U.S. Department
of Transportation
**National Highway
Traffic Safety
Administration**

NHTSA

DOT HS 812 285

June 2016

Assessment of Safety Standards for Automotive Electronic Control Systems

Disclaimer

This publication is distributed by the U.S. Department of Transportation, National Highway Traffic Safety Administration, in the interest of information exchange. The opinions, findings, and conclusions expressed in this publication are those of the author and not necessarily those of the Department of Transportation or the National Highway Traffic Safety Administration. The United States Government assumes no liability for its content or use thereof. If trade or manufacturers' names or products are mentioned, it is because they are considered essential to the object of the publication and should not be construed as an endorsement. The United States Government does not endorse products or manufacturers.

Suggested APA Format Citation:

Van Eikema Hommes, Q. D. (2016, June). *Assessment of safety standards for automotive electronic control systems*. (Report No. DOT HS 812 285). Washington, DC: National Highway Traffic Safety Administration.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 2016		3. REPORT TYPE AND DATES COVERED July 2012 – April 2015
4. TITLE AND SUBTITLE Assessment of Safety Standards for Automotive Electronic Control Systems			5. FUNDING NUMBERS Intra-Agency Agreement HS8AA1 DTNH22-12-V-00086
6. AUTHOR Qi D. Van Eikema Hommes			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) John A. Volpe National Transportation Systems Center U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 55 Broadway Cambridge, MA 02142			8. PERFORMING ORGANIZATION REPORT NUMBER DOT-VNTSC-NHTSA-13-03
9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS National Highway Traffic Safety Administration 1200 New Jersey Avenue SE. Washington, DC 20590			10. SPONSORING/MONITORING AGENCY REPORT NUMBER DOT HS 812 285
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT This document is available to the public through the National Technical Information Service www.ntis.gov			12b. DISTRIBUTION CODE

13. ABSTRACT

This report summarizes the results of a study that assessed and compared six industry and government safety standards relevant to the safety and reliability of automotive electronic control systems. These standards include ISO 26262 (Road Vehicles - Functional Safety), MIL-STD-882E (Department of Defense Standard Practice, System Safety), DO-178C (Software Considerations in Airborne Systems and Equipment Certification), Federal Motor Vehicle Safety Standards, AUTOSAR (Automotive Open System Architecture), and MISRA C (Guidelines for the Use of the C Language in Critical Systems). The assessment was carried out along the following 11 dimensions: (1) type of standard, (2) definition of safety and hazard, (3) identification of safety requirements, (4) hazard and safety analysis methods, (5) management of safety requirements, (6) risk assessment approach, (7) design for safety approach, (8) software safety, (9) system lifecycle consideration, (10) human factors consideration, and (11) approach for review, audit, and certification. The observed strengths and limitations of the standards studied in this report could support the future development of a robust functional safety approach for automotive electronic control systems.

14. SUBJECT TERMS Automotive Electronics, Automotive Electronic Control Systems, Functional Safety, Electronics Reliability, Automotive Electronics Safety Standards			15. NUMBER OF PAGES 49 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT

Foreword

NHTSA's Automotive Electronics Reliability Research Program

The mission of the National Highway Traffic Safety Administration is to save lives, prevent injuries, and reduce economic costs due to road traffic crashes. As part of this mission, NHTSA researches methods to ensure the safety and reliability of emerging safety-critical electronic control systems in motor vehicles. The electronics reliability research comprises methods and standards within and outside the automotive industry for assessing, identifying and mitigating potential and new hazards that may arise from the increasing use of electronics and electronic control systems in the design of modern automobiles.

Similar to the cybersecurity research program, NHTSA has established five research goals for the electronics reliability research program to ensure the safe operation of motor vehicles equipped with advanced electronic control systems. This program covers various safety-critical applications deployed on current generation vehicles, as well as those envisioned on future vehicles that may feature more advanced forms of automation and connectivity. These goals are:

1. Expand and share the knowledge base to ensure research plans for automotive electronics reliability are appropriate and promote tools for applied research in this area;
2. Strengthen and facilitate the implementation of safety-effective voluntary industry-based standards for automotive electronics reliability;
3. Foster the development of new system solutions for ensuring and improving automotive electronics reliability;
4. Research the feasibility of developing potential minimum vehicle safety requirements pertaining to the safe operation of automotive electronic control systems; and
5. Gather foundational research data and facts to inform future NHTSA policy and regulatory decisions.

This report contains the results of a research study that assessed and compared six industry and government safety standards relevant to the electronics reliability of automotive electronic control systems.

This publication is part of a series of reports that describe NHTSA's initial work in the automotive electronics reliability program. This research specifically supports the first and second goals of NHTSA's program by reviewing current safety standards and documenting their strengths, limitations, and opportunities for enhancement. The observed strengths and limitations of the standards studied in this report could support the future development of a robust functional safety approach for automotive electronic control systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ACRONYMS

ACC	adaptive cruise control
ASIL	Automotive Safety Integrity Level
AUTOSAR	Automotive Open System Architecture
C	controllability
DoD	Department of Defense
E	exposure
E/E	electrical/electronic
ETA	event tree analysis
FMEA	failure mode and effects analysis
FMVSS	Federal Motor Vehicle Safety Standard
FTA	fault tree analysis
HAZOP	hazard and operability analysis
HTS	Hazard Tracking System
IEC	International Electrotechnical Commission
MISRA	Motor Industry Software Reliability Association
NAS	National Academy of Sciences
QRA	quantitative reliability analysis
S	severity
STPA	system theoretic process analysis

EXECUTIVE SUMMARY

This report summarizes the technical support that the Volpe National Transportation Systems Center (Volpe) provided to the National Highway Traffic Safety Administration in the following focus areas concerning the safety and reliability of automotive safety-critical electronic control systems:

- Expand and share the knowledge base to ensure research plans for automotive electronics reliability are appropriate and promote tools for applied research in this area; and
- Strengthen and facilitate the implementation of safety-effective voluntary industry-based standards for automotive electronics reliability.

Specifically, this technical support effort compares and assesses the following relevant safety standards.

- ISO 26262: Road Vehicles - Functional Safety
- MIL-STD-882E: Department of Defense Standard Practice - System Safety
- DO-178C: Software Considerations in Airborne Systems and Equipment Certification
- FMVSS: Federal Motor Vehicle Safety Standard
- AUTOSAR: Automotive Open System Architecture
- MISRA C: Guidelines for the Use of the C Language in Critical Systems

ISO 26262 is the most relevant automotive electronic control system safety standard within the context of this study. This study also chose to review the five additional standards listed above because the ISO 26262 committee and industry practitioners frequently cited them in various discussions and research papers. MIL-STD-882E, DO-178C, and MISRA C are mature safety standards that have gone through a number of iterations. FMVSSs are vehicle safety regulations that NHTSA issues. AUTOSAR was developed around the same time as ISO 26262 and references ISO 26262 for safety considerations.

This study compared these standards along the following 11 dimensions.

1. Type of standard
2. Definition of safety and hazard
3. Identification of safety requirements
4. Hazard and safety analysis methods
5. Management of safety requirements
6. Risk assessment approach
7. Design for safety approach
8. Software safety
9. System lifecycle consideration

10. Human factors consideration
11. Approach to review, audit, and certification

This report provides a tabulated summary of the assessment results. Furthermore, Volpe made the following observations from the comparative assessment of the six relevant standards:

1. Process safety standards[1] or guidelines that follow a systems engineering[2] approach are different from FMVSSs and complement existing standards for safety assurance.
2. Existing process standards could be enhanced by providing a precise definition of "unreasonable risk" within the context of automotive safety.
3. *Hazard* definitions vary across different standards.
4. S*everity* alone can be used as the risk measure for software, similar to the approach outlined in DO-178C. Further, in cases when statistically valid failure probability or the probability of the occurrence of a mishap is not available, *Severity* could be used as the only measure.
5. *Exposure* and *Controllability* assessment used by the industry, as defined in the ISO 26262 standard, could be enhanced with the collection of additional data through design of specific experiments.
6. Existing process standards for software design could be enhanced by considering the overall safety of the control systems and software safety certification, in addition to the focus on specific aspects of the design solution (i.e., good architecture and coding standard).
7. Design-for-safety approach as specified in MIL-STD-882E provides a framework that could be leveraged for separate management of hazard tracking/safety requirements from regular system requirements, simpler risk assessment, and more emphasis on human factors.
8. The topic of health hazard analysis for drivers and service technicians could to further assessed for the appropriateness of including this topic in a process standard.
9. Existing process standards do not explicitly address environmental impacts on a vehicle throughout its lifecycle, including testing, manufacturing, operation, maintenance, etc.
10. Human factors studies could be better integrated into a comprehensive functional safety approach.

[1] Process safety standards define "what" needs to be done and which "roles" are involved to ensure the safety of a system.
[2] Systems engineering is an interdisciplinary field of engineering that focuses on how to design and manage complex engineering systems over their life cycles.

