Article

Towards Security-by-design in Automotive Development Process

Seungyeon Jeong[1], Sooyoung Kang[2], Seungjoo Kim*

[1] Department of Automotive Convergence, Korea University, Seoul 02841, South Korea; sodon5@korea.ac.kr
[2] Center for Information Security Technologies, School of Cybersecurity, Korea University, Seoul 02841, South Korea; bbang814@korea.ac.kr
* Correspondence: Center for Information Security Technologies, School of Cybersecurity, Korea University, Seoul 02841, South Korea; skim71@korea.ac.kr

Abstract: Although traditional automotive development has mainly focused on functional safety, as the number of automotive hacking cases has increased due to the growing Internet connectivity of automotive control systems, security is also becoming more important. Accordingly, various international organizations are preparing cybersecurity regulations or standards to ensure security in automotive development by emphasizing the concept of security-by-design(i.e. security engineering) which emphasizes trustworthiness from the beginning of development. The problem, however, is that no specific methodology has been suggested. In this paper, we propose a specific security-by-design methodology for automotive development based on Secure System Development Life Cycle (secure SDLC) standards and evidence-based standards. Our methodology could be easily used in the actual field as it is more general and detailed than existing secure SDLC standards and research. Also, since it satisfies all requirements of United Nations Economic Commission for Europe (UNECE) regulation, automobile manufacturers could respond to the upcoming cybersecurity regulation with our methodology.

Keywords: Automotive development, Secure SDLC, Evidence-based standard, ISO/SAE 21434, UNECE cybersecurity regulation

1. Introduction

Traditionally, development in the automotive industry has focused on functional safety. Functional safety is a concept including the functional correctness that determines whether a product operates exactly as designed, and the safety that determines whether errors occurred inside the product are exposed externally and can harm users [1]. Functional safety aims to prevent a situation in which malfunctions of E/E (Electric/Electronic) systems are exposed to the outside and cause injury to users and if it's not ensured, this could lead to human casualties as well as simple system operational errors [2]. Therefore, the International Organization for Standardization (ISO) established a standard related to automotive functional safety called ISO 26262 so that functional safety can be considered throughout the development process [3].

Unlike functional safety, security has not been a focus in automotive development. Security is a concept including confidentiality which ensures only authorized users have access to information assets of the system, integrity which ensures the system is fully preserved without inappropriate change or destruction of information and availability which ensures access to and use of system information at any time users want [4]. Security aims to prevent a situation where external security threats expand into the system and cause damage to users and if it is not guaranteed, it could lead to various accidents such as loss of life or privacy. Recently, with the advent of connected cars, the software proportion and internet connectivity of vehicles are growing, and the possibility of vehicles being exposed to security threats is increasing accordingly [5], [6]. As a result, the necessity of automotive security development is rising, and international organizations are showing efforts to emphasize it by enacting automotive cybersecurity regulations. Especially, the UNECE automotive

cybersecurity regulation (UNECE regulation) will be applied from 2022 based on new vehicles, and according to this regulation, vehicles that are not evaluated and certified with the regulation will not be allowed to be exported to Europe [7]. Therefore, developing secure vehicles is an important issue not only for various security threats but also for the automotive import and export economy that will be confronted right away.

The UNECE regulation proposes security-by-design as a core requirement, which is a concept of implementing a trustworthy product by considering all factors of functional correctness, safety, and security from the beginning of product development. In particular, since automotive development has a long life cycle and complex supply chain, it is very difficult to change the architecture after development. Therefore, security-by-design must be dealt with more importantly in automotive development, and it can be achieved by the secure SDLC. Secure SDLC is a systematic security development framework applied throughout the entire product development life cycle. It is used by many companies (e.g. Microsoft) or standard organizations (e.g. National Institute of Standards and Technology (NIST)), and research is also actively carried out [8] – [15].

However, the existing secure SDLC standards do not provide an overall and specific automotive security-by-design methodology, since they not only target software mostly but also emphasize different aspects of activities, such as eliciting systematic requirements or acquiring third-party components. In addition, existing research do not provide a specific methodology of achieving security-by-design either, since they have been conducted only for some phases, and even if they target the entire process, they provide only conceptual approaches or application targets are limited.

Therefore, in this paper, we propose Trustworthy Automotive SDLC as a specific methodology for security-by-design in the automotive development process. Trustworthiness is a concept providing the trust that the system will operate as we expected by considering all aspects of functional correctness, safety, and security of the system in development [16], and it should be particularly emphasized in the system where functional safety is important such as the automotive system.

In order to propose Trustworthy Automotive SDLC, we firstly derive activities related to automotive development from 4 major secure SDLC standards. These include Microsoft Security Development Lifecycle (Microsoft SDL), NIST Secure System Development Life Cycle (NIST SSDLC), The Open Web Application Security Project Comprehensive, Lightweight Application Security Process (OWASP CLASP) and Society of Automotive Engineers J3061 (SAE J3061). Afterward, we mapped each activity to the detailed items of evidence-based standards to derive Trustworthy Automotive SDLC in detail. Evidence-based standard is a standard composed of detailed evidences(detailed activities or outputs) required for a development process. Since it verifies the source of collected evidences, it ensures traceability between each evidence and each phase. In this paper, we consider 4 number of evidence-based standards: CC (Common Criteria, ISO/IEC 15408) which is a standard related to security evaluation of IT products, ISMS (Information Security Management System, ISO/IEC 27001) which is a standard related to security evaluation of development environment, PIMS (Privacy Information Management System, ISO/IEC 27701) which is a standard related to privacy and FSMS (Functional Safety Management System, ISO 26262) which is a standard related to automotive functional safety. Especially CC is very useful when you want to build secure SDLC since it specifies requirements for documents to be produced. Lastly, we pull out the requirements which are essential for automotive development, and compose Trustworthy Automotive SDLC based on them.

Trustworthy Automotive SDLC has a form that can be integrated into the existing functional safety System Development Life Cycle (safety SDLC) and takes into account all aspects required in automotive development: functional correctness, safety, and security. In addition, it provides a sufficient security level required in automotive development and suggests detailed activities. Thus, we believe that automobile manufacturers could ensure trustworthiness of the development process, and respond to the upcoming UNECE regulation at the same time with our methodology.

2. Literature review

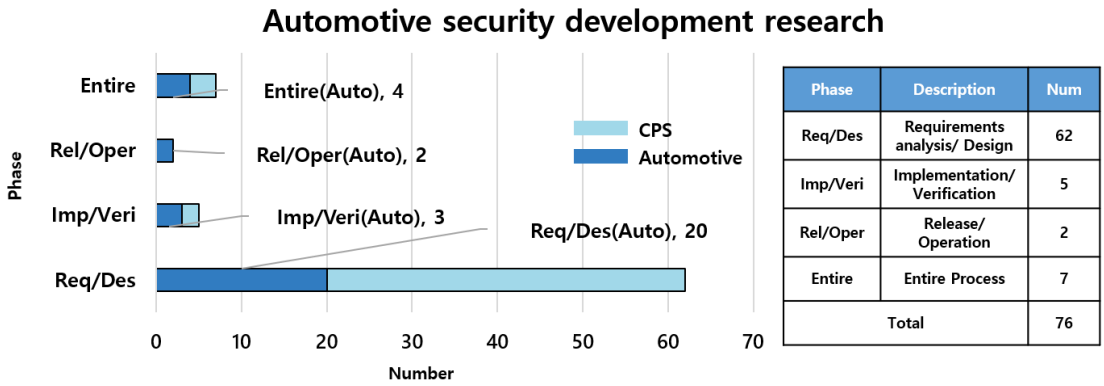
2.1. Research papers

To emphasize the need for Trustworthy Automotive SDLC, we analyzed research papers on automotive security development. We examined papers on secure SDLC in the last 10 years (2010 to 2020) from 5 well-known digital libraries: 1)ACM, 2)IEEE, 3)Springer, 4)Elsevier and 5)MDPI. In this procedure, we chose keywords related to automotive security development: 1) 'Automotive' or 'CPS (Cyber-Physical System)', 2) process-related keywords (e.g. 'security-by-design', 'process', 'life cycle', 'development'), phase-related keywords (e.g. as 'design') or activity-related keywords (e.g. 'dynamic analysis').

