DRAFT INTERNATIONAL STANDARD

ISO/DIS 24089

ISO/TC **22**/SC **32**

Secretariat: **JISC**

Voting begins on:
2022-01-11

Voting terminates on:
2022-04-05

Road vehicles — Software update engineering

Véhicules routiers — Ingénierie de mise à jour du logiciel

ICS: 43.040.15

This document is circulated as received from the committee secretariat.

Reference number
ISO/DIS 24089:2022(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road Vehicles*, Subcommittee SC 32, *Electrical and electronic components and general system aspects*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Road vehicles have been increasing their use of electronic control systems in recent years. With this has come increasing complexity of software in vehicles. As a result, software has become essential to the operation of road vehicles. This software is often updated to increase functionality and maintain the safety and cybersecurity of road vehicles. Therefore the establishment and application of software update engineering are important to ensure software quality, cybersecurity, and road vehicle safety.

Today, in-vehicle software is updated by skilled persons using specialized tools and equipment or by remote software updates. With the increased frequency of software updates it is important to have accurate current configuration information for individual vehicles to ensure software quality, cybersecurity, and road vehicle safety.

This document assigns accountability for the processes required to update software safely and securely, provides the fundamental requirements for performing those processes correctly, and producing high quality software update packages. By applying the software update engineering requirements and recommendations in this document, the following benefits can be expected:

- Safe and secure software update operations in road vehicles;
- Establishment of clear processes, including explicit goal setting, planning, auditing, process monitoring, process measurement, and process improvement;
- Shared awareness of safety and cybersecurity among related parties;
- Establish trust that software update engineering activities are based on clear and controlled processes.

Figure 1 shows the overview of this document.

Clause 4 and clause 5 define necessary organizational rules and processes.

Clause 6 and clause 7 define the software update capabilities for infrastructure and vehicles.

Clause 8 defines creation of software update packages.

Clause 9 defines the preparation and execution of software update campaigns.

In this document, clauses are structured using following idea,

- Process should be defined before actual operation
- Set of Processes should be managed as assets, i.e. documented and maintained
- Processes should be managed in higher level than each project, in order to reuse or refer established processes in other project.

<table>
<tr><td colspan="4">1. Scope</td></tr>
<tr><td colspan="4">2. Normative references</td></tr>
<tr><td colspan="4">3. Terms and definitions</td></tr>
<tr><td colspan="4">4. Organization level software update requirements</td></tr>
<tr><td colspan="4">5. Project level software update requirements</td></tr>
<tr><td>6. Infrastructure design and development</td><td>7. Vehicle and vehicle systems design and development</td><td>8. Software update package development</td><td>9. Software update campaign operations</td></tr>
</table>

Figure 1 – Overview of this document

Software update engineering activities occur throughout the life cycle of the vehicle.

It is important to apply software quality assurance, cybersecurity, and safety processes to software update engineering activities.

Road vehicles – Software Update Engineering

1 Scope

This document specifies requirements and recommendations for software update engineering in road vehicles on both the organizational and project levels.

These requirements and recommendations apply to the vehicle, its systems, and/or the infrastructure. Additionally, these requirements and recommendations apply to ECUs installed in road vehicles, and the creation of software update packages after the original development.

In addition, this document specifies requirements and recommendations for deployment of software update packages to road vehicles.

The development of software for vehicle functions, except for software update engineering, is outside the scope of this document.

This document enables a common understanding for communicating and managing activities and responsibilities among stakeholders.

This document is applicable to road vehicles that include ECUs whose software can be updated.

This document applies to organizations involved in software update engineering in road vehicles. Such organizations can include OEMs, suppliers, and their subsidiaries or contractual partners.

This document does not prescribe specific technologies or solutions for software update engineering.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 26262-6, *Road Vehicles--- Functional Safety --- Part 6: Product development at the software level*

ISO 26262-8, *Road Vehicles--- Functional Safety --- Part 8: Supporting processes*

ISO/SAE 21434, *Road vehicles --- Cybersecurity Engineering*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1 General terminology

3.1.1
corrective action
action to eliminate or contain a problem or failure

3.1.2
cybersecurity
road vehicle cybersecurity
condition in which assets are sufficiently protected against threat scenarios in road vehicles, vehicle systems and/or ECUs and infrastructure supporting software update engineering.

Note 1 to entry: In this document, for the sake of brevity, only the term cybersecurity is used.

[SOURCE: ISO/SAE 21434: modified - definition aligned to terms used in this document, added infrastructure, removed Note 1 to entry]

3.1.3
cybersecurity risk
effect of uncertainty on road vehicle cybersecurity expressed in terms of attack feasibility and impact

[SOURCE: ISO/SAE 21434]

3.1.4
electronic control unit
ECU
embedded device in a vehicle whose software can be updated

Note 1 to entry: Replacing an ECU can be a method of updating its software.

Note 2 to entry: ECUs can be implemented on a larger computing device as virtual ECUs.

3.1.5
functional safety risk
combination of the probability of occurrence of harm and the severity of that harm

[SOURCE: ISO 26262-1]

3.1.6
infrastructure
services that manage software update operations, software update campaigns, documentation and/or vehicle configuration information, including both digital and manual activities

Note 1 to entry: Infrastructure can include servers, tools, and/or manual activities used in the software update operation.