1 INTRODUCTION

1.1 Background

The proportion of electronic components used in the construction of motor vehicles has increased rapidly in recent years. Electronics play a crucial role in developing optimized technological solutions to improve the vehicle's drivability, enhance its safety features, and lower the environmental burden. Today's motor vehicles incorporate an increasingly complex array of electronics including sensors, actuators, microprocessors, instrumentation panels, controllers, and displays. Advances in electronics have contributed to the development and deployment of advanced safety features such as electronic stability control, tire pressure monitoring, lane departure warning, adaptive cruise control, forward crash warning, and automatic braking. Electronics have also improved the environmental performance of motor vehicles by delivering optimized control to a variety of vehicle systems, including more efficient operation of engines and other powertrain systems, heating and cooling systems, etc., that result in less fuel consumption and lower harmful emissions. Furthermore, electronic components tend to be lighter than the mechanical components they replace, again leading to improved energy efficiencies [1].

Modern motor vehicles leverage electro-mechanical components that produce precise performance control via algorithms implemented in software. With every model redesign, modern motor vehicles are incorporating more electronics and related software to their designs in order to remain competitive. As with many products today, the pace of innovation in vehicle design is accelerating. This requires continuous assessments of how customers' expectations evolve over time and how typical users will interact with new design features and capabilities. When a new feature that may impact the end-user's driving experience is introduced, designers consider all potential ramifications of such a change. The introduction of computers and software in motor vehicles has made possible designs that previously were physically impossible or impractical to build. However, with much fewer constraints than the physical systems it replaces, software also quickly grows in complexity, making it more challenging to fully understand the spectrum of software behavior and the ramifications of design changes under all conditions [2] [3] [4].

Further, automotive companies have unveiled research projects in recent years to develop self-driving vehicles. They are further implementing new technologies that enable the vehicle to perform various driving functions automatically [5]. Manufacturers continue to combine functionalities to achieve higher levels of automation. The widespread adoption of drive-by-wire technologies will likely accelerate the ongoing shift in automotive electronic control systems from providing individual function controls to providing integrated controls. This shift has the potential to improve safety and drivability. This will also drive the trend toward increased vehicle computerization. A common element in all levels of automation is safety-critical

electronic control systems. The capabilities of motor vehicles, coupled with vehicle-to-vehicle and vehicle-to-infrastructure communications, will provide the opportunity for further innovations that could also improve driver comfort, provide useful information and entertainment, and, most importantly, advance safety. With this emerging and rapid evolution of vehicle electronic control and connectivity come increased challenges in the areas of safety assurance.

A recent National Academy of Sciences study sponsored NHTSA found that the proliferation and increasingly interconnected electronics systems are creating opportunities to improve and enable many vehicle capabilities and changes in familiar driver interfaces. However, these advances in electronic systems also present challenges for system design [2]. Failures associated with electronics systems — including those related to software programming, intermittent electronic hardware faults, and electromagnetic disturbances — may not leave physical evidence to aid investigations into observed or reported unsafe vehicle behaviors. NAS recommended that NHTSA become more familiar with and engaged in standard-setting and other efforts involving industry that are aimed at strengthening the means by which manufacturers ensure the safe performance of their electronics systems. Subsequently, NHTSA established a research program on vehicle electronics and sought public comment on its research approach [6].

1.2 Research Objectives

In this study, Volpe provides technical support to NHTSA in the following focus area concerning the safety and reliability of automotive electronic control systems:

- Expand and share the knowledge base to ensure research plans for automotive electronics reliability are appropriate and promote tools for applied research in this area; and
- Strengthen and facilitate the implementation of safety-effective voluntary industry-based standards for automotive electronics reliability.

Specifically, Volpe compared and assessed the following relevant safety standards:

- ISO 26262: Road Vehicles - Functional Safety [7]. This voluntary industry standard is the first comprehensive and voluntary automotive safety standard that addresses the functional safety of electrical and/or electronic (E/E) and software-intensive features in today's road vehicles.
- MIL-STD-882E: Department of Defense Standard Practice - System Safety [8]. This system safety standard practice identifies the Department of Defense systems engineering approach to eliminating hazards, where possible, and minimizing risks where those hazards cannot be eliminated. It is a required practice for military automotive design.
- DO-178C: Software Considerations in Airborne Systems and Equipment Certification [9]. This is an industry-accepted guidance for software in airborne systems and equipment used in the Aviation industry.

- FMVSS: Federal Motor Vehicle Safety Standard [10]. These Federal safety standards are minimum-performance requirements for motor vehicles or regulated items of motor vehicle equipment. Each standard needs to meet the need for motor vehicle safety. Motor vehicle safety in an FMVSS means the performance of a motor vehicle or motor vehicle equipment in a way that protects the public against unreasonable risk of accidents occurring because of the design, construction, or performance of a motor vehicle, and against unreasonable risk of death or injury in an accident. Motor vehicle safety also includes the non-operational safety of a motor vehicle.
- AUTOSAR: Automotive Open System Architecture [11]. This voluntary automotive industry standard consists of a set of specifications that describe a software architecture, application interfaces, and a methodology. The key goals of this standard are to promote scalability to different vehicle and platform variants, transferability throughout the network, integration from multiple suppliers, maintainability throughout the entire product life-cycle, and software updates and upgrades over the vehicle's lifetime.
- MISRA C: Motor Industry Software Reliability Association's guidelines for the use of the C computer programming language in critical systems [12] [13]. This is a voluntary automotive industry standard for the use of C language in safety-related automotive embedded systems.

The assessment produced a list of observations that may aid the development of enhanced functional safety approaches for automotive electronic control systems.

2 DESCRIPTION OF RELEVANT SAFETY STANDARDS

This study reviewed six safety standards relevant to automotive electronic control system safety, as listed above. Among these standards, ISO 26262 is the most applicable safety standard within the context of this study. It is the first comprehensive and voluntary industry standard that addresses the functional safety[3] of automotive systems comprised of E/E and software elements.

This study also reviewed five additional standards outlined in the previous section because the ISO 26262 committee and industry practitioners frequently cited them in various discussions and research papers [14] [15]. MIL-STD-882E, DO-178C, and MISRA C are mature safety standards that have gone through a number of iterations. FMVSSs are performance standards, to which vehicles sold in the United States, already need to comply. AUTOSAR was developed around the same time as ISO 26262 and references ISO 26262 for safety considerations.

2.1 ISO 26262 Road Vehicles – Functional Safety

ISO 26262 is the first comprehensive automotive safety standard that addresses the functional safety of the growing number of E/E and software-intensive features in today's road vehicles [7]. ISO 26262 is an adaptation of the International Electrotechnical Commission 61508 standard [16] to road vehicles. IEC 61508 requires system designers and developers to consider all environmental factors that could result in an unsafe situation for the subject product. These factors include shock, vibration, temperature, and electromagnetic fields and their induced voltages and currents.

Published in November 2011, the first edition of the ISO 26262 standard has 10 parts. Parts 2 through 7 are the core of the standard. Parts 1, 8, 9, and 10 provide supporting information to various parts of the core content.

ISO 26262 recognizes and intends to address the important challenges facing today's road vehicle technologies including [7]:

- Safety of new E/E and software functionality in vehicles;
- Trend of increasing complexity, software content, and mechatronics implementation; and
- Risks from both systematic failure and random hardware failure.

ISO 26262 covers "*safety-related systems that include one or more E/E systems and that are installed in series production passenger cars with a maximum gross weight up to 3.5 tons* [7]." It addresses possible hazards caused by malfunctioning behavior of safety-related E/E systems including interactions of these systems. It does not address hazards such as electric shock, fire,

[3] Functional safety refers to the absence of unreasonable risk due to hazards caused by the malfunctioning behavior of electrical/electronic systems.

smoke, heat, radiation, toxicity, reactivity, corrosion, release of energy, and similar hazards unless directly caused by malfunctioning behavior of E/E safety related systems (see Part 1 [7]).

ISO 26262 prescribes a systems engineering process for safety engineering. It recognizes that safety is a system attribute and can be addressed using a systems engineering approach. It also emphasizes the importance of fostering a safety culture and implementing safety engineering management.

In 2011, Volpe performed a preliminary review and assessment of the standard based on the 2009 draft version that was the only publically available version at the time of the Volpe study [17]. The assessment here is based on the final draft version of the standard — the last draft before the publication of the first edition of the completed standard.[4]

In addition this study also reviewed the *Handbook for Functional Safety* (Microcontroller Application edition) published by JASPAR[5] General Incorporated Association [18]. This handbook provides detailed recommendations on how to improve the hardware architecture metrics calculation in ISO 26262 Part 5 for microcontrollers. Due to its detailed nature this handbook is considered as a supporting document for ISO 26262 and is not separately discussed in this comparison study.