As a result, we collected 76 papers [17] – [50], and Figure 1 is a graph classifying them based on the phases covered in each paper. We confirmed that there has been a steady effort to achieve security-by-design for CPS including vehicles and that it has been conducted for some phases or the entire development process. Especially in case of phase-focused papers, most of them are focused primarily on *model-driven security*, which refers to the concept of applying security to the model-based software development process, to achieve security-by-design in the design phase. Neureiter *et al*. [29] raised a need for a reference architecture to integrate and maintain security into existing smart grid architectures and presented a conceptual approach to enable security-by-design into the design phase in the development of smart grid systems by using the NIST logical reference model. Also, Geismann *et al*. [47], reviewed the literature on model-oriented security techniques for CPS and emphasized that the platform should also be considered in CPS which requires a high-reliability level. They identified security requirements for CPS and suggested a way to deal with them at the design phase by taking appropriate countermeasures for the software according to the type of software(platform-independent or platform-specific). However, these papers deal with limited security proof and can be applied only to a limited scale.

In case of process-focused papers, most have been conducted on safety & security co-engineering. Representatively, Schmittner *et al*. [22] raised a necessity to integrate independent safety and security development processes as the complexity and connectivity of the system increased. They reviewed existing standards to identify key activities, introduced an integrated lifecycle that harmoniously integrates safety and security activities, and conducted case studies using past events and incidents. Skoglund *et al*. [49], presented a safety and security-based co-engineering approach in terms of design and verification phase based on one automotive subsystem and presented the result that synergy in the verification phase is greater than that in the design phase. Bramberger *et al*. [50] introduced a procedure for proposing a development process incorporating security into a safety-oriented process using the existing tools which are used to build the development process and proved its usability based on a virtual scenario. However they proposed only conceptual approaches and not detailed activities [18], [19], [22], [30], [44], [49] or they targeted subsystem, not the entire system [43].

Therefore, since existing research have limited targets and scope and are difficult to apply to mass production processes of vehicles, we can conclude that there is a need for a detailed methodology for automotive security-by-design across the entire development process.



Phase	Description	Num
Req/Des	Requirements analysis/ Design	62
Imp/Veri	Implementation/ Verification	5
Rel/Oper	Release/ Operation	2
Entire	Entire Process	7
Total		76

Figure 1. Automotive security development research.

2.2. Secure SDLC standards

In the case of existing product development, security has been improved by secure SDLC [51]. Secure SDLC enables the development of secure products by considering security-related activities for all phases. However, the existing secure SDLC standards do not provide overall and sufficient details of the actual application. Therefore, we establish a universal security-by-design methodology by integrating existing secure SDLC standards. We targeted 4 secure SDLC standards: Microsoft SDL based on the software [52], NIST SSDLC based on the system [53], OWASP CLASP based on enterprise best practices [54], SAE J3061 based on the vehicle [55]. Since each secure SDLC standard emphasizes different aspects such as deriving systematic security requirements or acquiring third-party components, overall activities of the development process can be derived by integrating them. Table 1 shows the features of each standard.

Table 1. Features of secure SDLC standards.

	Microsoft SDL	**NIST SSDLC**	**OWASP CLASP**	**SAE J3061**
Target	Software	System	Best practices	Vehicle
Feature	Provide the developer-oriented process	Focuses on third-party components acquisition and system disposal	Provides real-world enterprise activities in the form of best practices	Provides rough activities of secure automotive development
Pros&Cons (P: Pros, C: Cons)	Provides tools for performing activities such as risk analysis (P)	Used to evaluate systems that require certification & accreditation (P)	Identifies the role of the personnel in charge of each activity (P)	Easy to apply security-related activities according to the automotive function safety SDLC (P)
	Not include disposal phase (C)	Lack of activity for implementation phase (C)	Lack of information because the project period expired (C)	Not include training and disposal phase (C)

2.3. Automotive cybersecurity regulation and standard

With the advent of the connected car, the portion and connectivity of automotive software increases, and the importance of automotive security is growing. Accordingly, various international organizations are enacting regulations or standards to ensure the security of automotive development as we mentioned earlier [56]. Especially, UNECE is enacting automotive cybersecurity regulation to ensure security throughout the entire development process, and it will be enacted from 2022 on new vehicles and 2024 on existing vehicles [57]. UNECE regulation is based on the ISO/SAE 21434, and ISO/SAE 21434 is an international standard for automotive cybersecurity established by ISO and SAE based on SAE J3061 and it will also be published in 2022 [58], [59]. Therefore, we propose Trustworthy Automotive SDLC that covers all the requirements of both UNECE regulation and ISO/SAE 21434.

3. Methodology

In this chapter, we explain the method of constructing Trustworthy Automotive SDLC suitable for automotive development, and Figure 2 shows the procedure. At first, we extract the overall activities related to automotive development from 4 representative secure SDLC standards which present only somewhat rough activities. Then we extract detailed items to comply with 4 evidence-based standards of each domain such as detailed activities to be performed or outputs to be derived. Also, we pull out requirements that are essential for automotive development. Each requirement is selected based on the characteristics of automotive development or the security level required for it. As a result, the activities related to the automotive development are detailed by the evidence-based standard and filtered based on the requirements, and finally, Trustworthy Automotive SDLC is derived.

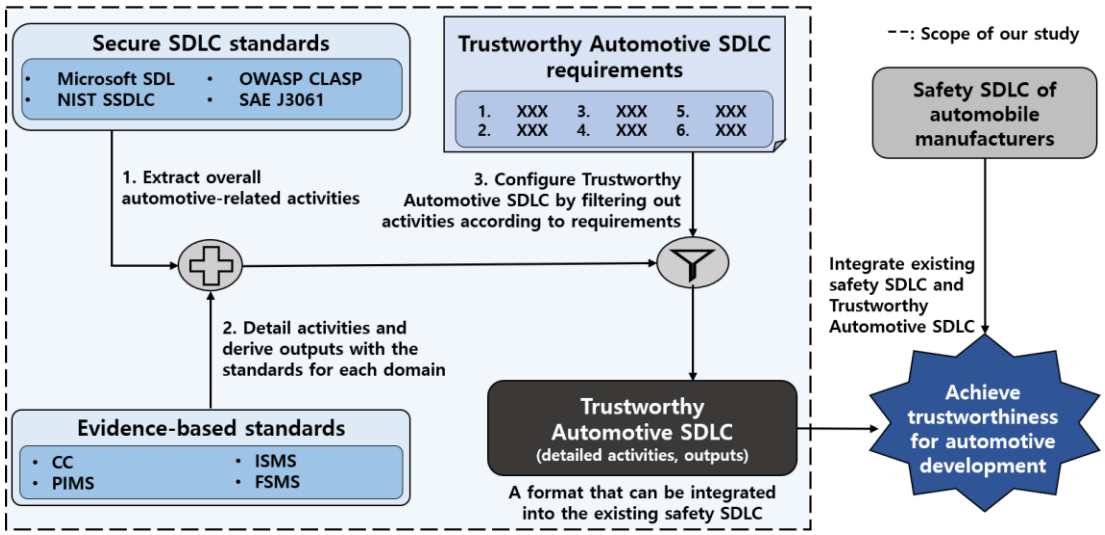


Figure 2. Procedure of Trustworthy Automotive SDLC construction.

Trustworthy Automotive SDLC enables developing a vehicle with security more than traditional development methodologies which considered security with only a few activities, such as penetration testing. Also, it makes automobile manufacturers that considered only functional safety achieve trustworthiness if it is integrated into the existing safety SDLC.