Note 2 to entry: Function(s) of the infrastructure can be implemented in the vehicle and/or externally.

3.1.7
in-vehicle resources
vehicle or ECU available properties relevant for software update engineering

Note 1 to entry: This can include available or remaining computational power, network capacity, RAM capacity, storage capacity and/or battery capacity

3.1.8
over the air
OTA
method for wireless delivery of a software update package

3.1.9
recipient
individual instance of a vehicle or component that receives a software update package during a software update campaign

3.1.10
safe vehicle state
vehicle operating mode based on conditions for performing software update campaigns safely

Note 1 to entry: Safe vehicle state can be different depending on the conditions required for the software update package.

3.1.11
skilled person
trained person, who is able to execute software update operations requiring specialised training

Note 1 to entry: A skilled person can include a mechanic in a workshop.

Note 2 to entry: A skilled person can be authorized or certified for their specialised training or be a skilled end user.

3.1.12
software
computer programs and associated data that may be dynamically written or modified during execution

[SOURCE: NIST SP 800-53]

3.1.13
software update campaign
sequence of identifying targets; validating, distributing, downloading, installing, and activating software update packages; and monitoring and documenting changes and results

Note 1 to entry: The software update campaign can include corrective or functional changes, such as upgrading, downgrading, and/or modifying feature-related configuration information.

3.1.14
software update engineering
application of a systematic, disciplined, quantifiable approach to the development and processing of software update packages

NOTE Development of the software included in the software update package is not part of software update engineering.

[SOURCE: IEEE 610.12-1990, modified]

3.1.15
software update method
mechanism for distribution of a software update package during a software update campaign

Note 1 to entry: The software update method can be wired (e.g. tool, USB flash drive), wireless (e.g. cellular or Wi-Fi) or hardware replacement.

3.1.16
software update operation
steps involved in the download, installation and activation of software update packages on a vehicle

3.1.17
software update package
set of software and associated metadata that is intended to be delivered to the target vehicles

Note 1 to entry: A software update package can contain updates for different ECUs.

3.1.18
software update project
set of software update engineering activities for one or more targets

Note 1 to entry: Target can refer to a class or model of vehicle or ECU.

3.1.19
software update support
capability to perform a software update campaign

3.1.20
tailoring
process by which individual requirements in specifications, standards, and related documents are evaluated and made applicable to one or more projects by selection, and in some exceptional cases, modification of existing or addition of new requirements

[SOURCE: ISO 27025]

3.1.21
target
class of vehicle or component defined by combination of hardware and/or software versions during software update campaign planning

3.1.22
update
<verb> bring up to date

3.1.23
vehicle configuration information
comprehensive accounting of hardware versions, software versions, and configuration parameters in a vehicle

[SOURCE: NIST SP 800-32, modified – added 'in a vehicle' for the scope of this document.]

3.1.24
vehicle state
current vehicle operating mode

EXAMPLE parked, stationary, driving, engine off

3.1.25
vehicle system
functional group of one or more ECUs and attached hardware that is not itself an ECU

Note 1 to entry: Attached hardware can be a sensor, actuator, light, etc.

Note 2 to entry: For example a braking system or infotainment system.

3.1.26
vehicle user
person operating, driving, owning or managing a vehicle

Note 1 to entry: Vehicle user can include skilled person.

3.2 Terminology for the software update operation

3.2.1
download
step in the software update operation when a vehicle receives a software update package

3.2.2
installation
step in the software update operation when the relevant parts of a software update package are installed on a recipient ECU but are not yet activated

3.2.3
activation
step in the software update operation when the relevant parts of an installed software update package are activated and become available for use on a recipient ECU

EXAMPLE 1 A new autonomous function is installed and set to active, but might run only after the vehicle user starts the function.

EXAMPLE 2 A software update package for a recipient ECU is installed, activated, and run immediately after activation without user interaction.

4 Organization level software update requirements

4.1 Objectives

The objectives of this clause are to ensure that the following are performed:

a) Establishment of organization-specific rules, processes, and procedures for software update engineering;

b) Adoption of quality management, functional safety, and cybersecurity management for software update engineering;

c) Instituting and maintaining a continuous improvement process for software update engineering;

d) Provisioning of the necessary resource requirements, information sharing and management of dependencies within the organization and with other related parties;

e) Performing an organizational audit for process compliance.

4.2 General

This clause covers the responsibility of the organization engaged in software update engineering to have governance in place so that the processes for software update engineering can conform to the requirements of this document. Governance includes compliance with required ISO standards as well as organizational activities such as continuous improvement, information sharing and supporting processes. This clause also establishes auditing requirements for this document.

4.3 Requirements and recommendations

4.3.1 Governance

4.3.1.1 If software update engineering applies, then changes in software in a vehicle or ECU shall be done in accordance with the requirements of this document.

EXAMPLE 1 Replace an ECU with the purpose of updating software.

4.3.1.2 The organization shall establish, document and maintain rules and processes for software update engineering to:

- enable the implementation of the requirements of this document;

- support the execution of the corresponding activities, including the assignment of resources and responsibilities across all those involved in the software update engineering activities;

- confirm compliance with the requirements of this document.

