Course #2

Content

- Definition and Scope
- Automotive Security
- Attack surface
- Standards and regulations
- Security principles
- Best Practices
- Trends and challenges

Why it matters

Definition and Scope

- **Embedded cybersecurity**: Security measures applied to embedded systems, which are computing devices integrated into larger systems (e.g., ECUs in vehicles).
- **Automotive cybersecurity**: Focuses on protecting in-vehicle networks, ECUs, and connected vehicle systems from cyber threats.
- The field encompasses both preventive measures and response capabilities

Evolution of Vehicle Electronics

- **Traditional ECUs (Electronic Control Units)**: Originally designed as isolated systems with minimal external connectivity
- **Modern vehicle architecture**: Features 200+ interconnected ECUs communicating across multiple networks
- Average modern vehicle now contains 300+ million lines of code (compared to ~10 million in early 2000s vehicles)

Key developments

- Vehicles have evolved from standalone systems with isolated ECUs to **highly interconnected platforms** with real-time communication capabilities.
- The rise of **Software-Defined Vehicles (SDVs)** has led to increased reliance on software, cloud connectivity, and OTA updates, introducing new cybersecurity challenges.

Software-Defined Vehicles (SDVs)

- Represents a fundamental shift where vehicle functions and features are primarily determined by software rather than hardware
- Enables features like adaptive driving assistance, personalized cabin experiences, and continuous improvement via updates
- Creates exponentially greater attack surface with each new software layer and connection point
- Introduces complexity challenges in security verification and validation

The 2015 Jeep Cherokee Hack

- A remote attack demonstrated by security researchers on a Jeep Cherokee, exploiting vulnerabilities in the infotainment system to gain control over critical vehicle functions.
- **Implications:** Led to recalls, regulatory responses, and increased focus on securing vehicle communication and software update mechanisms.

Increasing Attack Surfaces

Key Vulnerabilities

- V2X Communication (Vehicle-to-Everything)
- Telematics & Infotainment Systems
- Over-the-Air (OTA) Updates
- CAN Bus & Ethernet Attacks

V2X Communication

- V2V (Vehicle-to-Vehicle):
  - Enables collision warnings, emergency braking notifications, intersection movement assistance
  - Attack vectors include message spoofing, replay attacks, and Sybil attacks (creating multiple fake vehicle identities)
  - Privacy concerns regarding vehicle tracking and driver behavior monitoring
  - Potential risks include unauthorized access, spoofing, and man-in-the-middle (MITM) attacks.
- V2I (Vehicle-to-Infrastructure):
  - Connects vehicles to traffic signals, work zones, toll systems
  - Vulnerable to man-in-the-middle attacks where fake infrastructure could broadcast malicious data
  - Concerns about trust management in infrastructure deployment
  - Threats include fake traffic signal manipulation, denial-of-service (DoS) attacks on road infrastructure, and GPS spoofing.
- Security challenges:
  - Certificate management at scale (potentially millions of vehicles)
  - Balancing security needs with performance requirements
  - Message authentication with strict latency constraints (< 100ms)

Telematics & Infotainment Systems

- Connected vehicle services:
  - Remote start/unlock, location tracking, diagnostics
  - Often connected to manufacturer backend servers
  - Creates persistent internet connection to vehicle
- Infotainment vulnerabilities:
  - Often runs on standard operating systems with known vulnerabilities
  - Typically has connectivity to critical vehicle networks
  - Handles untrusted inputs (Bluetooth, USB, cellular data)
  - App stores and third-party applications introduce additional risks
- Cloud connectivity risks:
  - API vulnerabilities in backend systems
  - Authentication and authorization weaknesses
  - Data privacy concerns with location and usage information

Over-the-Air (OTA) Updates

- Benefits:
  - Rapid security patching without dealer visits
  - Continuous feature improvement and bug fixes
  - Reduced recall costs and improved customer satisfaction
- Security challenges:
  - Update authentication and verification mechanisms
  - Secure download and storage of updates before installation
  - Rollback protection and version control
  - Recovery mechanisms for failed updates
- Threat vectors:
  - Update server compromise
  - Man-in-the-middle attacks during download
  - Malicious updates via compromised signing keys
  - Denial of service during critical updates

CAN Bus & Ethernet Attacks

- CAN (Controller Area Network) vulnerabilities:
  - No built-in authentication or encryption
  - Broadcast nature means all nodes receive all messages
  - No sender identification mechanism
  - Limited bandwidth (up to 1 Mbps) constrains security add-ons
- Ethernet in automotive:
  - Enables higher bandwidth for ADAS, infotainment (100Mbps-10Gbps)
  - Allows network segmentation and improved security controls
  - Introduces IT-style attacks to automotive (ARP poisoning, MAC flooding)
  - New protocols like Automotive Ethernet have additional security considerations
- Network architecture security:
  - Domain isolation and gateway security
  - Intrusion detection systems for in-vehicle networks
  - Message authentication and encryption challenges

Tesla vs. Traditional OEMs

- Tesla's approach:
  - Centralized computing architecture with fewer ECUs
  - Cryptographically signed updates with mutual authentication
  - Secure boot chain and verification at multiple levels
  - Ability to rapidly deploy security patches
- Traditional approach limitations:
  - Multiple suppliers with different update mechanisms
  - Dealer-dependent update model with low customer compliance
  - Complex supply chain with varying security standards
  - Limited ability to quickly patch discovered vulnerabilities

