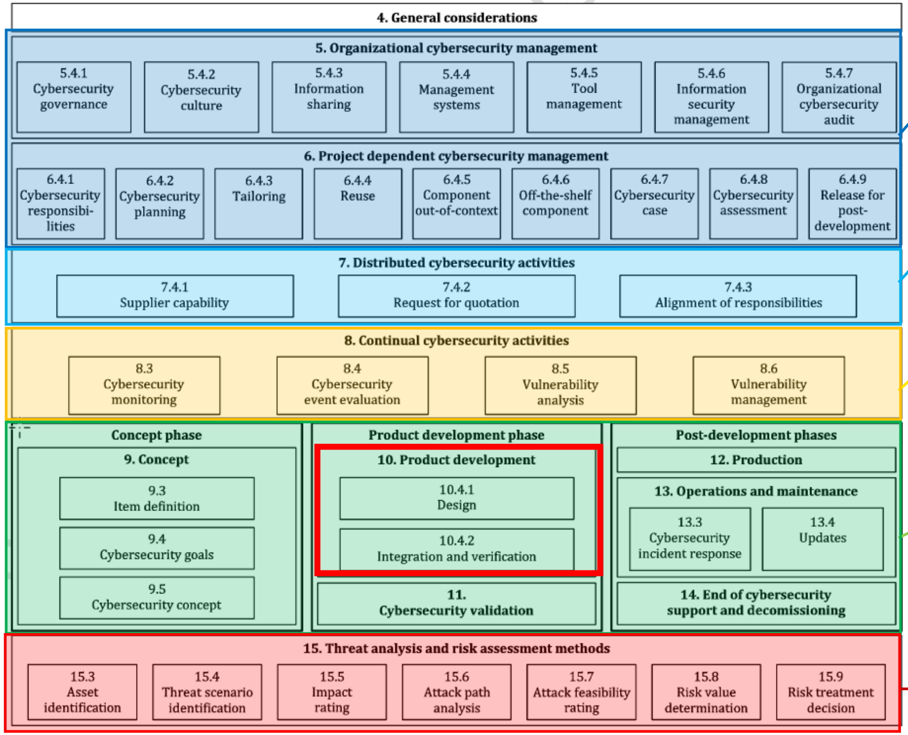
Clause 10:

Product development

Structure of ISO 21434



Overall & project specific management processes
(similar to ISO 26262) :
- Management Systems
- Policies
- Preparation for assessment

Distributed CS activities
- Define interfaces between customer, supplier, third parties..

Continual CS Activities :
- Requirements for continuous monitoring of CS relevant information
- Framework for analysis and management of vulnerabilities

Concept, Development and Post-Development
- Add-on of CS relevant activities during concept and development :
  - Establishment of CS goals and requirements
  - TARA and vulnerability analysis during development
- Consideration of post-development requirements (during of after production, decommissioning …)
- Definition of post development processes (Production, Incident response, Update)

TARA : Threat Analysis and Risk Assessment
- Describes the steps to perform a robust risk analysis on the system
- Complex process to be performed multiple times and for multiple assets

Clause 10: Product development

Definitions

Architectural design: representation that allows for identification of components, their boundaries, interfaces, and interactions

Item: component or set of components that implement a function at the vehicle level

Asset: an object or an item that has cybersecurity properties upon compromising can lead to severe damage to an item's physical value and its stakeholder

Component: part of an item that is ideally separated by logically and separable

Function(s): intended behavior of the item during the lifecycle phase and vehicle functionality that's defined for an item

Verification: confirmation, through the provision of objective evidence, that specified requirements have been fulfilled

Validation: confirmation, through the provision of objective evidence, that the cybersecurity goals of the item are adequate and are achieved

Penetration testing: cybersecurity testing in which real-world attacks are mimicked to identify ways to compromise cybersecurity goals