EXAMPLE Process definition, technical rules, guidelines, methods and templates.

NOTE These rules and processes cover vehicle system, and ECU which are affected by software update engineering activities, and infrastructure used for software update engineering activities.

4.3.1.3 In order to fulfil the requirements of this document, software update engineering shall conform with:

- ISO/SAE 21434 Road vehicles — Cybersecurity engineering;

- ISO 26262-6 Road Vehicles - Functional Safety – Part 6: Product development at the software level;

- ISO 26262-8 Road Vehicles - Functional safety - Part 8: Supporting processes.

4.3.2 Continuous improvement

4.3.2.1 The organization shall establish, perform, and maintain a continuous improvement process for software update engineering activities.

EXAMPLE 1 Applying lessons learned from previous or similar software update projects, field monitoring and observations.

EXAMPLE 2 Applying improvements to software update engineering processes based on experience

EXAMPLE 3 Communicating lessons learned appropriately within the organization

EXAMPLE 4 Evaluating the adequacy of current processes

4.3.2.2 The organization shall establish, maintain and perform a process to verify that after any change to its software update engineering processes, the process meets the requirements of this document.

4.3.3 Information sharing

4.3.3.1 The organization shall establish, perform and maintain a policy for sharing information inside and outside the organization concerning software update engineering activities.

NOTE The policy can include what information is shared, with whom the information is shared, when the information is shared, and how to permit sharing.

EXAMPLE Information being shared can include:

- update schedule;

- content description;

- possible implication of the software update campaign including safety or cybersecurity-relevant items;

- time the vehicle or its functions are unavailable during a software update campaign;

- reason for the software update campaign;

- treatment of sensitive or personal information;

- documentation about the software update campaign;

- license and intellectual property information

4.3.4 Auditing

4.3.4.1 An independent audit shall be performed on whether the organizational process for software update engineering achieves the objectives of this document.

NOTE 1 Such an audit can be included in, or combined with, an audit according to a quality management system standard.

EXAMPLE In a distributed development, right to audit can be included in contract.

NOTE 2 The person that performs the audit can be internal or external to the organization.

NOTE 3 To ensure the organizational processes remain appropriate for software update engineering, an audit can be performed periodically.

4.3.5 Supporting processes

4.3.5.1 The organization shall establish document management to handle the work products required by this document.

NOTE IATF 16949 can be applied.

4.3.5.2 The organization shall establish, implement and maintain a requirements management system for software update engineering activities.

4.3.5.3 The organization should consider privacy implications of the activities required by this document.

NOTE 1 Information on privacy can be found in ISO/IEC 27001 and ISO/IEC 27002.

NOTE 2 Activities in this document can involve personal information.

EXAMPLE 1 Customer ID included in software update campaign.

4.3.5.4 The organization shall establish, implement and maintain a configuration management process.

NOTE 1 Software update engineering activities involve configuration information for software update packages, vehicles and infrastructure.

EXAMPLE 1 ISO 10007 can be used for configuration management systems

EXAMPLE 2 ISO 15288 can be applied for configuration management on system life cycle management.

4.3.5.5 The organization shall establish, implement and maintain a quality management process for software update engineering activities.

EXAMPLE IATF 16949 and ISO 9001 can be used for quality management.

4.3.5.6 The organization shall establish, implement, and maintain a change management process for software update engineering activities.

EXAMPLE ISO 9001 and ISO 27001 can be used for change management.

4.4 Work products

4.4.1 Organizational rules and processes resulting from the requirements of 4.3.1.1, 4.3.1.2, 4.3.5.1, 4.3.5.3, 4.3.5.4, and 4.3.5.5.

4.4.2 Record of organizational management resulting from the requirements of 4.3.1.3, 4.3.5.2 and 4.3.5.5.

4.4.3 Documentation of continuous improvement resulting from the requirement of 4.3.2.1 and 4.3.2.2.

4.4.4 Information sharing policy resulting from the requirement of 4.3.3.1

4.4.5 Audit report, resulting from the requirement of 4.3.4.1

5 Project level software update requirements

5.1 Objectives

The objectives of this clause are to ensure that the following are performed:

a) Planning for a software update project, including assigning roles and responsibilities;

b) Managing and storing of information regarding a software update project;

c) Providing justifications for any tailoring of a software update project;

d) Confirming interoperability of the infrastructure and the vehicle capabilities for a software update project.

5.2 General

This clause covers the requirements to the organization for the software update projects including the planning for software update projects, and managing information related to the software update projects. In addition, this clause includes requirements on tailoring of the software update projects and interoperability between the parts of the software update projects.

5.3 Requirements and recommendations

5.3.1 Project management

5.3.1.1 The organization shall develop a plan for each software update project that covers all necessary activities.

NOTE 1 This plan can include activities for developing and/or adapting the infrastructure, vehicle capabilities and/or processes described in this document.

NOTE 2 A software update project can encompass multiple software update campaigns.

EXAMPLE 1 A software update project can be for one vehicle model or for one type of ECU.

5.3.1.2 The organization shall establish, implement, and maintain a process to manage and store records for each software update project.