Next GEN Automotive

Cybersecurity a universal challenge

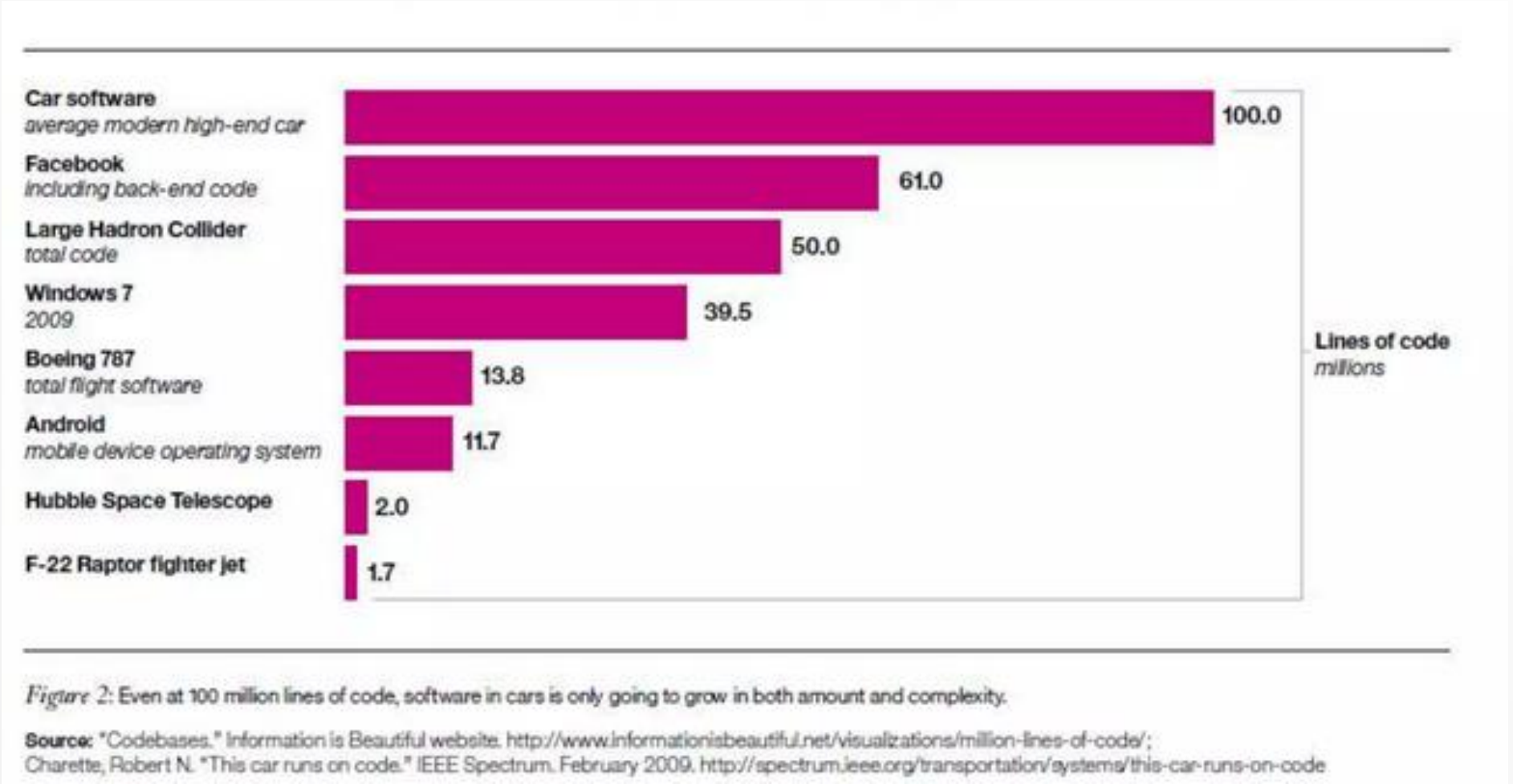
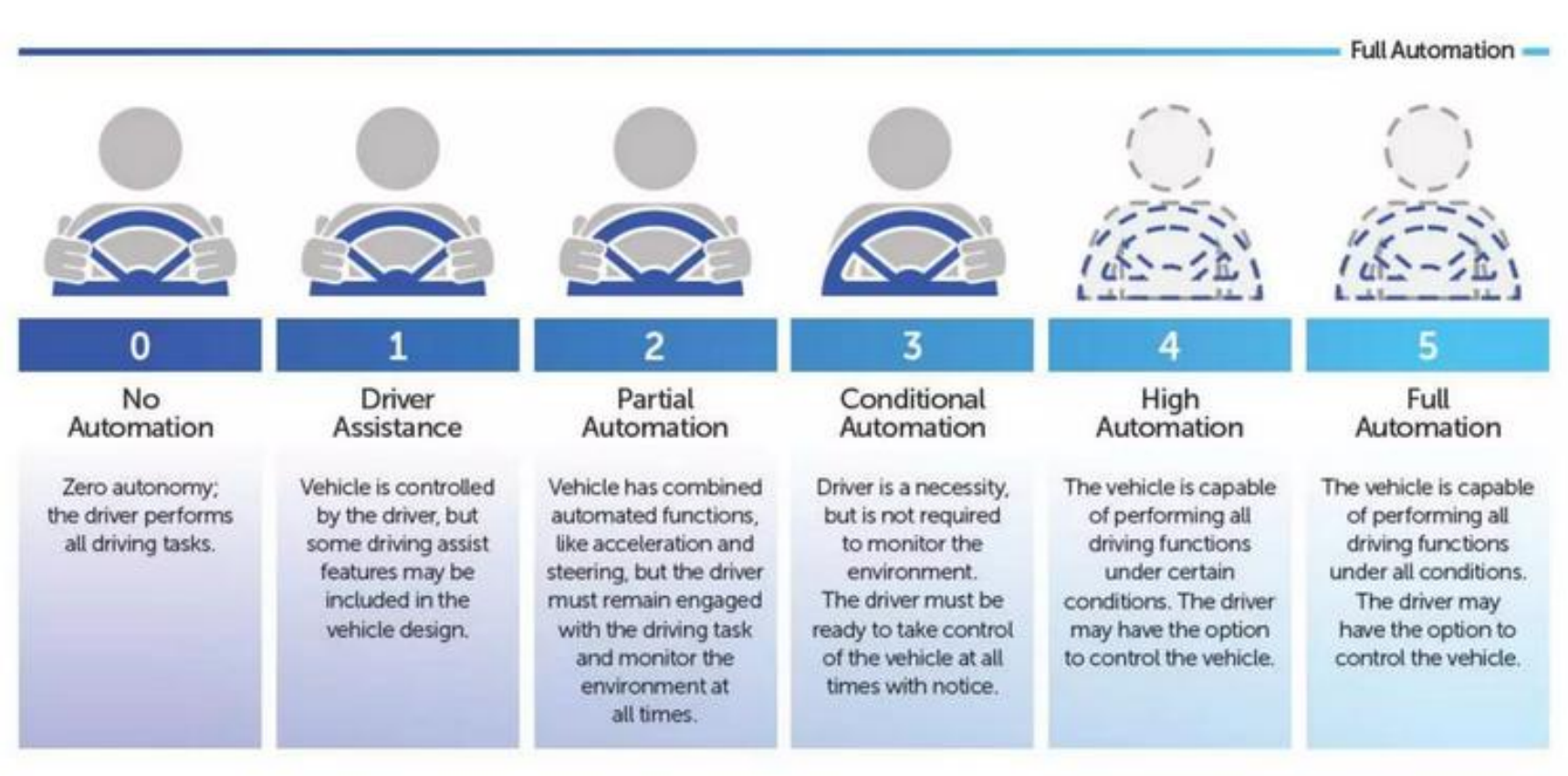


Figure 2: Even at 100 million lines of code, software in cars is only going to grow in both amount and complexity.

