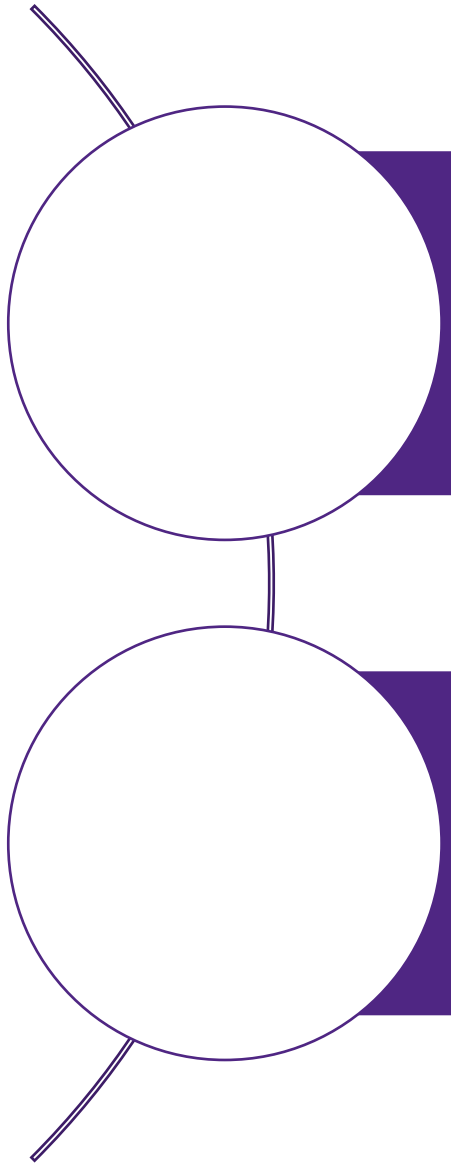
SYNOPSYS®
Silicon to Software™

Secure Automotive Software Development in the Age of ISO/SAE 21434

Dennis Kengo Oka
Principal Automotive Security Strategist

Chris Clark
Solutions Architect – Automotive Software & Security

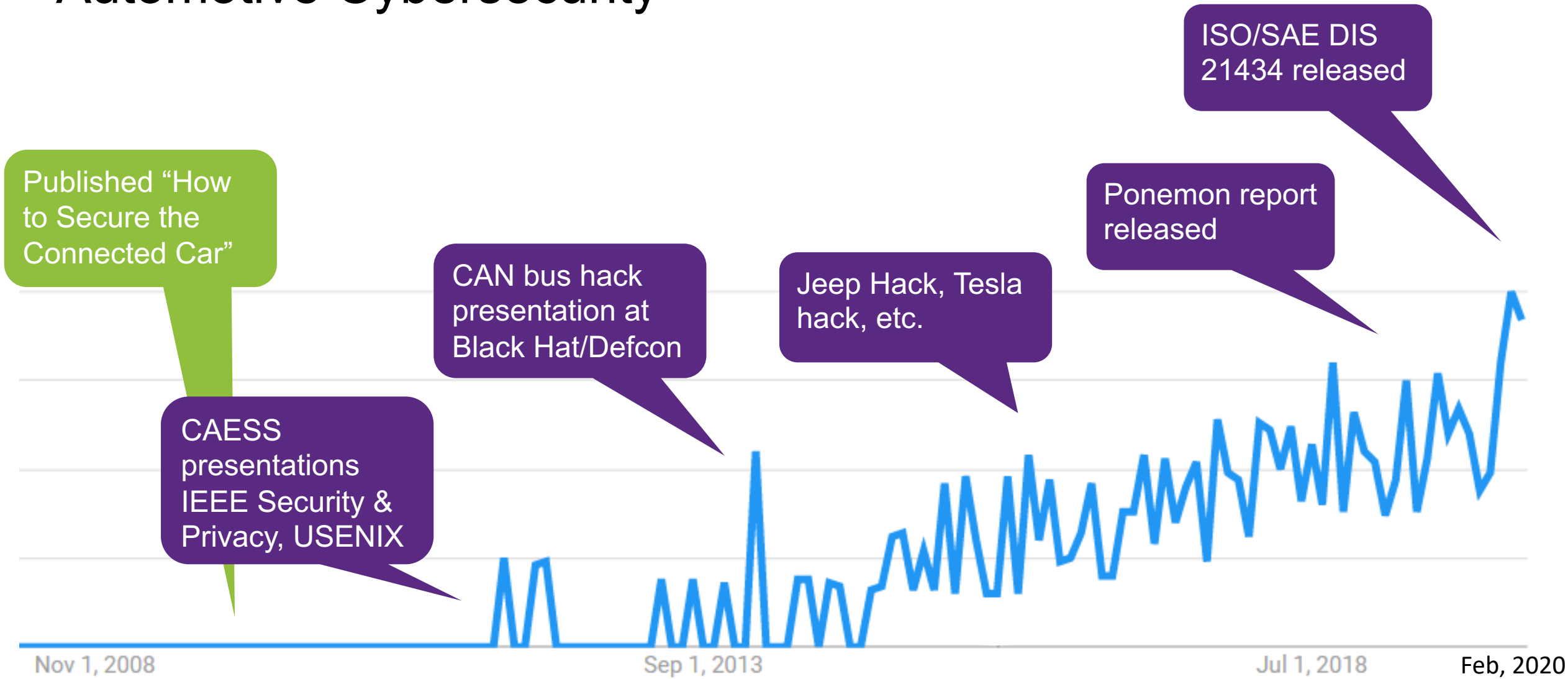


There are risks in automotive security that can cause serious consequences

New standards and regulations provide guidance for cybersecurity affecting processes and solutions

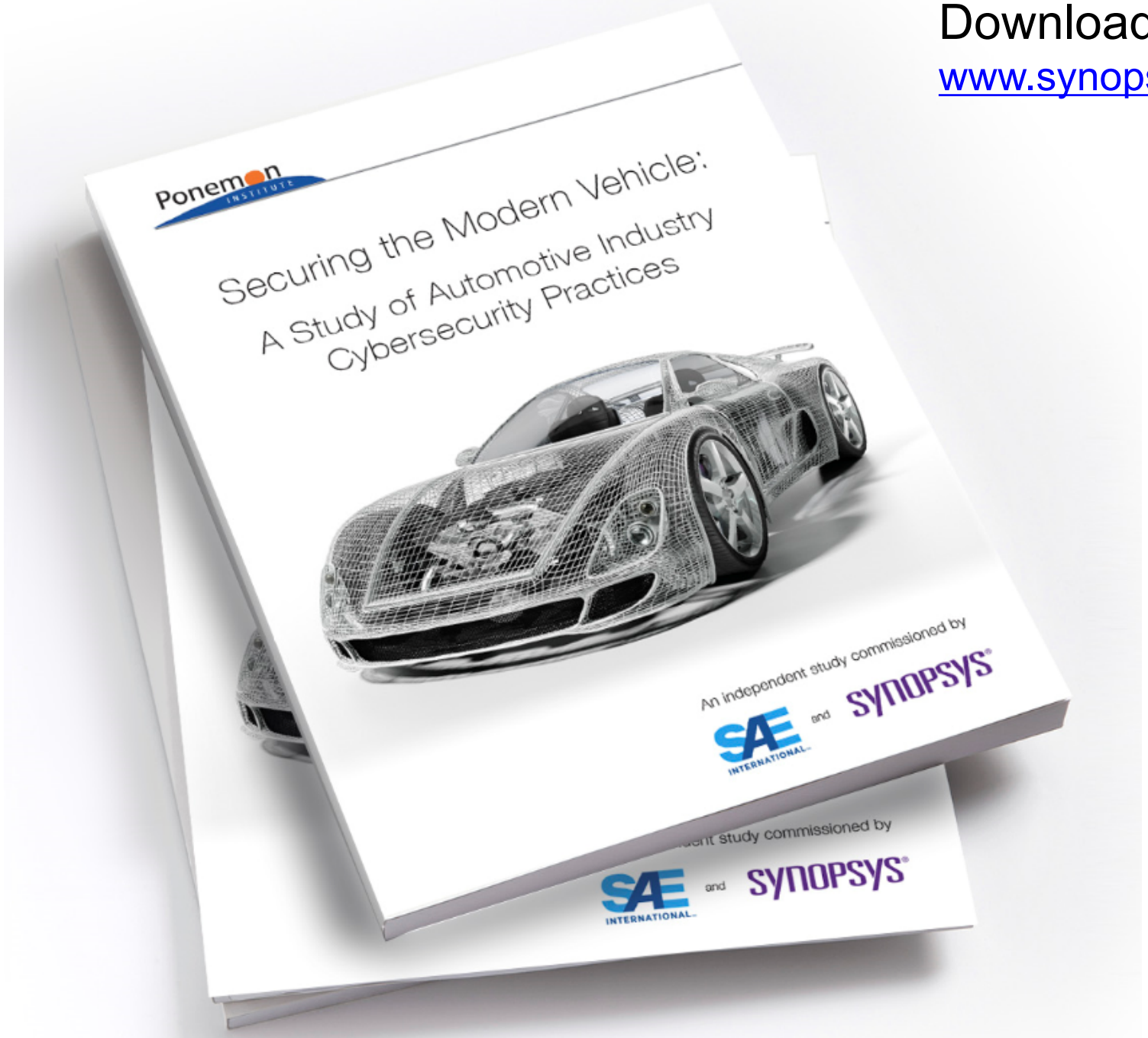
There are risks in automotive security that can cause serious consequences

Automotive Cybersecurity

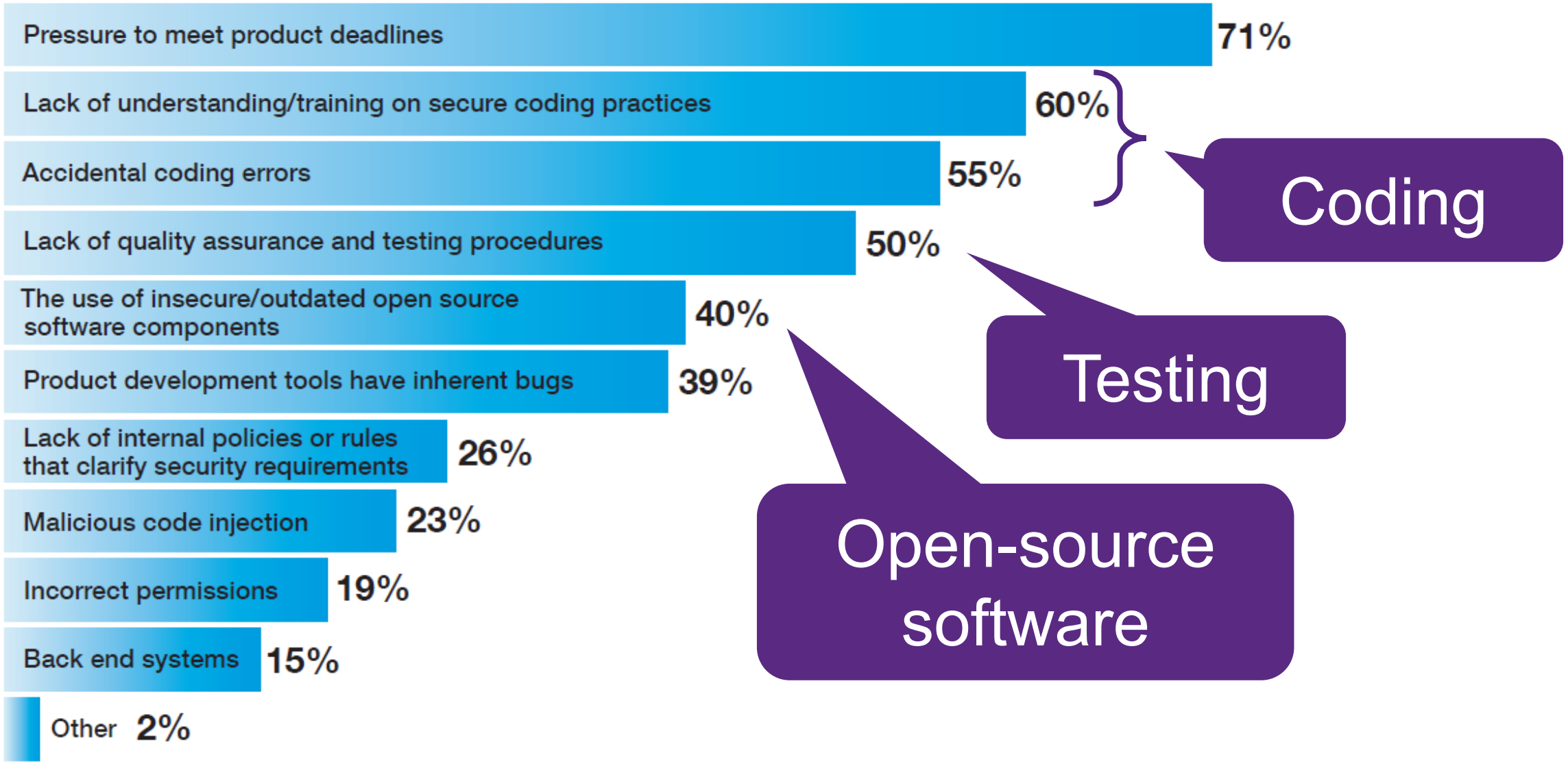


Ref: https://trends.google.com/trends/explore?date=all&q=automotive%20cybersecurity