5.3.1.3 The organization shall assign and document the roles and responsibilities for each software update project.

NOTE Documentation can be in the software project update plan required in 5.3.1.1

5.3.2 Tailoring and rationale

5.3.2.1 A software update project may be tailored.

EXAMPLE Management of functional safety risks in the context of ISO 26262.

5.3.2.2 If a software update is tailored, then a rationale shall be provided as to why the tailored activities adequate and sufficient to achieve the applicable objectives of this document.

NOTE 1 An activity is tailored if it is omitted or performed in a different manner compared to its description in this document.

EXAMPLE Activities that are not performed because they are performed by another entity in the supply chain are not considered as tailored, but as distributed cybersecurity activities (see ISO/SAE 21434 for a definition of Distributed Activities – Clause 15).

NOTE 2 Organizations can consult with their suppliers on tailoring of activities.

NOTE 3 A bodywork equipment builder can tailor a software update project to conform with functional safety standards such as ISO 13849 and/or IEC 61508.

5.3.3 Interoperability

5.3.3.1 The organization shall establish, implement and maintain a process to confirm the interoperability of the functions and capabilities developed under the requirements from Clause 6 and Clause 7.

NOTE Since the infrastructure and the vehicle system can be implemented separately, it is important to confirm the interoperability between them to have a successful software update campaign.

5.3.3.2 The organization shall establish, implement, and maintain processes to preserve the integrity of software, metadata and/or software update packages during transmission:

- within the infrastructure of organizations;

- between organizations within the supply chain;

- from organizations to vehicles;

- within the vehicle and between ECUs.

NOTE 1 The organization transmitting software to vehicles could be OEMs, suppliers or other contracted entities.

NOTE 2 Controls for integrity can be determined using the risk-based approach in ISO/SAE 21434.

5.4 Work products

5.4.1 Software update project plan resulting from the requirements of 5.3.1.1 and 5.3.1.3

5.4.2 Records of software update project resulting from the requirement of 5.3.1.2.

5.4.3 Rationale, resulting from the requirement of 5.3.2.2.

5.4.4 Documentation of confirmation of interoperability resulting from the requirements of 5.3.3.1 and 5.3.3.2.

6 Infrastructure design and development

6.1 Objectives

The objectives of this clause are to ensure that the following functionality is in the infrastructure:

a) Managing cybersecurity risks in the infrastructure;

b) Collecting and managing vehicle configuration information in the infrastructure;

c) Collecting and distributing information about software update campaigns in the infrastructure;

d) Creating, managing, and distributing software update packages in the infrastructure;

e) Managing failures during software update campaigns in the infrastructure.

6.2 General

This clause includes the requirements for the development of infrastructure that is used for software update campaigns. The requirements cover the functions that are assigned to the infrastructure for the software update campaigns, such as distribution, communication, cybersecurity and information storage. Software update functions described in this clause support the software update campaigns on the infrastructure. Such functions can be on or off vehicle depending on the architectural decisions of the organization.

6.3 Requirements and recommendations

6.3.1 Managing risk

6.3.1.1 Cybersecurity risks of software update campaigns in the infrastructure shall be managed.

EXAMPLE 1 ISO/IEC 27000 series of standards provides guidance on management of cybersecurity risk.

EXAMPLE 2 ISO/SAE 21434 provides guidance on management of cybersecurity risk for the vehicle.

6.3.2 Managing vehicle configuration information

6.3.2.1 The infrastructure shall have one or more functions for receiving, storing, and processing of the current vehicle configuration information.

6.3.2.2 The infrastructure shall have one or more functions to distribute vehicle configuration information to related parties.

NOTE 1 Related parties could be regulatory entities or suppliers.

NOTE 2 Vehicle configuration information can also be distributed by manual operations, such as paper-based.

NOTE 3 Vehicle configuration information can be distributed at any time.

6.3.2.3 The infrastructure shall have one or more functions to support the identification of dependencies of a software update package on other systems.

6.3.3 Communicating software update campaign information

6.3.3.1 The infrastructure shall have one or more functions to provide notifications as required by this document.

NOTE 1 This function can be used to notify vehicle users in lieu of an in-vehicle notification function.

NOTE 2 See requirements concerning notification in Clause 9.

6.3.3.2 The infrastructure shall have one or more functions for receiving, storing, processing, and distributing results of software update campaigns.

6.3.4 Processing software update packages

6.3.4.1 The infrastructure shall have one or more functions for creating, processing, receiving, storing, and distributing software update packages.

6.3.4.2 The infrastructure shall have one or more functions to correlate software update packages to targets.

6.3.4.3 The infrastructure shall have one or more functions to identify recipients based upon targets of a software update campaign.

6.3.4.4 The infrastructure should have one or more functions to determine whether there are sufficient in-vehicle resources to apply the software update package.

6.3.5 Managing failure during software update campaigns

6.3.5.1 The infrastructure should have one or more functions to initiate actions when the infrastructure is notified of a failure of a software update operation.

EXAMPLE Infrastructure sends notice of failure to dealership or local mechanic to pick up the vehicle.

NOTE The software update operation failure can be mitigated by actions of the vehicle and/or externally.