Source: "Codebases." Information is Beautiful website. http://www.informationisbeautiful.net/visualizations/million-lines-of-code/;
Charette, Robert N. "This car runs on code." IEEE Spectrum. February 2009. http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code

ADAS

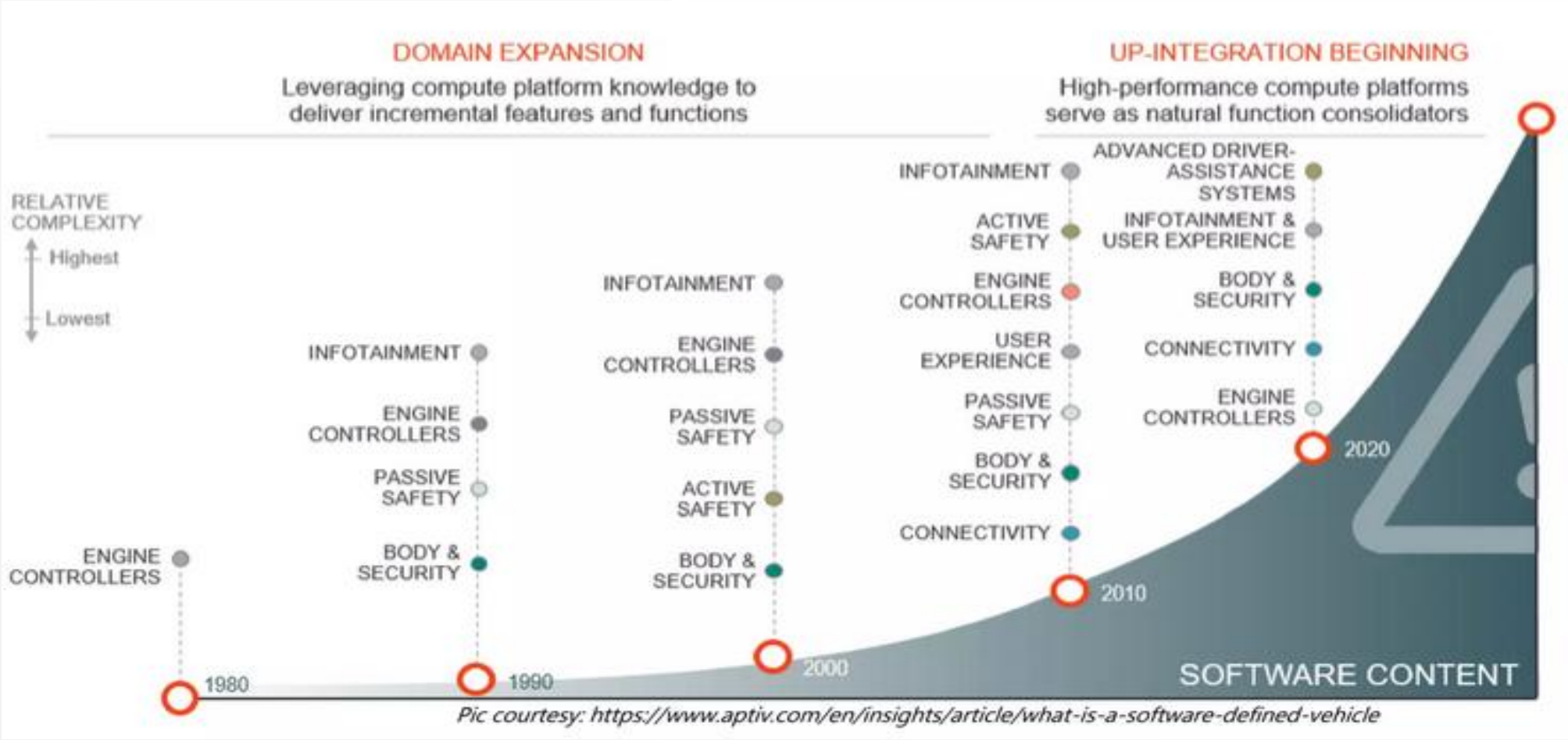
- System safety (Functional safety)
- Operational Design Domain
- Object and Event Detection and Response
- Fallback (Minimal Risk Condition)
- Validation Methods
- Human Machine Interface
- Vehicle Cybersecurity
- Crashworthiness
- Post-Crash ADS Behavior
- Data Recording
- Customer Education and Training
- Federal, State and Local Laws



Next Gen Automotive solutions

- 5G Connectivity
- Internet of Things (IoT)
- Vehicle Electrification
- Edge Computing
- Roadside Infrastructure Units (RSU)

Software defined vehicle



Pic courtesy: https://www.aptiv.com/en/insights/article/what-is-a-software-defined-vehicle

Security Standards & Regulations

UNECE WP.29 Cybersecurity Regulation

UN Regulation No. 155 requires:

- Mandatory Cybersecurity Management System (CSMS) certification for OEMs
- Vehicle type approval based on cybersecurity implementation
- Process for monitoring, detecting, and responding to cyber attacks
- Process for providing security updates throughout vehicle lifecycle