Download Link:
www.synopsys.com/auto-security



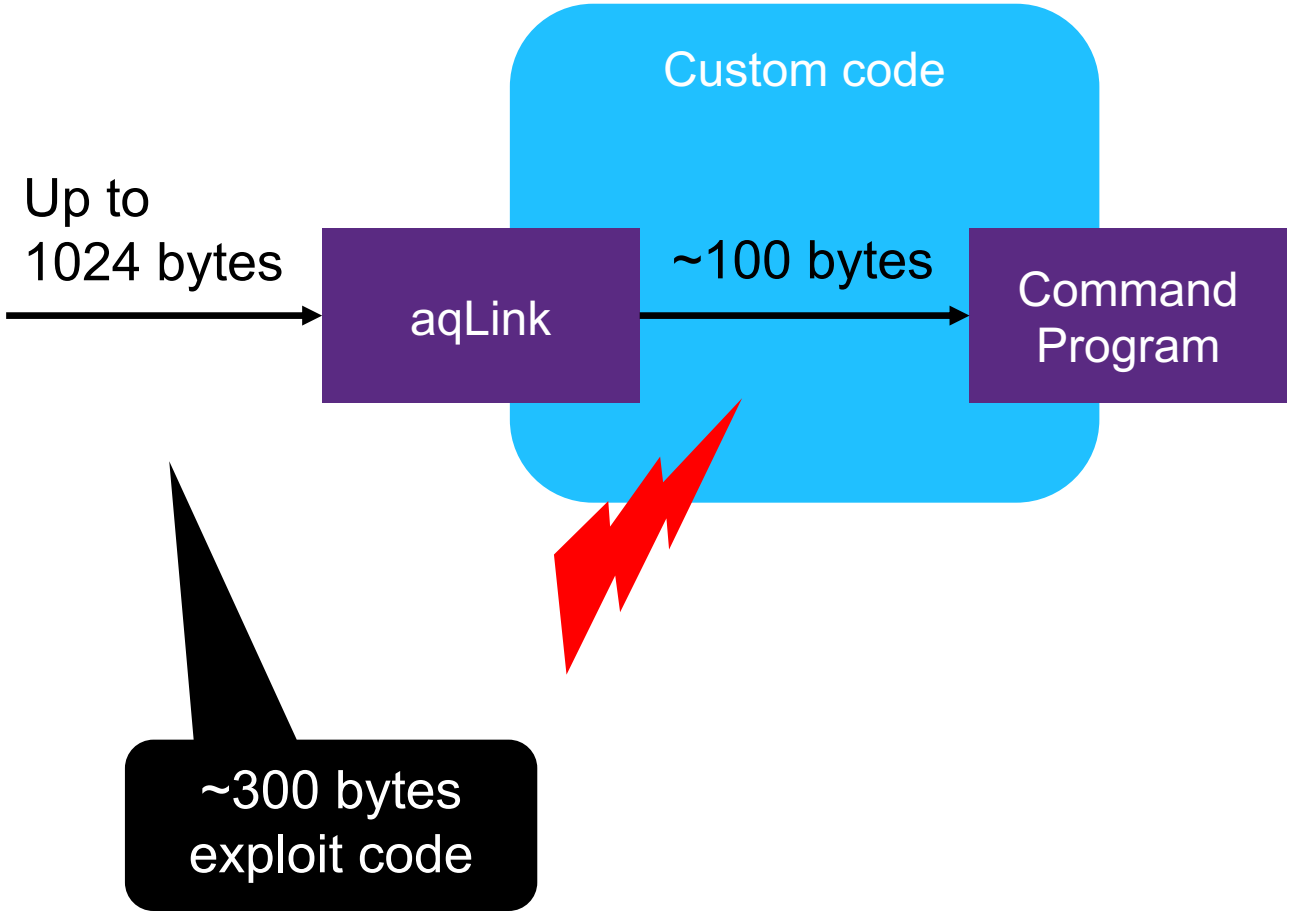
What are the primary factors that lead to vulnerabilities in automotive software/technology/components?



Ref: Ponemon Report "Securing the Modern Vehicle: A Study of Automotive Industry Cybersecurity Practices"

Coding Example – Buffer Overflow

- aqLink software to create link between telematics unit and remote telematics call center
- aqLink code explicitly supports packet sizes up to 1024 bytes
- Custom code that glues aqLink to the Command Program assumes packets ~100 bytes
- Buffer overflow vulnerability that is remotely exploitable (full control of sending messages on CAN bus)



Ref: Stephen Checkoway, et al. 2011. Comprehensive experimental analyses of automotive attack surfaces. In Proceedings of the 20th USENIX conference on Security

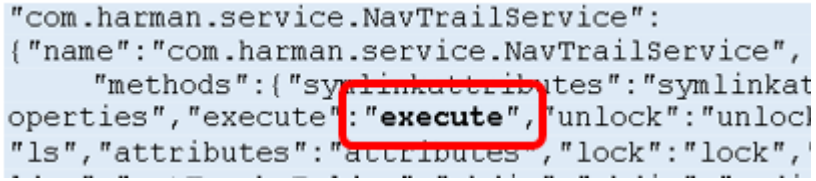
Testing Example – Lack of cybersecurity testing

- Vehicle connected to Cellular Sprint Network
  - 21.0.0.0/8 or 25.0.0.0/8 network address space
  - Sprint was not blocking intra-device communication
  - Scanned the network with nmap and found 2695 vehicles with listening D-Bus service (port 6667)

- Connect to D-Bus service on port 6667（inter-process communication (IPC), remote procedure call (RPC) mechanism
  - No authentication

```
telnet 192.168.5.1 6667
Trying 192.168.5.1...
Connected to 192.168.5.1.
Escape character is '^]'.
AUTH ANONYMOUS
OK 4943a53752f52f82a9ea4e6e00000001
BEGIN
```

- Exploit the D-bus service
  - NavTrailService has an “execute” method
  - Execute arbitrary command and gain remote root shell

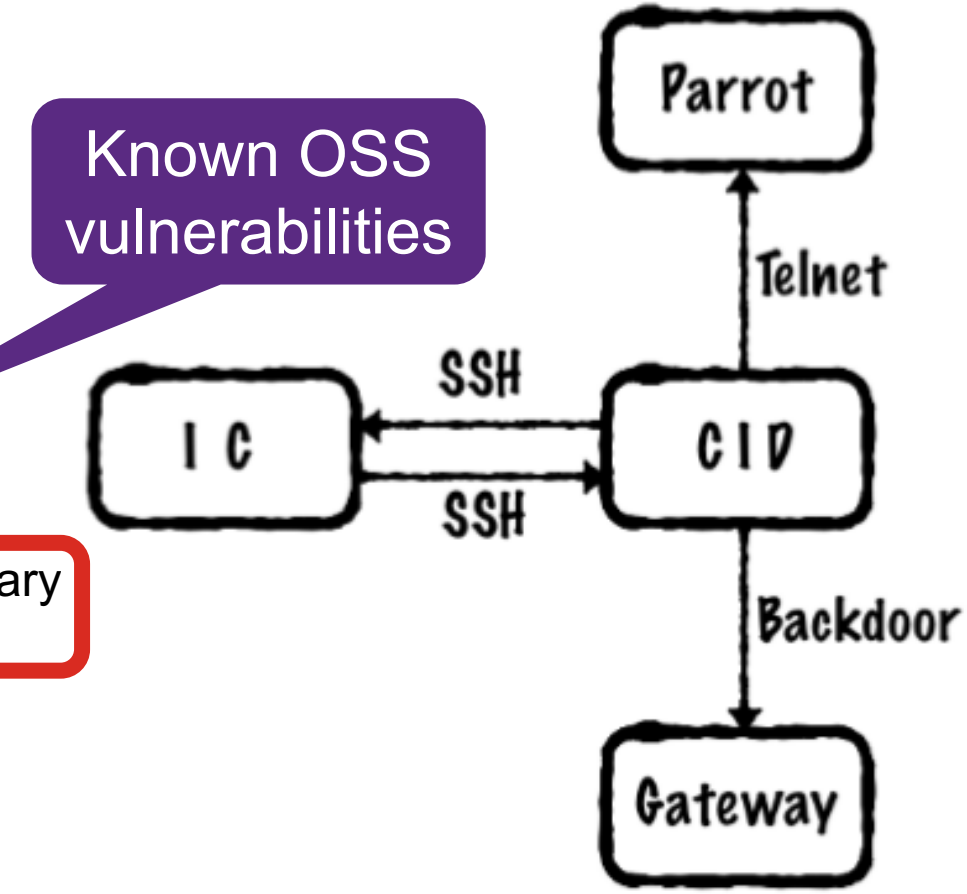


- Update V850 with arbitrary firmware (there is no digital signature verification of the V850 firmware)
  - Remotely send arbitrary CAN messages on the CAN bus

Ref: Charlie Miller, et al. Remote exploitation of an unaltered passenger vehicle. Black Hat 2015

Open Source Software Example – Known vulnerabilities

- Tesla web browser: "Mozilla/5.0 (X11; Linux) AppleWebKit/534.34 (KHTML, like Gecko) QtCarBrowser Safari/534.34“⇒ QtWebkit is around 2.2.x.
- Two vulnerabilities to achieve arbitrary code execution
  - Function JSArray::sort() (memory corruption vulnerability)
  - CVE-2011-3928 (use-after-free vulnerability)
- CID: Linux kernel 2.6.36
  - ARM Linux vulnerability CVE-2013-6282 (read or modify arbitrary kernel memory)
- Exploit web browser vulnerability to gain web browser privileges
- Exploit Linux kernel vulnerability to gain root privileges
- Flash modified firmware to Gateway to be able to send arbitrary CAN messages to control steering and brakes



Ref: Ling Liu, et al. Free-Fall: Hacking Tesla from Wireless to CAN Bus. Black Hat 2017