2.2 MIL-STD-882E U.S. Department of Defense Standard Practice - System Safety

MIL-STD-882E is the U.S. Department of Defense Systems Engineering approach for eliminating hazards, where possible, and minimizing risks where those hazards cannot be eliminated. By taking a systems approach,[6] this standard considers hazards in the entire lifecycle of systems/products/equipment/infrastructure, including design, development, test, production, use, and disposal. This standard concisely prescribes a systems engineering process for safety engineering.[7] The principle of this standard is that system safety should follow the systems engineering process,[8] and all engineers who are involved in the design of the system should consider the safety implications of their design choices in conjunction with the specialized safety engineers. This standard has gone through a number of revisions in order to adapt to changes in

[4] Experts in the ISO 26262 standard committee confirmed that the technical content of the final draft version is identical to the published edition.
[5] JASPAR (**J**apan **A**utomotive **S**oftware **P**latform and **Ar**chitecture) was originally established in 2004 to pursue increasing development efficiency and ensuring reliability, by standardization and common use of electronic control system software and in-vehicle networks, among car manufacturers, tier-1 suppliers, semiconductor manufacturers, and software developers.
[6] Systems approach is the understanding of a system by examining the linkages and interactions between the elements that compose the entirety of the system, and its interactions with other systems and its environment.
[7] Safety engineering assures that a life-critical system behaves as needed and provides acceptable levels of safety. It is related to systems engineering.
[8] The systems engineering process is a discovery process that begins by discovering the real problems to be resolved, identifies the most probable or highest impact failures that can occur, and involves finding solutions to these problems.

technology and lessons learned through experience [19] [20] and the latest published version is MIL-STD-882E [8].

2.3 DO-178C Software Considerations in Airborne Systems and Equipment Certification

DO-178C is an industry-accepted guidance for software in airborne systems and equipment in the Aviation industry [9]. The Federal Aviation Administration issued the Advisory Circular 20-115C in July 2013 to recognize this standard as one of acceptable means, but not the only means, for showing compliance with the applicable airworthiness regulations for the software aspects of airborne systems and equipment certification. Compliance to this standard means that the software satisfies airworthiness requirements with an acceptable level of confidence. As part of the airworthiness certification process, DO-178C provides guidelines to the generation of software lifecycle data and documentation needed in order to support the certification process. This includes software development plan, verification plan, configuration management plan, quality assurance plan, requirements standards, design standards, code standards, requirements data, design description, source code, executable object code, verification, problem reports, etc. It also provides a comprehensive list of considerations in order to avoid introducing errors into the software.

DO-178C considers software system development as a subset of the overall system development process. It assumes that safety-critical requirements for software systems are defined in the higher-level systems engineering activities and are given at the beginning of the software development process.

2.4 Federal Motor Vehicle Safety Standards (FMVSS)

In addition to authorizing other regulatory tools, the National Traffic and Motor Vehicle Safety Act (the "Safety Act") directs the Secretary of Transportation to issue Federal safety standards for motor vehicles.[9] The Safety Act states that FMVSSs need to be "practicable, meet the need for motor vehicle safety and stated in objective terms."

Motor vehicle safety in an FMVSS considers the performance of a motor vehicle or motor vehicle equipment in a way that protects the public against unreasonable risk of accidents occurring because of the design, construction, or performance of a motor vehicle, and against unreasonable risk of death or injury in an accident. Motor vehicle safety also includes the nonoperational safety of a motor vehicle.

To state the standard in objective terms, NHTSA generally establishes performance requirements for motor vehicles (or motor vehicle equipment) and establishes specific test conditions under which the motor vehicle (or motor vehicle equipment) needs to meet those requirements. The

[9] *See* 49 U.S.C. § 30101 et. seq., authority delegated from the Secretary of Transportation to the Administrator of NHTSA in 49 CFR Part 1.95.

Safety Act requires manufacturers to self-certify that their motor vehicles and items of motor vehicle equipment conform to these standards [2] [10].

Thus, the tests specified in the standards should enable both the manufacturer and the agency to determine whether a product complies. Other voluntary process standards discussed in this report use a different approach for evaluating safety (i.e., not necessarily based on running tests to determine pass/fail against performance criteria) and, therefore, do not have comparable objective measurements for determining conformance or non-conformance with the voluntary standard.

2.5 Automotive Open System Architecture

Industry practitioners often describe AUTOSAR as a means to support system safety by improving compatibility between electronic components within the vehicle. The following quote from the AUTOSAR website brochure explains what AUTOSAR is [11]:

> *"AUTOSAR (AUTomotive Open System ARchitecture) is a partnership between automotive manufacturers, suppliers and tool and semiconductor vendors. Since 2003, AUTOSAR has been working on the development of an open, standardized software architecture for automotive electronic control units (ECUs)."*

The AUTOSAR web publication further explains [11]:

> *"The AUTOSAR standard consists of a set of specifications that describe a software architecture, application interfaces and a methodology. The AUTOSAR layered software architecture enables the development of independent software components. These can be used in vehicles of different manufacturers, and in electronic components of different suppliers that can span multiple product generations. It results in a high reliability of the overall system with significant cost and capacity benefits."*

AUTOSAR aims to "improve complexity management of integrated E/E architectures through increased reuse and transferability of software modules between carmakers and suppliers [11]." As an industry standard, AUTOSAR's main objective is not safety. Nonetheless, it supports ISO 26262 by providing architectural concepts that can enhance safety, such as the memory partitioning concept, the time determinism concept, the program flow monitoring concept, and communication stack related features, etc. [11]. Section 4.11 of AUTOSAR also lists requirements for safety concerning the data consistency, hardware memory protection features, data corruption detection and protection, etc. Such design features alone cannot guarantee system safety, but they are powerful techniques to prevent and eliminate hazards upfront in the development process. The software architecture development process described in ISO 26262 and the architecture concepts in AUTOSAR are complementary methods that can improve the safety of automotive electronic control systems.

2.6 Guidelines for the Use of the C Language in Critical Systems (MISRA C)

MISRA C is a standard concerning the use of C language in safety-related automotive embedded systems [12] [13]. The C language is heavily used in the development of safety-critical software where reliability is a prime concern. ISO 26262 recommends following MISRA C for software coding. Therefore, it is a supporting standard for the software development portion of ISO 26262. While MISRA C is dedicated to the proper coding of the C language, AUTOSAR promotes the use of a common software architecture that enables compatibility and transferability among different system components.

MISRA C provides a set of rules that constrain the use of C language in order to avoid, as much as possible, certain limitations of this programming language that may lead to safety concerns. These limitations include programmer errors that may stem from misunderstanding of the language, compiler variations and errors, and potential poor performance of the language in run-time checking.

3 COMPARATIVE ANALYSIS OF STANDARDS

The following 11 dimensions were compared among the six standards.

1. Type of standard
2. Definition of safety and hazard
3. Identification of safety requirements
4. Hazard and safety analysis methods
5. Management of safety requirements
6. Risk assessment approach
7. Design for safety approach
8. Software safety
9. System lifecycle consideration
10. Human factors consideration
11. Review, audit, and certification

3.1 Type of Standard

Four different approaches to safety assurance exist among the six standards.

- Process prescription
- Methods listing
- Performance-based
- Design prescription

3.1.1 Process Prescription

A process prescription standard prescribes a safety engineering process. ISO 26262, MIL-STD-882E, and DO-178C all belong to this category, prescribing a top-down systems engineering[10] process for safety engineering. ISO 26262 and DO-178C both make safety engineering an integral part of the product development process. On the other hand, MIL-STD-882E specifies a system safety engineering process separate from but parallel to the product development process.

MIL-STD-882E and DO-178C only prescribe the safety engineering process. They do not specify the method used in each process step. The benefits of such an approach are:

- Project manager and contractors have the full freedom to select the most suitable safety engineering method for the project.

[10] "Top-down" systems engineering for the development of a "product" starts from the highest level (the system) to meet the performance requirements and the sub-systems and, ultimately, components are developed to a cascading set of requirements down from the highest level. Contrast that with the "bottom-up" approach that starts from components and sub-components (the lowest levels) and integrating them into sub-systems, and further integration into the product at the highest level.

- Standard imposes minimal restriction on how work is done and does not run the risk of inadvertently misguiding the user by over prescribing.
- Standard is flexible and may be more easily applied to novel technologies.

The drawback, however, may be that the standard only provides coarse high-level process guidelines and leaves the burden on developers and certifiers on each project to make certain that the detailed process is adequate to ensure safety for a particular project. In other words, with less specification in the standard (thus, more flexibility), the level of safety achieved by conforming to the standard might vary based on the developers' application of the standard.

3.1.2 Methods Listing

In addition to prescribing the safety engineering process, ISO 26262 also lists a number of commonly known hazard analysis and safety analysis methods in various parts of the standard. These methods include failure mode and effects analysis, fault tree analysis, and hazard and operability analysis, etc. (see Part 9 of ISO 26262). ISO 26262 also introduces hardware random failure metrics and other reliability engineering concepts.

The benefit of listing safety analysis methods is that it may help the practitioners to use the appropriate approach to ensure the safety of the process outcome. However, this approach also requires the standard to continuously examine the suitability of the available methods and incorporate additional advanced analysis methods into the list when appropriate.

3.1.3 Performance-Based

Performance-based standards set minimum performance requirements for specific systems. FMVSSs are generally performance standards for specific subsystems and/or the vehicle as a whole. The advantage of performance-based standards is that they are generally solution neutral. Performance targets can apply to both hardware and software components.

Process standards, similar to ISO 26262, approach safety differently as they do not prescribe performance targets for the completed product, but instead ask the designer to consider the safety implications of his/her design choices. Process standards may complement performance standards such as those in the FMVSSs.

