Autonomous Vehicle Cybersecurity Development Lifecycle (*AVCDL*)

Charles Wilson
Principal Engineer, Cybersecurity Development Lifecycle Practice

Version 31
7/15/2021 2:46:00 PM

Table of Contents

List of Tables

List of Figures

1. Introduction

Abstract

This material documents the Autonomous Vehicle Cybersecurity Development Lifecycle (***AVCDL***).

Questions, Errors and Other

This documentation is maintained by Charles Wilson. Please feel free to reach out should you have any questions.

Scannable email query QR

Note: QR code generated using.

License

Organization

This document is organized as follows:

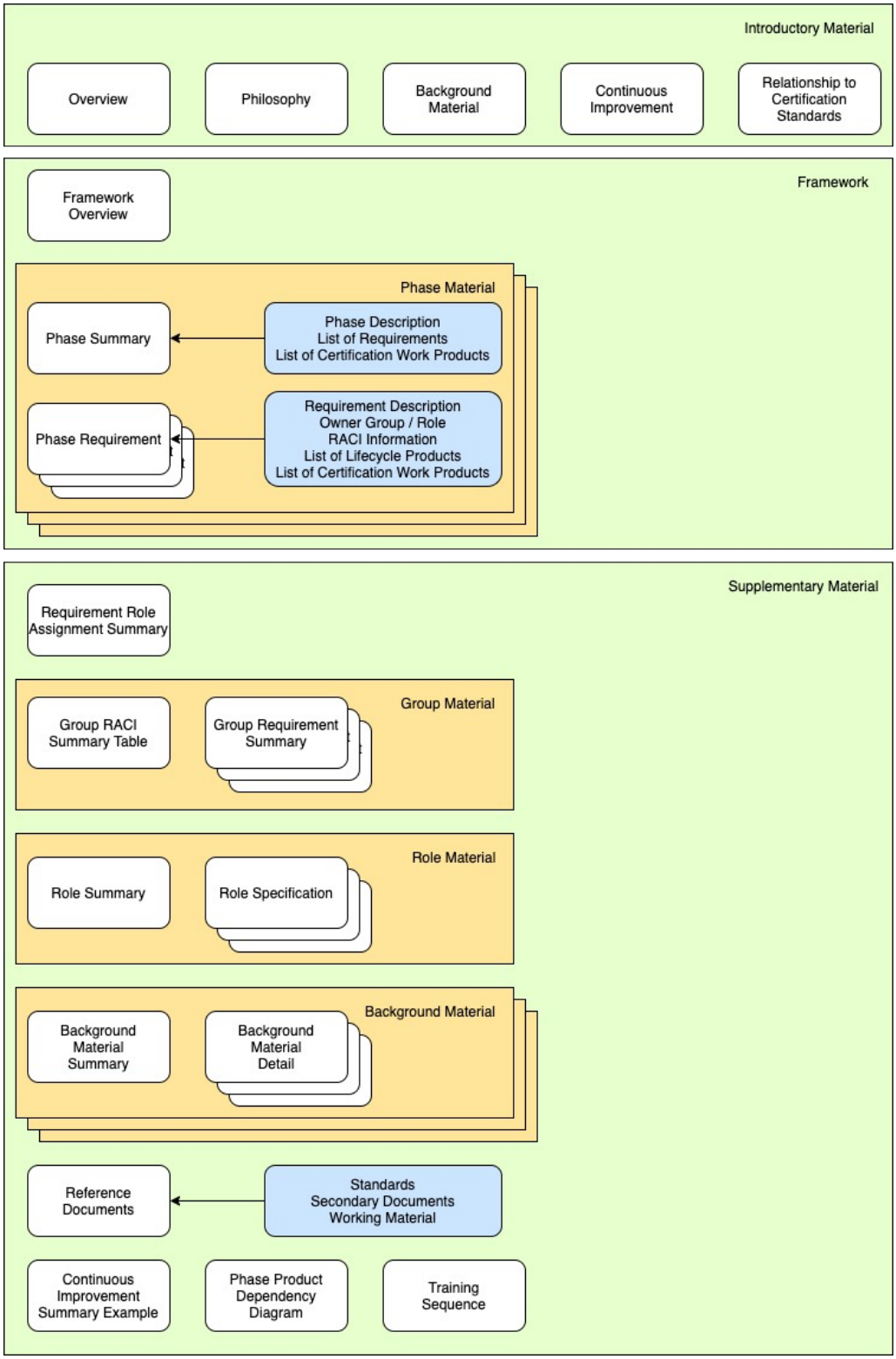

Figure 1 - Document Organization

As can be seen, there are three sections:

- Introductory material
- Framework (main AVCDL material)
- Supplementary material

Introductory Material

The introductory material provides a general overview of the **AVCDL**, its philosophy, and background material. It also discusses how continuous improvement is measured and tracked. The relationship between the **AVCDL** and **ISO 21434** is explored, as is how hardware and software are treated.

Framework

This is the core of the AVCDL. An overview of the AVCDL framework is presented. Following that each phase of the lifecycle is summarized, and the associated phase requirements are detailed. Reference to the associated **ISO 21434** work products and requirements are included.

Supplementary Material

The supplementary material falls into two broad categories. The first is background material. These include **NCWF** [10], **MSSDL**, and **SSDF**. The second is summary material. This includes RACI rollup, secondary document lists, a continuous improvement example, phase dependency graphs and recommended training sequence.

Note: Both the main **AVCDL** and its secondary documents are process in nature. They do not specify the tools and specific techniques used to accomplish their tasks. As such, the **AVCDL** is suitable for use by any organization engaged in development related to autonomous vehicles. It is presumed that each organization adopting the **AVCDL** will create a set of tertiary documents supporting the secondary ones. These tertiary documents will be procedural. Additionally, there may be multiple tertiary documents supporting a given secondary document. For example, a secondary document on secure code review might be supported by multiple tertiary documents, one for each programming language.

2. Overview

What it is

The **AVCDL** is a set of identified processes, requirements of those processes, generated products, and mappings from the generated products to their corresponding certification standard (**ISO/SAE 21434, UNECE WP.29**) work products: for the purpose of ensuring the creation of secure systems. It is intended to support auditing of the development process in the area of cybersecurity as specified in those standards.

What it isn't

The ***AVCDL*** does not attempt to specify:

- implementation methodology (waterfall, V-model, agile-scrum, agile-Kanban, TDD, BDD, spiral, …)
- specific development tools (source code control, build system, compilers, threat modeling tools, static analysis tools, …)
- remediation methodology

Where it fits

The **AVCDL** is not a standalone solution. It is intended to implement the non-governance elements of a larger product development lifecycle framework (**AVPDL**). Moreover, it is designed to overlay the system (**ISO/IEC 15288**) and software (**ISO/IEC 12207**) lifecycles and complement the safety (**ISO 26262**) lifecycle.