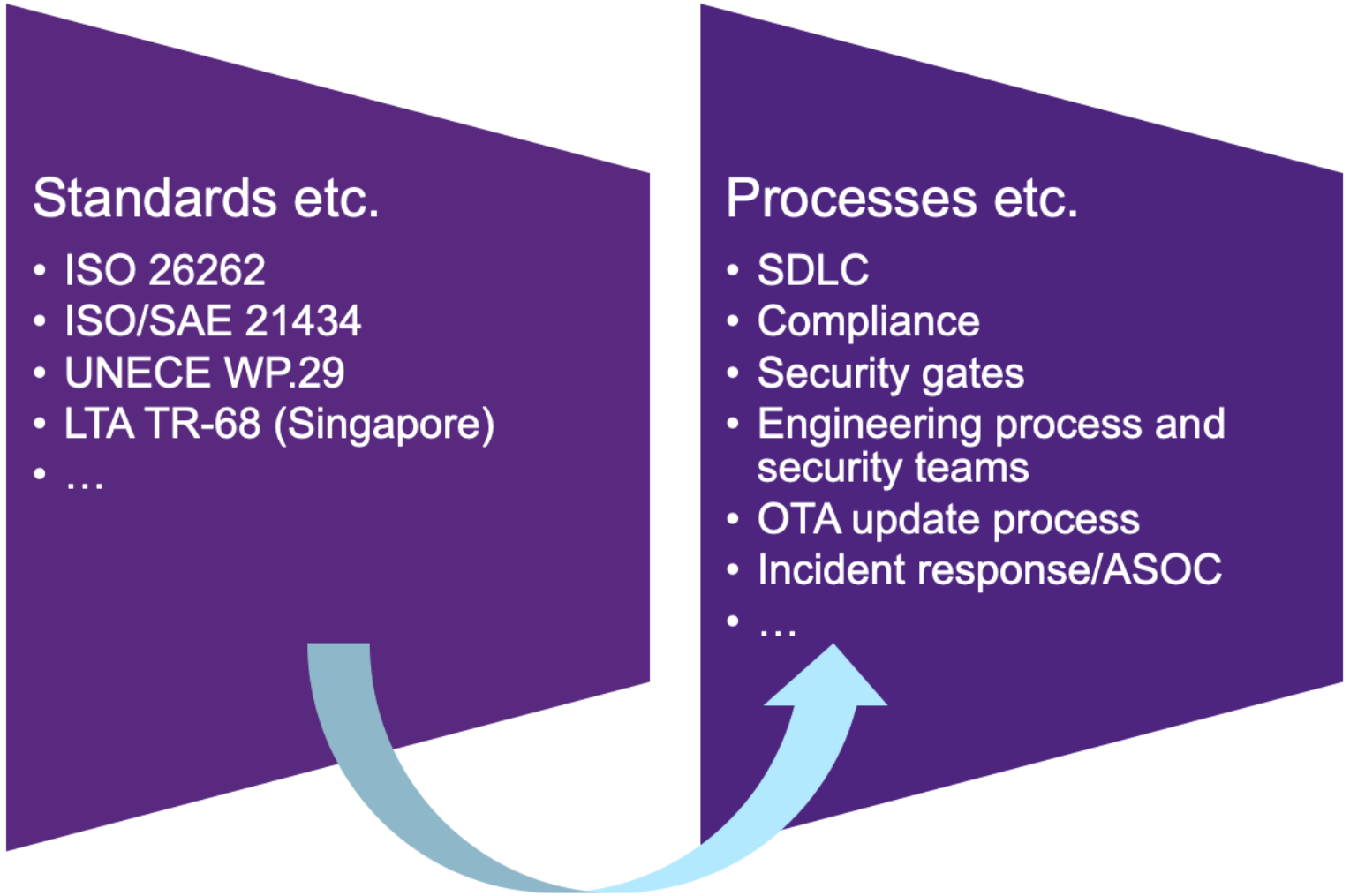
New standards and regulations provide guidance for cybersecurity affecting processes and solutions

Current Automotive Landscape

SDLC: software development lifecycle
OTA: over-the-air
ASOC: automotive security operations center

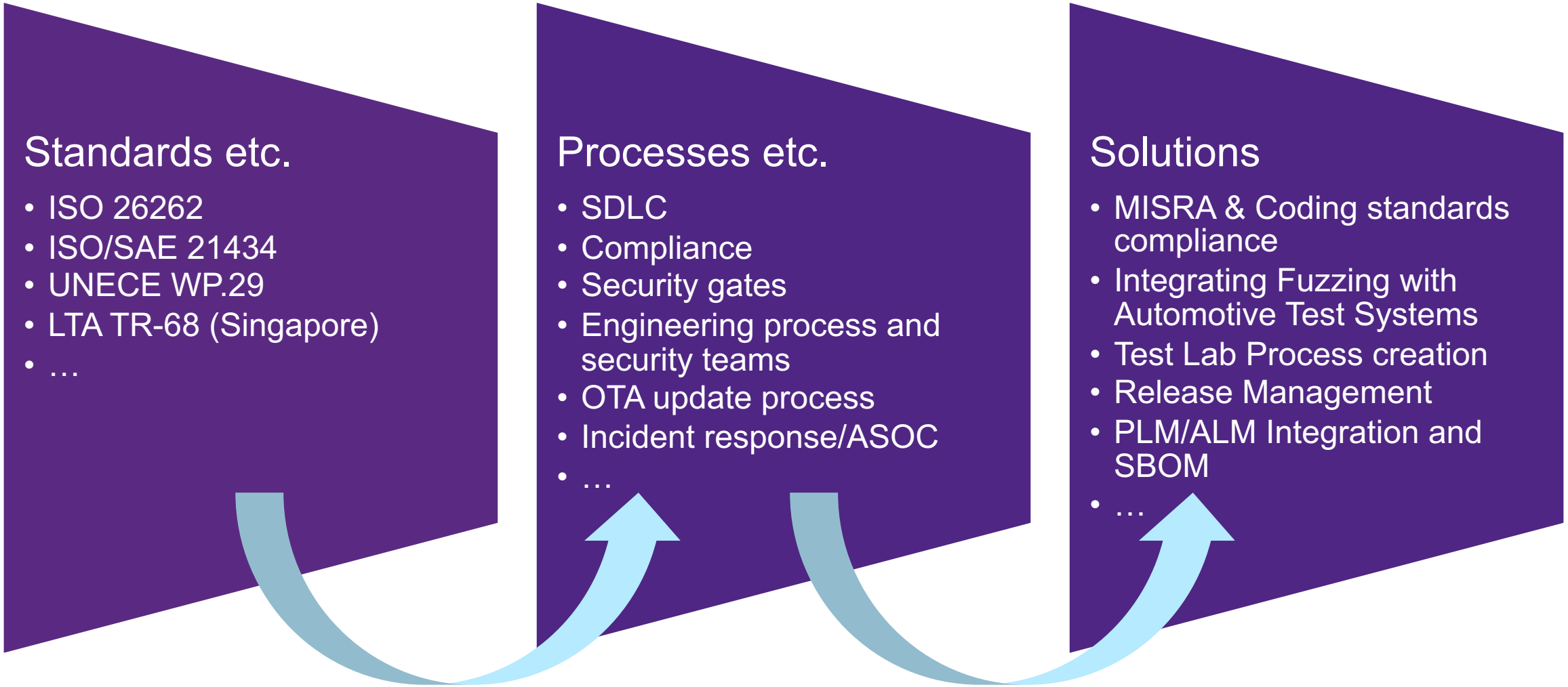
Current Automotive Landscape

SDLC: software development lifecycle
OTA: over-the-air
ASOC: automotive security operations center



Current Automotive Landscape

SDLC: software development lifecycle
OTA: over-the-air
ASOC: automotive security operations center

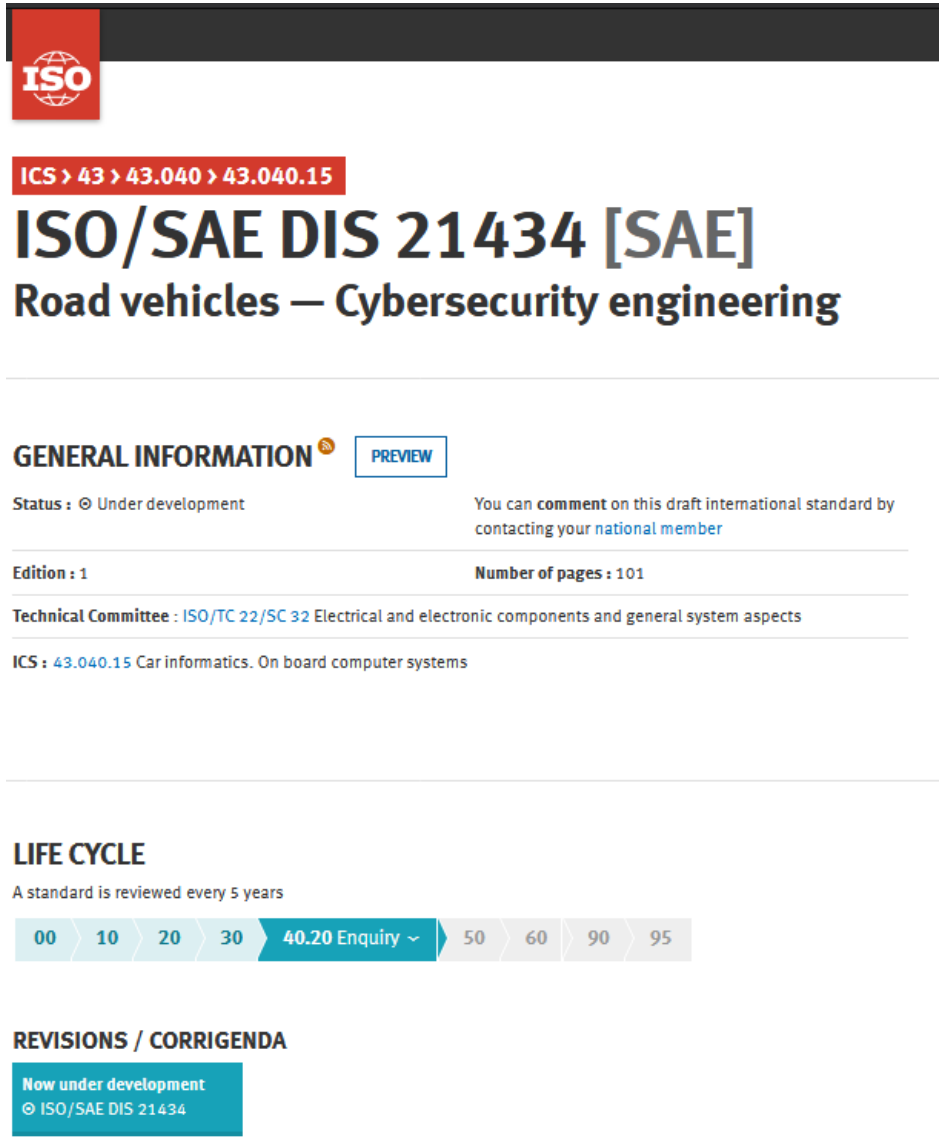


PLM/ALM: product/application lifecycle management
SBOM: software bill of materials