3.1. Requirements

The requirements of Trustworthy Automotive SDLC are as follows:

1. Targets on system
2. Includes the procedure of third-party components acquisition
3. Considers the aspect of trustworthiness
4. Satisfies the security level required for automotive development
5. Ensures traceability between phases
6. Provides detailed activities for every phase of the process (depth = 2)

First, vehicles are developed in the form of a system including software and hardware [60]. Therefore, Trustworthy Automotive SDLC must build a system-based process. Also, automobile manufacturers do not develop all the components of vehicles by themselves but utilize components acquired from tier-1 or tier-2 suppliers [61]. Therefore, in order to attain trustworthiness for third-party components, Trustworthy Automotive SDLC must perform activities for obtaining third-party components.

In third, trustworthiness which includes all aspects of functional correctness, safety, and security should be considered in automotive development [62]. Since functional correctness and safety have

been considered by the safety SDLC and Automotive Safety Integrity Level (ASIL) suggested by ISO 26262, we need to design a methodology that can combine security based on them. In particular, the portion where functional correctness, safety, and security goals conflict should be identified in the early phases of the development process [63].

In case of the safety level, ASIL required by ISO 26262 is selected differently depending on the function of the vehicle, and the safety SDLC develops a vehicle based on the ASIL assigned to each function. According to U.S. security company *Synopsys*, the safety SDLC should satisfy ASIL C on average for core functions [64]. With respect to security, we consider 2 aspects of the security level. Firstly, the Evaluation Assurance Level (EAL) required for the automotive system should be satisfied. EAL is the assurance level of CC, an international standard related to IT product security evaluation. According to [65], the ASIL C of ISO 26262 corresponds to the EAL 5 of CC. Therefore, in this paper, we determine that Trustworthy Automotive SDLC should cover the EAL 5 to ensure sufficient security of automotive development. In addition, Trustworthy Automotive SDLC should also reflect the requirements of automotive cybersecurity regulation which is essential for automobile manufacturers targeting not only domestic but also overseas markets. Thus, we consider UNECE regulation and ISO/SAE 21434, and in case of ISO/SAE 21434, we only consider the essential requirement RQ (Requirement).

Since the vehicle is a critical system in which a system's problem can lead to human casualties [61], [66], Trustworthy Automotive SDLC should ensure traceability of the system more rigorously by verifying the consistency and completeness between goals, requirements, architectures, and implemented system. In addition, Trustworthy Automotive SDLC should make it easy to apply itself to the real-world by proposing a broad and detailed methodology across the development process. In this paper, we define a detailed level of activity by *depth*. For example, phase such as 'verification' has a depth of 0, subphase such as 'test' has a depth of 1, and activity such as 'static analysis' has a depth of 2. Trustworthy Automotive SDLC has a format that can be applied to the actual product since it provides activities of all phases to the depth of 2 so that we can directly apply them to the detailed items of evidence-based standards.

3.2. Design methodology

To derive Trustworthy Automotive SDLC, we select 4 number of representative secure SDLC standards and derive universal security-by-design methodology. Subsequently, we map all activities to detailed items of evidence-based standards. There has been a steady stream of studies mapping the development process to evidence-based standards, but they were all focused on some phases (e.g. [67] for requirement analysis phase) or just suggested conceptual idea([68] – [77]). In this paper, we target the entire process and map detailed items of 4 evidence-based standards (CC, ISMS, PIMS, and FSMS) to each activity of the Trustworthy Automotive SDLC to refine all the activities, and Table 2 shows the features of the evidence-based standards. With this procedure, Trustworthy Automotive SDLC can further refine all activities in all aspects of security, privacy, and functional safety for products and development environments. Figure 3 shows the mapping relation between our Trustworthy Automotive SDLC and evidence-based standards and detailed mapping results are shown in Appendix A. Lastly, activities that satisfy the requirements of Section 3.1 are selected to derive Trustworthy Automotive SDLC.

Table 2. Features of evidence-based standards.

	CC	ISMS	PIMS	FSMS
Target	Security(product)	Security(environment)	Privacy	Functional safety
Description	The standard for evaluating the security and reliability of IT products	The standard for security and reliability certification of an organization's assets	The standard for providing detailed criteria for the performance of privacy impact assessment	The standard for presenting process of automotive functional safety development
Mapping phases	2,3,5-10	1-2,5,7-10	1-5,7-10	1-10
Source	ISO/IEC 15408	ISO/IEC 27001	ISO/IEC 27701	ISO 26262

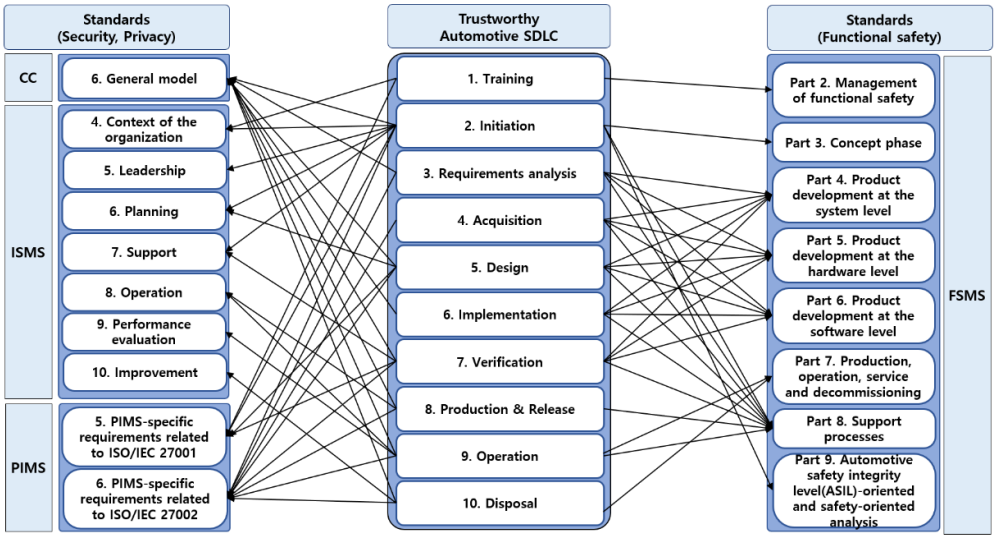


Figure 3. Mapping between Trustworthy Automotive SDLC and evidence-based standards.

4. Results

As a result, we derived Trustworthy Automotive SDLC is as shown in Figure 4 and it consists of a total of 50 activities for 10 phases. Trustworthy Automotive SDLC is a system-targeted methodology and performs the acquisition procedure for third-party components in acquisition phase. Also, as shown in Figure 3, it is possible to map all phases with each part of FSMS, so it can be merged into the existing safety SDLC which follows ISO 26262. In addition, conflicts between functional safety and security that occur in the procedure of merging the secure SDLC and the safety SDLC can be resolved by activities such as *3.1.2 conformity & conflict check on impact assessment results* or *3.2.2 conformity & conflict check on requirements*. Trustworthy Automotive SDLC also consists of detailed activities satisfying the security level required for automotive development and the requirements of UNECE regulation and ISO/SAE 21434.

4.1. Activities for each phase

Training phase. During the training phase, organization members gain the awareness of safety, security, and privacy, and get training about the relevant knowledge required as they go through the development process. Teams developing or managing functions related to security or safety (e.g. security team) have background knowledge, but most other teams (e.g. development team) often lack knowledge of trustworthiness. Therefore, it is necessary to train the subjects required for process execution through training by domain. Basic training covers topics for gaining awareness of safety, security, and privacy. In addition, by training basic knowledge on topics covered in the development process (e.g. risk analysis), it ensures that organization members do not feel uncomfortable in carrying out the process. Advanced training is an additional training which

trains how to carry out the detailed activities of Trustworthy Automotive SDLC. It is given to personnel according to their task. From a security perspective, this includes topics such as security design and secure coding for the development team, static and dynamic analysis for evaluation team, security, and privacy-related regulatory certification for security team. All training is managed by the road map, and by managing compliance with training according to the target audience, traceability of training is ensured.