6.4 Work products

6.4.1 Documentation of cybersecurity risk management resulting from 6.3.1.1.

6.4.2 Documentation of functions for vehicle configuration information resulting from 6.3.2.1 to 6.3.2.3.

6.4.3 Documentation of functions for software update campaign resulting from 6.3.3.1 and 6.3.3.2.

6.4.4 Documentation of functions for software update packages resulting from 6.3.4.1 to 6.3.4.4.

6.4.5 Documentation of functions for performing actions in the event of software update operation failure resulting from 6.3.5.1.

7 Vehicle and vehicle systems design and development

7.1 Objectives

The objectives of this clause are to ensure that the following are developed:

a) Managing safety and cybersecurity risks for the vehicle and/or its ECUs;

b) Managing vehicle configuration information in the vehicle and/or its ECUs;

c) Managing information about the software update campaigns in the vehicle and/or its ECUs;

d) Enabling the software update operation and verifying software update packages;

e) Managing failures during software update campaigns in the vehicle and/or its ECUs.

7.2 General

This clause contains the requirements for the functions needed for vehicles and ECUs to support software update campaigns. These functions include communications, generating necessary vehicle information and enabling the download, installation and activation of software in vehicles.

Software update functions described in this clause support the software update operation in the vehicle. Such functions can be on or off vehicle depending on the architectural decisions of the organization.

7.3 Requirements and recommendations

7.3.1 Managing risks

7.3.1.1 Functional safety risks of the software update operation in the vehicle shall be managed.

NOTE 1 Management includes identification, analysis, evaluation and treatment of risks.

NOTE 2 ISO 26262 provides guidance on achieving functional safety through appropriate requirements and processes.

EXAMPLE 1 An OEM performs a functional safety risk assessment of a braking system and decides based on that assessment whether a skilled person is necessary to apply the update or not.

EXAMPLE 2 The interfaces between the vehicle and the bodywork equipment are defined in such a way that vehicle safety is not impacted by a software update operation.

7.3.1.2 Safety risks due to reasonable and foreseeable misuse of the software update operation in the vehicle shall be managed.

NOTE Management includes identification, analysis, evaluation and treatment of risks.

EXAMPLE 1 ISO/PAS 21448 provides guidance on achieving safety of the intended functionality through appropriate requirements and processes.

EXAMPLE 2 A measure is put in place to prevent unintentional installation and/or activation of software by a vehicle user while driving.

7.3.1.3 Cybersecurity risks of the software update operation in the vehicle shall be managed.

NOTE 1 ISO/SAE 21434 provides guidance on implementing cybersecurity engineering to manage risks.

NOTE 2 Cybersecurity risks include the risk that vehicle configuration information might be modified without authorization.

7.3.2 Managing vehicle configuration information

7.3.2.1 There shall be one or more functions to collect vehicle configuration information.

NOTE These functions can be implemented in the vehicle and/or in the infrastructure.

7.3.2.2 There shall be one or more functions to identify the ECUs to which a software update package applies.

NOTE These functions can be implemented in the vehicle and/or in the infrastructure.

7.3.3 Communicating software update campaign information

7.3.3.1 There shall be one or more functions to provide information to related parties as required by this document.

NOTE 1 These functions support notification requirements in Clause 9.

NOTE 2 These functions can be implemented in the vehicle and/or in the infrastructure.

7.3.3.2 There should be one or more functions to obtain the confirmation of the vehicle user for a software update operation.

NOTE 1 Confirmation can be obtained for each single instance of software update campaign or a general confirmation may be obtained at the beginning of the relationship between the vehicle user and the organization initiating a software update campaign.

NOTE 2 These functions can be implemented in the vehicle and/or in the infrastructure.

EXAMPLE A vehicle user confirmation could be obtained via:

- an in-vehicle display;
- a mobile application;
- a website;
- a contractual agreement.

7.3.4 Processing software update packages

7.3.4.1 There shall be one or more functions to determine that all pre-conditions, including in-vehicle resources, are met in order to download, install, and activate the software.

NOTE These functions can be implemented in the vehicle and/or in the infrastructure.

EXAMPLE 1 Available battery capacity and remaining charge are sufficient to perform the software update operation.

EXAMPLE 2 Check if the existing software version of an ECU is compatible with the software update package.

7.3.4.2 There shall be one or more functions to handle interruptions in communications during download.

NOTE These functions can be implemented in the vehicle and/or in the infrastructure.

7.3.4.3 There shall be one or more functions to verify the integrity and authenticity of the downloaded software update package before the activation.

NOTE 1 This verification can be done earlier than activation.

NOTE 2 These functions can be implemented in the vehicle and/or in the infrastructure.

EXAMPLE Signature verification can be used for the integrity and authenticity check.

7.3.4.4 There shall be one or more functions to ensure a safe vehicle state at the start of and during the software update operation.

NOTE 1 Safety impacts of the software update package are identified under Clause 8.

NOTE 2 Disabling or restricting features and functions can allow the software update operation to proceed safely.

NOTE 3 These functions can be implemented in the vehicle and/or in the infrastructure.

EXAMPLE 1 Safe vehicle state can be ensured by a skilled person in a workshop.

EXAMPLE 2 The software update operation can be paused or aborted, because a safe vehicle state cannot be maintained.