We can visualize the relationship between the **AVCDL** and various standards as follows:

<table>
<tr><th>AVPDL</th><th>AVCDL</th><th>15288</th><th>12207</th><th>26262</th><th>21434</th></tr>
<tr><td rowspan="2">organization processes</td><td rowspan="2">N/A</td><td rowspan="2">technical processes</td><td rowspan="2">technical processes</td><td>management of functional safety</td><td>overall cybersecurity management</td></tr>
<tr><td>supporting processes</td><td>project dependent cybersecurity management</td></tr>
<tr><td>foundation phase</td><td>foundation phase</td><td>N/A</td><td>N/A</td><td>concept phase</td><td>concept phase</td></tr>
<tr><td rowspan="2">requirements phase</td><td rowspan="2">requirements phase</td><td>requirements definition</td><td>requirements definition</td><td>safety requirements</td><td>cybersecurity requirements</td></tr>
<tr><td>requirements analysis</td><td>system requirements analysis</td><td>hazard analysis / risk assessment</td><td>cybersecurity assessment</td></tr>
<tr><td>design phase</td><td>design phase</td><td>architectural design</td><td>system architectural design</td><td>architectural design</td><td>cybersecurity design</td></tr>
<tr><td rowspan="2">implementation phase</td><td rowspan="2">implementation phase</td><td>implementation</td><td>implementation</td><td>implementation</td><td>development</td></tr>
<tr><td>integration</td><td>system integration</td><td rowspan="3">integration and verification</td><td rowspan="3">integration and verification</td></tr>
<tr><td rowspan="3">verification phase</td><td rowspan="3">verification phase</td><td>verification</td><td>system qualification testing</td></tr>
<tr><td rowspan="2">transition</td><td>software installation</td></tr>
<tr><td rowspan="2">software acceptance support</td></tr>
<tr><td>release phase</td><td>release phase</td><td>validation</td><td>production</td><td>production</td></tr>
<tr><td rowspan="2">operation phase</td><td rowspan="2">operation phase</td><td>operation</td><td>software operation</td><td rowspan="3">operation, service, and decommissioning</td><td>continuous cybersecurity activities</td></tr>
<tr><td>maintenance</td><td>software maintenance</td><td>operation and maintenance</td></tr>
<tr><td>decommissioning phase</td><td>decommissioning phase</td><td>disposal</td><td>software disposal</td><td>decommissioning</td></tr>
<tr><td>supplier processes</td><td>N/A</td><td>agreement processes</td><td>agreement processes</td><td>supporting processes</td><td>distributed cybersecurity activities</td></tr>
</table>

Table 1 - Relationship Among Standards

Note: The **AVCDL** does not attempt to address either the organization or supplier-related processes because these are managed at the organizational level.

3. Philosophy

The creation of secure software is not simply a programming endeavor. It begins with ensuring all team members understand how software and systems are made secure; requires the additional stages in the design and testing phase; and permeates all typical development practices.

It is impossible to integrate security into a large operational system in a single pass. This is a reality stemming from the lack of security focus within the computing industry. In order to be effective, security must be seen as emergent property and not an adjunct capability. This stands in contrast to the recent trend toward minimal functional development. For a system to be secure it must be secure by design, not coincidence. Therefore, there are some foundational elements which should be in place prior to implementation.

Given the scope of the problem space, the most appropriate approach entails:

- creation of an **implementation framework**
- **continuous improvement** of the security posture

4. Background Material

The ***AVCDL*** is based on methodologies proven in industry as well as standards bodies' recommendations. These include **MSSDL** [6], **SSDF** [11], **ISO 21434** [4], and **ISO 26262**.

Microsoft SDL (MSSDL)

The archetype for the cybersecurity development lifecycle is the Microsoft SDL (**MSSDL**). It divides the development process into seven phases. These phases form a cycle of ever-improving security posture.

NIST SSDF (SSDF)

The NIST Secure Software Development Framework (**SSDF**) provides a more general approach which calls out several practices and provides references to the applicable standards. Within each of these are multiple practices and tasks.

The advantage of the **SSDF** over the **MSSDL** is that it provides a greater level of specificity and better supports existing international standards. It also calls out practices assumed, but not specified in the **MSSDL**.

ISO/SAE 21434 ('434)

Road Vehicles - Cybersecurity Engineering (**'434**) is intended to address the cybersecurity aspects of electrical and electronic (E/E) systems within road vehicles. Its goal is to enable organizations to:

- define cybersecurity policies and processes
- manage cybersecurity risk
- foster a cybersecurity culture

Like the **MSSDL**, the development process is divided into phases.

Unlike **MSSDL** and **SSDF**, which are lifecycle-focused, non-domain-specific documents, **'434** is a regulatory-focused, domain-specific (road vehicle E/E systems) work.

ISO 26262 ('262)

Road vehicles — Functional safety (**'262**) is intended to address the safety aspects of electrical and electronic (E/E) systems within road vehicles. Although not used as a primary source reference for the **AVCDL**, the **AVCDL** can be aligned to it. This allows for easier integration with existing development processes.

ISO/IEC/IEEE 12207 ('207)

Systems and software engineering – Software life cycle processes (**'207**) provides a set of processes required to systematically implement a development lifecycle. This should be considered the grounding process alignment document.

UNECE TRANS WP.29 GRVA (WP29)

The United Nations' Economic Commission for Europe's Inland Transport Committee's World Forum for Harmonization of Vehicle Regulations' Working Party on Automated/Autonomous and Connected Vehicles (**WP29** [15]) has created a set of guidance documents. Their intent is to

- [define] principles to address key cyber threats and vulnerabilities identified in order to assure vehicle safety in case of cyber-attacks
- [define] detailed guidance or measures for how to meet these principles … [including] examples of processes and technical approaches
- [consider] what assessments or evidence may be required to demonstrate compliance or certification with any requirements identified

These documents are intended to provide a common set of definitions and principles. For implementations, they point to specific standards, such as **'262**, **'434** and **ITU-T X.1500**. **UNECE** regulations **R155**, **R156**, and **R157** are the ones of specific interest. Within this document **WP.29** will be use to refer either the guidance documents or the specific regulations listed above.