3.1.4 Design Prescription

Design prescription standards focus on a specific aspect of the product design. For instance,

- AUTOSAR focuses on system architecture design, and
- MISRA C focuses on software coding rules.

Design prescription standards can be very powerful in preventing and eliminating hazards when applied to the appropriate system safety process steps. However, they focus on specific aspects

of the design solution. They may not be sufficiently general as safety standards for the overall safety of automotive electronic control systems.

3.2 Definition of Safety and Hazard

The definitions of *safety* and *hazard* are important as they define the underlying assumptions of the safety standard. Table 1 presents these definitions.

Table 1: Definition of Safety and Hazard

Standard	Safety	Hazard
ISO 26262	Absence of unreasonable risk.	Potential source of harm caused by malfunctioning behavior of the item. **Malfunctioning Behavior**: failure or unintended behavior of an item with respect to its design intent.
MIL-STD-882E	Freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.	A real or potential condition that could lead to an unplanned event or series of events (i.e., mishap) resulting in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.
DO-178C	No clear definition provided. Based on Failure Condition Category in the standard, the definition is similar to MIL-STD-882E.	No definition provided.
FMVSS	The rulemaking process (pursuant to the statutory definition of the need for motor vehicle safety) considers the potential hazards that might occur and their safety implications. However, once the standard is established, one complies with the standard based on meeting the established performance requirements rather than analyzing potential hazards and their safety implications on a case-by-case basis.	
AUTOSAR	Same as ISO 26262.	Same as ISO 26262.
MISRA C	No explicit definition.	No explicit definition.

While the definitions of *safety* in MIL-STD-882E and DO-178C focus directly on the impact on people and society, the definition of *Safety* in ISO 26262 and AUTOSAR is based on the absence of "unreasonable risk." No further definition of "unreasonable risk" is provided.

MIL-STD-882E defines *hazard* as a system condition that could lead to the loss of safety. This definition of hazard encompasses component failures and unsafe system interactions, both of which are mentioned in the NAS report as future challenges for automotive electronic control systems [2]. On the other hand, ISO 26262 limits *hazard* to be the effect of the malfunctioning behavior of the item, which is defined as not meeting the manufacturer's design intent. This definition of hazard may correspond to that of MIL-STD-882E. However, the hazard definition

in MIL-STD-882E has a broader focus where the design intent of the manufacturer should fully represent vehicle operators' need for safety and the safety interest of the public. We note that limiting the hazard definition to cases where the system/component fails to meet its design intent may not address safety issues arising from unsafe interactions between systems where no system/component may fail to perform its intended function.

3.3 Identification of Safety Requirements

The standards reviewed have various approaches to identify the safety requirements:

- MIL-STD-882E uses hazard analysis methods to generate safety requirements.
- ISO 26262 uses hazard analysis methods to generate system-level safety requirements, and safety analysis methods to generate lower-level safety requirements among other inputs.
- DO-178C assumes software requirements flow down from system-level activities. No guidance is provided on how to further decompose the requirements, or how to identify additional safety-critical requirements at each level of the system decomposition hierarchy.
- NHTSA's rulemaking process establishes the performance requirements in the FMVSSs. This is based on research and analysis of the crash data, the availability and capability of relevant countermeasures, and any other available information regarding a particular motor vehicle safety problem. The rulemaking process also establishes performance tests to evaluate a motor vehicle or motor vehicle equipment against the established performance requirements.
- AUTOSAR's focus is to generate architecture design requirements that will support the safety requirements in ISO 26262; not to generate safety requirements on its own.
- MISRA C focuses on coding standard, which is a very specific aspect of the overall electronic control system safety.

3.4 Hazard and Safety Analysis Methods

The standards reviewed take different hazard and safety analysis approaches to generate safety requirements at various levels of the system's decomposition hierarchy (i.e., top-down decomposition of the system into subsystems, functions, and then modules.)

3.4.1 Hazard Analysis in MIL-STD-882E

MIL-STD-882E requires the Functional Hazard Analysis (Task 208 on page 74 of reference [8]) method that assesses the safety implication of not meeting each of the specified system functions. The advantage of this method is that it can be performed with a high-level definition of the system early in the design and development process. However, this method assumes all functions are already identified, and only looks at the consequence of not meeting the predefined

requirements. It does not prompt the practitioners to identify additional missing requirements or functions that may lead to hazards.

3.4.2 Hazard and Safety Analysis in ISO 26262

In ISO 26262, hazard analysis is discussed at the item level — the highest level of system decomposition (Part 3 Clause 7.4.2.2 in [7]). The listed hazard analysis methods include brainstorming, checklists, quality history, FMEA, and field studies. At lower level of the system decomposition, ISO 26262 employs safety analysis (see Part 9 Clause 8 in [7]), including FMEA, FTA, ETA, HAZOP, etc. These are valid and commonly used methods in the automotive industry. However, these methods also have limitations when applied to modern complex electronic control systems.

Brainstorming, checklists, quality history, and field studies depend on past experiences. Brainstorming and checklists also depend on what the participating experts experienced and remembered. For new and rapidly changing technology with little history and experience to draw upon, these techniques may miss potential hazards.

FMEA, FTA, and ETA methods were developed 50 to 60 years ago when most of the engineered systems were mechanical, and random hardware failures were the dominant causes of safety issues. Modern vehicles with increasingly complex electronic control systems may get into unsafe states also due to software errors and potential unsafe system interactions.

FMEA, FTA, and ETA methods handle random hardware failure problems well. They handle these problems both qualitatively and quantitatively. Even though they may be used to address system interfaces with other surrounding systems, these methods do not provide sufficient guidance to identify causes of unsafe system interactions. Moreover, these methods are not immediately applicable to software because software does not have random failures like hardware [4]. In addition, automation and the human-machine interface can also contribute to system hazards [2], which also need special treatment and further guidance.

Another issue with FMEA, FTA, and ETA is that they are based on simple linear chain-of-event accident causality models [4]. In complex systems, hazards are not necessarily always caused by simple linear chain of events. Systemic issues and system interaction related problems may not be easily identified if the hazard causality were modeled as a simple linear chain-of-events.

FMEA is a bottom-up inductive method. Dealing with a complex system, it is a laborious task for an analyst to identify and analyze all possible component failures and combination failures that may cause system-level hazards [21]. It is even more difficult to foresee all the combinations of non-failure states of components that may lead to hazardous conditions.

FTA is a top-down approach focused on identifying faults and their causes. It alleviates the analysts' daunting task of finding all combinations of potential component failures that may lead to severe system-level failures [21] [22]. However, FTA does not provide as much guidance on

how to generate the lower-level events or failures that may lead to the higher-level system faults. Fault trees are typically records of the results of brainstorming or expert opinion, providing limited guidance or documentation on the hazard causal analysis thought process [23]. Once a fault tree is built, it can also be used for the Quantitative Reliability Analysis (QRA)—a method also listed in ISO 26262. The probability of lower-level events, often component failures, is aggregated up the tree to produce top-level event's probability, if component failures can be proven to be independent of one another. If the top-level event has a probability of occurrence below the acceptance level, then the system is considered safe. However, not all of the lower level events' probability data exist, especially for new automotive electronics technology, software, or interaction of human operators with the electronic systems. Software failure data may not be probabilistic in nature, and therefore cannot be used for probability math [24] [25]. This lack of data leaves room for subjectivity in the assessment of numerical probability values. This approach also has limitations in inadequately handling situations when none of the components fails, yet the interaction among components leads to potentially hazardous conditions.

HAZOP is also a referenced safety analysis method in ISO 26262. HAZOP was originally developed in the process industry [24] [25]. It is largely a qualitative analysis technique. The use of guidewords and the modeling of the actual process help the analyst to consider both component failures and system interactions. However, the adaptation of the method to assess potential hazards of automotive electronic control systems is not necessarily straight forward and may need modifications and testing to demonstrate its effectiveness. ISO 26262 does not provide guidance on how to apply this method to automotive systems.

ISO 26262 states that the above mentioned safety analysis methods are applicable for software development as well. However, the majority of software-related accidents is due to requirements flaws, not coding errors that lead to non-conformity of the requirements [4]. Software also does not fail randomly like hardware [4] [26] [27]. Therefore, these methods are not always considered adequate for software hazard analysis [4]. DO-178C also states that software reliability metrics should not be used for software safety assessments [9].

This section discussed the safety analysis methods listed in ISO 26262. This report acknowledges that Part 6 of the ISO 26262 standard provides a process for comprehensive checks of software requirements including, but not limited to, confirmation reviews, evidence of traceability, requirements management, and change control.

3.4.3 Hazard Analysis in FMVSS

FMVSSs evaluate the safety performance of a motor vehicle or motor vehicle equipment based on its performance under specified test conditions. Unlike the other standards discussed in this section, an FMVSS does not perform a hazard analysis when evaluating compliance of a particular motor vehicle or item of motor vehicle equipment. Instead, this hazard analysis is performed during the rulemaking process. It is based on research and analysis of the crash data,

the availability of relevant countermeasures, and any other available information regarding a particular motor vehicle safety issue.