7.3.5 Managing failure during software update campaigns

7.3.5.1 There shall be one or more functions to ensure vehicle safety if the software update operation fails.

NOTE 1 These functions can be implemented in the vehicle and/or in the infrastructure.

NOTE 2 These functions can be the responsibility of the skilled person.

NOTE 3 These functions can be developed as a result of the implementation of 7.3.1.

EXAMPLE Safety measures can include:

- changing the vehicle operating mode to one in which the vehicle is safe;
- launching fall-back operations.

7.3.5.2 There shall be one or more functions to arbitrate simultaneous access requests to maintain vehicle safety.

NOTE 1 These functions can be implemented in the vehicle and/or in the infrastructure.

NOTE 2 Arbitration can be limitation, acceptance, or rejection of simultaneous access requests.

EXAMPLE 1 Requests can be received simultaneously from a wired tool and a wireless tool.

EXAMPLE 2 Simultaneous requests can include multiple wireless requests.

7.4 Work products

7.4.1 Documentation of risk management resulting from 7.3.1.1 to 7.3.1.3.

7.4.2 Documentation of functions for vehicle configuration information resulting from 7.3.2.1 and 7.3.2.2.

7.4.3 Documentation of functions for communications related to software update campaigns resulting from 7.3.3.1 and 7.3.3.2.

7.4.4 Documentation of functions for software update operations resulting from 7.3.4.1 to 7.3.4.4.

7.4.5 Documentation of functions for managing failures of software update operations resulting from 7.3.5.1 and 7.3.5.2.

8 Software update package development

8.1 Objectives

The objectives of this clause are to ensure that the following are performed:

a) Identifying the target(s) and contents of the software update package;

b) Assembling the software update package containing the necessary software component(s) and metadata for the target(s);

c) Verifying and validating the software update package;

d) Approving release of the software update package.

8.2 General

This clause includes requirements for assembling the software update package and verifying and validating the software update package's contents, as well as identifying the types of vehicles or systems to receive the software update package. Software update package development is the process of putting all necessary elements into a form for the software update operation at the vehicle level. The software update package is approved for release based on the performed verification and validation.

8.3 Requirements and recommendations

8.3.1 Identification of targets and the contents for the software update package

8.3.1.1 The organisation shall determine the list of the target(s) for each software update package.

NOTE In the case of suppliers, the target may be ECUs. In the case of OEMs, the target may be the vehicles or ECUs.

8.3.1.2 The software and associated metadata for the identified target(s) shall be selected for the software update package**.**

NOTE 1 In the case of suppliers, the software and associated metadata may be for a single ECU. For OEMs, the software and associated metadata may cover the vehicle or multiple ECUs.

NOTE 2 Software Update of engine systems may require different conditions if performed in the workshop or by the vehicle user.

EXAMPLE 1 Metadata can include:

- safe vehicle state;

- conditions;

- compatibility information;

- dependencies between systems and/or ECUs;

- version information and/or release information;

- necessary in-vehicle resources.

EXAMPLE 2 Conditions can be parked, engine off, or availability of vehicle functionality etc.

EXAMPLE 3 Software update of infotainment systems may not affect the safety of the vehicle.

8.3.1.3 Compatibility of the software update package with the existing software and hardware of the target shall be identified.

8.3.1.4 Dependencies of the software update package with the existing software and hardware of the target shall be identified.

8.3.1.5 Necessary in-vehicle resources and conditions in the target shall be identified.

EXAMPLE In-vehicle resources can be necessary to complete the software update operation or run the new software.

8.3.2 Assembly of the software update package

8.3.2.1 Only the intended software and metadata shall be assembled into the software update package.

EXAMPLE Ensuring that only software authorized for release is included in the package.

8.3.2.2 A software update package shall be created, which contains the necessary software and metadata.

8.3.2.3 The organization shall define a unique identifier of the software update package.

8.3.2.4 Necessary actions regarding cybersecurity and safety for the software update package shall be determined.

NOTE This can include actions performed by the organisation, a skilled person or the vehicle.

8.3.3 Verification and validation of the software update package

8.3.3.1 The required verification and validation needed for a software update package shall be determined before the release of the software update package.

NOTE Dependencies of software inside the software update package can also affect verification and/or validation.

8.3.3.2 Compatibility of the software update package with the existing software and hardware of the target shall be validated.

NOTE See 9.3.3.10 for validation for the recipient.

8.3.3.3 Dependencies of the software update package with the existing software and hardware of the target shall be validated.

8.3.3.4 Necessary in-vehicle resources in the target shall be validated.

EXAMPLE Network capacity, RAM, storage requirements, etc.

8.3.3.5 The organization shall determine the implications for the software update package of failure during software update operation.

8.3.4 Approval for release of the software update package

8.3.4.1 A software update package shall be approved for release based on the performed verification and validation determined by 8.3.3.1.

EXAMPLE 1 Confirming the correct version is in the software update package.

EXAMPLE 2 Confirming validation and verification have been successfully completed.

8.4 Work products

8.4.1 Documentation of targets, contents, compatibility, dependencies, necessary conditions, and necessary in-vehicle resources in the target for the software update package resulting from 8.3.1.1 to 8.3.1.5.