Contributions Visualized

These sources come together as follows:

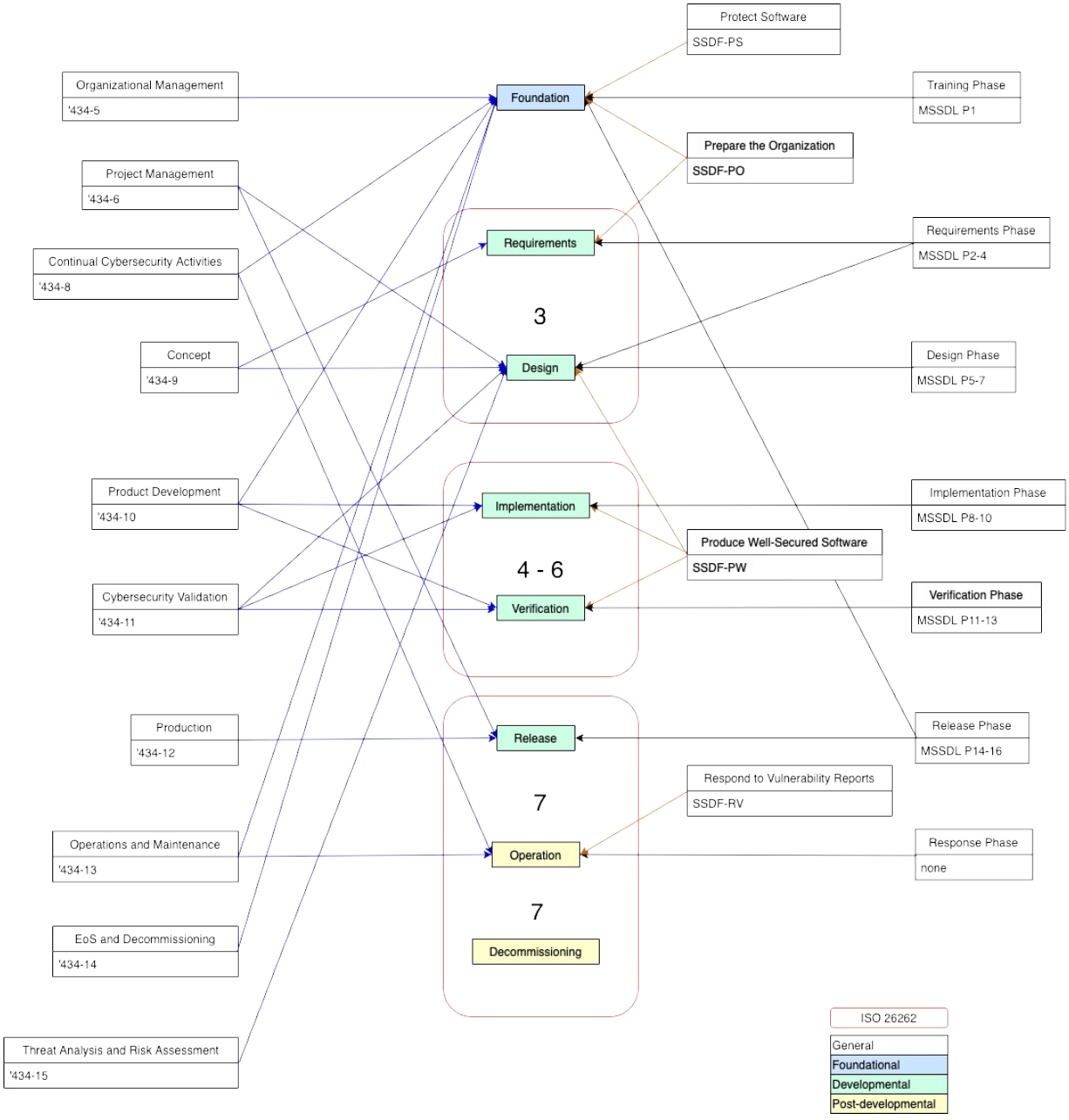

Figure 2 - AVCDL sources

Note: The red rounded rectangles indicate corresponding **'262** processes.

5. Continuous Improvement

The creation of secure software is not simply a programming endeavor. It begins with ensuring all team members understand how software and systems are made secure; requires the additional stages in the design and testing phase; and permeates all typical development practices.

As a system of systems, an autonomous vehicle will always be subject to variable-rate development. Each system within the system of systems is developed at a rate which may (and probably does) differ from that of other systems. That being the case, it is highly unlikely that the security profile of the system of systems will be the same as any individual system with it. This motivates the adoption of a process of continual improvement within each of the constituent systems, driving toward an ever-improved overall security profile.

Continuous improvement is not merely *pro forma*, but requisite. Whether we are considering the initial implementation of security or ongoing development, there will be the need to develop and refine the security model. This will need to be done at every level of the system (physical, network, protocol, and application) as well as across all the various sub-systems. Given the time-consuming nature of threat modeling, risk assessment, and threat mitigation; the only practical approach is to apply ever-increasing levels of security. This may manifest as an outside-in approach wherein the external attack surfaces are secured first with the interior system following; or by addressing the fundamental mechanisms by which data is managed within the system; or a combination of the two, dependent upon the maturity of the security in place in any given sub-system.

The ***AVCDL*** itself will also be subject to continuous improvement. This may manifest in changes to individual phases and their associated requirements. It may include changes in ownership of various phase requirements. There will always be new tools to consider in implementation of the ***AVCDL***.

In order to track progress of implementation of the ***AVCDL*** within the organization, **ISO 21827** Systems Security Engineering - Capability Maturity Model (**SSE-CMM** [13]) will be used as criteria for evaluation. Additionally, applicable elements of Cybersecurity Maturity Model Certification (**US DoD CMMC** [1]) level assignments of the requirements called out in ***Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations*** (**NIST SP 800-171**) [9] are identified and tracked for each ***AVCDL*** phase requirement.

An example tracking report is shown here.

6. Relationship to Standards

As stated in the **overview**, the ***AVCDL*** is a set of identified processes, requirements for those processes, generated products, and their mapping to the corresponding work products of various standards. This section will provide more detail about the relationship between the ***AVCDL*** and these standards.

Compliance versus Conformance

The ***AVCDL*** is designed to enable the organization to comply with the requirements of and enable the creation of products capable of satisfying the specifications for work products called by various certification standards. It does not conform to these standards in that it does not assume that any given certification standard's structure matches the form of any given organization's product development lifecycle framework, which encompasses development, safety, security and other needs. Any certification methodology with processes not conforming to the organization's actual development processes will be error-prone and unsuccessful.

Product Mapping

The ***AVCDL*** is designed to overlay an **ISO 15288** [3] **('288)** / **ISO 12207** [2] **('207)** lifecycle. As such, roughly fifty products are generated during any given product release's lifetime. The ***AVCDL*** provides mapping of these, which are the natural outcomes from implementation of cybersecurity best practices to their related standard work products. In this way, developmental friction is reduced as there is no expectation that individual contributors from any group (product management, risk, devops, development, …) be familiar with these standards and their details. This would, in fact, be seen as a negative as standards such as **'434** and **WP.29** are only the first of what will become many jurisdictional compliance standards needing to be complied with. By having a best practice lifecycle (***AVCDL***), we are more readily able to embrace compliance with regulatory standards as they appear.

7. Hardware-Software Relationship

Upon reading the ***AVCDL***, the question often arises as to why there is no specific reference to the hardware aspects of cybersecurity and how this relates to **'434** work products. This section will provide more detail about the relationship between the two, as well as the implications with respect to **'434**.