ISO/SAE 21434

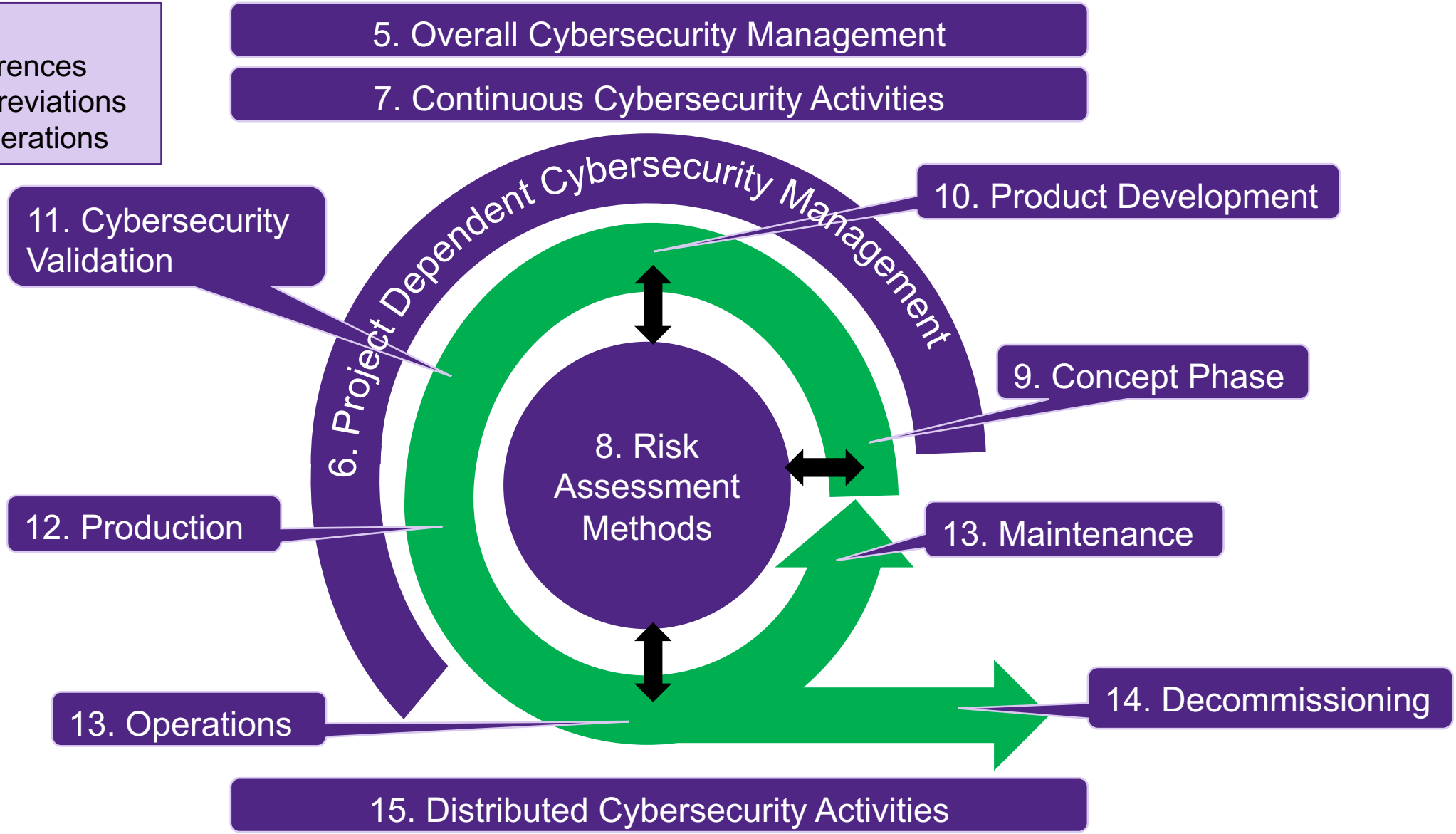
- **ISO/SAE 21434** Road Vehicles — Cybersecurity engineering
- Jointly published standard by SAE and ISO expected in late 2020 or early 2021

- Proposed contents:
  - Management of Cybersecurity
  - Continuous cybersecurity activities
  - Risk assessment methods
  - Concept phase
  - Product development
  - Production, Operations & Maintenance

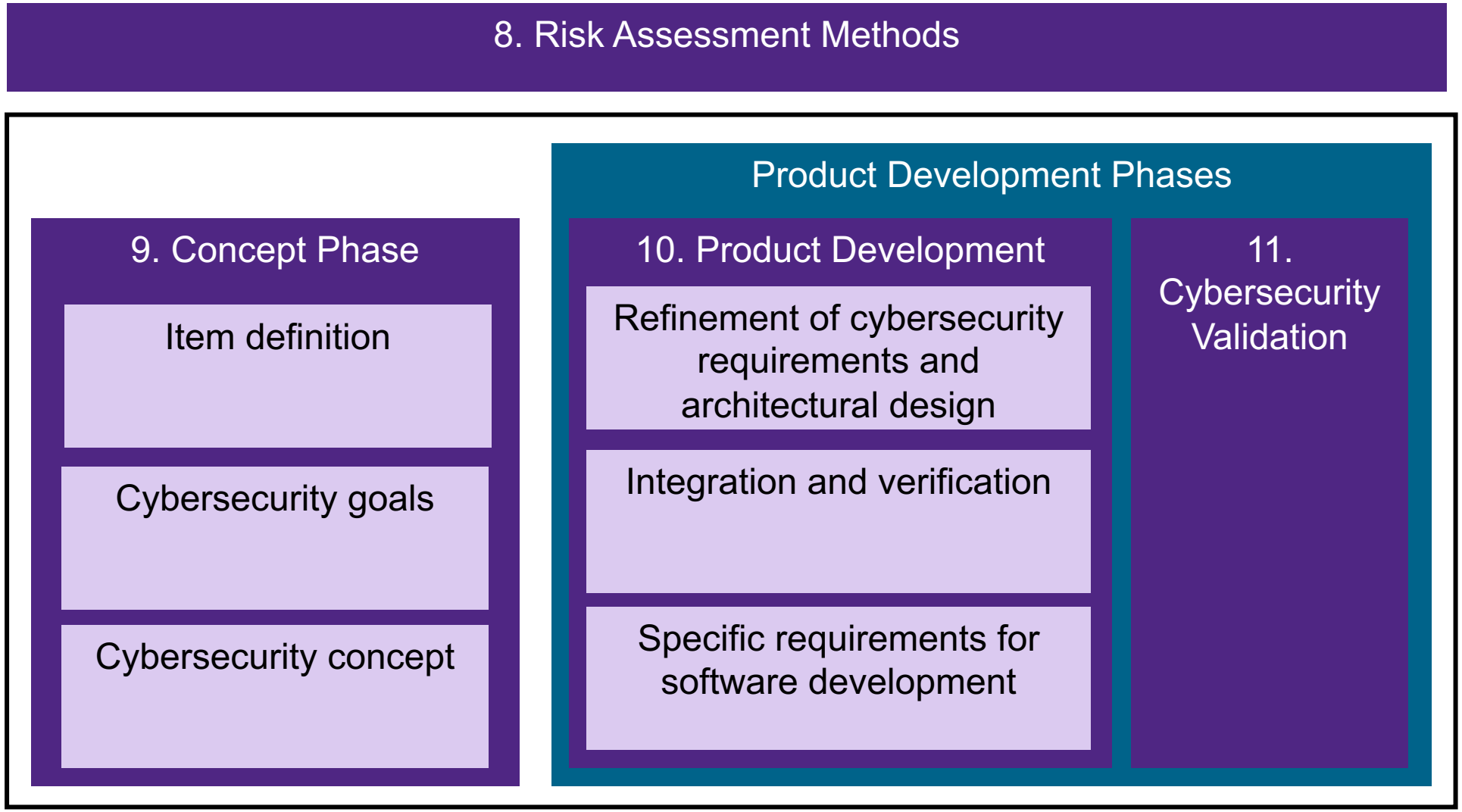


Overview of ISO/SAE 21434

1. Scope
2. Normative references
3. Terms and abbreviations
4. General considerations



Excerpt from ISO/SAE 21434



Ref: Excerpt from ISO/SAE DIS 21434: Road Vehicles – Cybersecurity Engineering

UNECE WP.29

- **ECE/TRANS/WP.29/GRVA/2019/2** - Recommendation on Cyber Security
- **ECE/TRANS/WP.29/GRVA/2019/3** - Recommendation on Software Updates

- Contents:
  - Guidance on process and procedures
    - Reference to standards e.g., ISO/SAE 21434
  - Best practices (Threats and Mitigations)
  - Proposal for a Cybersecurity Management System (CSMS) required for cyber security
  - Proposal for a Software Update Management System (SUMS) required for delivery of software updates
  - Certification regarding cyber security and software updates
    - CSMS Certificate of Compliance
    - SUMS Certificate of Compliance

United Nations ECE/TRANS/WP.29/GRVA/2019/2

Economic and Social Council

Distr.: General
19 November 2018

Original: English

Economic Commission for Europe

Inland Transport Committee

World Forum for Harmonization of Vehicle Regulations

Working Party on Automated/Autonomous and Connected Vehicles*

Second session
Geneva, 28 January-1 February 2019
Item 5 (b) of the provisional agenda
Automated/autonomous and connected vehicles:
Cyber security and data protection

Proposal for a Recommendation on Cyber Security

Submitted by the experts of the Task Force on Cyber Security and Over-the-air issues**

This proposal was prepared by the experts of the Task Force on Cyber Security and Ov
The-Air Software update issues in response to the mandate agreed by the World Forum
Harmonization of Vehicle Regulations (WP.29) as reflected in ECE/TRANS/WP29/1126, pa
28 and ECE/TRANS/WP29/1131, para. 27. It is based on informal document GRVA-01-
previously presented at the first session of the Working Party on Automated/Autonomous a
Connected Vehicles (GRVA) in September 2018. The Annex A of this document contains a dr
UN Regulations on cybersecurity. This draft Regulation contains four annexes (Annexes 1 to
belonging to Annex A.

* Formerly: **Working Party on Brakes and Running Gear (GRRF)**.
** In accordance with the programme of work of the Inland Transport Committee for 2018–2019
(ECE/TRANS/274, para. 123 and ECE/TRANS/2018/21/Add.1, Cluster 3), the World Forum
will develop, harmonize and update UN Regulations in order to enhance the performance of
vehicles. The present document is submitted in conformity with that mandate.

GE.18-19708(E)

Please recycle

United Nations ECE/TRANS/WP.29/GRVA/2019/3

Economic and Social Council

Distr.: General
19 November 2018

Original: English

Economic Commission for Europe

Inland Transport Committee

World Forum for Harmonization of Vehicle Regulations

Working Party on Automated/Autonomous and Connected Vehicles*

Second session
Geneva, 28 January-1 February 2019
Item 5 (c) of the provisional agenda
Automated/autonomous and connected vehicles:
Software updates (incl. Over-The-Air updates)

Draft Recommendation on Software Updates of the Task Force on Cyber Security and Over-the-air issues

Submitted by the experts of the Task Force on Cyber Security and Over-the-air issues**

This proposal was prepared by the experts of the Task Force on Cyber Security and Over-
The-Air Software update issues in response to the mandate agreed by the World Forum for
Harmonization of Vehicle Regulations (WP.29) as reflected in ECE/TRANS/WP29/1126, para.
28 and ECE/TRANS/WP29/1131, para. 27. It is based on informal document GRVA-01-18
previously presented at the first session of the Working Party on Automated/Autonomous and
Connected Vehicles (GRVA) in September 2018. The Annex A of this document contains a draft
UN Regulations on Software updates processes. This draft Regulation contains annexes (Annexes
1 to 4) belonging to Annex A.

* Formerly: **Working Party on Brakes and Running Gear (GRRF)**.
** In accordance with the programme of work of the Inland Transport Committee for 2018–2019
(ECE/TRANS/274, para. 123 and ECE/TRANS/2018/21/Add.1, Cluster 3), the World Forum
will develop, harmonize and update UN Regulations in order to enhance the performance of
vehicles. The present document is submitted in conformity with that mandate.

GE.18-19709(E)

Please recycle

Type Approval and UNECE WP.29

CSMS (Cyber Security Management System)

- Processes for:
  - Managing cyber security
  - Identifying, assessing, categorizing, treating risks
  - Testing of security
  - Monitoring, detecting, responding to cyber attacks, threats and vulnerabilities
- Entire life-cycle (development, production, operation)