Initiation phase. In the initiation phase, the development environment for the project is established, overall plans of the project are established, and goals for each domain are set. First, this

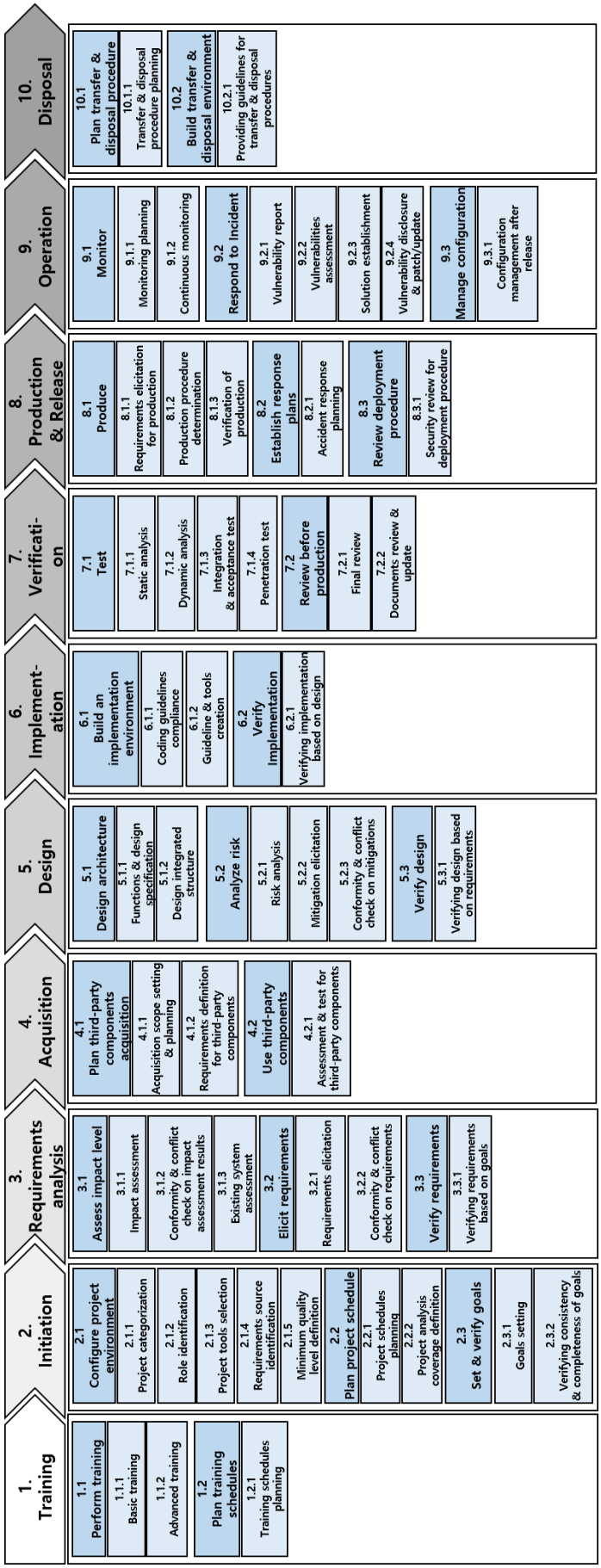


Figure 4. Trustworthy Automotive SDLC.

phase carries out project categorization by considering the information assets, product types, and project characteristics handled by the project. In addition, based on them, plans for the entire project including roles, tools, and minimum quality level are established. Particularly, minimum quality level are established. Particularly, minimum quality level is the minimum level of security, safety, and privacy that must be met by the project. If it is not satisfied, the following phases cannot be carried out. The initiation phase also establishes goals by domains and verifies consistency and completeness between goals to ensure traceability to the performance of subsequent phases. In the case of the automotive system, the development is performed separately according to the system, software, and hardware-level. Therefore, the corresponding phase should be performed according to each level. At the initiation phase on system-level, consistency of software and hardware development should be ensured by establishing the overall environment and plans covering the rest of the levels (i.e. hardware and software). As shown in Figure 4, this configuration is equally applied from the initiation to the production & release phase.

Requirement analysis phase. In the requirement analysis phase, impact assessment is performed on the project's security, safety, and privacy-related assets, and based on the results, requirements by domains are derived. This phase also ensures traceability between the initiation phase and the requirement analysis phase by verifying consistency and completeness between goals and requirements. As mentioned earlier, goals of safety and security can conflict [63]. Therefore, this phase identifies priority for safety and security impact level through the activity of *3.1.2 conformity & conflict check on impact assessment results*. At this point, security and safety impact on the reused existing system is also evaluated. Also, conformity and conflict of requirements by each domain are also checked by the activity of *3.2.2 conformity & conflict check on requirements*. Lastly, by verifying the consistency and completeness of requirements according to goals previously derived, traceability between the initiation phase and requirement analysis phase is ensured.

Acquisition phase. At the acquisition phase, the scope and plans of third-party component acquisition are established and the relevant requirements are defined. Also, based on the requirements, evaluation, and tests are performed on the specifications of third-party components. Third-party components are those acquired by tier-1 or tier-2 suppliers. In addition, in this phase, automobile manufacturers perform independent evaluations and tests based on the specifications of third-party components submitted in accordance with the automobile manufacturer's requirements. In particular, in the case of automotive development, the development of subsystems through partners plays a significant role in overall development, so it is essential to perform the corresponding phase.

Design phase. During the design phase, the architecture is designed and risk analysis for each domain is performed based on it. Since automotive development constructs one integrated system based on lots of subsystems, the corresponding phase considers the system integration procedure. Also, based on the integrated architecture, risk analysis is performed by domain to derive mitigation for possible threats of the system. As with the preceding phase, the design phase checks the conformity and conflict of domain-specific mitigation so that there is no conflict between the safety SDLC and the security development process. It also ensures traceability between the requirement analysis phase and the design phase by verifying the consistency and completeness of the architecture following the requirements.**Implementation phase.** In the implementation phase, the system is implemented based on the requirements and architectures derived ahead. At this phase, development team implements the system with coding guidelines, which may include previously established coding standards such as 'MISRA C'[78]. The development team also creates deployment guidelines or tools to enable users to build a trusted operation environment when they use the system. Furthermore, traceability between the design and the implementation phase is achieved by verifying consistency and completeness between the architectures and the implemented system.

Verification phase. During the verification phase, we perform tests and reviews based on the implemented system. Tests include static/dynamic analysis and acceptance tests. In addition, since the vehicle is an integrated system based on many subsystems, an integration test is performed to determine whether there is a problem in the integrating procedure. At this point, integration means not only subsystem-based integration but includes hardware and software-based integration into the system. Subsequently, the penetration test is performed to determine whether a security threat exists for the integrated system. Also, minimum quality levels or documents are reviewed based on whether any design change made in the implementation phase does cause security, safety, and privacy-related problems.

Production & Release phase. The production & release phase produces the system and establishes response plans for possible accidents of the system after deployment. It is necessary to ensure the security of the vehicle production procedure since the vehicle has a long and complicated production procedure. This phase also reviews whether security issues occur in the deploying procedure of a produced system.

Operation phase. The operation phase monitors potential vulnerabilities of the system after deployment and responds to what is found. It is necessary to establish a monitoring plan and continuously monitor the system based on it. The operation phase also collects vulnerability reports on accidents and evaluates them to derive countermeasures. Then, corresponding countermeasures are disclosed and distributed with patches or updates. In the case of vehicles, it is important to have a team in charge of responding 24 hours a day since problems can be directly led to a casualty accident. Also, if a change occurs in a system in operation, the traceability of the system should be achieved by managing the configuration of the changes.

Disposal phase. In the disposal phase, the use of the system is terminated and the transfer or disposal procedure is performed. System transfer is the case for transferring owner and system disposal is the case for discarding vehicles. Although automobile manufacturers do not carry out the procedure of transfer or disposal on their own, they should ensure the trustworthiness of them by providing information to relevant partners with guidelines. The applicable guidelines should include information about the complete sanitation of the user's personal information and preservation of future available internal information (e.g. the mileage of the vehicle). Depending on the purpose, vehicles could be considered in various situations such as transferring owners (e.g. sales of used vehicles) or used by a large number of users (e.g. rental cars). Therefore, in the disposal phase, the automobile manufacturers must provide information to the responsible companies so that they can perform the procedure suitable for the use or characteristics of the system.