AVCDL is about Process

The ***AVCDL*** is at the core a framework supporting a collection of processes implementing a set of requirements. As noted in the **background material**, it is built assuming the presence of the **'288** and **'207** standards within the organization. It further presumes that a best practices hardware-software development strategy akin the that described in **'262** (V-model). The phases and their requirements are sufficient to cover both hardware and software.

This is not to say that there is no consideration of hardware within the realm of cybersecurity as applied to the product's lifecycle. The cybersecurity requirements have explicit provision for hardware-specific requirements. Refer to the secondary document **Security Requirements Taxonomy** for additional details.

Any hardware-specific requirements are linked to specific product elements during the requirements phase as set out in the secondary document **Product-level Security Requirements**.

ISO 21434 Compliance

The ***AVCDL*** is designed to enable the production of a supporting case for certification under various standards including **'434** and **WP.29 (R155, R156, R157)**. **'434** has no hardware-specific work products or requirements. There is the desire that we be able to show that we apply cybersecurity to both hardware and software. This will be evidenced through application of cybersecurity concepts, goals and requirements using the processes and requirements set out in the ***AVCDL***.

8. Implementation Framework

8.1 Rectilinear Visualization

As indicated in the background section, the structure of the ***AVCDL*** is inspired by multiple standards and best practices documents.

It can be visualized as follows:

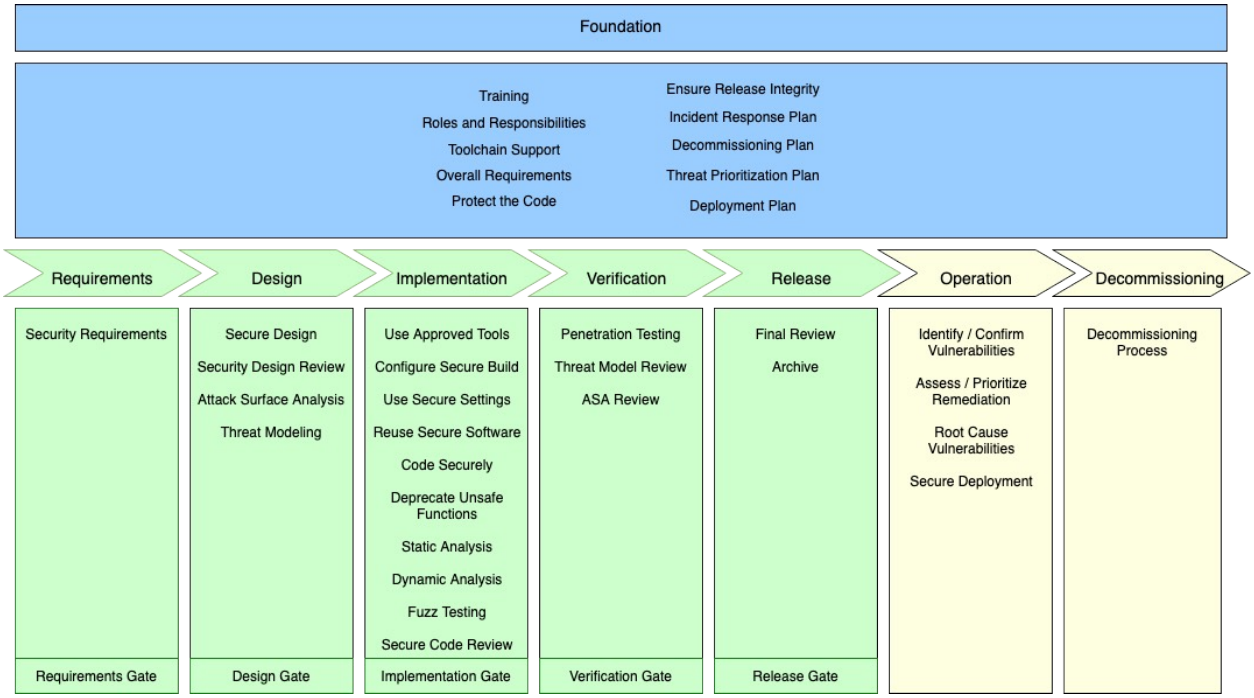

Figure 3 - AVCDL phases and requirements

Note: The rectilinear nature of this diagram is for convenience only and is not intended to imply any particular implementation or the use of any specific development methodology.

8.2 Cyclic Visualization

The ***AVCDL*** can also be visualized as a cyclic system (only phases shown):