Out-of-context: not developed in the context of a specific item

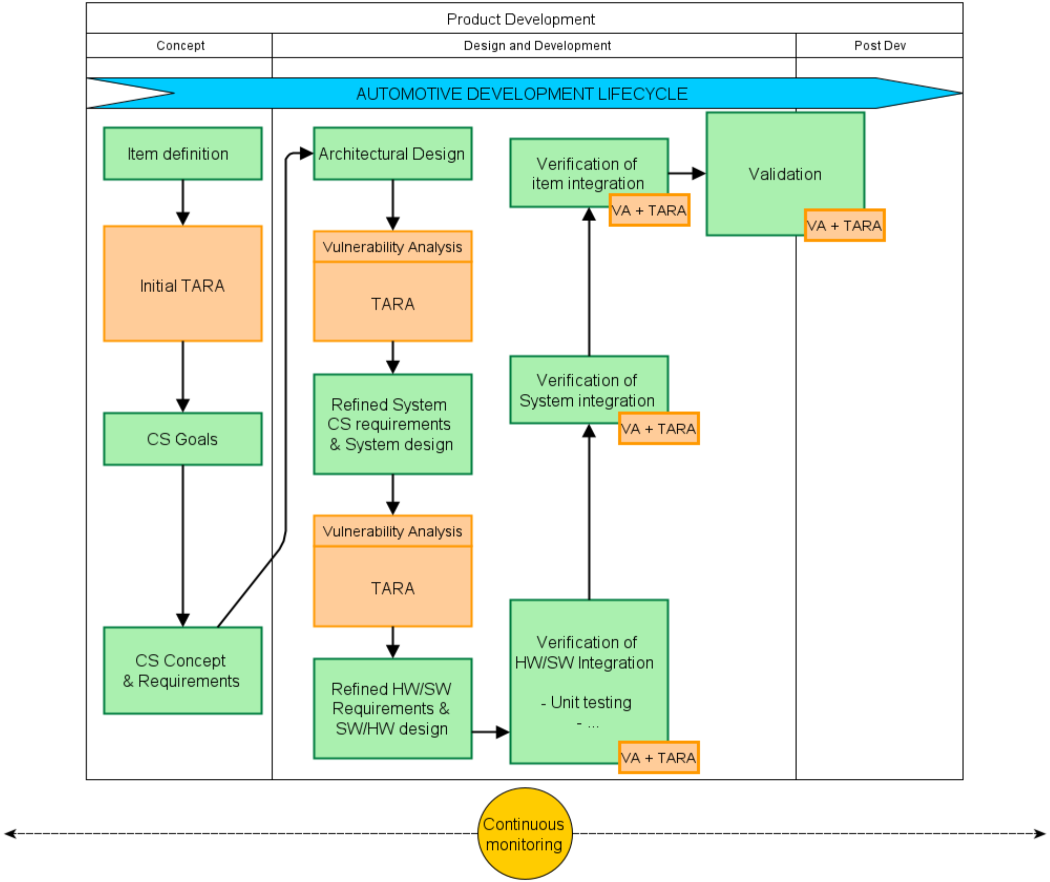
Clause 10: Product development

DEKRA DIGITAL

General

- This clause helps in understanding how the item is developed by considering all the cybersecurity and safety requirements from the concept phase
- Different phases of integration and verification activities are explained
- The dependencies between concept, development, and validation phases can be clearly understood
- SDLC (software development life cycle) V-Model is considered as an example in order to explain how the product development phases are carried over
- Note: Product development can be carried out using other methodologies apart from V-Model (ex: agile, continuous integration, waterfall, etc.)

Clause 10: Product development



Clause 10: Product development

DEKRA DIGITAL

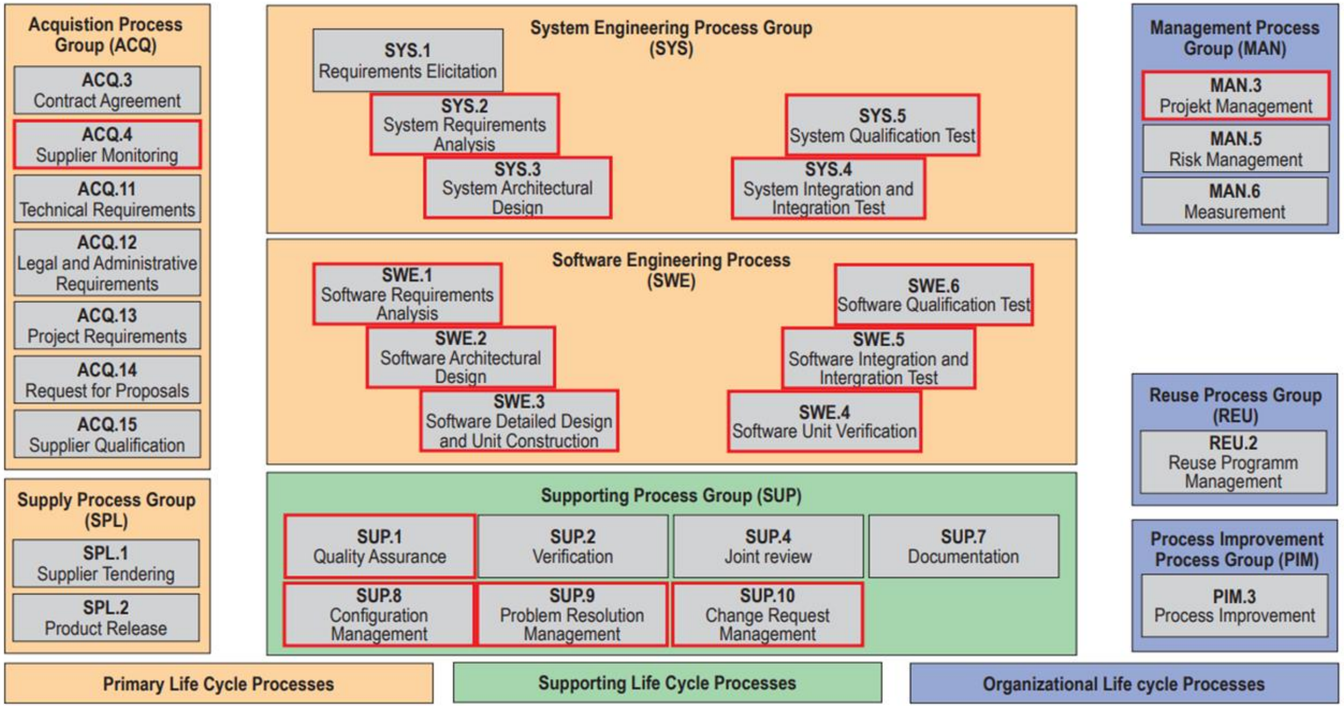
Automotive SPICE (ASPICE)