4.2. Detailed activities and evidences of each phase

We derived detailed activities by mapping 393 items of the evidence-based standards (63 in CC, 104 in ISMS, 54 in PIMS, and 172 in FSMS) with 50 activities of Trustworthy Automotive SDLC. As an example, the 'AGD class' of CC contains the contents of determining whether a user can securely build an operating environment through the operation manual when distributing a product to a user. This can be mapped to the *6.1.2 guideline & tools creation* of Trustworthy Automotive SDLC, through which the contents required by the AGD class user operation manual (e.g. 'The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.') can be reflected in this activity. As a result, the detailed activities of Trustworthy Automotive SDLC in terms of security, safety, and privacy are derived through this procedure. In addition, document templates can be produced(i.e. outputs) for each phase of Trustworthy Automotive SDLC in this procedure, and the list of outputs is shown in Table 3.

Table 3. Outputs of Trustworthy Automotive SDLC.

	Phase	Evidence
1	Training	· Training plan · List of attendance at training
2	Initiation	· Project plan · Project goal verification report
3	Requirement analysis	· Impact assessment report · Requirement definition report · Project requirement verification report
4	Acquisition	· Acquisition plan · Acquisition inspection report
5	Design	· Design specification · Architecture specification · System Risk Analysis report · Project design verification report
6	Implementation	· System implementation report · User manual or tool · Project implementation verification report
7	Verification	· Test result report · Vulnerability Analysis report · Final review report
8	Production & Release	· Production plan · Production verification report · Incident response plan
9	Operation	· System monitoring report · Incident response report
10	Disposal	· Guidelines for system transfer and disposal

Training phase. A training plan that includes the purpose of basic and advanced training, target audience, and topics covered in each training is produced. Also, a list of attendance at training is documented so that we can track the attendance or training list for each employee.

Initiation phase. A project plan is produced which includes the assets for each domain, roles (security team, development team, design team, etc.), tools, and the minimum quality level. In particular, the minimum quality level is included as it serves as a baseline for whether or not to carry out the phase of the project. Also, project goals and the results of verifying consistency and completeness between goals in terms of security, safety, and privacy are documented in a project goal verification report.

Requirement analysis phase. An impact assessment report that evaluates the impact of assets by the domain (such as ASIL in terms of safety) and a requirement definition report based on this are produced. Especially, the impact assessment report includes the results of the impact assessment on the reused existing system. In addition, we derive a project requirement verification report that includes the analysis results, such as whether the requirements are consistently and completely derived based on the project goals or whether there is a conflict between the requirements of each domain.

Acquisition phase. An acquisition plan including acquisition requirements and procedure for components developed by third-party is produced. Based on this document, an acquisition inspection

report including the results of evaluation and test performed independently on those components provided by other companies is also derived.

Design phase. A design specification for a unit (such as a module or interface) and an architectural specification for the entire system including all of them are derived. Also, a system risk analysis report including mitigations is produced based on the risk analysis results. This report includes the analysis result of the suitability of the mitigation for each domain and whether there is a conflict. A project design verification report is also produced based on the analysis results of the consistency and completeness of the requirements.

Implementation phase. A system implementation report including the source code based on the coding guidelines is produced, and the user manual or tools including information related to the installation environment are derived. At this phase, we also document the consistency and completeness analysis result of the implementation performed based on the architecture in the project implementation verification report.

Verification phase. We produce a test result report including static and dynamic analysis results. In the case of an independent penetration test, results are documented in a vulnerability analysis document, and the final review report is generated by reviewing whether all phases before production were performed properly.

Production & Release phase A production plan including the plan and procedure of the product production is derived, and the production verification report for the actual product is documented. In addition, an incident response plan is produced that includes a response plan for possible incidents in the future.

Operation phase. A system monitoring report is derived based on monitoring performed on servers or firewalls. In addition, an incident response report including the evaluation results of the discovered vulnerabilities, the disclosure of the vulnerabilities and the patch/update management plan is produced.

Disposal phase. Guidelines for system transfer and disposal are derived for users and related companies so that the system can be securely transferred or disposed of even in situations that are not managed by the automobile manufacturer.

5. Discussion

5.1. Comparison with secure SDLC standards and reseach papers

Trustworthy Automotive SDLC(TA_SDLC) is a security-by-design methodology which presents more detailed activities than the existing secure SDLC standards, and more suitable for automotive development. Table 4 shows the result of a comparative analysis of existing research papers and secure SDLC standards by each requirement of TA_SDLC. Research papers targeting the entire development process from Section 2.1 are selected as the target research papers [18], [19], [22], [30], [44], [49] and Microsoft SDL, NIST SSDLC, OWASP CLASP, and SAE J3061 are selected as the target secure SDLC standards. The result has shown that research papers and secure SDLC standards are insufficient in presenting the security-by-design methodology suitable for automotive development. Notably, we found that all of them did not satisfy with the 4th and 6th requirements.

In the case of the 4th requirement, it is important that the development process not just presents activities, but activities that ensure sufficient security. In particular, since the security level for existing products has been rigorously assessed through CC, it is efficient to construct the process that ensures the security level required by automotive development on this basis. Since TA_SDLC has been established to ensure all detailed activities and outputs required for CC EAL 5, the automotive system produced by the process could get a certification of CC and other security assessments related

to CC. Also regarding the 6th requirement, secure SDLC standards have 36 activities with a depth of 2 throughout 9 phases in maximum, but TA_SDLC consists of 50 activities with a depth of 2 throughout 10 phases. Therefore, the activities of TA_SDLC are broad and detail than the existing secure SDLC standards, so it can be used systematically and easily in the actual field.

Table 4. Limitations of research papers and secure SDLC standards

Requir-ements	Research papers						Secure SDLC standards				TA_ SDLC
	[18]	[19]	[22]	[30]	[44]	[49]	Microsoft SDL	NIST SSDLC	OWASP CLASP	SAE J3061	
1	O	O	O	O	O	O	X	O	X	O	O
2	X	X	X	X	X	X	X	O	O	O	O
3	O	O	O	O	O	O	X	X	X	O	O
4	X	X	X	X	X	X	X	X	X	X	O
5	O	X	O	X	O	X	O	O	X	O	O
6	6/13	0/0	7/28	0/0	0/0	2/8	8/34	9/36	9/21	9/26	10/50

5.2. Analysis of UNECE regulation requirements

TA_SDLC provides a development process suitable for automobile manufacturers targeting domestic as well as overseas markets by enabling them to respond to the upcoming UNECE regulation. To verify this, we choose 16 requirements related to the development process in UNECE regulation and map them to the activities of TA_SDLC to determine whether they were satisfied. As a result, as shown in Table 5, all of the requirements suggested by UNECE regulation could be satisfied with TA_SDLC. Therefore, we can confirm that the automotive security-by-design methodology we proposed has a form that can be certified against UNECE regulation.

Table 5. Mapping between UNECE regulation requirements and TA_SDLC activities.