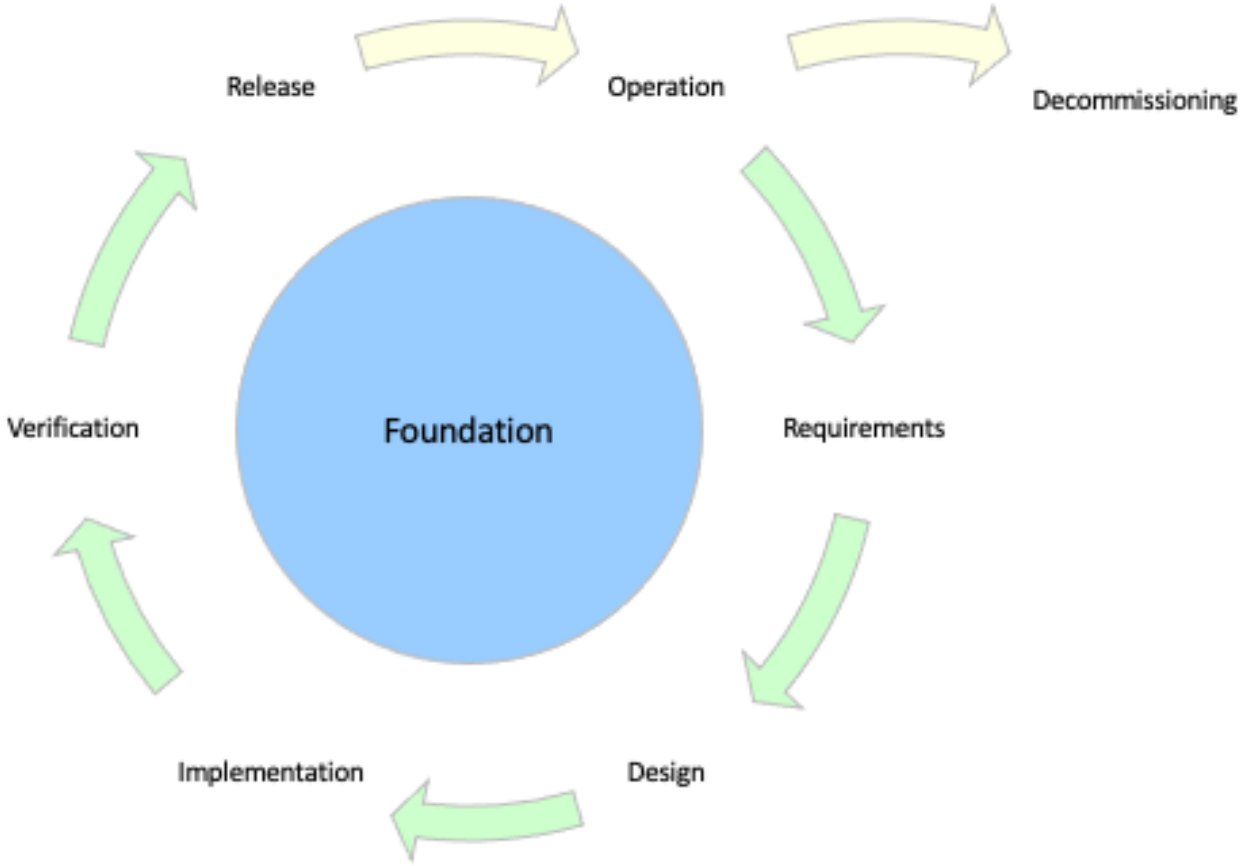

Figure 4 - AVCDL phases cyclic

8.3 Framework Categories

The process elements fall into three broad categories:

- **Foundational** process elements (shown in blue) form a foundation for secure development and take place outside the normal development cycle. These may be done in parallel and are subject to refresh as the security landscape changes. The basis for this phase comes from **MSSDL** and **'434**, and the practices from **SSDF**.

    - Foundation

- **Intra-developmental** process elements (shown in green) serve to augment the existing development processes. The phases come from **MSSDL** and **'434**, and the practices from **SSDF**. To a large extent, the framework on which these phase augmentations hang already exists as part of the typical non-security-aware development process.

    - Requirements
    - Design
    - Implementation
    - Verification
    - Release

    Note: In addition to its **'434**-supporting requirements, each of the intra-developmental phases has an exit gate requirement. There are no **'434** work products in the gate requirements as these are intended to verify the completion of the other requirements within the phase.

- **Post-developmental** phases (shown in yellow) take place once the product has been released.

    - Operation
    - Decommissioning

8.4 Implementation Methodology

It is important to note that the **AVCDL** does not mandate an implementation methodology (waterfall, XP, BDD, TDD, scrum, Kanban, spiral, …). The phases are dictated by the work product dependencies which are more fully explored in the AVCDL Product Dependencies section.

8.4.1 Linear Methodologies

If a linear implementation methodology (waterfall, V-model, …) is employed, the phase diagram (above) can be used directly.

8.4.2 Cyclic Methodologies

If a cyclic implementation methodology (scrum-based agile, spiral, …) is employed, the intra-developmental phases overlay as follows:

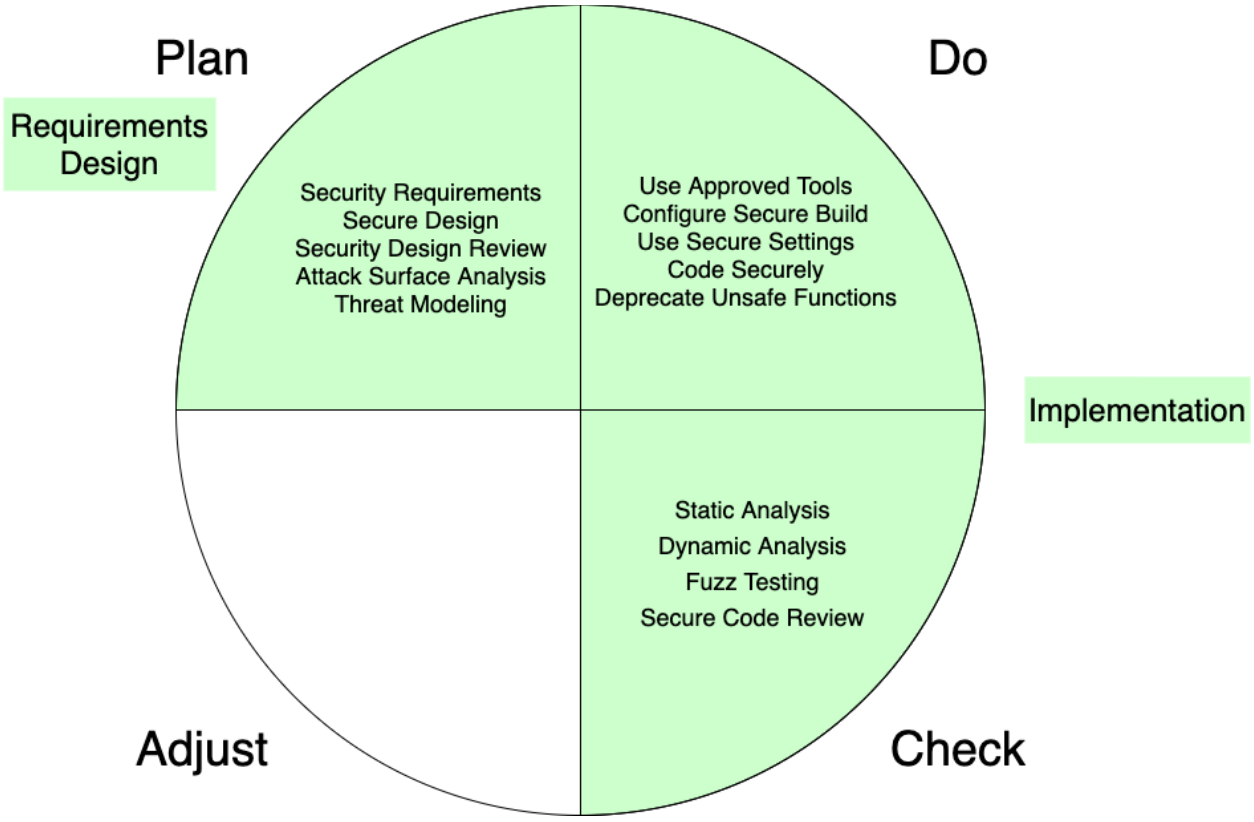

Figure 5 - AVCDL-PDCA Phase Requirement Mapping

Since cyclic implementation methodologies use short activity windows (sprints), there is a need to offset the security activities. The following diagram illustrates how the activity cycle unfolds into linear time.