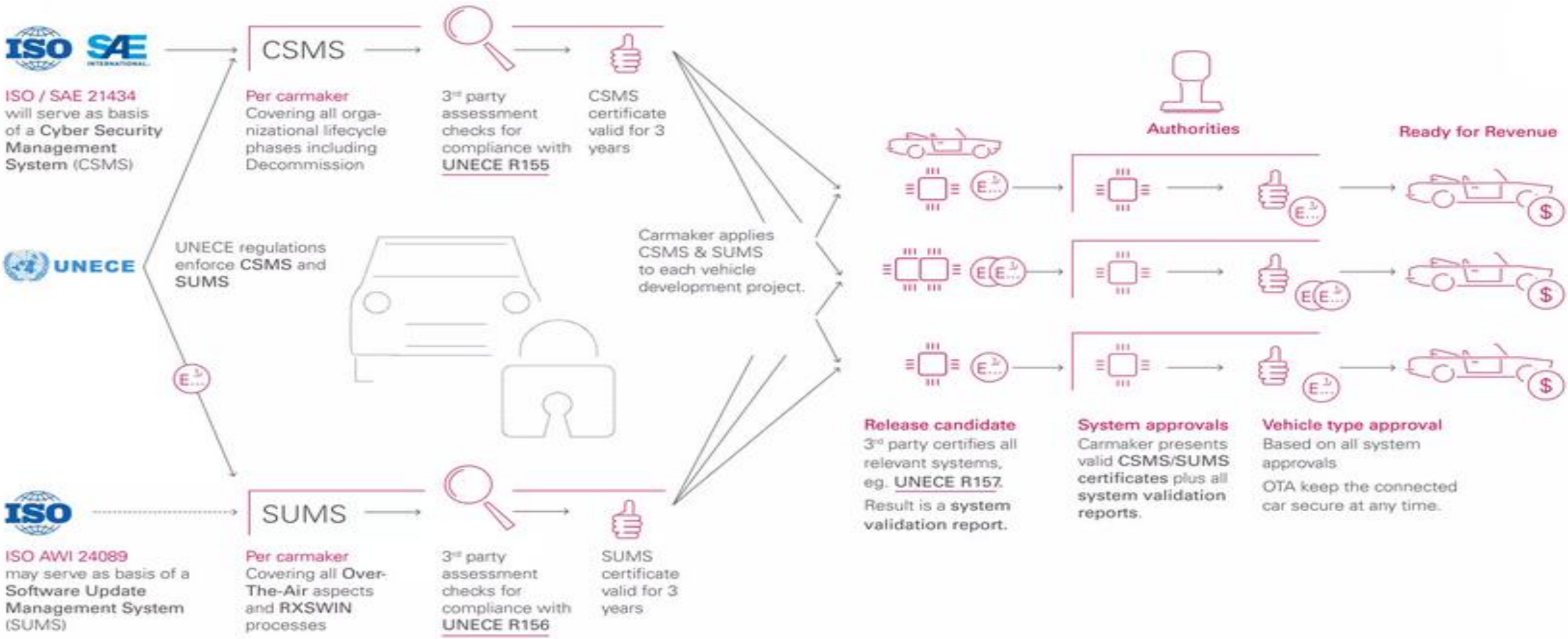
Regulatory impact:

- Applies to 54+ countries including EU, Japan, Korea
- Affects vehicles in categories M (passenger) and N (commercial)
- Required for new vehicle types since July 2022
- Required for all new vehicles since July 2024

CSMS requirements include:

- Risk assessment and management processes
- Security testing and validation procedures
- Security event monitoring capabilities
- Incident response procedures
- Supply chain security management

CSMS and SUMS – How do the Pieces fit together?



Carmaker's CSMS and SUMS are to be certified. Vehicles type certification checks for both general CSMS/SUMS compliance and vehicle-specific validation report prior to permit.

ISO/SAE 21434 – Road Vehicles Cybersecurity Engineering

Framework components:

- Organization-wide cybersecurity governance
- Project-specific cybersecurity management
- Continuous risk assessment throughout development
- Concept, product development, production, operation, and maintenance phases

Key processes:

- Threat analysis and risk assessment (TARA)
- Cybersecurity goals and claims definition
- Security controls implementation and verification
- Vulnerability management and incident response

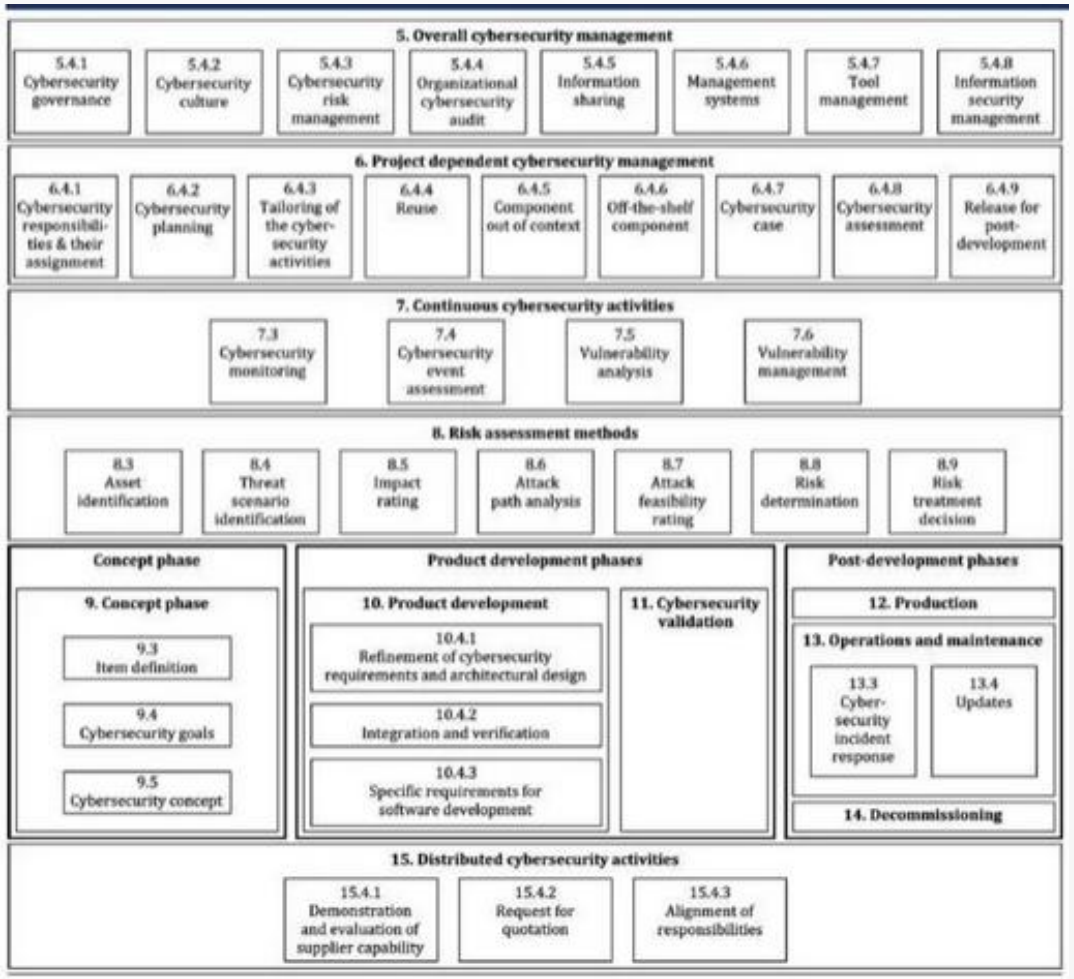
Integration with other standards:

- Automotive SPICE (ASPICE) for process assessment
- ISO 26262 for functional safety considerations
- ISO/PAS 21448 (SOTIF) for safety of intended functionality