- ASPICE stands for **A**utomotive **S**oftware **P**rocess **I**mprovement and **C**apability d**E**termination
- ASPICE is a standard framework that helps in bringing coordination between different stakeholders such as OEMs, Tier-1, suppliers, technology providers, etc. with respect to definition, implementation, process setup for software and hardware development
- ASPICE can be seen as
  - Measurement framework
  - Process assessment model
  - Process reference model
  - Best practices for embedded system development
- The technical standard for ASPICE is ISO/IEC 15504, tailored specifically to the automotive industry

Clause 10: Product development

Automotive SPICE (ASPICE)

- According to VDA, **Verband der Automobilindustrie e.V.** (Association of the Automotive Industry) scope, ASPICE Assessments predominantly focus on 16 Key Processes



Clause 10: Product development

ASPICE for OEMs

- OEM or an Original Equipment Manufacturer could use the ASPICE framework to assess their supplier's quality and capability during selection
- ASPICE helps OEM in defining system development process which improves the process capability

ASPICE for Suppliers

- Suppliers or service providers use ASPICE framework in the system development process to ensure the product quality
- It shortens the release schedule and reduces cost impact on the product development
- Early identification of quality issues in the later stages of product development

Clause 10: Product development

ASPICE and Functional Safety (ISO 26262)

- ISO 26262 covers functional safety standards for vehicles
- It incorporates safety analysis methods that account for random and systematic errors in electrical and electronic systems and is broadly adopted worldwide
- ASPICE and ISO 26262 are complementary and do overlap in places, they ultimately serve different purposes
- Key distinctions between ASPICE and ISO 26262

ASPICE	ISO 26262
Framework and process for system development	Standards to safety-critical vehicle systems
Requirements analysis considers cost and schedule impacts on development	No concern with schedule or cost and safety is the primary focus
Process implementation at organization, project and system levels	Safety assessments at organization, project and system levels