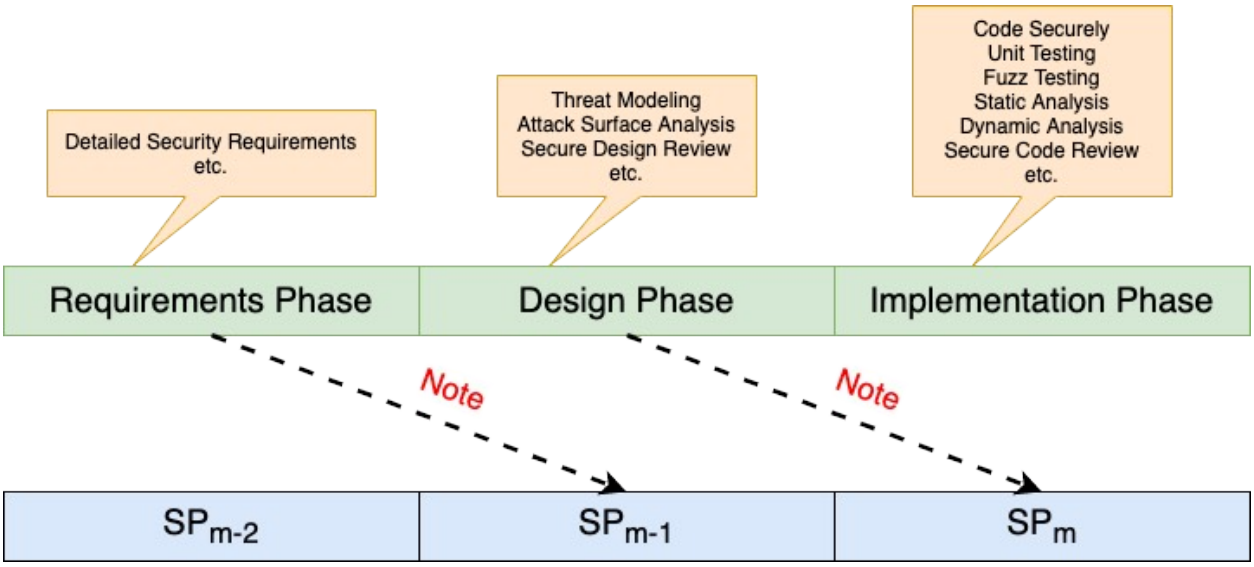

Figure 6 - Sprint-level Alignment

Note: The detailed requirements activity is performed one sprint in advance of design, and design one sprint in advance of implementation. A sprint SP_0 is assumed to bootstrap the process.

9. Process Phases and Requirements

The following is a summary of the AVCDL phases and their associated requirements.

- Foundation
  - Foundation-1 Training
  - Foundation-2 Roles and Responsibilities
  - Foundation-3 Toolchain Support
  - Foundation-4 Definition of Security Requirements
  - Foundation-5 Protect the Code
  - Foundation-6 Ensure Release Integrity
  - Foundation-7 Incident Response Plan
  - Foundation-8 Decommissioning Plan
  - Foundation-9 Threat Prioritization Plan
  - Foundation-10 Deployment Plan

- Requirements
  - Requirements-1 Definition of Security Requirements
  - Requirements-2 Requirements Gate

- Design
  - Design-1 Take Security Requirements and Risk Information into Account During Software Design
  - Design-2 Review the Software Design to Verify Compliance with Security Requirements and Risk Information
  - Design-3 Attack Surface Reduction
  - Design-4 Threat Modeling
  - Design-5 Design Gate

- Implementation
  - Implementation-1 Use Approved Tools
  - Implementation-2 Configure the Compilation and Build Process to Improve Executable Security
  - Implementation-3 Configure the Software to Have Secure Settings by Default
  - Implementation-4 Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality
  - Implementation-5 Create Source Code Adhering to Secure Coding Practice
  - Implementation-6 Deprecate Unsafe Functions
  - Implementation-7 Static Analysis
  - Implementation-8 Dynamic Program Analysis
  - Implementation-9 Security Code Review
  - Implementation-10 Fuzz Testing

  - Implementation-11 Implementation Gate

- Verification
  - Verification-1 Penetration Testing
  - Verification-2 Threat Model Review
  - Verification-3 Attack Surface Analysis Review
  - Verification-4 Verification Gate

- Release
  - Release-1 Final Security Review
  - Release-2 Archive
  - Release-3 Release Gate

- Operation
  - Operation-1 Identify and Confirm Vulnerabilities on an Ongoing Basis
  - Operation-2 Assess and Prioritize the Remediation of all Vulnerabilities
  - Operation-3 Analyze Vulnerabilities to Identify Their Root Causes
  - Operation-4 Secure Deployment

- Decommissioning
  - Decommissioning-1 Decommissioning Protocol

9.1 Foundation Phase

Predecessor: N/A

Successor: Requirements Phase

These process elements form a foundation for secure development and take place outside the normal development cycle. These may be done in parallel and are subject to refresh as the security landscape changes.

[AVCDL-Foundation-1] Training (SSDF PO.1 / MSSDL P1)

This training ensures that the **AVCDL** and its requirements are understood by those interacting with it.

[AVCDL-Foundation-2] Roles and Responsibilities (SSDF PO.2)

It is critical to the success of any **AVCDL**-based project that the roles and responsibilities be defined and assigned prior to the phase to which they apply. These individuals serve as gatekeepers of security issues at the various phase gates.

[AVCDL-Foundation-3] Toolchain Support (SSDF PO.3 / MSSDL P8)

Software supporting secure development must be evaluated, installed, and trained for.

[AVCDL-Foundation-4] Definition of Security Requirements (SSDF PO.4 / MSSDL P3)

Before designing a secure system, it is necessary to have a clear and coherent set of security requirements.

These are the global security requirements as opposed to the more fine-grained requirements called out during the requirements phase.

[AVCDL-Foundation-5] Protect the Code (SSDF PS.1)

The code storage and access should be set up in such a way as to prevent inadvertent or intentional unauthorized changes, inappropriate access, or theft.

[AVCDL-Foundation-6] Ensure Release Integrity (SSDF PS.2)

To some extent, this could be considered part of toolchain support. Operations such as code signing, and root-of-trust fall into this process element.

[AVCDL-Foundation-7] Incident Response Plan (MSSDL P14)

The incident response plan covers the mechanisms needed for dealing with both internal and externally discovered security issues.

[AVCDL-Foundation-8] Decommissioning Plan

A framework must be in place for the eventual removal from service of an in-use system. This should cover the proper handling for any sensitive data embodied in the system.

[AVCDL-Foundation-9] Threat Prioritization Plan (MSSDL P3)

A mechanism for quantifying potential risks and prioritizing their disposition must be established. This may take the form ranging from a gross-level quantization (bug bar) to a formal methodology (TARA).

[AVCDL-Foundation-10] Deployment Plan

A framework must be in place for the loading of software onto the system. This should include both the initial loading / configuration and updating. This should cover the proper handling for any sensitive data to be embodied in the system.

Foundation phase product dependencies are visualized in Figure 8.

