU, WiFi drivers, Bluetooth Drivers to name but a few), hardware redundancy, Hypervisors, RTOS, Graphics Sharing schemes – again to name but a few.

From the hardware prospective, the white paper “Hardware Convergence & Functional Safety” [(20)] provides a prospective of the various architectures when implementing a HUD, Cluster and Infotainment systems. The industry needs to balance costs and complexity when implementing a specific solution so perhaps there is not a single solution, which fits all.

When we are looking at the entire eco-system though, there are, some additional generic elements we need to consider are:

1. Robust development platforms and processes are a must have
2. Appropriate tools selection.

    This is not something that should be deferred into a program. We often are focused on selecting what language we want to program in or what processor we want to use, we ignore the need for the tools pertinent to our ISO 26262 needs. Delaying such activity means delays in the development process as things may have to be repeated once the tools are selected later. We must remember that tools directly impact quality, productivity, schedule, budget and certifiability

3. Robust architecture & design processes to ensure design errors are rooted out early

4. Strong test methodologies – involve testing in the design process on the left-hand side of the V-Model. Not something to be left to implement later in the design cycle

   Further, investments in auto-mated testing can yield significant returns. Automated testing allows larger portion to be tested quickly for every build. In fact, coverage for Regression testing, stress testing, randomized testing, use case/functional testing can all be significantly increased thus reducing risks in any software release

5. Using pre-certified components.

   This allows a solid stepping stone to achieving the level of functional safety required

6. Keep business logic as simple as possible.

   This avoids having to test for complex corner cases or logic paths thus avoiding untested routes escaping in the release

7. Traceability, both top-to-bottom and bottom-to-top, is required for ISO 26262 certification

   This relies on robust audit reviews, program management processes, traceability in both directions. Tying up requirements to implementation is necessary. However, even more important (because it is sometime missed) is that on-the-fly development changes are reflected correctly back up to the requirements

8. Proper & adequate Quality Assurance

   Quality assurance provides a distinct purpose for automotive certification - to independently provide proof that ISO 26262 guidelines were followed

CONCLUSION

As autonomy increases, there are both business and technology drivers to consider for functional safety implementation. There are levels of autonomy (1-5) that impact the marketability of vehicles and their ability to mitigate risk with variable driver inputs to the systems. Thus, in addition to the obvious safety gains, there is a significant economic reason for implementing safety critical to ensure that the vehicle is less at fault today. Regulations will be tightened to ensure safety on the road and insurance companies will start considering the safety level of vehicles in this constant battle to assign responsibility for accidents – driver or the car - in the autonomous world. Whilst there may be a compulsion to ignore or moderate this need because there are no mandatory drivers, the time for action is now in preparation for the future.

There is a distinct possibility that in the future that the insurance companies will insure the car and not the driver. When the level of autonomy increases to 3 and beyond, the premiums users pay for insuring the car will be dictated by the safety reputation that manufacturer/vehicle has in the industry. Therefore, the users will be obliged to take the safety record into consideration when deciding on which vehicle to purchase. All this will directly impact revenue.

Therefore, foresight of such possibilities should be enough for the industry to consider functional safety not as a burden to mitigate in the cheapest manner but an integrated necessity to confront face on.

Full ASIL D Functional Safety capability is possible for digital User Interfaces and has been part of Aviation certification process for decades.

Consolidated hardware architecture across multiple displays with mixed criticality can provide cost savings and architecture simplification.

Further, in the world of autonomy, the functional safety requirements in clusters, HUDs and Infotainment systems have begun to blur as functional safety features are required across all these elements. Industry needs to consider all systems to be implemented in a functionally safe manner. We cannot continue to pick and choose from the standard to meet the bare minimum requirements. The industry needs to commit wholeheartedly to a full safety capable system stack for both safety and economic reasons.