Clause 10: Product development

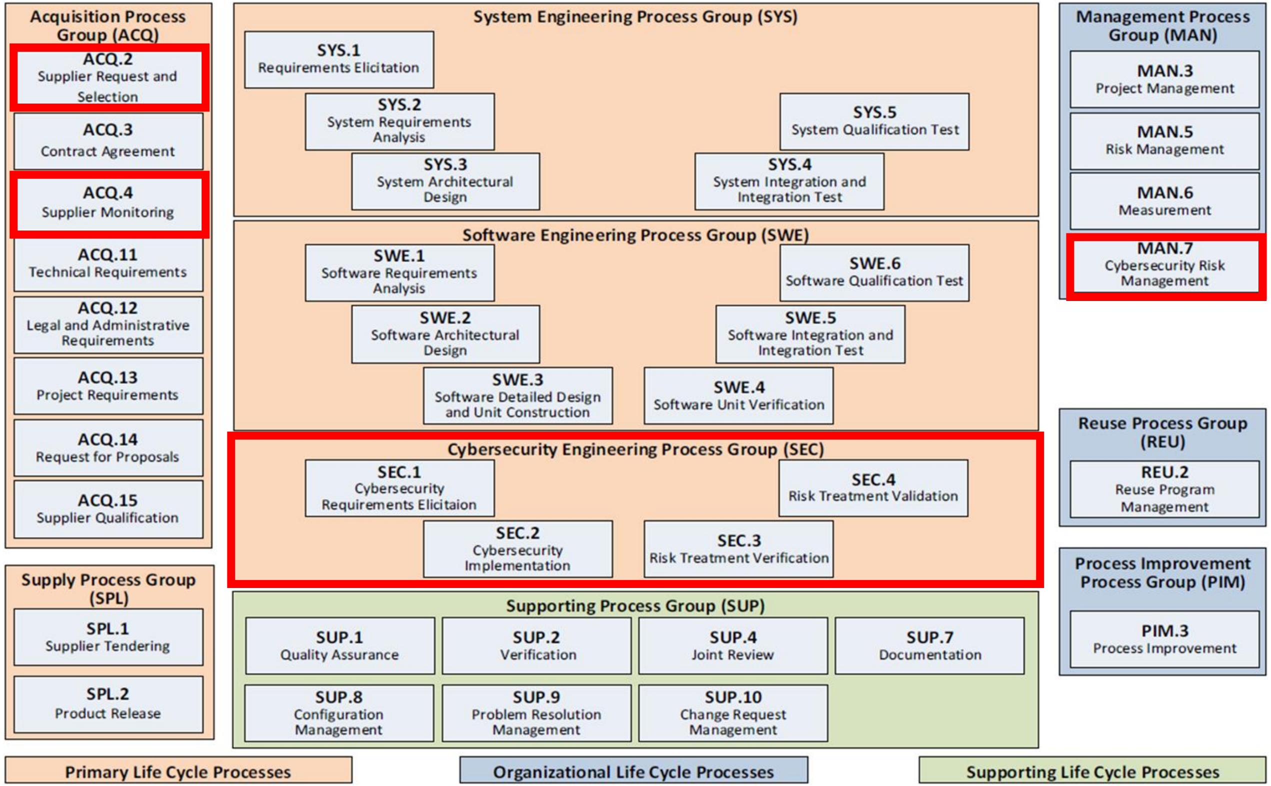
ASPICE for Cybersecurity

- Created to support UNECE R155 using ASPICE as a proven assessment model
- To identify process-related product risks in Cybersecurity projects
- Additional 7 processes have been added specifically for Cybersecurity
- ASPICE for Cybersecurity enables the evaluation of cybersecurity-relevant development processes using stand alone assessment or combined with VDA scope
- ASPICE for Cybersecurity cannot be seen as a standalone model but as an extension

Clause 10: Product development

ASPICE for Cybersecurity

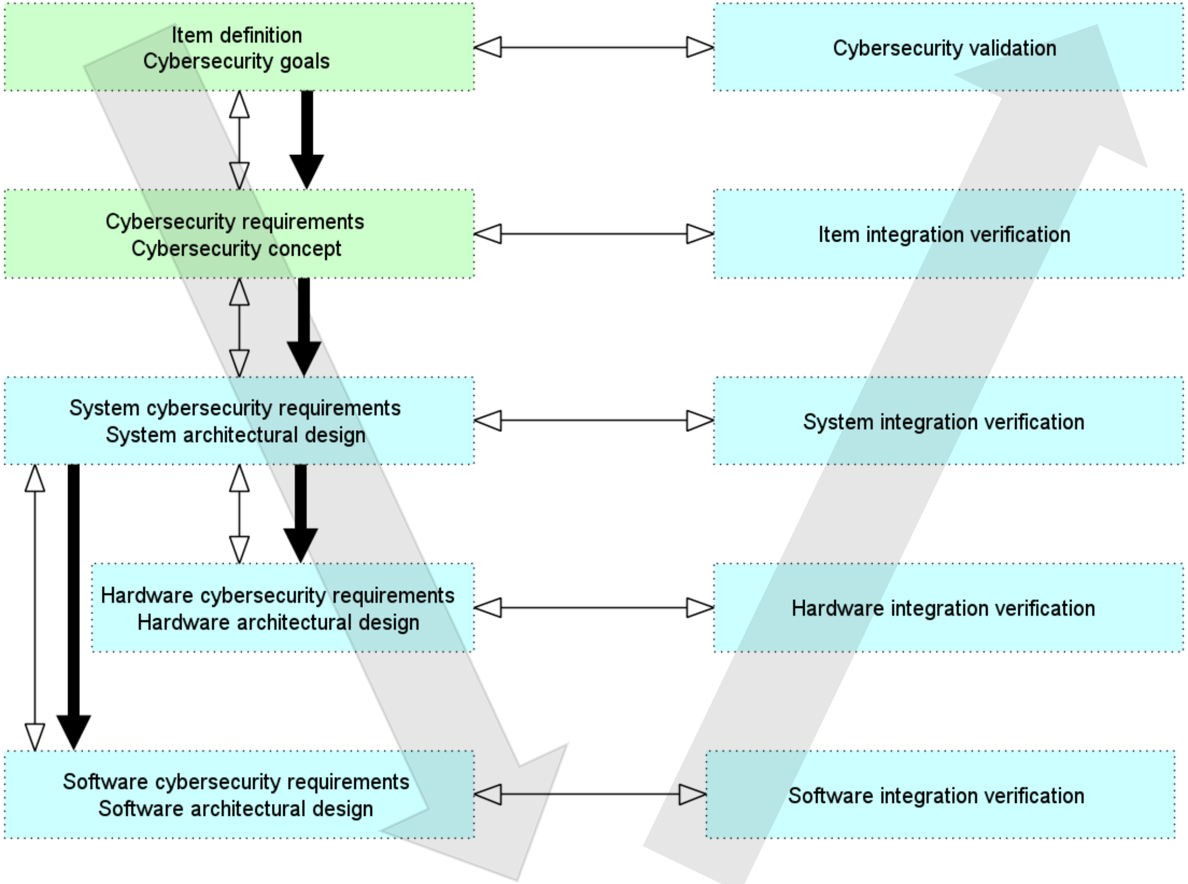
- Scope of ASPICE for Cybersecurity
- The red highlighted processes are newly added or modified for Cybersecurity requirements