ISO 21434 Required Work Products

[WP-05-04] Evidence of tool management
[WP-08-01] Sources for cybersecurity monitoring
[WP-08-02] Triage triggers of cybersecurity information
[WP-08-03] Cybersecurity event triage
[WP-08-04] Cybersecurity event assessment
[WP-08-05] Vulnerability analysis
[WP-08-06] Evidence of managed vulnerabilities
[WP-08-X1] Apply incident response protocols
[WP-10-03] Documentation of the modeling, design, or programming languages and coding guidelines
[WP-10-06] Integration and verification specification
[WP-12-01] Production control plan
[WP-13-01] Cybersecurity incident response plan
[WP-13-X2] Update plan
[WP-14-01] Procedures to communicate end of cybersecurity support
[WP-14-X1] Decommissioning implications

9.1.1 Training [AVCDL-Foundation-1]

Owner

Group: Security

NCWF Role: Cyber Instructor

Administration

security	devops	development	risk
R	I	C	I

There should be a general security awareness training covering the motivation for cybersecurity and its relationship to safety.

There are five distinct areas of training (as specified in MSSDL P1):

- Secure design
- Threat modeling
- Secure coding
- Security testing
- Privacy

Aside from the awareness training, the other training classes have different target audiences. They may be presented concurrently. Ideally, they should be presented prior to the phase to which they apply. There should be an annual limited-scope refresher for each.

The overall training sequence is covered in the **Training Path** section.

There is also the need to track individual and aggregate training participation.

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	List of programming languages / compilers
Risk	**none**

AVCDL Products

- Training Catalog
- System to Track Training Participation

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	AT.2.056, AT.2.057
3	**none**
4	**none**
5	**none**

9.1.2 Roles and Responsibilities [AVCDL-Foundation-2]

Owner

Group: Security

NCWF Role: Systems Requirements Planner

Administration

security	devops	development	risk
R	C	C	-

NIST SP 800-181 [***National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework (NCWF)***] provides an exhaustive breakdown of cybersecurity roles and responsibilities. It provides a common, consistent lexicon that categorizes and describes cybersecurity work. We will draw upon these to establish those needed in support of the ***AVCDL***.

Note: Additional information on ***NCWF*** can be found on their site.

Note: There will be tasks and abilities called out for roles in NCWF which are not leveraged. Additionally, there will be areas where there is not a 1-to-1 mapping.

Note: The role assignments are shown at the top of each requirement page and collected here.

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

none

AVCDL Products

- Roles and Responsibilities Document

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

none

9.1.3 Toolchain Support [AVCDL-Foundation-3]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
C	R	C	-

All software used in the product's development (tools and code [as source or binaries]) must be vetted by development, cybersecurity, and legal to ensure that it is appropriate for use in the development of safety-related systems. A catalog of this software should be created for use by later processes.

The following additional software needs to be in place to support secure development:

- threat modeling
- attack surface analysis
- compile-time security analysis
- static security analysis (including MISRA and SEI CERT)
- open source and third-party software tracking
- security incident tracking

Note: Training for each must be provided prior to time of use.

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

Group	Inputs
Devops	Component tracking system
Development	List of development tools
Risk	**none**

AVCDL Products

- List of Approved Tools and Components

ISO 21434 Required Work Products

[WP-05-04] Evidence of tool management

[WP-10-03] Documentation of the modeling, design, or programming languages and coding guidelines

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	CM.2.061, CM.2.064, CM.2.065, CM.2.066, MA.2.112
3	AU.3.049, AU.3.052, CM.3.068, MA.3.116, CA.3.162
4	CM.4.073
5	AU.5.055, CM.5.074, RM.5.152

9.1.4 Definition of Security Requirements [AVCDL-Foundation-4]

Owner

Group: Security

NCWF Role: Systems Requirements Planner

Administration

security	devops	development	risk
R	I	I	-

Before designing a secure system, it is necessary to have a clear and coherent set of security requirements.

These are the global security requirements as opposed to the more fine-grained requirements called out during the requirements phase.

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

none

AVCDL Products

- Global Security Goals
- Global Security Requirements

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

[7.2.2.2(g)] The processes used to monitor for, detect and respond to cyber-attacks, cyber threats and vulnerabilities on vehicle types and 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.

CMMC Applicable Practices

Level	Practice
1	**none**
2	CA.2.157
3	**none**
4	**none**
5	**none**

9.1.5 Protect the Code [AVCDL-Foundation-5]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
C	R	-	-

Processes and controls need to be in place to accomplish the following goals:

- secure code storage
- IP / open source / third party material segregation
- Deterministic builds
- Disaster recovery
- Security audit

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

Group	Inputs
Devops	Secure IT infrastructure
Development	**none**
Risk	**none**

AVCDL Products

- Code Protection Plan

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	AC.1.001, AC.1.002, AC.1.003. AC.1.004, IA.1.076, IA.1.077, PE.1.131, PE.1.133, PE.1.134
2	AC.2.007, AC.2.008, AC.2.009, AC.2.010, IA.2.081, IA.2.082
3	AC.3.017, AC.3.018, AC.3.019, AC.3.014, AM.3.036, IA.3.083, IA.3.084, IS.3.086
4	AM.4.226
5	**none**

9.1.6 Ensure Release Integrity [AVCDL-Foundation-6]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
C	R	C	-

Steps need to be in place to accomplish the following tasks:

- code signing
- hash tracking
- credential management
- root-of-trust
- secure deployment support

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

Group	Inputs
Devops	• Code signing • Credential management • Deployment infrastructure
Development	**none**
Risk	**none**

AVCDL Products

- Release Integrity Plan

ISO 21434 Required Work Products

[WP-12-01] Production control plan

WP.29 CSMS Requirements

[7.2.2.2(e)] The processes used for testing the cyber security of a vehicle type.

CMMC Applicable Practices

Level	Practice
1	AC.1.001, AC.1.002, AC.1.003. AC.1.004, PE.1.131, PE.1.133, PE.1.134
2	AC.2.007, AC.2.008, AC.2.009, AC.2.010
3	AC.3.017, AC.3.018, AC.3.019, AC.3.014, AM.3.036
4	AM.4.226
5	**none**

9.1.7 Incident Response Plan [AVCDL-Foundation-7]

Owner

Group: Security

NCWF Role: Partner Integration Planner

Administration

security	devops	development	risk
R	-	C	-

Monitoring sources may include:

- external sources
  - government sources
  - commercial or non-commercial sources
  - researchers
  - organization's supply chain
  - organization's customers

- internal sources
  - vulnerability analysis results
  - information from the field (vulnerability scanning reports, repair information, consumer usage information)

The incident response plan should include:

- An identified sustained engineering (SE)
- On-call contacts with decision-making authority 24/7/365
- Security servicing plans for both issues related to internally and externally supplied software

---

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Triage information required
Risk	**none**

AVCDL Products

- Cybersecurity Monitoring Plan
- Incident Response Plan

ISO 21434 Required Work Products

[WP-07-01] Sources for cybersecurity monitoring
[WP-07-02] Triage triggers of cybersecurity information
[WP-08-03] Cybersecurity event triage
[WP-08-04] Cybersecurity event assessment
[WP-08-05] Vulnerability analysis
[WP-08-06] Evidence of managed vulnerabilities
[WP-08-X1] Apply incident response protocols
[WP-13-01] Cybersecurity incident response plan

WP.29 CSMS Requirements

[7.2.2.2(g)] The processes used to monitor for, detect and respond to cyber-attacks, cyber threats and vulnerabilities on vehicle types and 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.