3.4.4 System Theoretic Process Analysis for Complex Systems

Recognizing the limitations of traditional hazard analysis methods, many researchers and practitioners are working on improving existing methods or developing new methods to meet the challenges of modern complex electronic control systems. One such new method is the system theoretic process analysis [4], a systematic process that guides the analysts through the identification of hazards and causal factors. It views system safety as a control problem and addresses it following the feedback control system theory [28] [29] [30] [31]. It can lead the analysts to discover safety-critical component failures as well as potential unsafe system interactions. It can help model software requirements so as to reduce software design errors. STPA can also be easily used in the early stage of the design when only high-level system concept is available and later stages as more design details emerge. Moreover, STPA guides the analysts through an organized process to identify causal factors at different levels of hierarchy in the system, and hence ensures safety requirements are traceable throughout the system. STPA can be used for:

- Generating safety requirements at all levels of the system hierarchy for hardware, software, and human factors consideration.
- Comparison of design concepts early in order to choose a safer design concept up front.
- Prompting the consideration of hazard prevention and elimination before other countermeasures.
- Interface analysis for system modification and component reuse.
- System safety analysis including random hardware failure, software design error, and unsafe system interactions.

Although STPA has shown a lot of promise in other industries [32] [33] [34], applications to motor vehicles are still limited in numbers [35] [36]. Similar to all analysis methods, the effectiveness of STPA also depends on the analyst's domain knowledge and expertise.

3.5 Management of Safety Requirements

ISO 26262 and DO-178C manage safety requirements as a part of the overall system requirements. On the other hand, MIL-STD-882E requires the use of a Hazard Tracking System including identified hazards, associated mishaps, risk assessments, identified risk mitigation measures, selected mitigation measures, hazard status, verification of risk reductions, and risk acceptance [8]. HTS is used by the government and contractors for safety management, and it is separate from the rest of the system requirements.

If HTS and requirements management system are separate, it would be important to make sure that the two refer back to each other on safety-critical design requirements if the designers were incorporating safety into the system up front in the development process.

3.6 Risk Assessment Approach

Explicit risk assessment is presented in ISO 26262, MIL-STD-882E, and DO-178C as follows:

- ISO 26262 uses the Automotive Safety Integrity Level to assess the risk. Once the top-level safety goals are established, the ASIL is assessed for each of the safety goals based on three categories: *Severity* (S) of the accident if the safety goal is not met, probability of *exposure* (E) regarding hazardous operational situations, and the *controllability* (C) of the hazardous situation. The final ASILs for the safety goals are assessed based on a tabular combination of S, E, and C.
- MIL-STD-882E combines *severity* of the accident and *the probability* of occurrence of the hazard to create the risk index for hardware systems. For software, it uses the *severity* and the *software control category* to assess the risk. *Software control category* is "an assignment of the degree of autonomy, command and control authority, and redundant fault tolerance of a software function in context with its system behavior [8]."
- DO-178 specifies software levels only based on the impact of software anomalous behavior on the overall system safety (*severity*).

Table 2 summarizes these three different approaches. ISO 26262, MIL-STD-882E, and DO-178C all agree that the hazard severity is a necessary dimension of the risk assessment for both hardware and software systems. Severity of the hazardous event is usually relatively easy to define based on the resulting harm or loss. On the other hand, ISO 26262, MIL-STD-882E, and DO-178C differ in other dimensions and in the treatment of hardware versus software systems.

Table 2: Comparison of Risk Assessment Approaches

Measure	ISO 26262 Hardware and Software	MIL-STD-882E for Hardware	MIL-STD-882E for Software	DO-178C (Software only)
Severity	√	√	√	√
Probability of Operational Scenario (Exposure)	√			
Probability of Mishap Occurrence		√		
Controllability	√			
Software Control Category			√	

3.6.1 Hardware Risk Assessment—Exposure and Probability of Mishap Occurrence

In addition to severity, ISO 26262 uses the probability of operational scenario—*exposure.* MIL-STD-882E uses the probability of the occurrence of the hazardous event as an additional dimension for hardware risk assessment. This approach results in clear risk assessment so long as statistically valid probability for the operational scenario or hazardous event can be obtained. However, historical information about older systems is not always statistically valid for new systems because the design and the operating context can change [23]. Therefore, new systems may not have any historical data to rely on.

When statistically valid data are not available, subject matter experts often estimate the probability of an event subjectively. This is a recommended practice in both ISO 26262 and MIL-STD-882E [7] [8]. Psychologists studying human decision-making have shown that humans are not good at predicting truly random events, especially rare events [37]. For example, the availability of an event in the risk analyst's mind, and how vividly the event is described, heavily influence the subjective probability assessment.

When the probability of occurrence is uncertain, a conservative assessment should be taken. The most conservative probability value for either *exposure* or *probability of occurrence* is one hundred percent (100%). In other words, the most conservative approach to risk assessment is to use only the *severity* dimension for risk assessment when a statistically valid probability value does not exist.

3.6.2 Hardware Risk Assessment—Controllability

Controllability is a risk assessment dimension unique to ISO 26262, which is defined in Part 1 of the standard as [7]:

> *"Ability to avoid a specified harm or damage through the timely reactions of the persons involved, possibly with support from external measures.*
>
> *NOTE 1 Persons involved can include the driver, passengers or persons in the vicinity of the vehicle's exterior.*
>
> *NOTE 2 the parameter C in hazard analysis and risk assessment represents the potential for controllability."*

ISO 26262 Part 3 Clause 7.4.3.7 further clarifies that "*the evaluation of the controllability is an estimate of the probability that the driver or other persons potentially at risk are able to gain sufficient control of the hazardous event, such that they are able to avoid the specific harm* [7]."

The evaluation of *C* requires data that could be obtained from extensive testing with statistically valid sample selection, sample size, and testing condition. Operator's reaction to the same warning message or hazardous condition depends on the complex interaction of many factors such as the operator's demographic and physical condition, the specific user interface design, the

level of automation, etc. (see Appendix B of Part 3 [7], and also [38] [39] [40]). The two most obvious challenges are:

1. Resource and time needed to run human factors testing for each new vehicle design are often extensive. Historical data should be used with caution unless it can be proven that relevant factors are equivalent in both designs.
2. Tests should reflect the specific design implementation on how the users interact with the system and how they are informed of the hazardous system states. Such details about the design are usually not available during early phase of the system design process when ASILs are evaluated.

3.6.3 Hardware Risk Assessment—Overall

The overall risk assessment discussion here focuses on the ISO 26262 approach as it is the most relevant to automotive electronics safety. The adaptive cruise control system can be used as an example to show the ASIL assessment process. Take the operational scenario of exiting the highway on a curved exit ramp with ACC engaged. Due to the limitation of the front radar, the ACC system may not be able to safely reduce the vehicle speed if the ramp curvature is tight. The resulting hazard is that the vehicle may rear-end the vehicle traveling ahead of it. Based on Table B.1 in Part 3 Appendix B [7], this hazard has severity **S3**. Based on Table B.2 in Part 3 Appendix B [7], the exposure for the exiting a highway ramp scenario is **E2**. Based on Table B.3 in Part 3 Appendix B [7], the "unavailability of a driver assisting system" gets **C0** or **C1** for controllability. Combining S3, E2, and C0/C1, the ASIL value for the rear-ending hazard when leaving ACC on while exiting a curved highway ramp is **QM**—a high-severity hazard is reduced to a non-safety-critical issue. It is unclear what evidence has been produced to demonstrate that the "unavailability of a driver assisting system" is only a C0 or C1 controllability, while many examples in cockpit automation illustrate concerns for the safety of shared responsibility between human pilots and automated controllers [4] [38] [39] [40]. It should be noted that ISO 26262 requires the overall ASIL assessment for a hazard to take the value of the worst-case operational scenario. Therefore, the final ASIL for the said hazard may be worse than QM after examining all of the operating scenarios.

Nonetheless, the above ACC example illustrates that the use of *exposure* and *controllability* effectively provides an opportunity to reduce the risk measure for a high severity hazard. We note that, in some instances, an analyst may need to make assessments regarding exposure and controllability of a high severity hazard in the absence of valid statistics and rigorous testing. In this situation, an analyst's conclusions based on the limited available information may be the basis for re-categorizing a high severity hazard to a lower severity category. Otherwise, the most conservative approach is to use the *severity* dimension alone. Some in the industry argue that a conservative approach to risk assessment may have cost implications on the system, while others argue the economy of scale for large volume production vehicles can counter-balance the cost

implication. Furthermore, it has also been argued that designing safety into the system in the first place may not lead to an increase in the cost of the overall system [4] [8].