Clause 10: Product development

Concept phase

Development phase



Clause 10: Product development

Objectives

- Define cybersecurity specifications for the related components which are not present in the existing architectural design
- Verification of defined specifications from the higher levels of architectural abstraction
- Identifying weaknesses in the component based on vulnerability analysis and management
- Evidence (e.g. documentation) to prove the implementation and integration of components are according to desired cybersecurity specifications

Clause 10: Product development

Prerequisites

- For the components in which the development is yet to be started following information shall be available
  - item definition
  - cybersecurity concept
  - architectural design
  - established cybersecurity controls
- Cybersecurity specifications from higher levels of architectural abstraction are considered for the components under development along with the following relevant information
  - cybersecurity requirements
  - interface specification (external)
  - information assumed on the operational environment

Clause 10: Product development

DEKRA DIGITAL

Design

- Architectural design is a blueprint for an item development based on the requirements which can have a collection of both hardware and software components
- In the design phase, considering cybersecurity aspects for the safe design and its verification is crucial

Key requirements:

1. Cybersecurity specifications (requirements and architectural design) shall be defined based on:
   - **Higher levels of architectural abstraction**
   - **Component wise requirement specification and secure external connections or interfaces**
   - **Existing architectural design**, if applicable – e.g. interfaces between components and subcomponents can be reused based on the requirements and their usage
   - For hardware dependent components, information such as **configuration, calibration values, etc.,** can be considered to fulfill the cybersecurity requirements
   - **Known weaknesses and vulnerabilities** from reused components can be considered and managed accordingly.
   - If identified weaknesses can be resolved by changes of the architectural design, then vulnerability analysis would not be needed

Clause 10: Product development

2. Defined **cybersecurity requirements shall be allocated to components** of the architectural design by ensuring cybersecurity control are in place for the component post development
3. For cybersecurity specification and implementation, in any phase (modeling, design, implementation), **programming language** or notations which are used must contain the following aspects while selecting
   - An unambiguous and comprehensible definition in both syntax and semantics
   - Support for the achievement of modularity, abstraction, and encapsulation
   - support for the use of secure design and implementation techniques e.g.: ASPICE
   - ability to integrate already existing components.
   - resilience of the language against vulnerabilities due to its improper use.
     coding guidelines for the software development based on the chosen programming language.

Clause 10: Product development

DEKRA DIGITAL

4. Some of the **criteria for suitable design, modeling and programming languages**:
   - Use of language subsets
   - Enforcement of strong typing and/or
   - Use of defensive implementation techniques
5. In order to avoid or minimize the weaknesses, a verified, **established and trusted design** and implementation principles must be chosen
6. The defined cybersecurity specifications **shall be verified** to ensure **completeness, correctness, and consistency** with the cybersecurity specifications from higher levels of architectural abstraction. Verification methods can include:
   - review
   - analysis
   - simulation, and/or
   - prototyping

Clause 10: Product development

DEKRA DIGITAL

Integration and verification

- Integration and verification activities are mainly done in order to verify whether the developed component fulfills the defined cybersecurity specification and to potentially identify new vulnerabilities

Key requirements:

1. Integration and verification can be done on component level, as well as vehicle level
2. Some of the following verification methods can be used in the verification process:
   - requirements-based test
   - interface test
   - resource usage evaluation
   - verification of the control flow and data flow
   - dynamic analysis, and/or
   - static analysis.
3. Test cases and test environment for integration and verification activities can be selected based on the verification objectives that need to be fulfilled

Clause 10: Product development

4. Evaluation of test coverage is done based on the coverage metrics defined. This helps in knowing the sufficiency of the test activities
5. Following testing methods can be considered :
   – functional testing
   – vulnerability scanning
   – fuzz testing, and/or
   – penetration testing
7. Methods for deriving test cases can include:
   – analysis of requirements
   – generation and analysis of equivalence classes
   – boundary values analysis, and/or
   – error guessing based on knowledge or experience

Clause 10: Product development

DEKRA DIGITAL

Item integration

- Testing of the considered item
- Verification of CS requirements established at concept phase
- (On road / On real life simulation)
- Design review

System integration

- Interface testing
- Resource consumption testing
- Performance testing
- Functional and vulnerability testing
- Design review

HW integration

- Verification of the Hardware CS Requirements (component test)
- Design reviews
- HiL environment
- Functional and vulnerability testing

SW integration

- Verification of the Software CS requirements (module test, white-box testing)
- Code analysis
- Code review, inspection
- Design review
- SIL environment

Clause 10: Product development

DEKRA DIGITAL

Cybersecurity assurance levels (CAL)

- In product development CAL is applied for methods and measures such as verification methods, integration, deriving test cases
- The components shall be developed in accordance with the highest CAL for those requirements
- The independence scheme based on ISO 26262 (ASIL) can be used in order to have an unbiased viewpoint and avoid conflict of interest for cybersecurity activities
- Three levels I1, I2 and I3 are used to indicate the level of independence that can be applied to cybersecurity activities
  - I1: the activity is performed by a different person in relation to the person(s) responsible for the creation of the considered work product
  - I2 : the activity is performed by a person who is independent of the team that is responsible for the creation of the considered work product(s), i.e., by a person reporting to a different direct superior
  - I3: the activity is performed by a person who is independent, regarding management, resources, and release authority, from the department responsible for the creation of the considered work product(s)

Clause 10: Product development

Example assigning CAL levels for independent cybersecurity activities by considering Headlamp systems as an example

Activity	Requirements	CAL1	CAL2	CAL3	CAL4
Verification	Architectural design: • Review • Analysis	I1	I1	I2	I2
	Completeness and correctness of cybersecurity goals	I1	I1	I2	I2
	Implementation and integration of components according to: • Cybersecurity specifications • Modeling and coding guidelines	I1	I1	I2	I2
Assessment	• Cybersecurity assessment to check for process implementation agianst cybersecurity plan • Item or component decided for develpment should fulfill all the cybersecurity objectives	-	I1	I2	I3