CITATIONS

[1] - Association for Safe International Road Travel (ASIRT), *Annual Global Road Crash Statistics 2017* https://www.asirt.org/road-safety-facts/

[2] - Bureau of Transportation Statistics (BTS), Transportation Statistics Annual Report, 2017, P6-24 (Table 6-6) https://www.bts.dot.gov/sites/bts.dot.gov/files/docs/browse-statistical-products-and-data/transportation-statistics-annual-reports/215041/tsar-2017-rev-2-5-18-full-layout.pdf

[3] - US Department of Transportation (US DoT) https://www.transportation.gov/www.transportation.gov/connections/fall-back-then-check-vehicle-recalls

[4] - ISO 26262 Standard, Figure 1, https://www.iso.org/obp/ui/#iso:std:iso:26262:-9:ed-1:v1:en

[5] – C. Smart, Sep 23, 2009, The Linux Foundation http://www.linux-mag.com/id/7536/ - https://www.linuxfoundation.org/projects/linux/

[6] –Aviation Today http://interactive.aviationtoday.com/avionicsmagazine/february-2017-march-2017/do-178c-software-for-nextgen-avionics-uavs-and-more/_fragment.html

[7] - C. Barker, Dec 4, 2012, Computer Disasters, 10 Air Disasters Caused By Computer Errors: https://www.bestcomputersciencedegrees.com/10-air-disasters-caused-by-computer-errors/

[8] - National Highway Traffic and Safety Administration (NHTSA) https://www.nhtsa.gov/risky-driving/distracted-driving - https://www.nhtsa.gov/recalls

[9] – Sun, Brain, Kroening, Hawthorn, Wilson, Shcanda, Jimenez, Daniel, Bryan, Broster - Jul. 5, 2017, Functional Requirements-Based Automated Testing for Avionics https://arxiv.org/pdf/1707.01466.pdf

[10] – D. Lopez, Oct. 3, 2016, NXP Blog "How Functional Safety is Handled" https://blog.nxp.com/automotive/three-things-to-know-about-functional-safety

[11] - EETE Automotive. (2014, April 29). *Innovation in the car: 90% comes from electronics and software*. Retrieved 05 02, 2016, from Automotive IT News: http://www.automotiveitnews.org/articles/73473/innovation-in-the-car-90-comes-from-electronics-an/

[12] – R. Charette, Feb. 1, 2009, IEEE Spectrum: This Car Runs on Code https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code

[13] - Simon, F. (2010). Challenges in the Design of Automotive Software. *Design, Automation and Test in Europe* (p. 3). Brussles: European Design and Automation Association (EDAA). Retrieved May 1, 2016, from https://www.date-conference.com/proceedings-archive/PAPERS/2010/DATE10/PDFFILES/03.8_1.PDF

[14] - SAE Ground vehicle Technical Committees: Functional Safety Committee Fact Sheet: https://www.sae.org/works/committeeResources.do?resourceID=386637

[15] - Howell Instruments GL Studio Case Study: https://www.disti.com/user-interface/case-studies/howell-instruments/

[16] - Maquet Cardiovascular Press Release: https://www.prnewswire.com/news-releases/maquet-cardiovascular-announces-fda-510k-clearance-and-ce-mark-for-the-cardiosave-intra-aortic-balloon-pump-133635028.html

[17] - Cath Lab Digest, Volume 20 – Issue 1, January 2012: Cardiosave IAB{: An Early Adopter's Report https://www.cathlabdigest.com/articles/Cardiosave-IABP-Early-Adopter%E2%80%99s-Report

[18] - Wall Street Journal, March 27, 2018: "In Car Makers' Digital Dash, Little Room for Error" https://www.wsj.com/articles/the-latest-technology-is-awesome-just-not-always-in-your-car-1522152001

[19] - DiSTI Press Release, "DiSTI's GL Studio Received ISO 27272 ASIL D Certification", April 8, 2015: https://www.disti.com/distis-gl-studio-receives-iso-26262-asil-d-certification/

[20] – J. Carroll MOBICA, C. Giordano DiSTI, June 27, 2016: Whitepaper "Hardware Convergence & Functional Safety" (https://www.disti.com/user-interface/videos-literature/

AUTHORS

Dr. Harwinder Jutla – Solution Architect, DiSTI Corporation, UK
hjutla@disti.com / https://www.linkedin.com/in/harwinder-jutla-166a4542/

Christopher Giordano, CpE – Vice President UX/UI Technology, DiSTI Corporation, Orlando, FL, USA
cgiordano@disti.com / https://www.linkedin.com/in/cpgiordano/

Scott VanderWeide, MSCpE – Lead Embedded Engineer, DiSTI Corporation, Orlando, FL, USA
svanderweide@disti.com / https://www.linkedin.com/in/scott-vander-weide-0934445/

CONTACT

The DiSTI Corporation
11301 Corporate Blvd, Suite 100
Orlando, FL 32817
+1 (407)-206-3390
www.disti.com

NOTICE