3.6.4 Software Risk Assessment

The approaches to software risk assessment vary among the standards reviewed:

- ISO 26262 assigns ASIL to software requirements following the ASIL decomposition process (Part 9 Clause 5 of [7]). The software requirement may receive an ASIL that is either the same or lower than the ASIL associated with its associated safety goal.
- MIL-STD-882E recognizes that "*determining the probability of failure of a single software function is difficult at best and cannot be based on historical data*" [8]. Therefore, MIL-STD-882E does not use a software failure probability concept. Instead, MIL-STD-882E uses *software control category* in addition to *severity* to classify the software related risk. Caution needs to be exercised when using *software control category*. For instance, semi-autonomous operations by software may not necessarily be less risky than a full-autonomous operation. There are examples in aviation where the operators experience mode confusion relying on semi-autonomous operations that result in hazardous situations [38] [39] [40]. Redundant fault tolerant software also needs verification that they are truly independent. This may be very hard to do in early stages of the design and development process when the risk assessment is carried out. Again in the case for software, the truly conservative approach may be to follow the *Severity* assessment only [4] [8].
- DO-178C states that "*it is important to realize that the likelihood that the software contains an error cannot be quantified in the same way as for random hardware failures*" [9]. The risk assessment is only based on the severity of the impact in DO -178. Furthermore, software reliability models and calculations are not accepted by the standard since they are also based on the probability of random failures (DO-178C Section 12.3.3 [9]).

DO-178C presents the most conservative approach to software risk assessment using only the *Severity* measure.

3.7 Design for Safety Approach

The design for safety approaches varies among the standards reviewed.

MIL-STD-882 provides a clear statement about its design for the safety approach in section 4.3.4 [8]:

> *"The goal should always be to eliminate the hazard if possible. When a hazard cannot be eliminated, the associated risk should be reduced to the lowest acceptable level within the constraints of cost, schedule, and performance by applying the system safety design order*

of precedence. The system safety design order of precedence identifies alternative mitigation approaches and lists them in order of decreasing effectiveness:

a. Eliminate hazards through design selection
b. Reduce risk through design alteration
c. Incorporate engineered features or devices
d. Provide warning devices
e. Incorporate signage, procedures, training, and Personal Protective Equipment."

This approach is similar to the design for safety philosophy proposed by Leveson [4] [23], which promotes hazard prevention and elimination before the consideration of safety devices. Safety devices add cost and complexity. The safety of the safety device will also have to be analyzed to ensure overall safety of the system. Therefore, it is desirable to first attempt steps *a* and *b* listed above.

Even though ISO 26262 does not provide a direct statement, its design for safety approach is implied by two terms—*safety measure* and *safety mechanism*. *Safety measure* is defined as "activity or technical solution to avoid or control systematic failures and to detect random hardware failures or control random hardware failures, or mitigate their harmful effects" [7]. For example, design methods and design processes are safety measures. *Safety mechanism* is defined as the "measure implemented by an E/E function or element, or in other technologies, to detect or control failures in order to achieve a safety state of the item, or maintain a safety state of the item, or both" [7]. For example, a diagnostic component that monitors the fault of a sensor is a safety mechanism.

Safety measures could potentially include preventing and eliminating hazards (steps *a* and *b* in the above list). The systems engineering process and associated steps and deliverables described in the ISO 26262 are safety measures. However, process steps and deliverables are only *what* need to be done. In many places in the standard, ISO 26262 provides limited guidance on *how* to accomplish these process steps effectively. Safety mechanisms are engineered features or devices that would mitigate a hazardous state (Step *c* in the list above). ISO 26262 contains extensive discussions about developing safety mechanisms in each process step, immediately after the system-level hazard analysis is complete.

DO-178C and FMVSS do not have explicit design for safety approaches. AUTOSAR and MISRA C are standards describing specific design for safety approaches. AUTOSAR specifies good system architecture as a way to prevent hazards. MISRA C constrains C coding practices in order to reduce error in the software.

3.8 Software Safety

ISO26262, MIL-STD-882E, and DO-178C ensure software safety by following the systems engineering process.[11] AUTOSAR and MISRA C address narrower scopes by specifying means to achieve software safety—good architecture (AUTOSAR) and good coding standard (MISRA C).

In addition to the systems engineering process, ISO 26262 suggests good software design practices in the architecture design, unit design, and testing phases. For instance, architectural design considerations include verifiability, testability, modularity, minimum complexity, etc. (Part 6 Clause 7.4.2 [7]). Table 3 of Part 6 in the standard provides a list of methods that may help improve software modularity. However, the standard does not provide guidance on how to arrive at the rest of the architectural properties or advice on how to measure whether or not these architectural attributes have been achieved to a satisfactory level.

Following good software design practices has safety benefits. However, an additional area for consideration is how to generate a comprehensive list of safety-related software requirements. Software design practices such as coding standards, architecture consideration, and testing methods are means to achieve design requirements. A software system, no matter how modular, testable, maintainable, or how carefully it adheres to coding standards, such as MISRA C, can only be as good as what the requirements have specified. The effectiveness of the entire systems engineering process on the final product lies heavily on the definition of requirements [4] [41].

Both ISO 26262 and DO-178C focus on addressing "software failures." ISO 26262 uses the concept of *software fault*. A fault is defined as an "abnormal condition that can cause an element or an item to fail" [7]. Failure is defined as "termination of the ability of an element to perform a function as required" [7]. DO-178C assumes all hazards are caused by *software anomalous behavior*, which is defined as "behavior that is inconsistent with specified requirements" [9]. Therefore, both the *software fault* in ISO 26262 and the *software anomalies* in DO-178C focus on the consequential software failures.

However, software failures are only part of the causes of software-related safety issues. In fact, the majority of software-related accidents are due to requirements flaws (i.e., incomplete or insufficient requirements), not coding and other implementation errors [4]. If a requirement is not correctly identified or is missing, the resulting unsafe behavior of the software cannot be addressed by the approaches used to deal with software failure and software reliability [4] [27]. Additional effort is needed to identify as complete a set of safety-related software requirements as possible. ISO 26262 and DO -178C can be further strengthened by clarifying the connection between safety requirements and proposed engineering processes.

[11] The NASA Software Safety Guidebook (www.system-safety.org/Documents/NASA-GB-8719.13.pdf) is another document that may be relevant to this study. However, as a guidebook, it carries less weight than the standards listed above. In addition, it was not suggested by the industry practitioners. Nor was it referenced by the NAS report [2]. Therefore, this study did not review it.

3.9 System Lifecycle Consideration

ISO 26262, MIL-STD-882E, and DO-178C are system design standards, and all three contain lifecycle considerations. System lifecycle considerations in ISO 26262 include production, operation, service (maintenance and repair), and decommissioning (Part 7 of [7]). MIL-STD-882E discusses managing lifecycle risk in section 4.3.8, after the system is operating in the field [8]. DO-178C considers aspects of the software lifecycle including security, maintenance, etc. [9]. However, in all three standards, the discussions regarding how to generate lifecycle safety requirements are very high level, with limited guidance provided.

MIL-STD-882E does not discuss how the manufacturing process might introduce safety concerns into the system, while ISO 26262 does. In the automotive industry, mass production and manufacturing process control is a major effort. As an important phase of the product lifecycle, it makes sense to consider how manufacturing may contribute to system safety.

Even though ISO 26262 considers manufacturing's impact on safety, it provides no guidance on how to identify important safety-critical parameters for manufacturing process control. A commonly used industry practice in identifying critical manufacturing factors is Process FMEA. Referencing earlier discussions on hazard analysis methods, other hazard analysis techniques may also be helpful.

MIL-STD-882E considers health hazard analysis for human operators and maintainers (Task 206 in [8]). It also considers environmental hazard analysis (Task 210 in [8])—identifying hazards to the environment throughout all phases of the system lifecycle. ISO 26262 does not explicitly cover these two topics. The health hazard analysis seems applicable to the scope of ISO 26262 and may have been partially considered in Part 7 of the standard.

AUTOSAR also considers lifecycle of the software system from reusability and modifiability perspective.

3.10 Human Factors Consideration

The NAS and NHTSA state that human factors is an important dimension to vehicle safety, especially when facing the increasing amount of automation enabled by electronic control systems [2] [42]. However, the treatment of human factors consideration varies among the standards reviewed.

ISO 26262 provides very limited guidance on human factors consideration for safety, except in the following two places:

- *Controllability* factor used in ASIL assessment.
- Consideration for driver is mentioned in Part 5 Clause 7.4.3.4 for the ability to notify driver in the event of a system failure.

MIL-STD-882E mentions *human factors engineering* throughout the standard and process steps. It considers the human operator as an integral part of system safety. For instance, human interface study is mentioned as part of the hazard analysis in Section 4.3.2 of MIL-STD-882E. Tasks 101, 102, 103, 106, etc. list human-system integration as an important process step [8]. However, similar to the rest of the standard, no specific method or approach is proposed.

DO-178C only mentions flight crew in the Failure Condition Category Descriptions (Table 2-1 in [9], but not in the rest of the standard. This study did not look at the overall systems engineering activities that provide airworthiness certification. Further investigation may reveal that the human factors consideration for software design is possibly addressed elsewhere in the airworthiness certification process.

The electronics in modern motor vehicles can automate many of the tasks traditionally performed by human drivers. The aviation industry has gone through similar stages in introducing automation into the cockpit to assist pilots and reduce their workload. Automation in the cockpit has also been linked to a number of aviation accidents. Learning from those lessons [38] [39] [40] may help improve the safety of electronic control systems in motor vehicles.

3.11 Review, Audit, and Certification

MIL-STD-882E is not a safety certification standard. Task 104 states that the process should support government reviews, audits, and boards, but details are left up to the program manager and contractors [8].