Clause 10: Product development

DEKRA DIGITAL

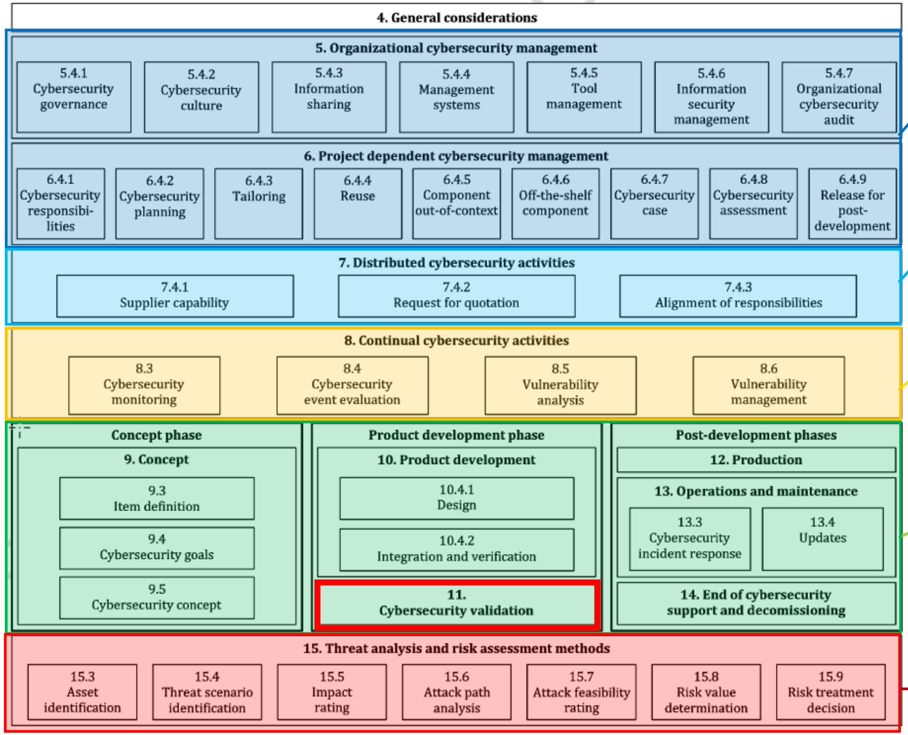
Summary of work products

- [WP-10-01] Cybersecurity specifications
- [WP-10-02] Cybersecurity requirements for post-development
- [WP-10-03] Documentation of the modelling, design or programming languages and coding guidelines, if applicable
- [WP-10-04] Verification report for the cybersecurity specifications
- [WP-10-05] Weaknesses found during product development
- [WP-10-06] Integration and verification specification
- [WP-10-07] Integration and verification report

Clause 11:

Cybersecurity validation

Structure of ISO 21434



Overall & project specific management processes

(similar to ISO 26262) :

- Management Systems
- Policies
- Preparation for assessment

Distributed CS activities

- Define interfaces between customer, supplier, third parties..

Continual CS Activities :

- Requirements for continuous monitoring of CS relevant information
- Framework for analysis and management of vulnerabilities

Concept, Development and Post-Development

- Add-on of CS relevant activities during concept and development :
  - Establishment of CS goals and requirements
  - TARA and vulnerability analysis during development
- Consideration of post-development requirements (during of after production, decommissioning …)
- Definition of post development processes (Production, Incident response, Update)

TARA : Threat Analysis and Risk Assessment

- Describes the steps to perform a robust risk analysis on the system
- Complex process to be performed multiple times and for multiple assets

Clause 11: Cybersecurity validation

DEKRA DIGITAL

Objectives:

- Confirmation of the fulfillment of the cybersecurity goals and claims established in the concept phase
- Validation is typically be done by analysis and by testing
- All the risks have been managed in a way that the residual risk is accepted
- All vulnerabilities have been identified and managed as needed
- Weaknesses identified during the validation activities are analyzed for vulnerabilities and identified vulnerabilities are managed
- Validation activities can include testing processes such as security, penetration form of testing to increase the effectiveness
- Validation report to demonstrate that the CS goals have been reached and all potential risks have been managed.

Clause 11: Cybersecurity validation

VALIDATION

Planning
- Test plan
- Test strategy
- Quality assurance plan

Design
- Test cases
- Test data
- Checklists
- Test environment configuration

Execution and Reporting
- Test protocols
- Defect reports
- Test report results

Clause 11: Cybersecurity validation

Test parameters (T1 & T2):

- T1 -testing parameter set 1:
  - functional testing based on requirements
  - vulnerability scanning for known vulnerabilities
  - fuzz testing with a random selection of inputs
  - penetration testing assuming moderate attacker expertise, knowledge of the item or component and/or resources
- T2- testing parameters set 2:
  - functional testing based on requirements and interactions between components
  - vulnerability scanning for known vulnerabilities
  - fuzz testing with an increased number of test case iterations and/or adaptive selection of inputs
  - penetration testing assuming higher attacker expertise, knowledge of the item or component and/or resources

Clause 11: Cybersecurity validation

Example assigning CAL levels to determine parameters for testing methods in validation phase

Activity	Requirements	CAL1	CAL2	CAL3	CAL4
Functional testing	• To minimize the unidentified weaknesses and vulnerabilities remaining in the component • Adequacy of the cybersecurity goals with respect to the threat scenarios and corresponding risk	T1	T1	T2	T2
Vulnerability scanning		T1	T1	T1	T1
Fuzz testing		-	T1	T2	T2
Penetration testing		-	-	T1	T2

Clause 11: Cybersecurity validation

DEKRA DIGITAL

Penetration testing

- Penetration or pen-testing is a form of ethical hacking conducted to find vulnerabilities present in a system that can be exploited by any external agents
- The output of penetration testing can be used to harden systems to deny entry to external agents
- Pen testing can also be termed as a simulated cyber attack on your own system to know the loopholes and manage it accordingly
- Black, grey, and white box testing of the system are also part of pentesting

Clause 11: Cybersecurity validation

Penetration testing phases

Preliminary analysis
- Target system intelligence
- Gather data for attack

Scanning
- Vulnerability analysis
- Target environment configuration
- Scan for vulnerabilities

Access gain
- Launch attack on the target system
- Gain control over the system

Access maintain
- To be persistently within the target environment
- Stealth mode in host environment
- Asset modification

Covering tracks
- Removal of changes to return to the state of non-recognition by the host
- Modification of entries such as log files