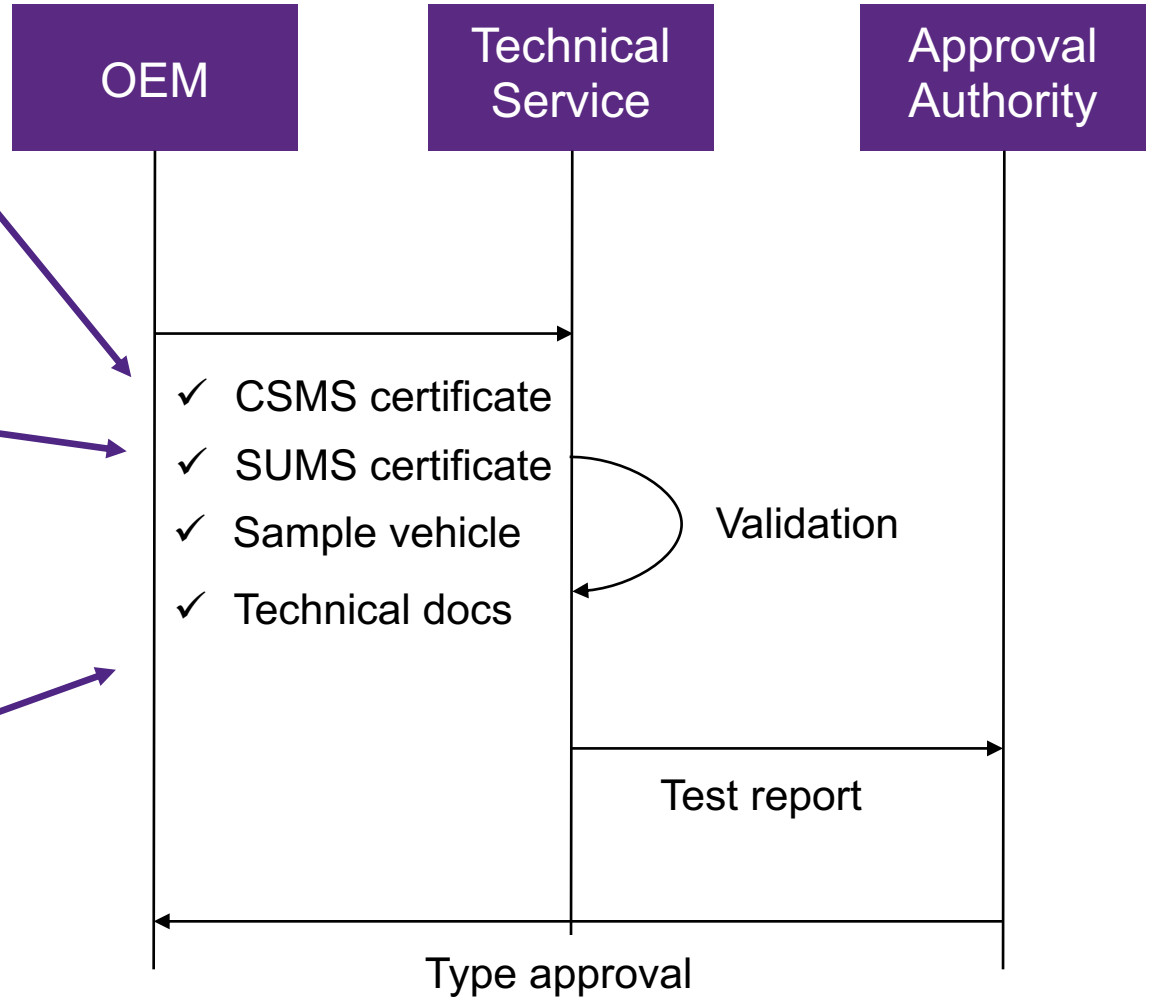
SUMS (Software Update Management System)

- Demonstrate that:
  - Process to ensure software updates will be protected before update process is initiated
  - Update processes used are protected (e.g., development of update delivery system)
  - Processes to validate and verify software functionality for the software used in the vehicle are appropriate