8.4.2 Software update package with only intended and necessary contents and actions resulting from 8.3.2.1 to 8.3.2.4.

8.4.3 Documentation of verification and validation resulting from 8.3.3.1 to 8.3.3.5.

8.4.4 Documentation of approval for release resulting from 8.3.4.1.

9 Software update campaign operations

9.1 Objectives

The objectives of this clause are to ensure that the following activities are performed:

a) Preparing software update campaigns;

b) Executing software update campaigns;

c) Completing software update campaigns.

9.2 General

This clause includes requirements on identifying the targets of a software update campaign, obtaining vehicle configuration information, resolving targets into recipients, issuing the software update package and relevant communications through the end of a software update campaign.

9.3 Requirements and recommendations

9.3.1 Software update campaign preparation

9.3.1.1 The organization shall determine the purpose(s) of the software update campaign.

NOTE The list of target(s) is determined in 9.3.1.11 and is not in the purpose.

EXAMPLE 1 A list of new features.

EXAMPLE 2 Fixing a newly discovered vulnerability

9.3.1.2 The organization shall assign roles and responsibilities for each software update campaign.

9.3.1.3 The organization shall select the software update packages for the software update campaign.

9.3.1.4 The organization shall confirm that each selected software update packages for the software update campaign has been approved for release. (See 8.4.4).

9.3.1.5 The organization shall determine which hardware versions and software versions in the targets are to be replaced by the software update campaign.

9.3.1.6 The organization shall determine which software update method(s) are used for the software update campaign.

NOTE 1 It is possible to choose one or more software update methods.

NOTE 2 It is possible to choose different software update methods for different software update packages in the same software update campaign.

EXAMPLE 1 A skilled person performs the software update operation in a workshop.

EXAMPLE 2 A vehicle performs a wireless software update operation.

9.3.1.7 The organization shall determine the list of the target(s) for each software update campaign.

EXAMPLE 1 A vehicle model and year.

EXAMPLE 2 A specific ECU in different vehicle models.

9.3.1.8 The software update campaign plan shall specify the necessary conditions, in-vehicle resources and/or external resources to execute the software update campaign.

NOTE These resources can be determined from 8.3.3.4.

EXAMPLE Necessary external resources could be available or remaining network capacity, workshop availability, connectivity and/or mobile network reception.

9.3.1.9 The organization shall determine the implications of dependencies on other ECUs and software versions during the software update campaign execution, based on the results from 8.3.3.3.

EXAMPLE The engine ECU software has a dependency on the transmission control unit software.

9.3.1.10 The software update campaign plan shall specify appropriate procedures/measures for corrective actions in the event of a software update operation failure in a vehicle.

NOTE 1 A corrective action could be technical or involve the vehicle user.

NOTE 2 If several targets or recipients of the software update campaign have failures, the organization can suspend or cancel the entire software update campaign.

EXAMPLE 1 A corrective action can be changing the vehicle operating mode to one in which the vehicle is safe.

EXAMPLE 2 A corrective action can be to notify the vehicle user to stop the vehicle in a safe area and contact dealer.

9.3.1.11 The measures for cybersecurity on the software update campaign shall be analyzed and determined based on the results of 8.3.3.5.

9.3.1.12 It shall be determined whether the software update campaign needs an operation or action that requires special equipment or training to complete the software update operation.

EXAMPLE Manual calibration, initialization or mechanical parts replacement.

9.3.1.13 For each software update campaign, the need for vehicle user confirmation shall be determined.

NOTE Confirmation can be obtained for each single instance of software update campaign or a general confirmation can be obtained at the beginning of the relationship between the vehicle user and the organization.

9.3.1.14 The necessary information to be communicated concerning the software update campaign, the related parties, and the communication method(s) shall be determined.

NOTE 1 The related parties can include external (vehicle users, government) and/or internal (customer service).

NOTE 2 The information can include content of the software update campaign, the user manual, or actions.

NOTE 3 Methods of communication can include paper based, an in-vehicle display, or a web page.

9.3.1.15 For each software update campaign, a software update campaign plan shall be created, which contains: 9.3.1.1 to 9.3.1.14 and Clause 8.

9.3.2 Software update campaign execution

9.3.2.1 The current vehicle configuration information necessary to resolve targets into recipients shall be obtained for each software update campaign.

NOTE 1 Vehicle configuration information can be for an entire vehicle or for an ECU.

NOTE 2 Vehicle configuration information can be obtained from an entity other than directly from the vehicle.

EXAMPLE A supplier obtains vehicle configuration information from an OEM.

9.3.2.2 The targets of the software update campaign shall be resolved into the recipients.

EXAMPLE A vehicle model and year is resolved into specific VINs of individual vehicles.

9.3.2.3 Before starting the software update operation, the processes which are required in clause 8(software update package development) and clause 9.3.1(software update campaign preparation) shall be completed.

9.3.2.4 The software update package shall be distributed to the vehicle(s) according to the software update campaign plan.

9.3.2.5 The software update campaign should ensure the necessary conditions, in-vehicle resources and/or external resources to perform the software update operations according to the software update campaign plan (9.3.1.15).

9.3.2.6 The software update operation shall arbitrate simultaneous access requests to maintain vehicle safety.

9.3.2.7 The integrity and authenticity of the software update package shall be verified before activation in a recipient of the software update operation.