	UNECE regulation requirements (7.2. Requirements for the CSMS (Cyber Security Management System))	TA_SDLC activities (Activity number)
1	The vehicle manufacturer shall have a CSMS in place and shall comply with this Regulation.	Total
2	The vehicle manufacturer shall demonstrate that their CSMS applies to the development phase.	Total
3	CSMS shall include the processes used within the manufacturer's organization to manage cyber security.	Total
4	CSMS shall include the processes used for the identification of risks to vehicles.	3.1.1/3.1.2/5.2.1/ 5.2.3
5	CSMS shall include the processes used for the assessment, categorization and treatment of the risks identified.	5.2.2/5.2.3
6	The vehicle manufacturer shall demonstrate how their CSMS will manage dependencies that may exist with contracted suppliers, service providers or manufacturer's sub-organizations.	4.1.1/4.1.2/4.2.1
7	CSMS ensure security shall include the processes used for testing the cyber security of a vehicle.	7.1.1/7.1.2/7.1.3/7.1.4
8	CSMS ensure security shall include the processes in place to verify that the risks identified are appropriately managed.	7.2.1/9.1.1/9.1.2/9.3.1
9	The vehicle manufacturer shall demonstrate that their CSMS applies to the production phase.	8.1.1/8.1.2/8.1.3
10	The vehicle manufacturer shall demonstrate that the processes that include the capability to analyze and detect cyber threats, vulnerabilities and cyber-attacks from vehicle data and vehicle logs.	9.1.1/9.1.2/9.2.1
11	CSMS shall include the processes used for ensuring that the risk assessment is kept current.	9.2.4/9.3.1
12	CSMS shall include the processes used to monitor for, detect and respond to cyber-attacks, cyber threats, and vulnerabilities on vehicles.	9.1.1/9.1.2/9.2.1/9.2.2/9.2.3/

		9.2.4/9.3.1
13	CSMS shall include the processes used to assess whether the cyber security measures implemented are still effective in the light of new cyber threats and vulnerabilities that have been identified.	7.2.1/9.2.3
14	The vehicle manufacturer shall demonstrate that the processes used within their CSMS will ensure that cyber threats and vulnerabilities which require a response from the vehicle manufacturer shall be mitigated within a reasonable timeframe.	9.2.1/9.2.2/9.2.3/9.2.4
15	The vehicle manufacturer shall demonstrate that the processes that include vehicles after first registration in the monitoring.	9.1.1/9.1.2
16	The vehicle manufacturer shall demonstrate that their CSMS applies to the post-production phase.	8.1.1/8.1.2/8.1.3/8.2.1/8.3.1/ 9.1.1/9.1.2/9.2.1/9.2.2/9.2.3/9 .2.4/9.3.1/10.1.1/10.2.1

6. Case study

We conducted a case study with a domestic automobile manufacturer company A to prove the effectiveness of TA_SDLC. Company A is an automobile manufacturer targeting not only domestic but also overseas markets and preparing to respond to UNECE regulation. We grasped the current development process and activities of company A based on expert interviews & surveys and analyzed to what extent the process satisfies the activities of TA_SDLC. In addition, we derived improvements in company A's development process for achieving automotive security-by-design based on the analysis result.

Table 6. TA_SDLC activities fulfillment.

	Phase	Number of activities (A/TA_SDLC)
1	Training	2/3
2	Initiation	8/9
3	Requirement analysis	3/6
4	Acquisition	3/3
5	Design	5/6
6	Implementation	2/3
7	Verification	3/6
8	Production & Release	4/5
9	Operation	4/7
10	Disposal	0/2
	Total	34/50

As a result of mapping the development process of company A to TA_SDLC, we found that 34 out of 50 activities were performed by company A as shown in Table 6, and that company A should perform 16 additional activities to achieve security-by-design. Afterward, based on the result, we derived improvements in company A's development process as shown in Table 7.

Table 7. Improvements in company A's development process for automotive security-by-design.

Phase		Improvements
1	Training	Schedules and policies for training should be established, and the compliance of training for each member should be managed.
2	Initiation	The scope of analysis for each domain(safety, security) should be established. For example, in the case of an engine, safety should be considered for both the engine and the software which controls the engine, but only the software needs to be considered in terms of security.

3	Requirement analysis	The impact assessment should be performed and the conformity and conflict of the requirements by domain should be assessed. Also, the consistency and completeness of the requirements according to the objective should be verified.
4	Acquisition	-
5	Design	The conformity and conflict of the mitigations by domain should be assessed.
6	Implementation	Consistency, completeness of the implemented system should be verified.
7	Verification	Dynamic analysis should be performed on the implemented system, and a final review by domain should be executed before system deployment. Also, the documentation produced in the previous phase should be reviewed and updated.
8	Production & Release	Security review should be performed on the deployment procedure, and the trustworthiness of the supply chain and users must be checked.
9	Operation	After identifying the monitoring targets for the system operating environment, data collection, and reporting strategies, continuous monitoring of the system should be performed.
10	Disposal	A plan for system transfer and disposal should be established based on the relevant laws or policies that the system complies with. Also, guidelines should be established and provided for trustworthy transfer and disposal procedures to be provided to suppliers.

This is a result of applying TA_SDLC to actual automobile manufacturers, and we presented the possibility that based on our methodology, automobile manufacturers can improve the development process of their companies to satisfy UNECE regulation and to develop automobiles with reliability at the same time.

7. Conclusion

Traditional automotive development has focused on functional correctness and safety but not addressed security with emphasis. However, as the number of automotive hacking cases increase due to the recent increase in internet connectivity of vehicles, various international organizations are preparing cybersecurity regulations to achieve the security of automotive development. Typically, UNECE regulation will be applied on a new vehicle from 2022 and it emphasizes security-by-design which takes into account trustworthiness from the beginning of development. However, it does not provide a specific methodology for achieving security-by-design, and this is also true for previous research papers. Therefore, to solve this problem, this paper proposes Trustworthy Automotive SDLC, a concrete methodology for automotive security-by-design.

In this paper, we first derived activities related to automotive development from 4 major secure SDLC standards and detailed them with 4 evidence-based standards CC, ISMS, PIMS, and FSMS in terms of security, privacy, and functional safety for the product and development environment. Additionally, based on the mapped results, we configured a detailed Trustworthy Automotive SDLC suitable for automotive development. We also demonstrated the effectiveness of applying Trustworthy Automotive SDLC by case study. Trustworthy Automotive SDLC considers all aspects of functional correctness, safety, and security which are important for automotive development. Furthermore, by consisting of activities embodied through existing secure SDLC standards and evidence-based standards, it can be used for the upcoming UNECE regulation. In future work, we will apply our methodology to the actual field and prove the effectiveness of it with the result.

In future work, we will propose tools or methodologies required for carrying out each activity of Trustworthy Automotive SDLC in more detail. Then we will apply our methodology to the actual automobile manufacturers to prove the effectiveness of it.

Acknowledgments: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2015-0-00403)supervised by the IITP(Institute for Information &communications Technology Planning &Evaluation)

Appendix A

Appendix A shows a table of detailed mapping results of trustworthy automotive SDLC and evidence-based standards as shown in Figure 8. We provide the result of CC, ISMS, PIMS, and FSMS by clause of each standard and in the case of FSMS, we chose part 6 as an example.

Table 8. Detailed mapping result of TA_SDLC and evidence-based standards.

TA_SDLC			CC	ISMS	PIMS	FSMS
Phase	**Activity**		**ISO/IEC 15408-3:2008**	**ISO/IEC 27001:2013**	**ISO/IEC 27701:2019**	**ISO 26262-6:2018**
1	1.1	1.1.1	-	4.1, 7.3	5.2.1, 5.5.3	-
		1.1.2	-	-	-	-
	1.2	1.2.1	-	-	-	-
2	2.1	2.1.1	ASE_INT, ASE_CCL	4.3	5.2.3	-
		2.1.2	AGD_PRE, AGD_OPE	5.1, 5.3	5.3.1, 5.3.3, 6.3.1, 6.4.3	-
		2.1.3	ALC_TAT	7.1	5.5.1	-
		2.1.4	ASE_CCL, ASE_OBJ	5.2	5.3.2, 6.2.1, 6.11.2	-
		2.1.5	-	-	6.11.2	-
	2.2	2.2.1	ALC_LCD	6.2	5.4.2	5.4
		2.2.2	ASE_INT, ASE_CCL	-	6.5.1, 6.5.2	-
	2.3	2.3.1	ASE_OBJ	4.2	5.2.2, 5.4.2, 6.11.2	-
		2.3.2		-		-
3	3.1	3.1.1	-	-	-	4.4
		3.1.2	-	-		
		3.1.3	-	-		-
	3.2	3.2.1	ASE_ECD, ASE_REQ, ASE_TSS	-	6.11.1	6.4
		3.2.2		-		
	3.3	3.3.1		-		
4	4.1	4.1.1	-	-	6.12.1	-
		4.1.2	-	-		-
	4.2	4.2.1	-	-	6.12.2	-