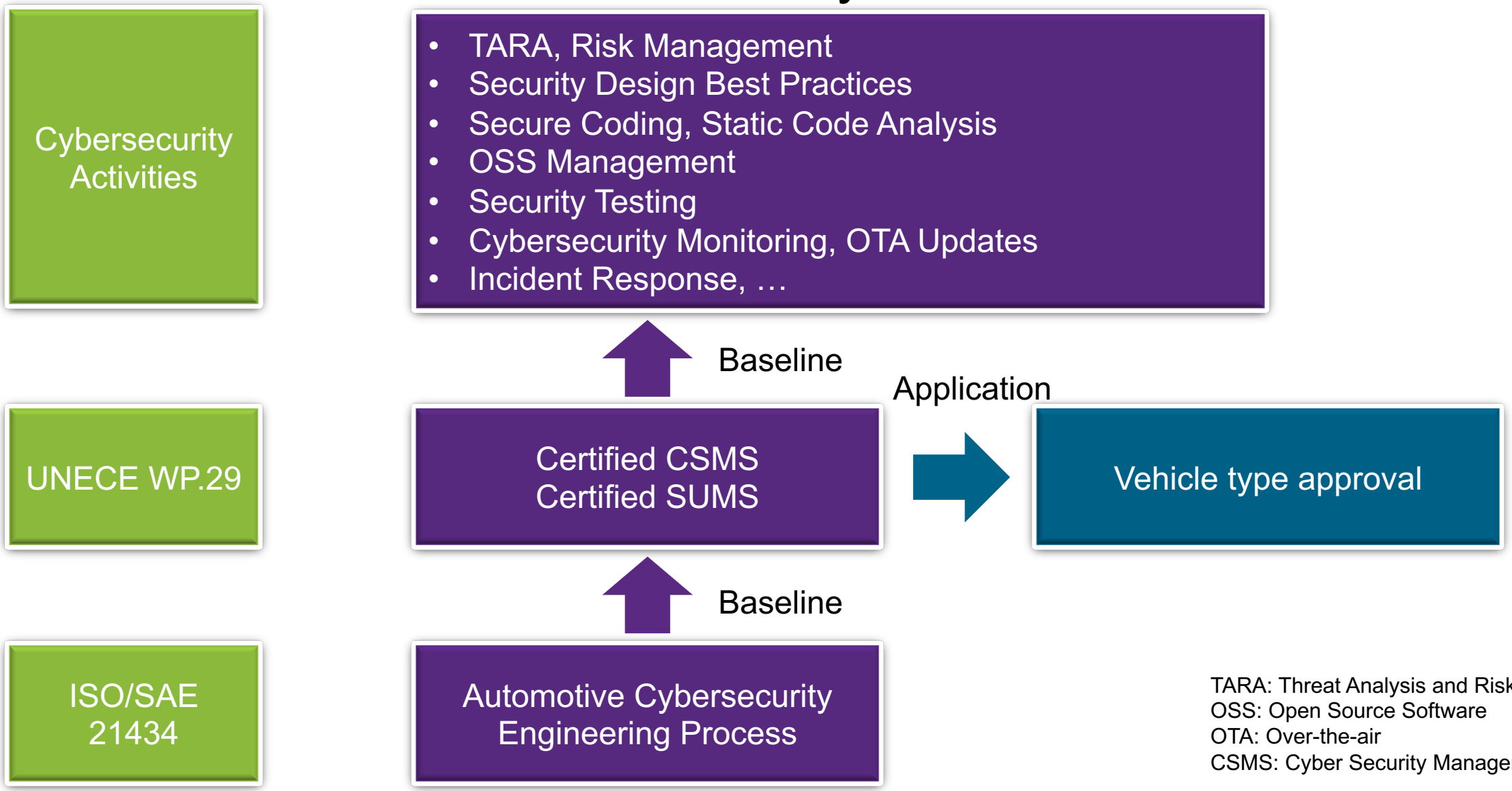
Security of Vehicle Types

- Application to vehicle type at time of type approval

Type Approval Process

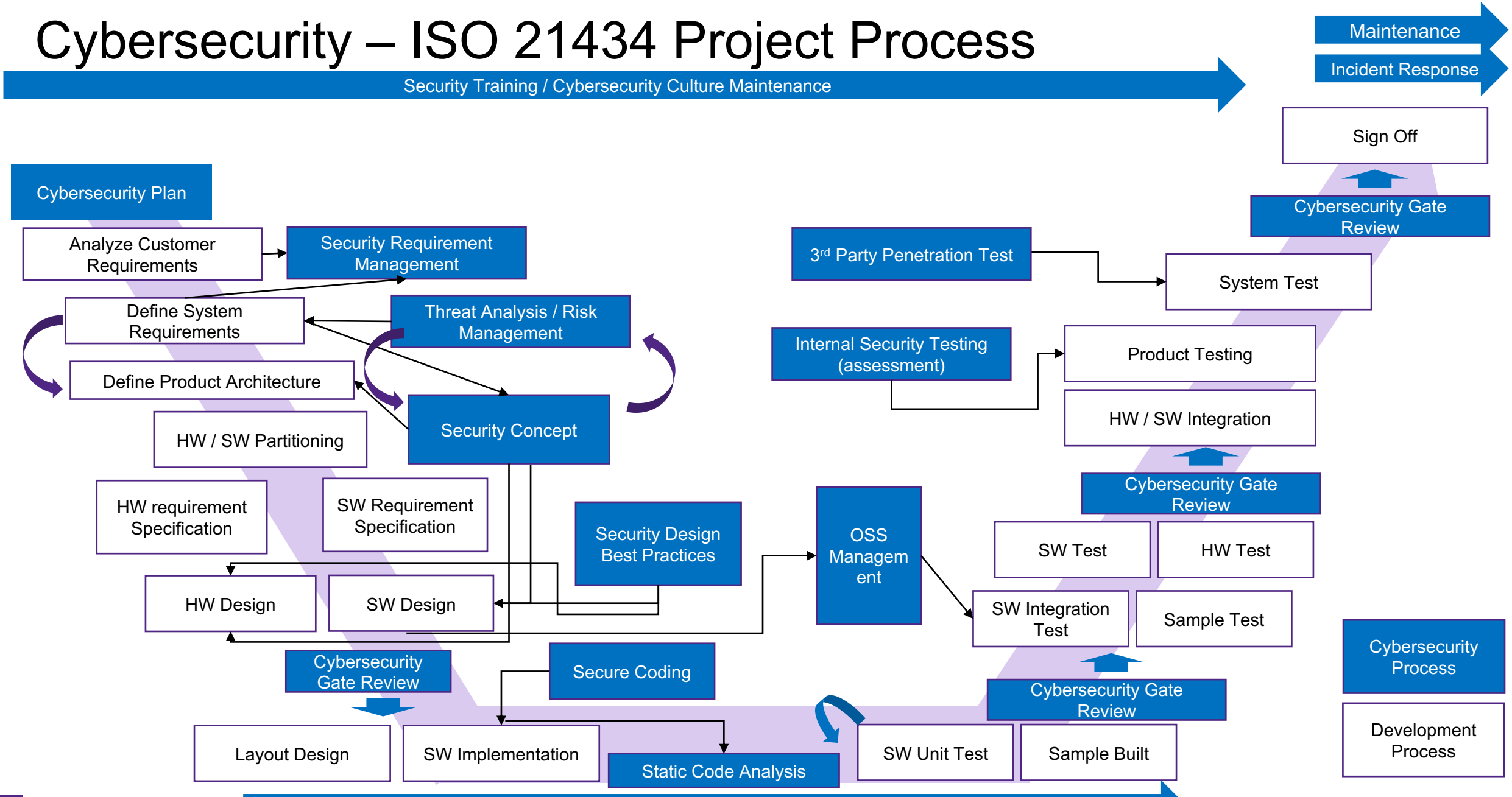


Overview of Process and Security Solutions



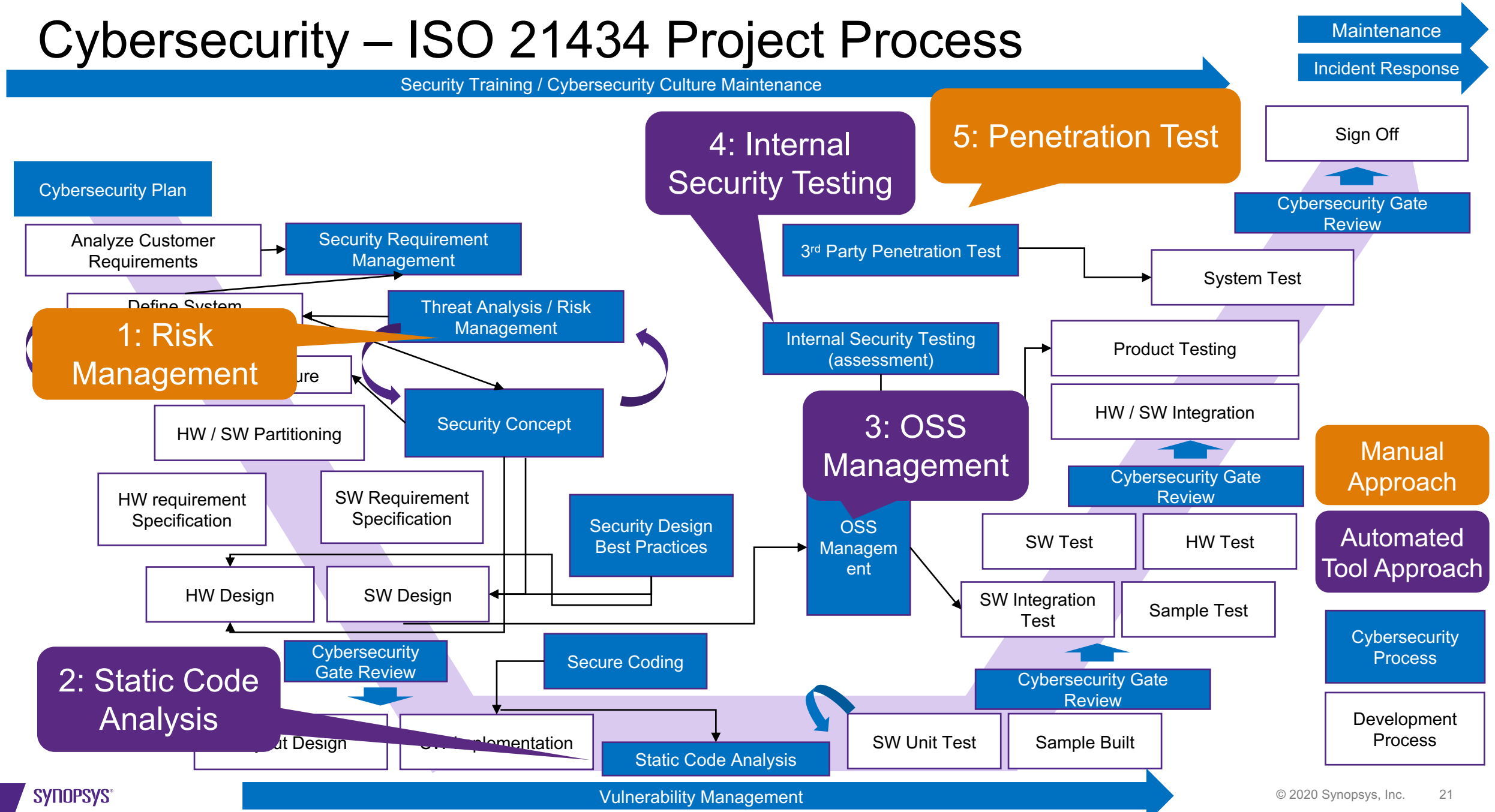
TARA: Threat Analysis and Risk Assessment
OSS: Open Source Software
OTA: Over-the-air
CSMS: Cyber Security Management System

Cybersecurity – ISO 21434 Project Process

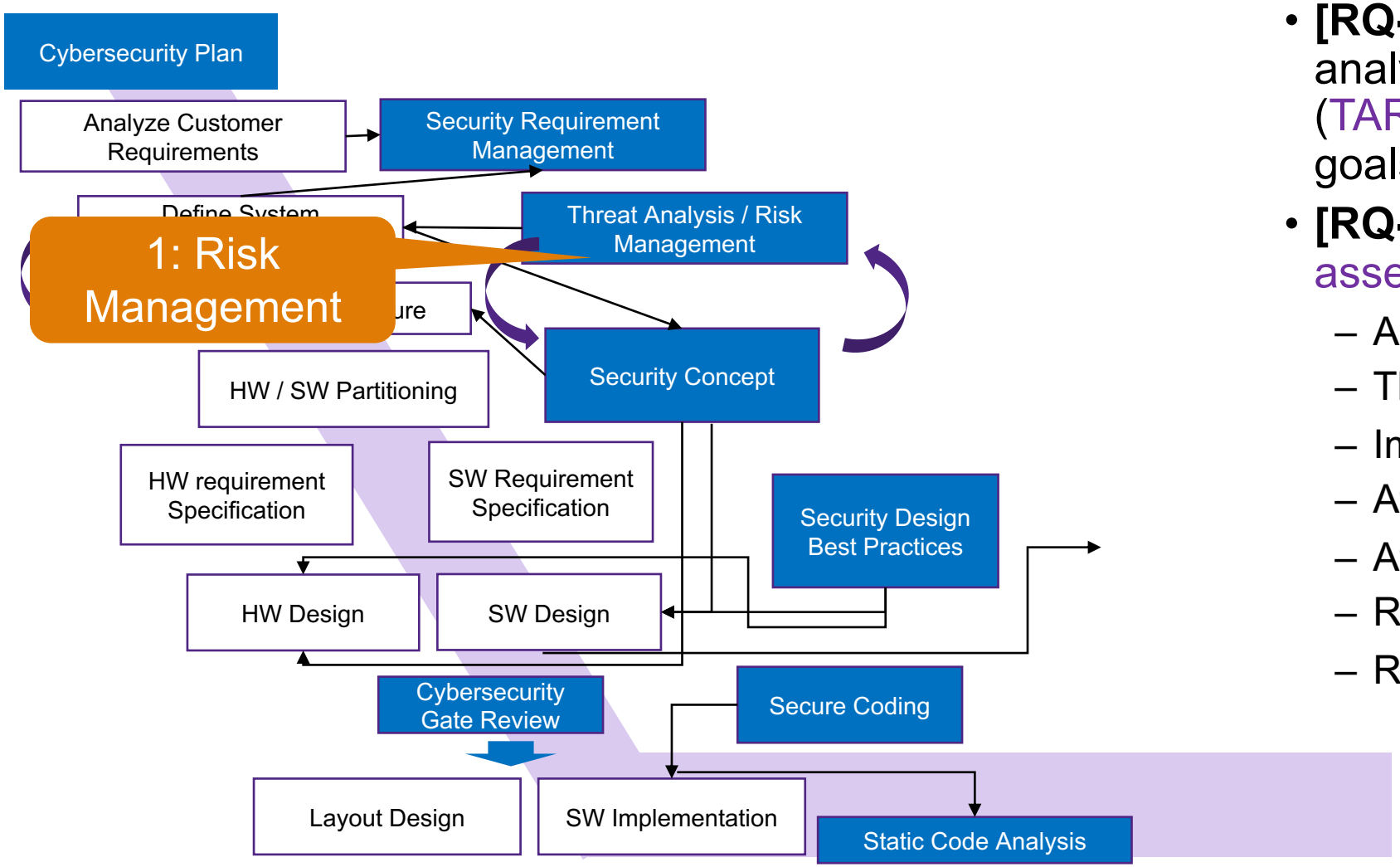


OSS: Open Source Software

Cybersecurity – ISO 21434 Project Process



1: Risk Management



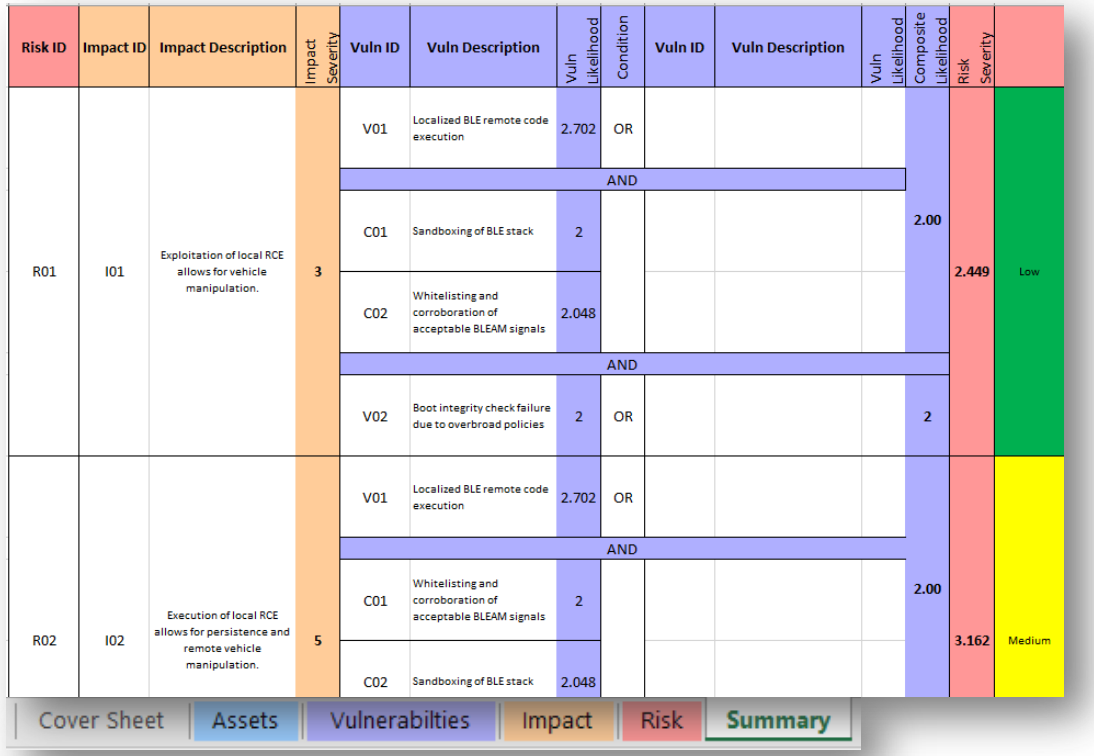
- **[RQ-09-05]:** Perform threat analysis and risk assessment (TARA) to identify cybersecurity goals
- **[RQ-08-xx]:** Various risk assessment steps are defined:
  - Asset identification
  - Threat scenario identification
  - Impact rating
  - Attack paths (trees) analysis
  - Attack feasibility rating
  - Risk determination
  - Risk treatment decision

1: Risk Management: TARA

- Risk management tooling and resources:
  - Requirements engineers, security experts (automotive, security expertise)
  - Tools
    - Templates
    - Vulnerability/risk databases
    - “Risk analysis” tools

- Assess the risk associated with threats by combining the impact (safety, financial, operational and privacy) with the likelihood (e.g., derived from time, expertise, knowledge of target, window of opportunity and equipment)

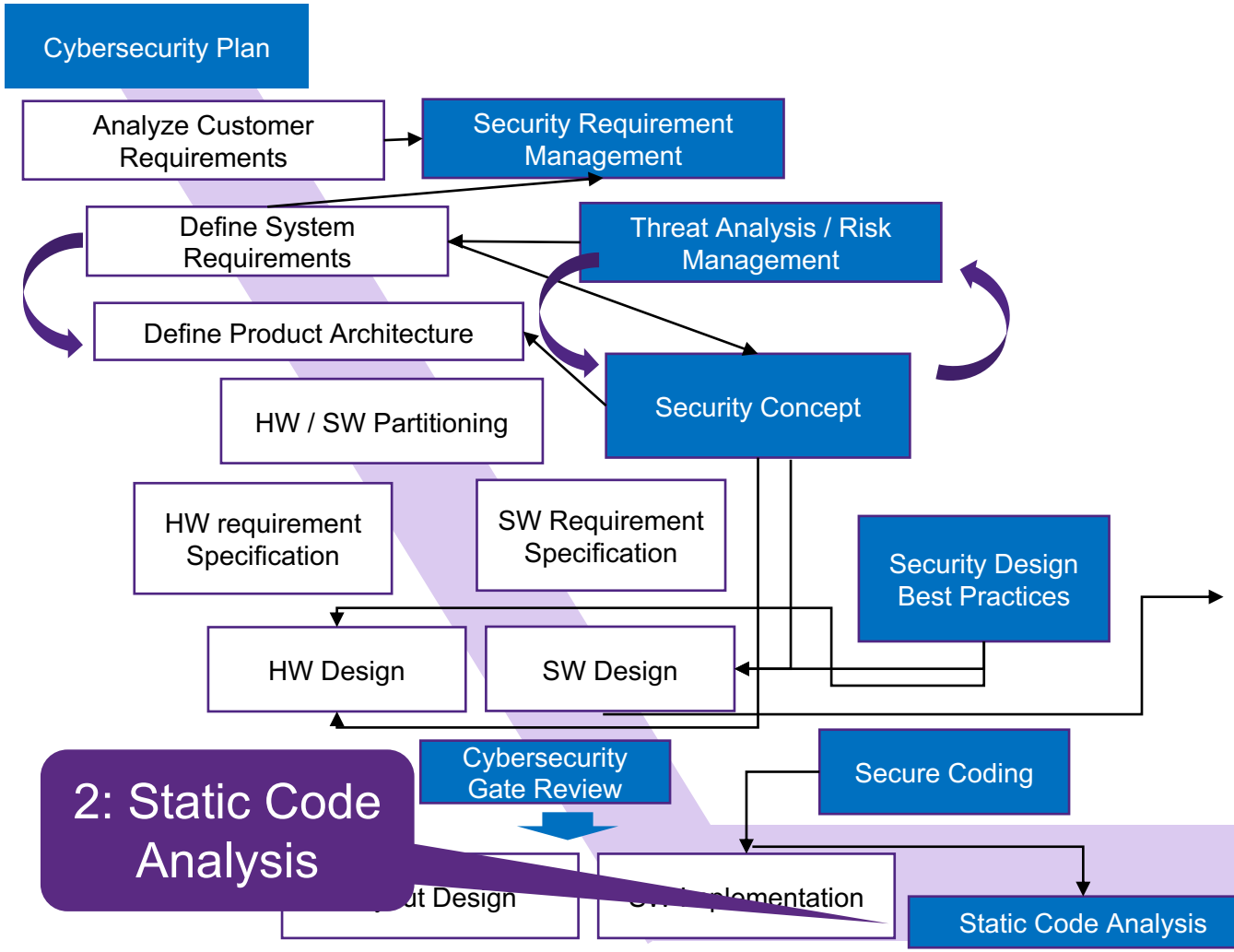
- Establish a process, decide risk assessment methodology and define responsibilities