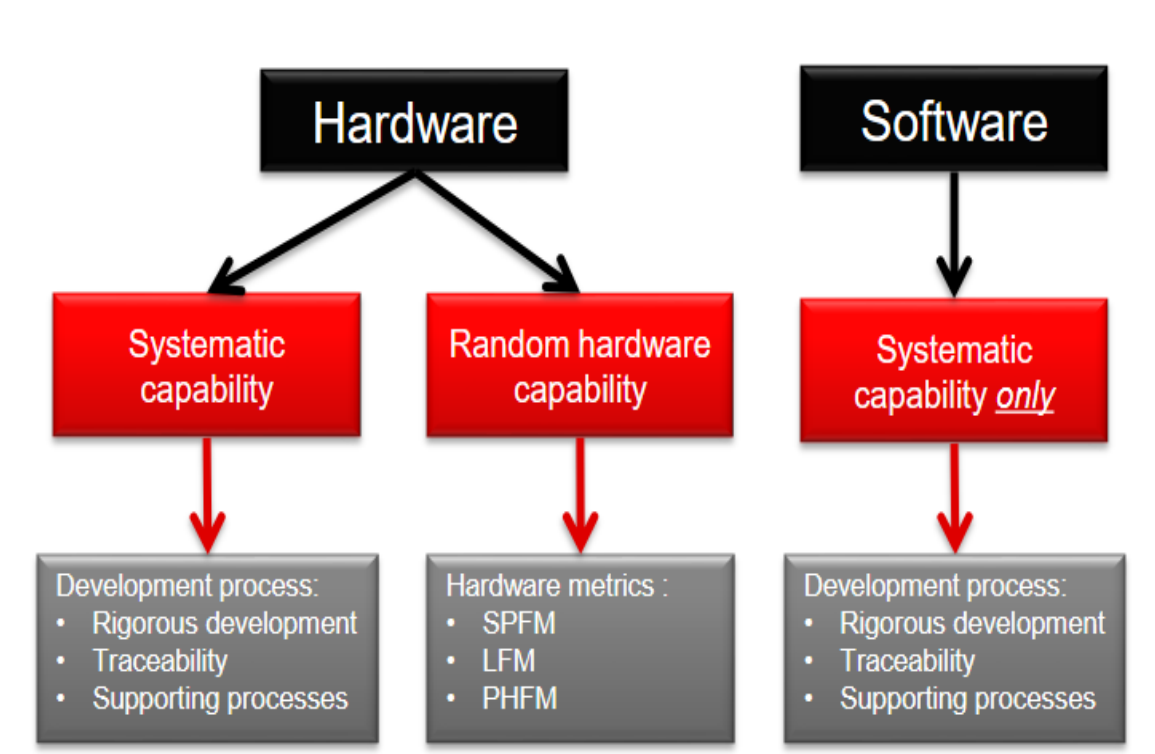
Documentation requirements:

- Cybersecurity case demonstrating adequate risk treatment
- Evidence of security testing and validation
- Clear traceability from threats to implemented controls

ISO 21434 overview



ISO 26262 – Road Vehicles Functional Safety



Core Framework: Automotive-specific adaptation of IEC 61508 for E/E systems safety

ASIL Classification: Risk-based approach (A-D) based on Severity, Exposure, Controllability

Key Processes:

- Hazard Analysis and Risk Assessment (HARA)
- Functional and Technical Safety Concepts
- Safety-driven design and implementation
- Comprehensive verification and validation

Lifecycle Coverage: 10-part standard covering management, concept, system, HW/SW development, production and operation

Implementation Approach: Safety goals → Safety requirements → Safety mechanisms → Safety validation

Industry Impact: De facto standard for automotive functional safety, often contractually required by OEMs

Automotive SPICE (ASPICE)

Framework Purpose: Process assessment model for automotive software and embedded systems development

Capability Levels: 0 (Incomplete) through 5 (Optimizing), assessed as F/L/P/N achievement

Process Areas:

- Acquisition (ACQ) and Supply (SPL)
- System (SYS) and Software Engineering (SWE)
- Supporting (SUP) and Management (MAN)

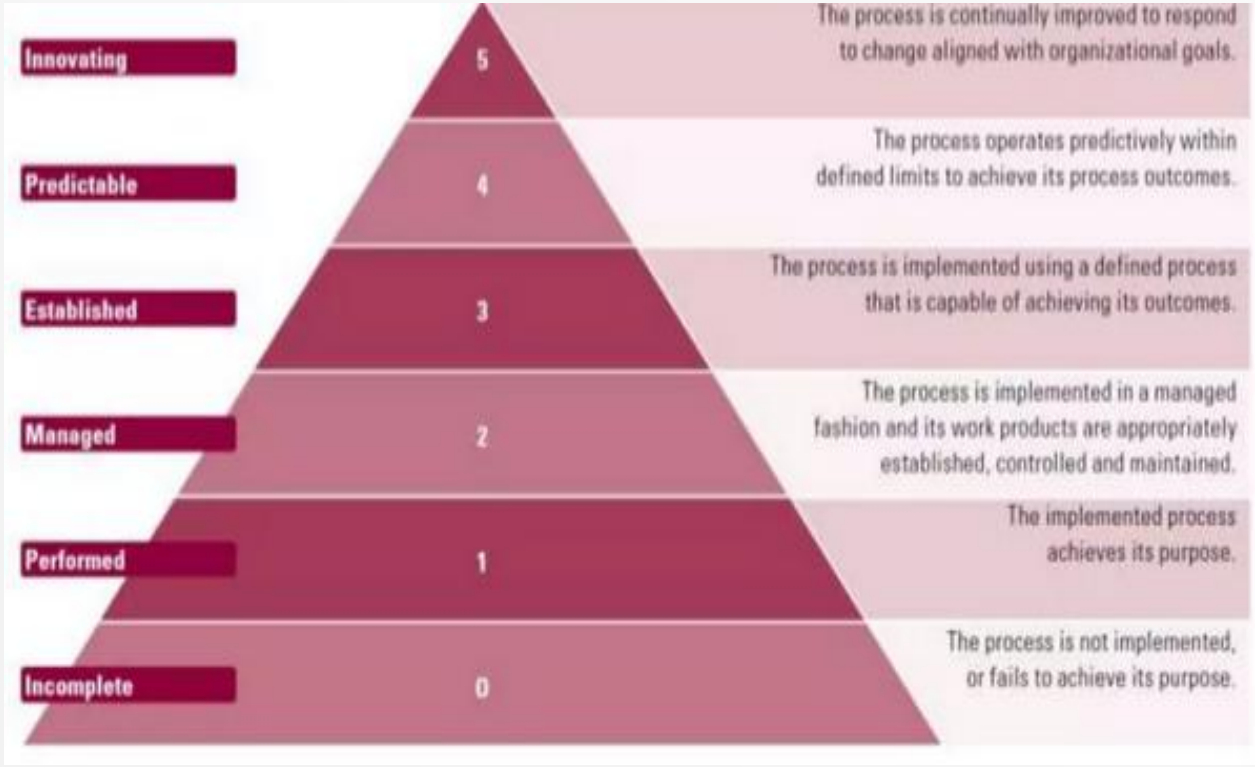
Assessment Methodology: Evidence-based evaluation through documentation, interviews, and observations