5	5.1	5.1.1	ADV_ARC, ADV_FSP, ADV_TDS	-	6.5.1, 6.5.2	7, 8
		5.1.2	ADV_ARC, ADV_TDS	-		
	5.2	5.2.1	ASE_INT, ASE_REQ, ASE_OBJ, ASE_SPD, AVA_VAN	6.1.2		
		5.2.2	ASE_REQ, ALC_DVS	6.1.3	5.4.1	
		5.2.3	ASE_REQ, ALC_DVS	-		
	5.3	5.3.1	ADV_FSP	-		
6	6.1	6.1.1	ALC_TAT	-		8
		6.1.2	ALC_DEL, AGD_PRE	-		
	6.2	6.2.1	ADV_TDS, ADV_SPM, ADV_IMP	-		
7	7.1	7.1.1	ADV_IMP, ATE_DPT, ATE_COV, ATE_FUN, ATE_IND	-	6.11.2, 6.11.3	9, 11
		7.1.2	ATE_DPT, ATE_COV, ATE_FUN, ATE_IND	-		
		7.1.3	ATE_DPT, ATE_COV, ATE_FUN, ATE_IND	-		
		7.1.4	AVA_VAN	-		
	7.2	7.2.1	ASE_OBJ, ASE_REQ, ADV_FSP, ASE_TSS, ADV_SPM, ADV_TDS, ADV_IMP, ALC_CMC, AVA_VAN	-	6.11.2	
		7.2.2	ADV_ARC, ALC_CMC, ALC_DEL, ALC_DVS, ALC_LCD, ALC_TAT, ATE_COV, ATE_DPT, ATE_FUN, ALC_LCD	7.5	5.5.5, 6.9.1, 6.9.6	
8	8.1	8.1.1	ALC_DVS	-		-
		8.1.2		-		
		8.1.3		-		

	8.2	8.2.1	ALC_DVS	8.1	6.9.2	
	8.3	8.3.1	AGD_OPE, ALC_DVS, ALC_DEL	-	6.9.5	
9	9.1	9.1.1	ALC_CMS, ALC_CMC, ALC_DVS	9.1	6.9.4	-
		9.1.2	AGD_OPE			
	9.2	9.2.1	ALC_CMS	8.2, 9.1	6.9.6, 6.13.1	
		9.2.2				
		9.2.3		8.3, 9.1		
		9.2.4				
	9.3	9.3.1	ALC_CMS, ALC_CMC	9.3, 10.2	6.9.1, 6.9.3	
10	10.1	10.1.1	ALC_DVS	10.1	6.5.3	-
	10.2	10.2.1	ALC_DVS	-		

References

1. Bell, R. Introduction to IEC 61508. *ACM International Conference Proceeding Series* **2006**, Vol. 162, pp.3–12.
2. Barr, M. Bookout vs. Toyota. *case No. CJ-2008-7969, District Court of Oklahoma County, http://www. safetyresearch. net/Library/Bookout_v_Toyota_Barr_redacted. pdf, consultado el* **2013**, *10*.
3. Debouk, R. Overview of the 2nd Edition of ISO 26262: Functional Safety-Road Vehicles. *General Motors Company, Warren, MI, USA* **2018**.
4. Craigen, D.; Diakun-Thibault, N.; Purse, R. Defining cybersecurity. *Technology Innovation Management Review* **2014**, *4*.
5. Mössinger, J. Software in automotive systems. *IEEE software* **2010**, *27*, 92–94.
6. Miller, C.; Valasek, C. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* **2015**, 91.
7. Dehm, M. Road Vehicles' Life-Cycle : Mapping of Relevant Standards and Regulations for Automotive Cybersecurity, *The 18th International Common Criteria Conference(ICCC)* **2019.**
8. Khattri, H.; Kumar, N.; Mangipudi, V.; Mandujano, S. Hsdl: A security development lifecycle for hardware technologies. *2012 IEEE International Symposium on Hardware-Oriented Security and Trust* **2012**.
9. Salini, P.; Kanmani, S. Survey and analysis on Security Requirements Engineering. *Computers & Electrical Engineering* **2012**, *38*, 1785–1797.
10. Khou, S.; Mailloux, L.O.; Pecarina, J.M.; Mcevilley, M. A Customizable Framework for Prioritizing Systems Security Engineering Processes, Activities, and Tasks. *IEEE Access* **2017**, *5*, 12878–12894.
11. Mohammed, N.M. Exploring software security approaches in software development lifecycle: A systematic mapping study. *Computer Standards & Interfaces* **2017**, *50*, 107–115.
12. Loruenser, T. CryptSDLC: Embedding cryptographic engineering into secure software development lifecycle. *Proceedings of the 13th International Conference on Availability, Reliability and Security* **2018**.
13. Ruggieri, M.; Hsu, T.T.; Ali, M.L. Security Considerations for the Development of Secure Software Systems. *IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON),***2019**.
14. Casola, V.; Benedictis, A.D.; Rak, M.; Villano, U. A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach. *Journal of Systems and Software* **2020**, *163*, 110537–110537.
15. Venson, E.; Guo, X.; Yan, Z.; Boehm, B. Costing secure software development: A systematic mapping study. *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1–11.
16. Avizienis, A. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing* **2004**, *1*, 11–33.
17. Michailidis, A.; Spieth, U.; Ringler, T.; Hedenetz, B.; Kowalewski, S. Test front loading in early stages of automotive software development based on AUTOSAR. *2010 Design, Automation & Test in Europe Conference & Exhibition* **2010**, pp. 435–440.
18. Takahira, R.Y. Scrum and Embedded Software development for the automotive industry., **2014**.
19. Young, W.; Leveson, N.G. An integrated approach to safety and security based on systems theory. *Communications of the ACM* **2014**, *57*, 31–35.
20. Kriaa, S.; Pietre-Cambacedes, L.; Bouissou, M.; Halgand, Y. A survey of approaches combining safety and security for industrial control systems. *Reliability Engineering & System Safety* **2015**, *139*, 156–178.
21. Wolff, C. AMALTHEA-Tailoring tools to projects in automotive software development. *IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* **2015**, 2.
22. Schmittner, C.; Ma, Z.; Schoitsch, E. Combined safety and security development lifecylce. *IEEE 13th International Conference on Industrial Informatics (INDIN)* **2015**.
23. Sabaliauskaite, G.; Adepu, S.; Mathur, A. A six-step model for safety and security analysis of cyber-physical systems. *International Conference on Critical Information Infrastructures Security*. Springer, **2016.**
24. Pricop, E.; Mihalache, S.F.; Fattahi, J. Innovative fuzzy approach on analyzing industrial control systems security. *Recent Advances in Systems Safety and Security*. Springer**, 2016**, pp. 223–239.
25. Brunner, M. Towards an integrated model for safety and security requirements of cyber-physical systems. *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)***2017**.
26. Zhang, Y. Test and Evaluation System for Automotive Cybersecurity. *IEEE International Conference on Computational Science and Engineering (CSE)* **2018**.