Risk ID	Impact ID	Impact Description	Impact Severity	Vuln ID	Vuln Description	Vuln Likelihood	Condition	Vuln ID	Vuln Description	Vuln Likelihood	Composite Likelihood	Risk Severity	
R01	I01	Exploitation of local RCE allows for vehicle manipulation.	3	V01	Localized BLE remote code execution	2.702	OR				2.00	2.449	Low
				AND									
				C01	Sandboxing of BLE stack	2							
				C02	Whitelisting and corroboration of acceptable BLEAM signals	2.048							
				AND									
				V02	Boot integrity check failure due to overbroad policies	2	OR				2		
R02	I02	Execution of local RCE allows for persistence and remote vehicle manipulation.	5	V01	Localized BLE remote code execution	2.702	OR				2.00	3.162	Medium
				AND									
				C01	Whitelisting and corroboration of acceptable BLEAM signals	2							
				C02	Sandboxing of BLE stack	2.048							

Cover Sheet | Assets | Vulnerabilties | Impact | Risk | Summary

2: Static Code Analysis



- **[RQ-10-12]:** Static code analysis can be used as a verification activity to search the source code text for patterns matching known faults, common weakness patterns or compliance with coding guidelines

- **[RQ-10-20, RQ-10-21]:** Static code analysis tools can check against coding guidelines: MISRA C, CERT C, AUTOSAR etc.

2: Static Code Analysis

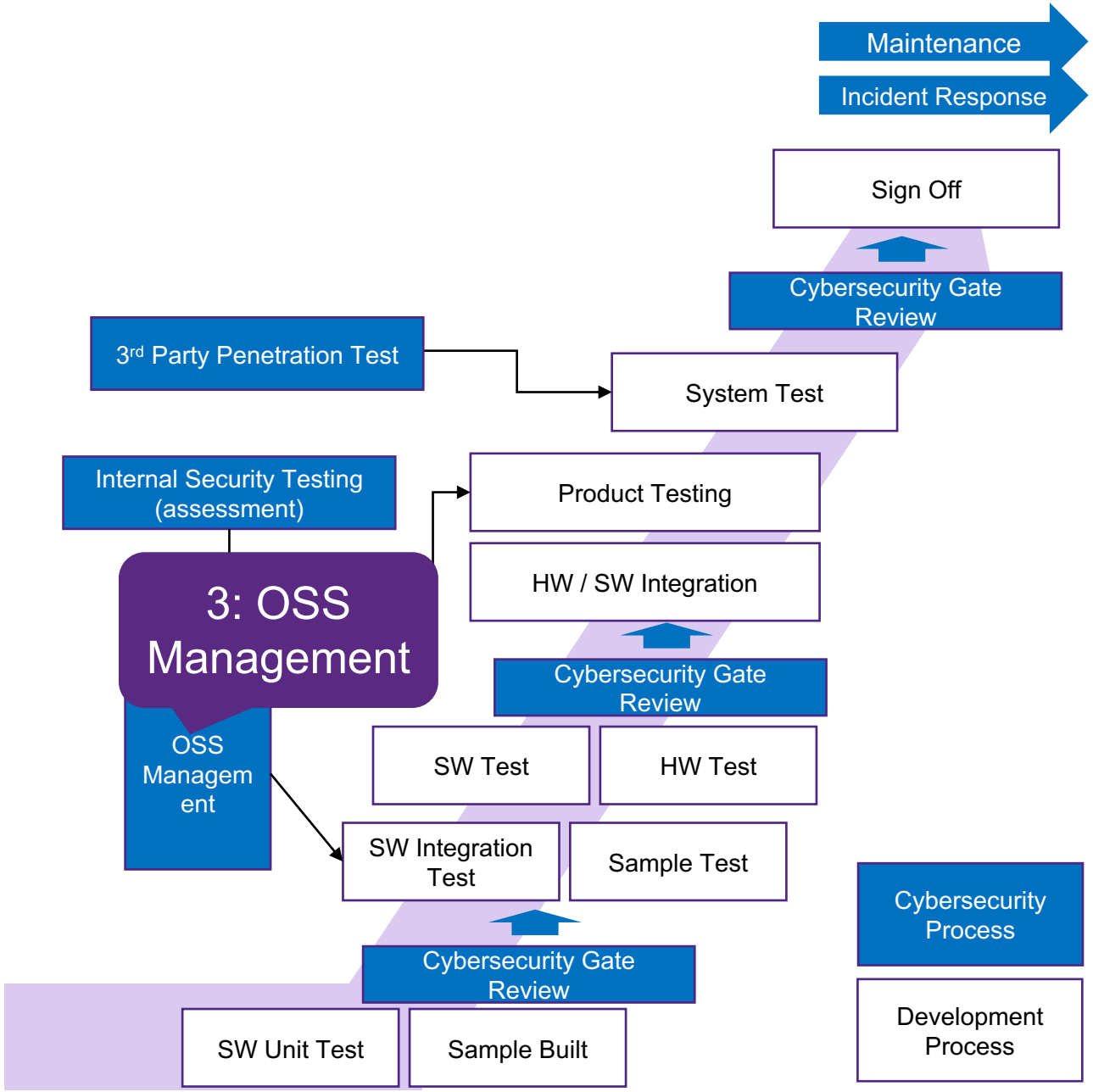
- Static code analysis tooling
  - Use automated tools for developers
  - Dashboards for project managers

- How to make the most of the tools
  - Considerations for roll-out to large development teams (~1000+ developers): scalability, performance
  - Prepare enablement materials for developers (‘how to’ guides for specific projects)
  - Integration with other tools in the software development lifecycle (automation)

3: OSS Management

- **[RQ-06-16]:** When integrating an off-the-shelf component, OSS scanning can be used to gather cybersecurity-relevant information about the component
  - Known vulnerabilities (CVEs)

- OSS management focuses on OSS scanning which has several goals:
  - Used open source software is identified and is checked against vulnerability databases
  - Ensure no source code that falls under license restrictions is used
  - Ensure latest version of open source software is being used



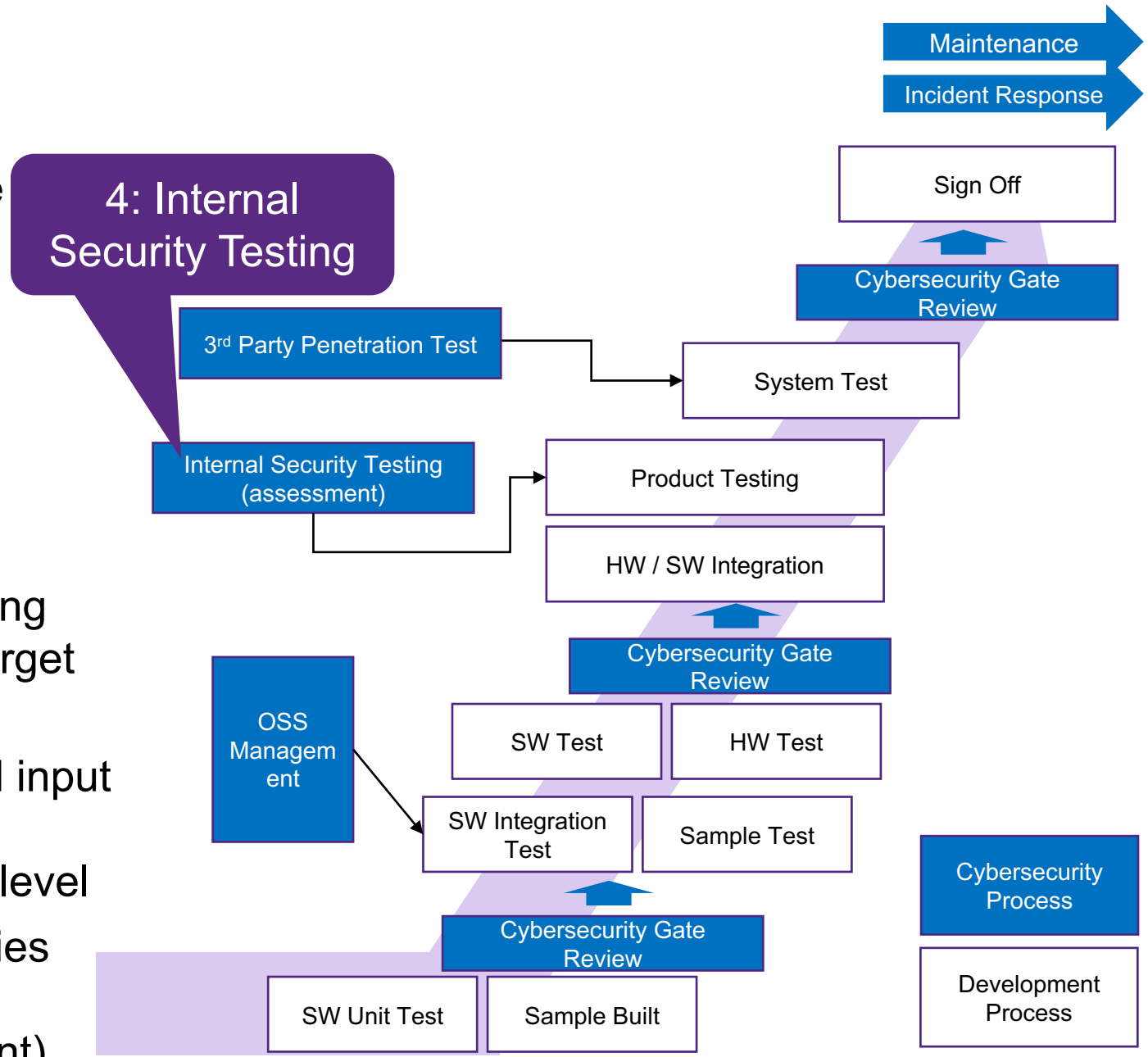
3: OSS Management

- Software composition analysis tooling
  - Use automated tools for developers, system integrators
  - Dashboards for project managers

- How to make the most of the tools
  - Establish a process for OSS management (when/what to scan, OSS policies, approval process etc.)
  - Integration with other tools in the software development lifecycle (automation)
  - Tools to provide alerts on newly identified vulnerabilities as input to the continuous cybersecurity monitoring activity (cf. **[RQ-07-01]**)

4: Internal Security Testing

- **[RC-10-03]:** Component testing should be performed to search for unidentified vulnerabilities
- Test methods can include:
  - Vulnerability scanning
  - Fuzz testing
  - Penetration testing
- Vulnerability scanning focuses on identifying potential vulnerabilities by scanning the target system for (unauthorized) access
- Fuzz testing involves providing malformed input to a system and identifying unknown vulnerabilities at system level or interface level
- **[RQ-10-18]:** Weaknesses and vulnerabilities identified in [RC-10-03] shall be managed according to 7.6 (Vulnerability Management)