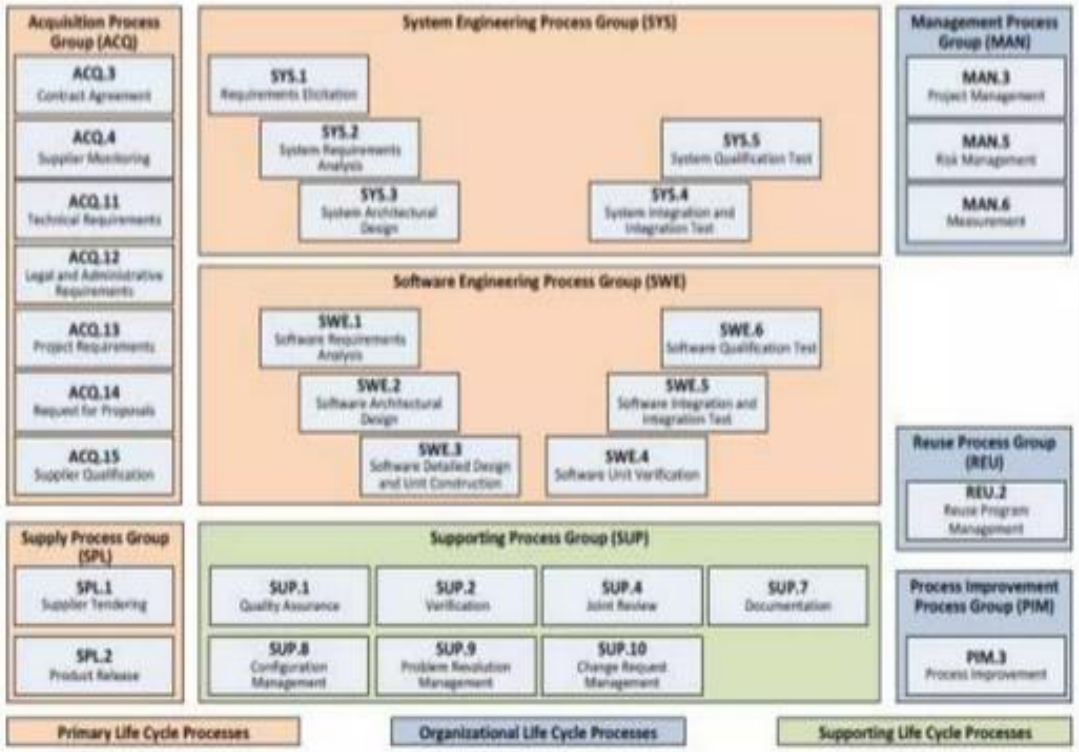
OEM Application: Used to qualify suppliers, establish minimum capability requirements, and drive improvement

Key Value: Enables standardized process maturity comparison across suppliers and projects

Integration: Aligns with functional safety and security standards for efficient implementation

ASPICE overview





ISO 21448 (SOTIF) - Safety Of The Intended Functionality

Purpose: Addresses safety hazards not caused by failures (complementing ISO 26262)

Core Concepts:

- Performance limitations and boundary conditions
- Foreseeable misuse scenarios
- Known safe, known unsafe, and unknown unsafe regions

Methodology:

- Hazard identification beyond traditional failure modes
- Scenario-based testing and validation
- Comprehensive safety argumentation

Application Focus: Critical for ADAS, autonomous driving, and AI-based systems

Validation Approach: Combines simulation, testing, field monitoring, and statistical validation

Safety Case: Documented evidence demonstrating system safety despite performance limitations

Emerging Importance: Essential standard for complex systems with emergent behaviors

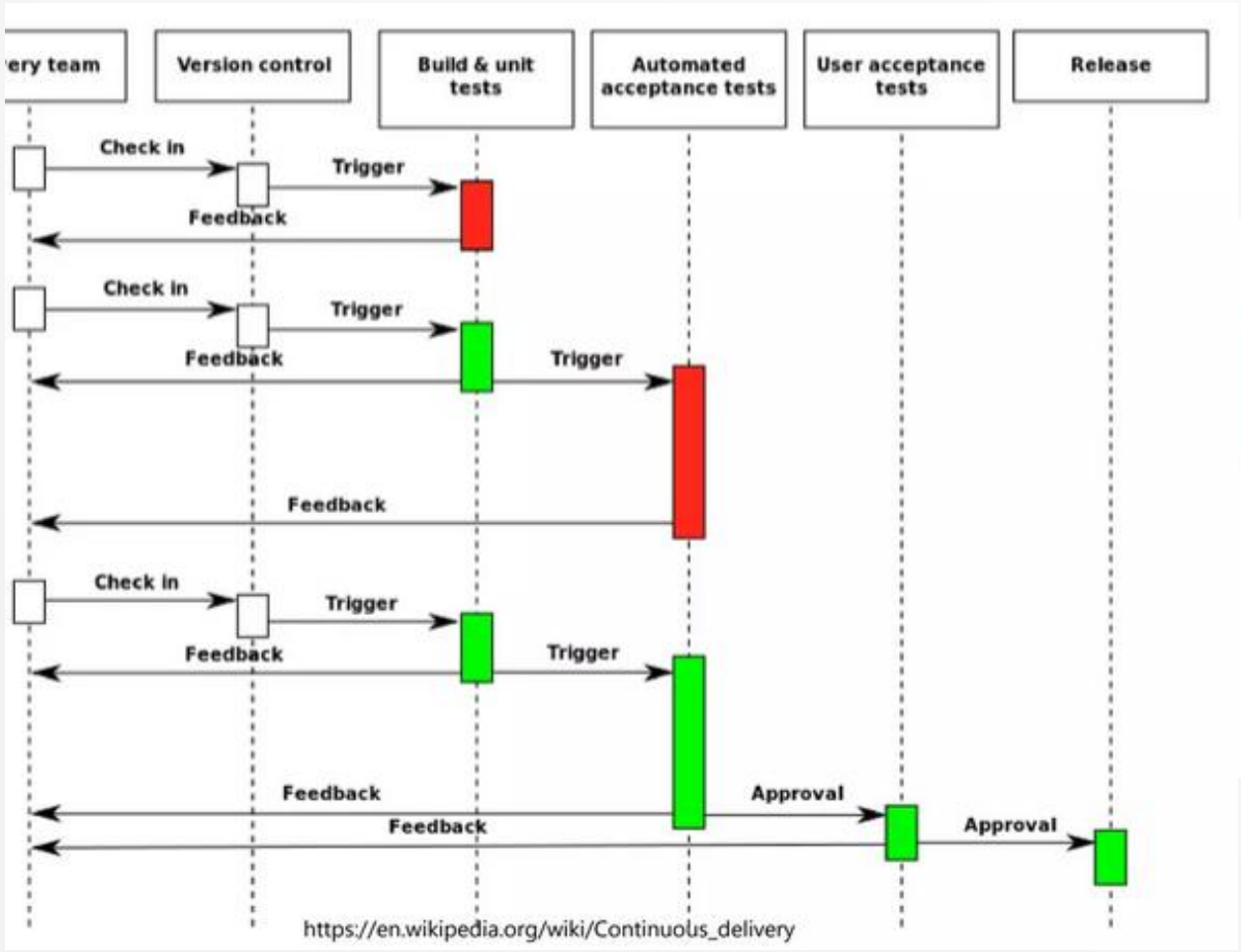
Relationship Between Automotive Standards