NOTE This verification can be performed in the vehicle and/or the infrastructure.

EXAMPLE Guidance can be found in ISO/SAE 21434

9.3.2.8 The dependencies of the software update package(s) with existing hardware and software of the recipient shall be confirmed before activation.

NOTE This confirmation can be performed by the vehicle or the infrastructure.

EXAMPLE Confirming the dependencies with a tool.

9.3.2.9 The vehicle user should be informed about the availability of software update campaigns affecting their vehicles.

EXAMPLE 1 Sending the vehicle user written notification.

EXAMPLE 2 Displaying notification on an in-vehicle display.

9.3.2.10 Before activation in the software update operation in a recipient, the vehicle user should be informed about related information including:

- purposes of the software update campaign including the criticality of the software update campaign;

- instructions for safely performing the software update operation;

- changes in vehicle functions due to the software update campaign;

- vehicle functions not available during the software updates operation and its implications for the vehicle user;

- estimated time the vehicle or specific vehicle function(s) will not be available during the software update operation.

EXAMPLE 1 ECU functions can be made unavailable by: transiting into boot mode, resetting or rebooting an ECU.

EXAMPLE 2 Information that the advanced emergency braking system is not functional during the software update operation.

EXAMPLE 3 Information that the vehicle is not safe to drive.

9.3.2.11 The software update campaign should obtain confirmation from the vehicle user before activation in the software update operation.

NOTE 1 Confirmation can be obtained for each single instance of software update campaign or a general confirmation may be obtained at the beginning of the relationship between the vehicle user and the organization initiating a software update campaign.

NOTE 2 The vehicle user can have the capability to reject or postpone the software update campaign and can be informed of the corresponding risk.

EXAMPLE A vehicle user confirmation could be obtained via:

- an in-vehicle display;

- a mobile application;

- website;

- contractual agreement.

9.3.2.12 If the vehicle user is required to perform an action during software update campaign execution, then the vehicle user shall be notified.

EXAMPLE If the vehicle needs to be turned off and on to complete the software update operation, then a notification on an in-vehicle display.

9.3.2.13 Software update campaign activities that require special equipment or training shall be performed by a skilled person.

EXAMPLE 1 A skilled person servicing the vehicle at its location or in a workshop.

EXAMPLE 2 Software update packages are only available to a skilled person.

9.3.2.14 A corrective action shall be performed in accordance with the software update campaign plan when a software update operation fails.

9.3.2.15 The status of software update campaign execution of each recipient shall be obtained.

NOTE The status can be provided by recipients via established infrastructure to the organization.

EXAMPLE 1 The state of software update operation or the result of software update operation

EXAMPLE 2 The status of each recipient can be used to calculate the progress of the software update campaign and it can be the "number of recipients which completed the software update operation" divided by "number of recipients".

9.3.2.16 The results of the software update operation for each recipient should be reported in an appropriate timeframe to the organization initiating the software update campaign and to the related parties.

NOTE 1 The reporting can be postponed to avoid driver's distractions.

NOTE 2 The result of each vehicle can be "success, failure, interruption or cancellation, etc.".

9.3.2.17 The identified information about the content of the software update campaign shall be communicated to the related parties, including any changes to the user manual.

NOTE Information is identified in 9.3.1.14.

EXAMPLE 1 A software update campaign adds a completely new function and it requires new vehicle operation, then vehicle user is informed about the new operation.

EXAMPLE 2 The manual might be updated electronically inside the vehicle, changes sent out via email, made available on a server for download or ordered by the vehicle user to be sent again in paper from.

9.3.3 Software update campaign completion

9.3.3.1 The records of the application of software update campaign shall be managed and stored.

EXAMPLE The purpose of the campaign, target vehicles and systems, updated contents, start and end date, and results, etc. can be recorded for each software update campaign.

9.3.3.2 The end of the software update campaign should be communicated to the vehicle user and related parties.

NOTE It is not necessary to predetermine the end of a software update campaign during preparation or execution.

9.4 Work products

9.4.1 Documentation of software update campaign preparation resulting from 9.3.1.1 to 9.3.1.15.

9.4.2 Documentation of software update campaign execution resulting from 9.3.2.1 to 9.3.2.17.

9.4.3 Documentation related to the completion of the software update campaign resulting from 9.3.3.1 to 9.3.3.2.

Bibliography

[1] IATF 16949, *a technical specification for automotive sector quality management systems*

[2] ISO/IEC 27001, *Information technology — Security techniques — Information security management systems — Requirements*

[3] ISO/IEC 27002, *Information technology — Security techniques — Code of practice for information security controls*

[4] ISO/PAS 21448, *Road vehicles — Safety of the intended functionality*

[5] ISO 13489, *Safety of machinery — Safety-related parts of control systems*

[6] IEC 61508, *Functional safety of electrical/electronic/programmable electronic safety-related systems*

[7] ISO 10007, *Quality management — Guidelines for configuration management*

[8] ISO 9001, *Quality management systems — Requirements*

[9] ISO 15288, *Systems and software engineering — System life cycle processes*

[10] ISO 27025, *Space systems — Programme management — Quality assurance requirements*