4: Internal Security Testing

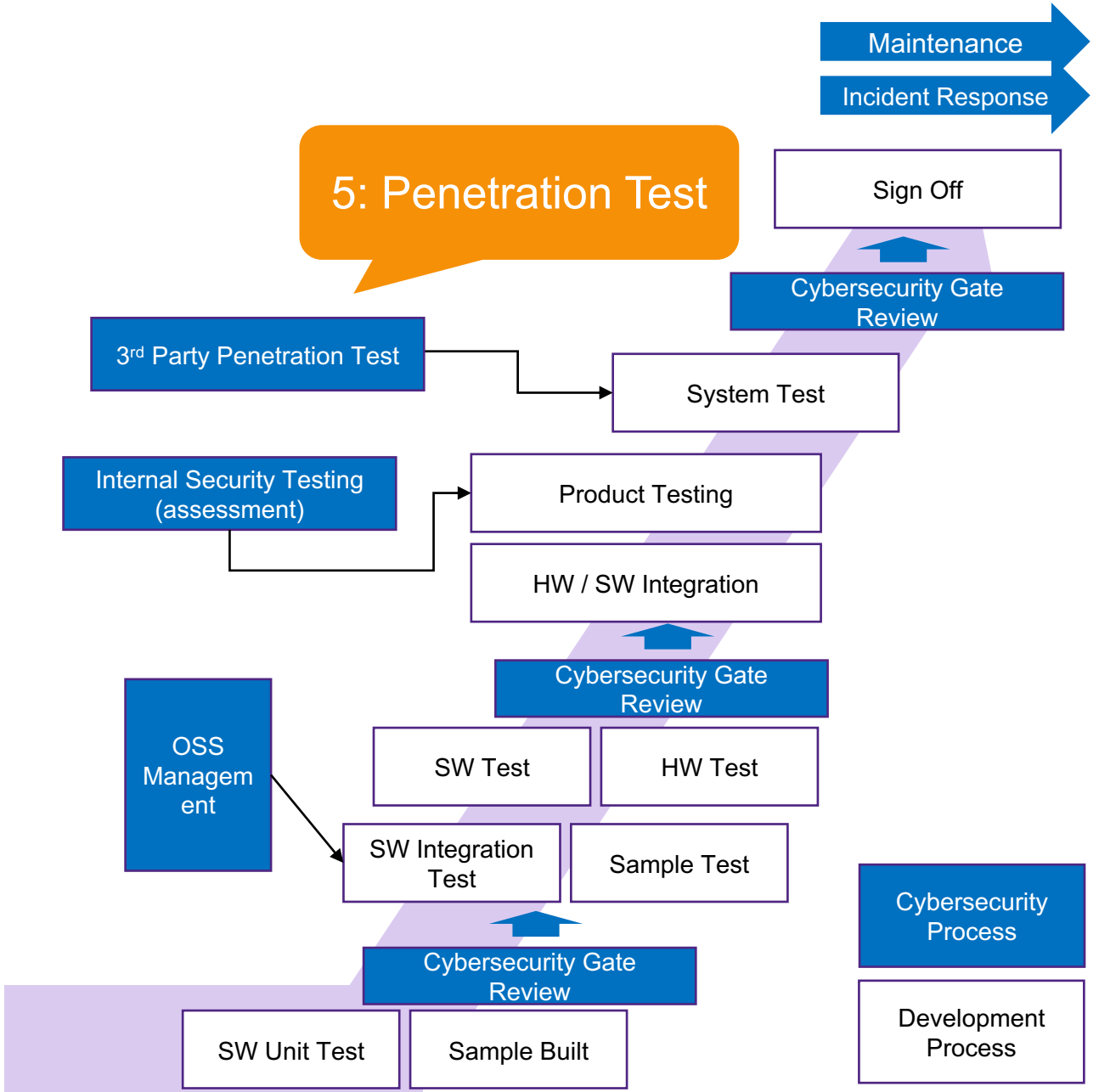
- Internal security testing tooling
  - Use automated tools for developers, QA teams, security teams
    - Vulnerability scanners
    - Fuzz testing tools
  - Dashboards for project managers

- How to make the most of the tools
  - Establish processes for security testing (which tools, when/what to test, test environments, how often)
  - Integration with other tools in the software development lifecycle
  - Building a cybersecurity testlab to enable automated testing

5: Penetration Test

- **[RQ-11-01, RC-11-01]:** Penetration testing should be performed to validate the cybersecurity goals as part of the validation activities

- Penetration Test testing typically includes:
  - Intelligence gathering
  - Vulnerability analysis
  - Exploitation
  - Post exploitation analysis
  - Report with reproduction steps
- There are different approaches for penetration testing:
  - White-box, Gray-box, Black-box approaches



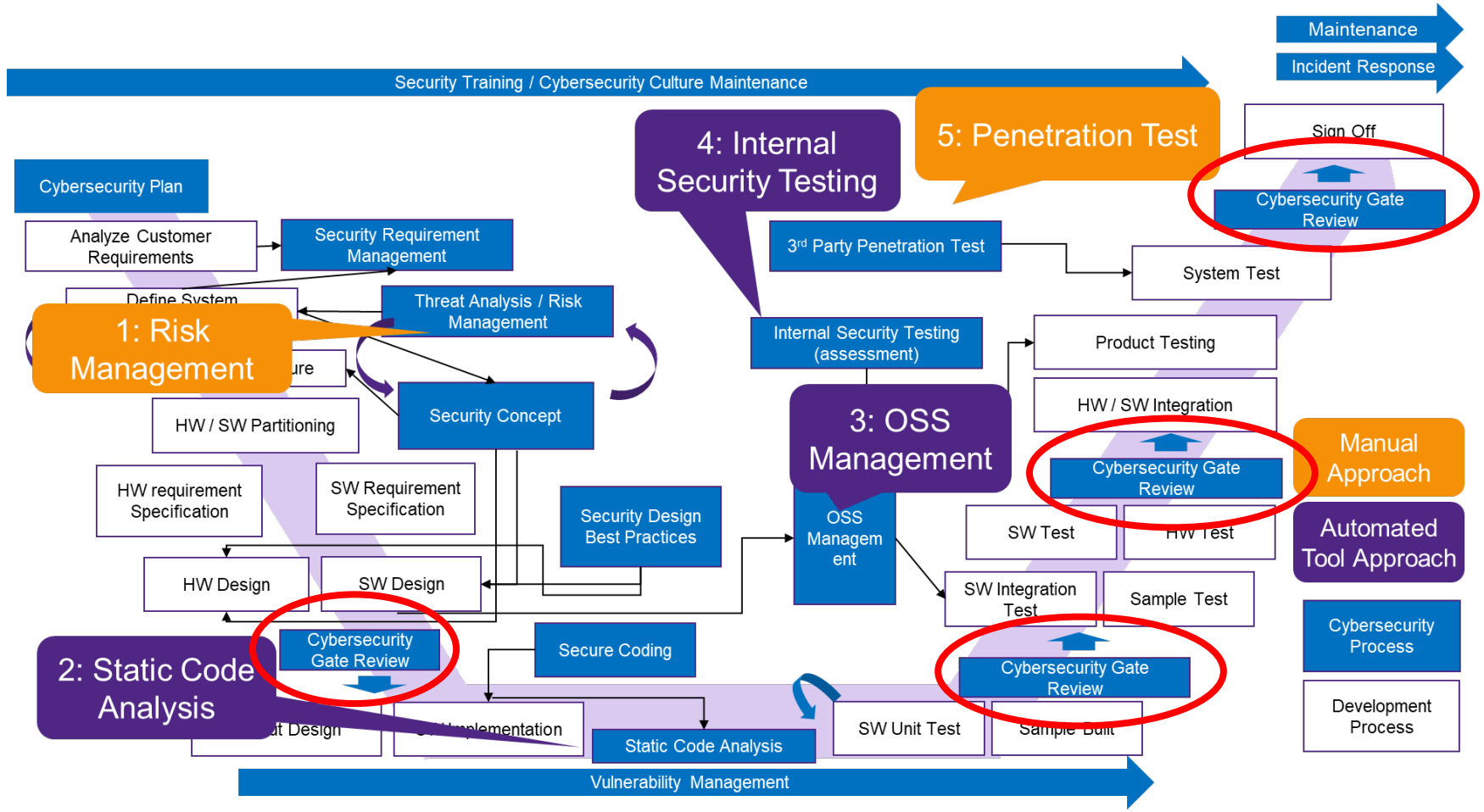
5: Penetration Test

- Penetration test tooling and resources
  - Special security experts (pentesters)
    - Automotive/embedded expertise
    - Communication protocols experise
    - Wireless expertise
    - Software expertise
  - Tools
    - Debuggers, logic analyzers
    - nmap
    - BladeRF
    - IDA Pro

- Penetration test is the “last gate”
  - Goal: find as few vulnerabilities as possible since it may be difficult to remediate any vulnerabilities found
  - As many vulnerabilities as possible should be found and addressed earlier in the development lifecycle

Cybersecurity Gate Reviews

- The security gates should be independent verification steps, out of the product teams' control
- Avoiding gates to become to bottlenecks, there needs to be security approval/review process
- Gate reviews are used to validate proper security for the products, with complying to security activities from ISO/SAE 21434



CAL - Cybersecurity Assurance Levels (Annex E)

- CAL can be used to help provide assurance that the assets of an item or component are adequately protected against the relevant threat scenarios
- CAL can affect:
  - the number, scope and extent of the cybersecurity activities
  - methods for design and verification
  - testing methods including depth of testing
- For each increasing CAL, the corresponding requirements represent an increase in the assurance of the item or component by the design, verification, and other cybersecurity activities

Example of CAL determination based on impact and attack vector

	Attack Vector:	Physical	Local	Adjacent	Network
Impact:	Negligible	--	--	--	--
	Moderate	CAL 1	CAL 1	CAL 2	CAL 3
	Major	CAL 1	CAL 2	CAL 3	CAL 4
	Severe	CAL 2	CAL 3	CAL 4	CAL 4

CAL - Cybersecurity Assurance Levels (2)

- Example usage of CAL in product development:

Methods	CAL 1	CAL 2	CAL 3	CAL 4
Static code analysis	✔	✔	✔	✔
Vulnerability scanning	✔	✔	✔	✔
Fuzz testing	✔	✔	✔	✔
Penetration testing (Basic Level)	✔	✔		
Penetration testing (Enhanced-Basic Level)			✔	✔

E.g., 8 hours

E.g., 16 hours

E.g., 40 hours

E.g., 160 hours

Call to Action

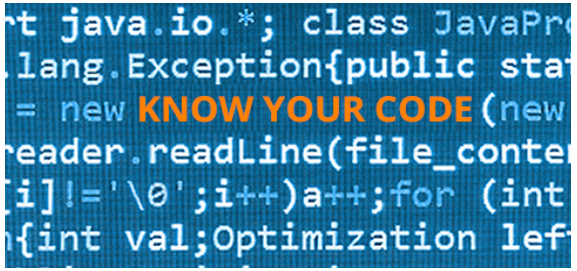
Investigate how to incorporate activities into the development process

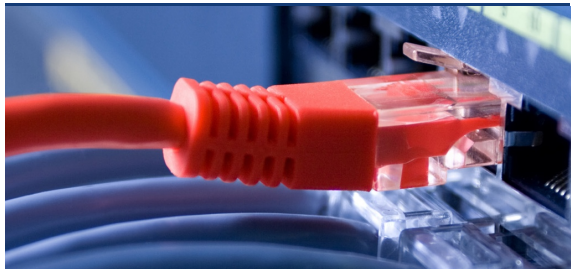
- Risk Management
- Static Code Analysis
- OSS Management
- Internal Security Testing
- Penetration Testing

Create and update internal process documents and requirements

Use appropriate tools and technology to automate the process

Synopsys Automotive Software Cybersecurity & Quality





Coverity Static Analysis	Defensics Fuzz Testing	Black Duck OSS Management	Security Services
Find critical defects and vulnerabilities in code	Find vulnerabilities before hackers	Find known vulnerabilities in OSS	Best practices consulting
Automotive compliance (MISRA, ISO26262)	Fuzzing for automotive protocols	Generate SBOM for supply chain management	Services for TARA, security test, penetration test
Security: CERT-C and CWE Top 25	CAN, Ethernet, WiFi, Bluetooth, IPv4, mp3, mp4	Alerts for newly detected vulnerabilities	Gap analysis/remediation planning

SYNOPSYS®
Silicon to Software™

Thank You