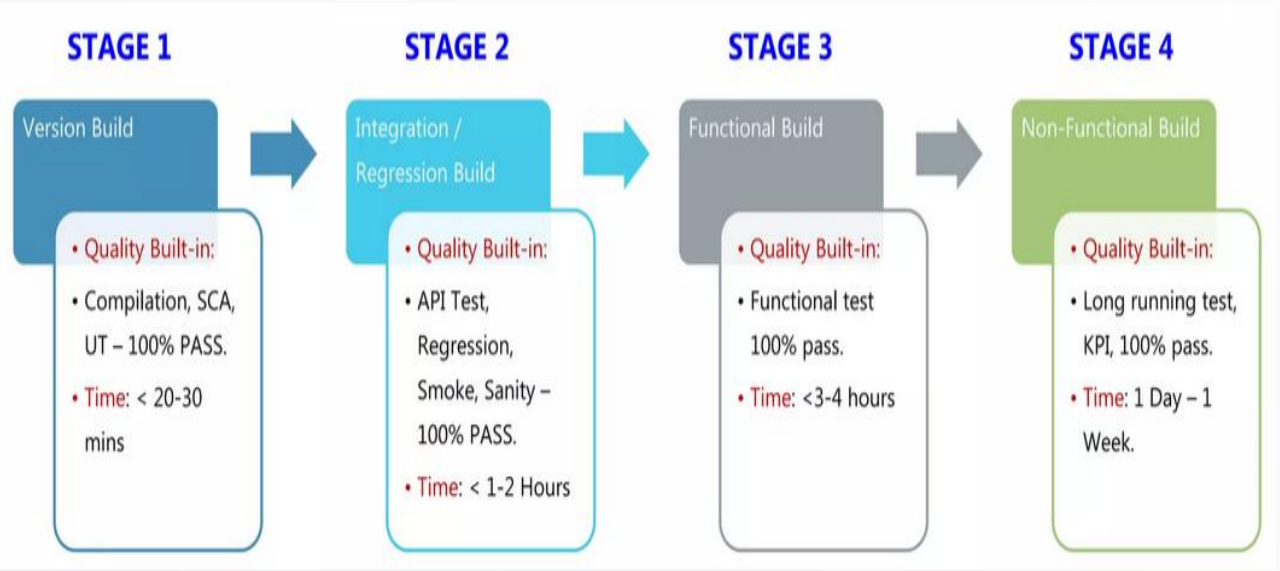
- Integrated Framework:
  - ISO 26262: Addresses systematic and random hardware failures
  - ISO 21434: Covers cybersecurity threats and vulnerabilities
  - ISO 21448: Handles performance limitations and misuse
  - ASPICE: Ensures mature development processes
  - WP.29: Provides regulatory type approval framework
- Complementary Coverage: Safety + Security, Process + Product, Known + Unknown Risks
- Implementation Synergies:
  - Shared activities (risk assessment, requirements management)
  - Aligned documentation and evidence collection
  - Coordinated organizational roles and responsibilities
- Unified Approach: Cross-standard implementation from requirements through certification
- Industry Trend: Moving toward integrated compliance and certification processes
- Future Direction: Harmonized standards addressing increasingly complex automotive systems

Development perspective

Category	Standard	Description
Automotive SW Development Standards	ISO 26262	Functional Safety – Road Vehicles, A risk-based functional safety standard. Applies to the electric and electronic systems in vehicles including ADAS components
	SOTIF/ISO 21448	Safety of Intended Functionality - It considers situations that cause safety hazards that do not result from system failures
	ISO 21434/SAE J3061	Road vehicles — Cybersecurity engineering,
	UNECE WP.29	Automotive Regulation – Defines Functional Requirements for automated/ autonomous vehicles, Cyber security on OTA , Data Storage in automated driving vehicle and New Test/Assessment Methods
	TR68:Part 3:2019	Cybersecurity principles and assessment framework (Singapore), similar to ISO 21434
	CERT	CERT is a secure coding standard that supports C, C++, and Java
	MISRA	Motor Industry Software Reliability Association - Coding Standards for C,C++
	AUTOSAR	Automotive Open System Architecture (AUTOSAR) – Coding standard for C++14 to ensure that automotive software is safe, secure, and reliable.
Automotive Quality Standards	ASPICE	Automotive Software Performance Improvement and Capability determination (ASPICE) as a standard provides the framework for defining, implementing, and evaluating the process required for system development focused on software and system parts in the automotive industry.
	IATF 16949	International Standard For Automotive Quality Management Systems, The global automotive industry standard for quality management systems

Continuous Delivery





Compliance Throughout the Vehicle Lifecycle

- Concept phase:
  - Define asset identification and cybersecurity goals
  - Perform preliminary TARA
  - Establish cybersecurity concept with security requirements
- Development phase:
  - Implement security controls based on requirements
  - Conduct component and integration testing
  - Verify controls against security requirements
- Production phase:
  - Ensure production environment security
  - Protect against supply chain tampering
  - Verify software integrity during manufacturing
- Post-production phase:
  - Monitor for new vulnerabilities
  - Test and deploy security updates
  - Maintain incident response capabilities