DO-178C Section 10 specifies how software is qualified as a part of the airworthiness certification process. Specifically [9]:

> *"The airborne community and certification authorities use several terms related to aircraft approval for flight with its associated equipment. The terms used are 'certification,' 'approval,' and, with respect to tools, 'qualification.'*
>
> *'Certification' applies to aircraft, engines, or propellers; and, in respect of some certification authorities, auxiliary power units. The certification authorities consider the software as part of the airborne system or equipment installed on the certified product; that is, the certification authorities do not certify the software as a unique, stand-alone product.*
>
> *Systems and equipment, including embedded software, should be "approved" in order to be accepted as a part of a certification. Approval by the certification authorities is given dependent upon a successful demonstration or by review of the products of the software lifecycle. Any such approval currently has significance only within the context of a specific certification."*

Therefore, software approval is based on the documents produced to support the software lifecycle specified in this standard.

ISO 26262's approach is very similar to DO-178C. It requires the compilation of all work products from the system safety engineering process specified in the standard to be used to build up a "safety case." The safety case is then used to demonstrate the adherence to the ISO 26262 process.

Although the definition of "safety case" in ISO 26262 is appropriate in that context, the use of this term in broader sense can lead to misunderstandings. In system safety engineering, the term "safety case" is often used to denote a quantitative argument that the system will be acceptably safe in a given operating context [43]. Leveson states that trying to prove a system is safe can lead us into the trap of confirmation bias, even though no one may be able to prove that a system is completely safe [44]. A term such as "lifecycle document" used in DO-178C may be considered less confusing.

ISO 26262 Part 2 Table 1 describes the required confirmation review for various ASILs. In particular, it requires "independence with regard to the developers of the item, project management and the authors of the work product [7]." This suggestion has merits, but in reality it may be hard to execute. Reviewers truly independent from the design may also be inexperienced with the system design, and hence may not be able to identify potential safety risks when presented with the documents and analysis. Confirmation bias is human nature, and thus the development team will usually present strong supporting evidence that the safety goal has been achieved [44]. Facing such evidence, truly independent reviewers may not be able to find the unknown-unknowns [45] if they are not deeply ingrained in the technology and design. Furthermore, a company may not have many technical experts in each area. These experts may be assigned to assist multiple projects in the company. A truly independent reviewer may be hard to find. Alternatively, a potentially more effective way to enforce safety may be to let the engineering team ensure that solid systems engineering best practices are used in all stages of the design process (hazard analyses, risk assessment, requirements definition, etc.).

4 CONCLUSIONS

The results of the standards comparison by each of the 11 factors are summarized in the Appendix. This assessment serves as a basis for the following observations on functional safety approaches:

1. Process safety standards that follow a systems engineering approach are different than FMVSSs and complement existing standards for safety assurance.
2. Existing process standards could be enhanced by providing a precise definition of "unreasonable risk" within the context of automotive safety.
3. *Hazard* definitions vary across different standards.
4. S*everity* alone can be used as the risk measure for software, similar to the approach outlined in DO-178C. Further, in cases when statistically valid failure probability or the probability of the occurrence of a mishap is not available, *severity* could be used as the only measure.
5. *Exposure* and *controllability* assessment used by the industry, as defined in the ISO 26262 standard, could be enhanced with the collection of additional data through design of specific experiments.
6. Existing process standards for software design could be enhanced with consideration for the overall safety of the control systems and software safety certification, in addition to the focus on specific aspects of the design solution (i.e., good architecture and coding standard).
7. Design-for-safety approach as specified in MIL-STD-882E provides a framework that could be leveraged for separate management of hazard tracking/safety requirements from regular system requirements, simpler risk assessment, and more emphasis on human factors.
8. The topic of health hazard analysis for drivers and service technicians could be further assessed for the appropriateness of including this topic in a process standard.
9. Existing process standards do not explicitly address environmental impacts on a vehicle throughout its lifecycle, including testing, manufacturing, operation, maintenance, etc.
10. Human factors studies could be better integrated into a comprehensive functional safety approach.

5 REFERENCES

[1] Center for Automotive Research. (2011). *Automotive technology: Greener vehicles, changing skills - Electronics, software & controls report*. Washington, DC: Employment and Training Administration. Available at www.drivingworkforcechange.org/reports/electronics.pdf

[2] Transportation Research Board. (2012). *The safety challenge and promise of automotive electronics: Insights from unintended acceleration* (TRB Special Report 308). Washington, DC: National Academies Press.

[3] Slater, R., Augustine, N., Goldman, P., Good, M., Martin, R., O'Neill, B., & Widnall, S. (2011). *A road forward: The report of the Toyota North American Quality Advisory Panel*. New York: Toyota Motor North America. Available at www.safetyresearch.net/Library/Toyota_Quality_Report.pdf

[4] Leveson, N. G. (2012). *Engineering a safer world*. Cambridge, MA: MIT Press.

[5] Testimony of The Honorable David L. Strickland, Administrator, National Highway Traffic Safety Administration, Senate Committee on Commerce, Science, and Transportation. Hearing on *"The Road Ahead: Advanced Vehicle Technology and Its Implications,"* May 15, 2013. Available at www.commerce.senate.gov/public/_cache/files/566d5c52-be38-4245-ab44-1ccbf2f198db/F77B78433264822E2A26AAA04B89B650.strickland.pdf

[6] 79 FR 60574, Request for Comment on Automotive Electronic Control Systems Safety and Security..National Highway Traffic Safety Administration, Automotive Electronic Control Systems Safety and Security, October, 7, 2014. Available at https://federalregister.gov/a/2014-23805.

[7] ISO 26262 Road Vehicles - Functional Safety, Final Draft (FDIS), 2011. Geneva.

[8] MIL-STD-882E: Department of Defense Standard Practice: System Safety, 2012. Available at www.system-safety.org/Documents/MIL-STD-882E.pdf

[9] DO -178C: Software Considerations in Airborne Systems and Equipment Certification. (2011). Washington DC: Radio Technical Commission for Aeronautics, Inc., & the European Organisation for Civil Aviation Equipment.

[10] National Highway Traffic Safety Administration. (1999). Federal Motor Vehicle Safety Standards (Online). Washington, DC: Author. Available at www.nhtsa.gov/cars/rules/import/FMVSS/index.html.

[11] AUTOSAR: Automotive Open System Architecture, (Online]. Available: www.autosar.org.

[12] Motor Industry Software Reliability Association. (2004). *MISRA-C: Guidelines for the use of the C language in critical systems*. Warwickshire, UK: Author.

[13] Motor Industry Software Reliability Association. (2007). *Guidelines for the application of MISRA-C: 2004 in the context of automatic code generation*. Warwickshire, UK: Author.

[14] Murray, B. (2011, March 4). *Software safety assurance processes and challenges in the automotive and aviation industries*. (PowerPoint presentation). East Hartford, CT: United Technologies Research Center. Available at http://onlinepubs.trb.org/onlinepubs/UA/030411Murray.pdf

[15] Czerny, B., D'Ambrosio, J., Jacob, P., & Murray, B. (2003). *Identifying and understanding relevant system safety standards for use in the automotive industry.* (SAE Technical Paper 2003-01- doi:10.4271/2003-01-1293..

[16] International Electrotechnical Commission. (2010). IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems, Part 4, Definitions and abbreviations, Edition 2.0. Geneva: Author. Available at https://webstore.iec.ch/preview/info_iec61508-4%7Bed2.0%7Db.pdf

[17] Van Eikema Hommes, Q. D. (2012). *Review and assessment of the ISO 26262 draft road vehicle - Functional safety*. (SAE Technical Paper 2012-01-0025). doi:10.4271/2012-01-0025.

[18] JASPAR General Incorporated Association. (2013). Handbook for Functional Safety (Microcontroller Application Edition] ver. 1.0.0. Publication information not available.

[19] McAllister, B., & Turner, J. (2005). Evolution of MIL-STD-882E. (PowerPoint presentation. Wright-Patterson Air Force Base, OH: Air Force Materiel Command. Available at www.dtic.mil/ndia/2005systems/wednesday/mcallister.pdf

[20] Smith, R. E. (2008, October 22). Update on Revisions of MIL-STD-882. (PowerPoint presentation at NDIA 11th Annual Systems Engineering Conference System Safety – ESOH & HSI Session 3C4, San Diego, CA). Tysons Corner, VA: Booz Allen Hamilton Inc. Available at www.dtic.mil/ndia/2008systems/RobertSmith.pdf

[21] Vesely, W. E., Goldberg, F. F., Roberts, N. H., & Haasl, D. F. (1981). *Fault tree handbook*. (Publication No. NUREG-0492). Washington, DC: U.S. Nuclear Regulatory Commission. Available at http://pbadupws.nrc.gov/docs/ML1007/ML100780465.pdf

[22] International Electrotechnical Commission. (2006). IEC 61025: Fault Tree Analysis, Second Edition. Geneva: Author. Available at https://webstore.iec.ch/preview/info_iec61025%7Bed2.0%7Den_d.pdf

[23] Leveson, N. G. (1995). *Safeware: System safety and computers*. Boston: Addison-Wesley Publishing Company.