Penetration test report
- Document the process and tools used
- Detailed analysis of exploits and vulnerabilities in the system
- Weaknesses and probable solutions to manage it

Clause 11: Cybersecurity validation

DEKRA DIGITAL

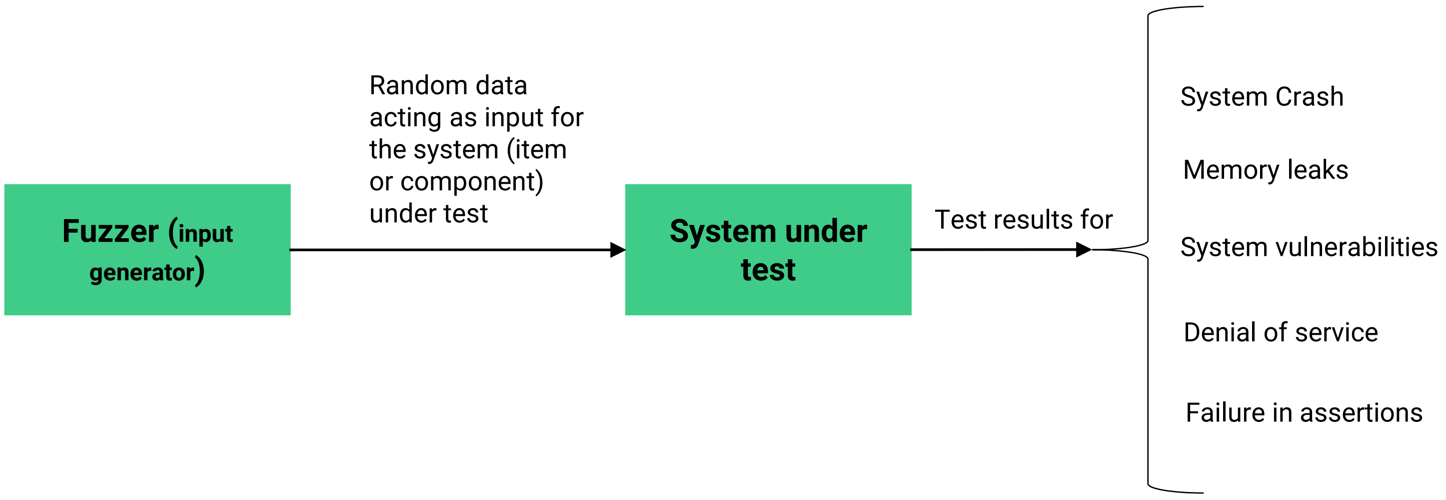
Fuzz Testing

- Fuzz testing is an automated software testing technique
- Involves inputting massive amounts of random data, called fuzz, to the system being tested to make it crash or break through its defenses
- Helps to find hackable software bugs by randomly feeding invalid and unexpected inputs and data into a component or item in order to find coding errors and security loopholes
- It exploits the system's vulnerabilities which can be used by hackers, so it can be fixed in due time

Clause 11: Cybersecurity validation

DEKRA DIGITAL

Fuzz Testing



Clause 11: Cybersecurity validation

Vulnerability scanning

- Vulnerability scanning is an automated process of proactively identifying network, application, and security vulnerabilities
- A vulnerability scan detects and classifies system weaknesses in networks and communications equipment and predicts the effectiveness of countermeasures
- Vulnerability scanning uses a piece of software running from the standpoint of the person or organization inspecting the attack surface in question
- Vulnerability scanning can be conducted in the following ways
  - **Environmental scans**: The type of system we are dealing with for the vulnerability scan such as cloud-based, IoT devices etc.
  - **Internal scans**: These scans identify the vulnerabilities inside the network to avoid damage, and allow the organization to tighten systems, application security that is not exposed by external scans
  - **External scans**: Target the areas of an IT ecosystem that are exposed to the internet, or not restricted for internal use. Including applications, ports, services, and networks that are accessed by external customers or users.

Clause 11: Cybersecurity validation

Vulnerability scanning

- How does vulnerability scan work?
  - Depending on the type of scan the vulnerability platform uses, various techniques and tactics will be leveraged to elicit a response from devices within the target scope. Based on the device’s reactions, the scanner will attempt to match the results to a database and assign risk ratings (severity levels) based on those reactions.
  - Vulnerability scanners can be configured to scan all network ports, detecting and identifying password breaches as well as suspicious applications and services. The scanning service reports security fixes or missing service packs, identifies malware as well as any coding flaws, and monitors remote access.

Clause 11: Cybersecurity validation

Summary of work products

- [WP-11-01] Validation report

DEKRA

DEKRA DIGITAL

Thank you for your attention