Security Principles & Best Practices

Secure Boot & Secure Firmware Updates

- Secure boot process:
  - Root of Trust (RoT) verification using immutable hardware-based security
  - Chain of trust from bootloader to OS to applications
  - Cryptographic verification at each boot stage
  - Integrity checking of critical software components
- Secure update mechanisms:
  - Cryptographic signing of update packages
  - Version rollback protection
  - Secure storage of update before installation
  - Atomic updates with recovery mechanisms
  - Verification before execution

Hardware Security Modules (HSM)

- Dedicated cryptographic processors for secure key storage and encryption operations.
- HSM capabilities:
  - Secure key storage and management
  - Hardware-accelerated cryptographic operations
  - Tamper resistance and detection
  - Secure boot support
- Examples:
  - Infineon AURIX TPU,
  - NXP CSE,
  - Renesas RH850 SCU

Trusted Execution Environments (TEE)

- Isolated execution environment for running security-critical code separately from the main OS.
- TEE features:
  - Isolated execution environment separate from main OS
  - Protected memory regions
  - Secure storage capabilities
  - Reduced attack surface for critical functions
- Examples:
  - Arm TrustZone,
  - Intel SGX

Securing the Supply Chain with SBOM

- SBOM (Software Bill of Materials)
- Tracks all software components, including third-party libraries, to detect vulnerabilities.
  - Comprehensive inventory of all software components
  - Identification of open source components and versions
  - Documentation of dependencies and licensing information
  - Enables vulnerability tracking and management
- Aligns with **Executive Order 14028 on cybersecurity**, emphasizing transparency and risk management.

Executive Order 14028 requirements

- Executive Order 14028 requirements:
  - Mandates SBOM for software sold to federal government
  - Increasing adoption in automotive industry
  - Standard formats include: SPDX, CycloneDX, SWID
- Supply chain security practices:
  - Supplier security assessment and requirements
  - Secure development practices throughout supply chain
  - Third-party component evaluation and testing
  - Continuous monitoring of component vulnerabilities

Penetration Testing & Threat Modeling

- TARA methodology:
  - Asset identification and valuation
  - Threat scenario identification
  - Impact assessment
  - Vulnerability analysis
  - Risk determination and prioritization
- Penetration testing approaches:
  - Black box (no prior knowledge)
  - White box (complete system knowledge)
  - Gray box (partial knowledge)
- Testing focus areas:
  - Communication interfaces (CAN, Ethernet, Bluetooth, cellular)
  - Diagnostic protocols and services
  - Software update mechanisms
  - External connectivity points
- Red team exercises:
  - Simulate real-world attackers
  - Test detection and response capabilities
  - Evaluate security controls effectiveness
  - Identify procedural and technical gaps

Firmware Update Exploitation

- Vulnerable update process:
  - Update downloaded over unencrypted channel
  - Insufficient signature verification
  - No version control or anti-rollback protection
- Attack methodology:
  1. Intercept update traffic via man-in-the-middle attack
  2. Modify update package with malicious code
  3. Forge or bypass signature verification
  4. Send modified update to vehicle
  5. Exploit elevated privileges once executed
- Security controls to prevent attack:
  - Mutual authentication between vehicle and update server
  - Transport layer security for update download
  - Multi-stage verification of update package
  - Hardware-backed secure storage for verification keys
  - Secure boot to prevent unauthorized code execution

The 2015 Jeep Cherokee Hack

- A remote attack demonstrated by security researchers on a Jeep Cherokee, exploiting vulnerabilities in the infotainment system to gain control over critical vehicle functions.
- **Implications:** Led to recalls, regulatory responses, and increased focus on securing vehicle communication and software update mechanisms.

Industry Trends & Future Challenges

Shift to Software-Defined Vehicles

- Architectural evolution:
  - Moving from 200+ distributed ECUs to centralized high-performance computers
  - Separation of hardware and software development cycles
  - Containerization and virtualization of vehicle functions
  - Microservices-based architecture replacing monolithic systems
- Cloud-based architectures:
  - Offloading computation to cloud services
  - Connected vehicle platforms with continuous integration
  - Fleet-wide data collection and analysis
  - Cybersecurity implications for distributed systems
- Security challenges:
  - Increased complexity with millions of lines of code
  - Hypervisor and container security considerations
  - More frequent updates increasing potential vulnerability windows
  - Balancing innovation speed with security requirements

DevSecOps in Automotive Development

- Integration of security in CI/CD pipelines:
  - Automated security testing in development workflow
  - Static and dynamic code analysis
  - Composition analysis for third-party components
  - Continuous vulnerability scanning
- Shift-left security approach:
  - Security requirements defined early in development
  - Threat modeling during design phase
  - Developer security training and awareness
  - Security champions embedded in development teams
- Automated security validation:
  - Fuzzing and penetration testing in CI pipeline
  - Security regression testing for updates
  - Compliance checking against standards
  - Security metrics and dashboards

AI & ML in Automotive Cybersecurity

- Anomaly detection applications:
  - Baseline normal vehicle network behavior
  - Identify unusual message patterns or timing
  - Detect potential intrusions in real-time
  - Reduce false positives with context-aware models
- Threat intelligence:
  - Predict emerging threats based on patterns
  - Prioritize vulnerabilities based on exploitation likelihood
  - Automate incident classification and response
  - Enable proactive security posture
- Challenges:
  - Ensuring AI model security itself
  - Balancing detection sensitivity with false positives
  - Resource constraints in embedded environments
  - Training data quality and quantity

Post-Quantum Cryptography

- Quantum computing threat:
  - Potential to break current asymmetric cryptography (RSA, ECC)
  - Risk to digital signatures for secure boot and updates
  - Long vehicle lifecycles require forward security planning
  - NIST standardization efforts underway
- Implementation considerations:
  - Performance impact on resource-constrained ECUs
  - Key size and computational requirements
  - Backward compatibility with existing systems
  - Transition strategies from current cryptography
- Automotive-specific challenges:
  - Limited update capabilities in deployed vehicles
  - Real-time performance requirements
  - Diversity of computing platforms across vehicle
  - Need for crypto-agility in future designs

Tesla's Intrusion Detection System

- Tesla's neural network-based approach:
  - Uses vehicle's existing compute resources
  - Monitors internal network traffic patterns
  - Identifies anomalous behavior in real-time
  - Updates detection models via OTA updates
- Implementation details:
  - Baseline established during normal operation
  - Contextual awareness of driving conditions
  - Low false positive rate through continuous learning
  - Integration with central security operations
- Benefits demonstrated:
  - Rapid detection of potential intrusions
  - Ability to contain and isolate affected systems
  - Forensic data collection for incident analysis
  - Continuous improvement through fleet-wide learning

Q&A