[24] Kletz, T. (1999). *HAZOP and HAZAN: Identifying and assessing process industry hazards*, 4th Edition. Philadelphia: Taylor and Francis.

[25] International Electrotechnical Commission. (2001). IEC 61882: Hazard and Operability Studies (HAZOP Studies)--Application Guide. Geneva: Author.

[26] Pan, J. (1999). *Software reliability*. Pittsburgh, PA: Carnegie Mellon University. Available at http://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/

[27] Lyu, M. R. (1995). *Handbook of software reliability engineering*. New York: McGraw-Hill Publishing.

[28] Gopal, M. (1993). *Modern control system theory*. New York: John Wiley & Sons.

[29] Goodwin, G. C., Graebe, S. F., & Salgado, M. E. (2001). *Control system design* Upper Saddle River, NJ: Prentice Hall.

[30] Bateson, R. (1999). *Introduction to control system technology*. Upper Saddle River, NJ:Prentice Hall.

[31] Ozbay, H. (2000). *Introduction to feedback control theory*. Boca Raton, FL: CRC Press.

[32] Ishimatsu, T. N., Leveson, N., Thomas, J., Katahira, M., Miyamoto, Y., & Nakao, N. (2010). Modeling and Hazard Analysis Using STPA, in Fourth Global Conference on Space Safety, *International Association for the Advancement of Space Safety*, Huntsville, AL, May 19-21, 2010.

[33] Balgos, V. H. (2012). A System Theoretic Application to Design for the Safety of Medical Diagnostic Devices. (Master's degree thesis). Cambridge, MA: Massachusetts Institute of Technology.

[34] Hickey, J., & Van Eikema Hommes, Q. D. (2013). Effectiveness of accident models: System theoretic model vs. the Swiss cheese model: A case study of a US Coast Guard aviation mishap, *International Journal of Risk Assessment and Management, 17*, no. 1.

[35] Van Eikema Hommes, Q. D. (2012). Applying System Theoretic Hazard Analysis Method to Complex Automotive Cyber Physical Systems (IDETC 2012-70527) In *ASME 2012 International Design Engineering Technical Conferences & Computer and Information in Engineering Conference*, Chicargo, IL, August 12-15, 2012.

[36] Goerges, S. L., and Van Eikema Hommes, Q. D. (2014). System Theoretic Approach for Determining Causal Factors of Quality Loss in Complex System Design . In *DETC2014-34156, Proceedings of the ASME 2014 International Design Engineering Technical Conferences, IDETC/CIE 2014, August 17-20*, Buffalo, NY, 2014.

[37] Kahneman, D. (2013). *Thinking, fast and slow*. New York: Farrar, Straus, and Giroux.

[38] Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach*. Mahwah, N. J.: Lawrence Erlbaum Associates.

[39] Starter, N. B., & Woods, D. D. (1995). *How in the world did we ever get into that mode? Mode error and awareness in supervisory control*. *Human Factors, 37*, no. 1, pp. 5-19.

[40] Norman, D. (2002). *Design of everyday things* New York: Basic Books.

[41] International Council for Systems Engineering (INCOSE), INCOSE Systems Engineering Handbook, V 3.2.2, INCOSE-TP_2003-002-03.2.2, INCOSE, 2011.

[42] Blanco, M., Atwood, J., Vasquez, H. M., Trimble, T. E., Fitchett, V. L., Radlbeck, J., ... & Morgan, J. F. (2015, August). Human factors evaluation of level 2 and level 3 automated driving concepts. (Report No. DOT HS 812 182). Washington, DC: National Highway Traffic Safety Administration.

[43] Kelly, T. (1998). Arguing Safety - A Systematic Approach to Managing Safety Cases, (Ph.D. thesis). York, UK: University of York.

[44] Leveson, N. (2011). *The use of safety cases in certification and regulation*. *Journal of System Safety,* Nov/Dec 2011.

[45] U.S. Department of Defense. (2002, February 12). DOD News Briefing - Secretary Rumsfeld and General Myers. (Website no longer available).

APPENDIX: SUMMARY OF STANDARDS COMPARISON

This appendix summarizes the results of comparing the five standards (MIL-STD-882E, DO-178C, ISO 26262, AUTOSAR, and MISRA C) by each of the 11 factors listed in the first column.[12]

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Type of Standard	Process	Process	Process and method	Design (architecture)	Design (coding)
Definition of Safety	Freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.	No clear definition provided. Based on Failure Condition Category (Figure 37), the definition is similar to MIL-STD-882E.	Absence of unreasonable risk.	Same as ISO 26262.	No explicit definition.

[12] The results of FMVSS are not included (side-by-side) in this comparison table because things like the definition of "motor vehicle safety" and the way that the process incorporates hazard analyses just aren't used in the same context as these process standards. The process standard generally establishes some definition or method for things like safety or hazard; then, it asks the analyst/engineer to apply those to their specific product (then develop a mitigation strategy). This is very different from FMVSS where NHTSA is making those determinations to develop a performance metric that is generally applicable to everyone's products.

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Definition of Hazard	A real or potential condition that could lead to an unplanned event or series of events (i.e., mishap) resulting in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.	No definition provided.	Potential source of harm caused by malfunctioning behavior of the item Malfunctioning Behavior: failure or unintended behavior of an item with respect to its design intent.	Same as ISO 26262.	No explicit definition.

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Identification of Safety Requirements	Functional hazard analysis on predefined nominal system functions is used to identify safety hazards.	Assumes software requirements are flowed down from system-level activities, and provides no guidance on how to further decompose the requirements and identify additional safety-critical requirements at each level of the system decomposition hierarchy.	At high level of the system, use hazard analysis method. At lower levels of the system, use safety analysis methods.	Focused on architecture design requirements. Depend on ISO 26262 to identify safety requirements.	Focused on coding standard.

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Hazard and Safety Analysis Methods	Functional hazard analysis on predefined nominal system functions is used to identify safety hazards.	Not discussed.	Hazard Analysis includes: brainstorming, checklists, quality history, Failure Models and Effects Analysis, and field studies. Safety Analysis include: Failure Modes and Effects Analysis, Fault Tree Analysis, Event Tree Analysis, and Hazard and Operability Analysis, Markov Model, Reliability Block Diagram, etc.	Not discussed.	Not discussed.
Management of Safety Requirements	Hazard Tracking System used separately from other requirements management system.	Safety requirements and regular system requirements are managed together.	Safety requirements and regular system requirements are managed together.	Not discussed.	Not discussed.

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Risk Assessment	For general system: Severity and Probability of occurrence For software: Severity and Software Control Category (no probability) Uses the term of "acceptable risk," but mentions user involvement in decision. Accepts both qualitative and quantitative probability assessment.	Only considers severity, no probability assessment. Hazards are considered to be caused by software behavior inconsistent with specified requirements, assuming all safety requirements are identified already.	Uses three dimensions—Severity, Exposure, and Controllability to generate ASIL-Automotive Safety Integrity Level. Uses the term of "acceptable risk," without sufficiently precise definition. Accepts both qualitative and quantitative probability assessment. Suggests ASIL decomposition for all hardware and software.	Not applicable.	Not applicable.

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Design for Safety Approach	Thoroughly discussed starting with prevention and elimination.	Not explicitly discussed.	No explicit discussion. Focus on *safety mechanism.*	Good architecture may prevent hazards.	Good coding practice will reduce errors in software.
Software Safety	Same as Hardware Development Process, following the systems engineering process.	Follows the systems engineering process. Uses the concept of *Software Anomaly*.	Follows the systems engineering process. Uses the concept of *Software Fault*.	Not applicable.	Coding standards.
System Lifecycle Consideration	Prompts considerations for various aspects after system is in operation, but does not mention safety considerations for the manufacturing process.	Focuses on software lifecycle considerations such as coding, configuration management, etc.	Considers lifecycle of the system, including manufacturing, but does not explicitly discuss the safety of human operators and maintainers, and the environmental hazard.	Focuses on reusability, modifiability.	Not applicable.

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Human Factors Considerations	Emphasized throughout the standard.	Not discussed.	Not emphasized. Only controllability assessment in ASIL relates to human factors.	Not discussed.	Not discussed.

	MIL-STD-882E	DO-178C	ISO 26262	AUTOSAR	MISRA C
Review, Audit, and Certification	Supports government reviews, audits, and boards, but details to be specified in Request for Proposal and Statement or Work for each project. Specifics are left up to the program manager and contractor.	The certification authorities consider the software as part of the airborne system or equipment installed on the certified product, and do not certify the software as a unique, stand-alone product. Systems and equipment, including embedded software, should be "approved" and then accepted as a part of a certification. Approval is given dependent upon successful demonstration or review of products of the software lifecycle.	Independent confirmation reviews, safety audits, and safety assessments are required for various ASIL. No safety certification requirement.	Not for safety approval or certification.	Supports software quality system by contributing to the documentation; but, not used for software safety certification.

DOT HS 812 285
June 2016

U.S. Department
of Transportation

**National Highway
Traffic Safety
Administration**

12358-060216-v2