27. Abdo, H. A safety/security risk analysis approach of Industrial Control Systems: A cyber bowtie-combining new version of attack tree with bowtie analysis. *Computers & Security* **2018**, *72*, 175–195.
28. Yi, S. A safety-security assessment approach for communication-based train control (cbtc) systems based on the extended fault tree. *27th International Conference on Computer Communication and Networks (ICCCN)* **2018**.
29. Neureiter, Christian; ENGEL, Dominik; USLAR, Mathias. Domain specific and model based systems engineering in the smart grid as prerequesite for security by design. *Electronics*, MDPI, **2016**, 5.2: 24.
30. Koschuch, M. Safety & Security in the Context of Autonomous Driving. *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)* **2019**.
31. Chowdhury, T. Safe and secure automotive over-the-air updates. *International Conference on Computer Safety, Reliability, and Security*. Springer, **2018.**
32. Asplund, F.; McDermid, J.; Oates, R.; Roberts, J. Rapid Integration of CPS Security and Safety. *IEEEEmbedded Systems Letters* **2019**, *11*, 111–114.
33. Lisova, E.; Sljivo, I.; Causevic, A. Safety and Security Co-Analyses: A Systematic Literature Review. *IEEE Systems Journal* **2019**, *13*, 2189–2200.
34. Geismann, J.; Gerking, C.; Bodden, E. Towards ensuring security by design in cyber-physical systems engineering processes. *Proceedings of the 2018 International Conference on Software and System Process* **2018**.
35. Huang, K.; Zhou, C.; Tian, Y.C.; Yang, S.; Qin, Y. Assessing the Physical Impact of Cyberattacks on Industrial Cyber-Physical Systems. *IEEE Transactions on Industrial Electronics* **2018**, *65*, 8153–8162.
36. Fowler, D.S. A Method for Constructing Automotive Cybersecurity Tests, a CAN Fuzz Testing Example. *IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)* **2019**.
37. Oka, D.; Kengo, T.; Makila, R.; Kuipers. Integrating Application Security Testing Tools into ALM Tools in the Automotive Industry. *IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)* **2019**.
38. Verma, S. Combined Approach for Safety and Security. *International Conference on Computer Safety, Reliability, and Security*. Springer, **2019.**
39. Apvrille, L.; Li, L.W. Harmonizing safety, security and performance requirements in embedded systems, *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE* **2019.**
40. Dobaj, J. Towards Integrated Quantitative Security and Safety Risk Assessment. *International Conference on Computer Safety, Reliability, and Security. Springer*, **2019**.
41. Uslar, Mathias; ROSINGER, Christine; SCHLEGEL, Stefanie. Security by Design for the Smart Grid: Combining the SGAM and NISTIR 7628. *2014 IEEE 38th International Computer Software and Applications Conference Workshops*. IEEE, **2014**. p. 110-115.
42. Placho, Teresa. Management of automotive software updates. *Microprocessors and Microsystems*, **2020**, 78: 103257.
43. Kranabitl, Philipp. Automotive Powertrain Development Process. *Systems Engineering for Automotive Powertrain Development*, **2020**, 1-20.
44. Schmittner, Christoph. A preliminary view on automotive cyber security management systems. *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE).* IEEE, **2020**. p. 1634-1639.
45. Schmittner, Christoph. ThreatGet: Threat modeling based approach for automated and connected vehicle systems. *AmE 2020-Automotive meets Electronics; 11th GMM-Symposium*. VDE, **2020**. p. 1-3
46. Chattopadhyay, Anupam; LAM, Kwok-Yan; TAVVA, Yaswanth. Autonomous vehicle: Security by design. *IEEE Transactions on Intelligent Transportation Systems, 2020. Proceedings of the 2018 International Conference on Software and System Process*. **2018**.
47. Geismann, Johannes; GERKING, Christopher; BODDEN, Eric. Towards ensuring security by design in cyber-physical systems engineering processes. *Proceedings of the 2018 International Conference on Software and System Process*. **2018**. p. 123-127.
48. Veichtlbauer, Armin. Towards applied security-by-design for DER units. *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, **2016**. p. 1-4.
49. Skoglund, M.; Warg, F.; Sangchoolie, B. In Search of Synergies in a Multi-concern Development Lifecycle: Safety and Cybersecurity. *International Conference on Computer Safety, Reliability, and Security.* Springer, **2018**.

50. Bramberger, R.; Martin, H.; Gallina, B.; Schmittner, C. Co-engineering of Safety and Security Life Cycles for Engineering of Automotive Systems. *ACM SIGAda Ada Letters* **2020**, *39*, 41–48.
51. Win, B.D.; Scandariato, R.; Buyens, K.; Grégoire, J.; Joosen, W. On the secure software development process: CLASP, SDL and Touchpoints compared. *Information and Software Technology* **2009**, *51*, 1152–1171.
52. Microsoft. Security Development Lifecycle - SDL Process Guidance Version 5.2, **2012.**
53. Kissel, R.; others. Sp 800-64 rev. 2. security considerations in the system development life cycle, **2008.**
54. OWASP. Comprehensive, lightweight application security process, **2006.**
55. Sae j3061-cybersecurity guidebook for cyber-physical automotive systems. *SAE Vehicle Electrical System Security Committee* **2016**.
56. Schmittner, C.; Macher, G. Automotive Cybersecurity Standards-Relation and Overview. *International Conference on Computer Safety, Reliability, and Security*. Springer, **2019.**
57. UNECE. Draft Cyber Security Regulation - final clean version, **2020.**
58. Hunjan, H. ISO/SAE 21434 Automotive Cyber-Security Engineering. *Presentation, Renessas Electronics LTD* **2018**.
59. Schmittner, C.; Griessnig, G.; Ma, Z. Status of the development of ISO/SAE 21434. *European Conference on Software Process Improvement.* Springer, **2018.**
60. Blyler, J. Software-Hardware Integration in Automotive Product Development, **2014.**
61. LDRA. BUILD SECURITY INTO THE CONNECTED CAR DEVELOPMENT LIFE CYCLE, **2017**.
62. Schoitsch.; Erwin.; others. The need for safety and cyber-security co-engineering and standardization for highly automated automotive vehicles. *Advanced Microsystems for Automotive Applications 2015* Springer: Cham**, 2015**; pp. 251–261.
63. Sabaliauskaite, G.; Mathur, A.P. Aligning cyber-physical system safety and security. *Complex Systems Design & Management Asia.* Springer, Cham, **2015**. 41-53.
64. Synopsys. What is ASIL?
65. Schmittner, C.; Ma, Z. Towards a framework for alignment between automotive safety and security standards. *International Conference on Computer Safety, Reliability, and Security.* Springer, **2014.**
66. Miller, J.D. Automotive System Safety: Critical Considerations for Engineering and Effective Management, **2019**.
67. Mellado, D.; Fernández-Medina, E.; Piattini, M. A common criteria based security requirements engineering process for the development of secure information systems. *Computer Standards & Interfaces* **2007**, *29*, 244–253.
68. Yin, L.; Qiu, F.L. A novel method of security requirements development integrated common criteria. *International Conference On Computer Design and Applications* **2010**, *5*.
69. Mellado, D.; Blanco, C.; Sánchez, L.E.; Fernández-Medina, E. A systematic review of security requirements engineering. *Computer Standards & Interfaces* **2010**, *32*, 153–165.
70. Houmb, S.H.; Islam, S.; Knauss, E.; Jürjens, J.; Schneider, K. Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec. *Requirements Engineering* **2010**, *15*, 63–93.
71. Mesquida, A.L.; Mas, A. Implementing information security best practices on software lifecycle processes: The ISO/IEC 15504 Security Extension. *Computers & Security* **2015**, *48*, 19–34.
72. Li, H. Fesr: A framework for eliciting security requirements based on integration of common criteria and weakness detection formal model. *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)* **2017**.
73. Barafort, B.; Mesquida, A.L.; Mas, A. Integrating risk management in IT settings from ISO standards and management systems perspectives. *Computer Standards & Interfaces* **2017**, *54*, 176–185.
74. Barafort, B.; Mesquida, A.L.; Mas, A. Integrated risk management process assessment model for IT organizations based on ISO 31000 in an ISO multi-standards context. *Computer Standards & Interfaces* **2018**, *60*, 57–66.
75. Lee, Y.; Lee, J.; Lee, Z. Integrating Software Lifecycle Process Standards with Security Engineering. *Computers & Security* **2002**, *21*, 345–355.
76. Horie, D. A new model of software life cycle processes for consistent design, development, management, and maintenance of secure information systems. *Eighth IEEE/ACIS International Conference on Computer and Information Science* **2009**.

77. Amara, N.; Huang, Z.; Ali, A. Modelling Security Requirements for Software Development with Common Criteria. *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, **2019**.
78. Hatton, L. Safer language subsets: an overview and a case history, MISRA C. *Information and Software Technology* **2004**, *46*, 465–472.