CMMC Applicable Practices

Level	Practice
1	**none**
2	IR.2.092
3	**none**
4	IR.4.100
5	IR.5.106

9.1.8 Decommissioning Plan [AVCDL-Foundation-8]

Owner

Group: Security

NCWF Role: Partner Integration Planner

Administration

security	devops	development	risk
R	C	C	-

The decommissioning plan covers the proper handling for any sensitive data embodied in the system. This includes:

- credentials
- certificates
- PII
- logs

It is critical that the decommissioning plan include partner elements within the system.

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

Group	Inputs
Devops	Decommissioning / RMA process
Development	List of data stored on systems
Risk	**none**

AVCDL Products

- Decommissioning Plan

ISO 21434 Required Work Products

[WP-14-01] Procedures to communicate end of cybersecurity support
[WP-14-X1] Decommissioning requirements

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AM.3.036
4	AM.4.226
5	**none**

9.1.9 Threat Prioritization Plan [AVCDL-Foundation-9]

Owner

Group: Security

NCWF Role: Systems Requirements Planner

Administration

security	devops	development	risk
R	-	I	I

A mechanism for quantifying potential risks and prioritizing their disposition must be established. This may take the form from a gross-level quantization (bug bar) to a formal methodology (TARA).

Since an autonomous vehicle is a safety-critical system, a formal threat quantification system is warranted.

This plan will be applied to take a threat potential (output of the threat modeling process) and yield a prioritized set of threat issues which must be addressed in order to ensure the safe operation of the vehicle.

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

none

AVCDL Products

- Threat Prioritization Plan

ISO 21434 Required Work Products

[WP-08-06] Evidence of managed vulnerabilities

WP.29 CSMS Requirements

[7.2.2.2(f)] The processes used for ensuring that the risk assessment is kept current.

CMMC Applicable Practices

Level	Practice
1	**none**
2	CA.2.159
3	RM.3.146
4	**none**
5	**none**

9.1.10 Deployment Plan [AVCDL-Foundation-10]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
C	R	C	-

The deployment framework must consider the following:

- secure software deployment
- initial and update scenarios
- deployment failure handling

The deployment plan covers the proper handling for any sensitive data embodied in the system. This includes:

- credentials
- certificates
- PII
- logs

Training Provided

none

Phase Requirement Dependencies

none

External Group Product Dependencies

Group	Inputs
Devops	Deployment infrastructure / process
Development	List of material to be deployed
Risk	**none**

AVCDL Products

- Deployment Plan

ISO 21434 Required Work Products

[WP-12-01] Production control plan
[WP-13-X2] Update plan

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AM.3.036
4	AM.4.226
5	**none**

9.2 Requirements Phase

Predecessor: Foundation Phase or Operation Phase

Successor: Design Phase

The requirements phase of development is a reiteration of **AVCDL-Foundation-4** **Definition of Security Requirements** but with higher resolution. In an Agile-based development process, this is to be expected.

[AVCDL-Requirements-1] Security Requirements Definition (SSDF PO.4 / MSSDL P2)

The requirements are created with consideration of the global security requirements. They provide constraints specific to the work under consideration.

[AVCDL-Requirements-2] Requirements Gate (MSSDL P3)

Requirements phase exit is conditional (formally gated) on completion of all AVCDL phase requirements and work products for this phase.

Requirements phase product dependencies are visualized in Figure 9.

ISO 21434 Required Work Products

[WP-09-01] Item definition

[WP-10-02] Cybersecurity requirements for post-development

9.2.1 Security Requirements Definition [AVCDL-Requirements-1]

Owner

Group: Security

NCWF Role: Security Architect

Administration

security	devops	development	risk
R	-	I	-

Requirements need to both consider the global security requirements and add constraints necessary to the specifics of the work under consideration. As with the global-level requirements called out in AVCDL-Foundation-4, these requirements should be derived using the **security requirements taxonomy** in order to expose gaps up-front (prior to threat modeling, attack surface analysis, ...).

Requirements should be traceable through the product operation phase to allow for improvement should deficiencies be discovered.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Foundation-4] Definition of Security Requirements

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	High-level design
Risk	**none**

AVCDL Products

- Product-level Security Goals
- Product-level Security Requirements

ISO 21434 Required Work Products

[WP-09-01] Item definition
[WP-10-02] Cybersecurity requirements for post-development

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	CA.2.157
3	SC.3.177, SC.3.181, SC.3.183, SC.3.185, SC.3.186, SC.3.187, SC.3.190, SC.3.191
4	**none**
5	**none**

9.2.2 Requirements Gate [AVCDL-Requirements-2]

Owner

Group: Security

NCWF Role: Secure Software Assessor

Administration

security	devops	development	risk
R	-	R	-

Requirements phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase. The security advisor assigned to the release must certify that the project team has satisfied security requirements for this phase.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Requirements-1] Security Requirements Definition

External Group Product Dependencies

none

AVCDL Products

- Requirements Phase Gate

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055

9.3 Design Phase

Predecessor: Requirements Phase

Successor: Implementation Phase

The changes to the design phase include the incorporation of security requirements and analysis of the design from a security perspective.

[AVCDL-Design-1] Apply Security Requirements and Risk Information to Design (SSDF PW.1 / MSSDL P4, MSSDL P5)

The design should take into consideration established security requirements and risk information.

[AVCDL-Design-2] Security Design Review (SSDF PW.2)

Help ensure the software will meet the security requirements and satisfactorily address the identified risk information.

[AVCDL-Design-3] Attack Surface Reduction (MSSDL P6)

Attack surface analysis guides the disabling or access restricting of system services. It applies the principles of least privilege and layered defense.

[AVCDL-Design-4] Threat Modeling (MSSDL P7)

Threat modeling realizes an abstraction of the system as a set of interacting processes managing resources passing data between them. It is on these data flows that automated threat modeling tools reason.

[AVCDL-Design-5] Design Gate (MSSDL P3)

Design phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase.

Design phase product dependencies are visualized in Figure 10.

ISO 21434 Required Work Products

[WP-06-03] Cybersecurity assessment report
[WP-09-02] Threat analysis and risk assessment
[WP-09-03] Cybersecurity goals
[WP-09-04] Cybersecurity claims
[WP-09-05] Verification report
[WP-09-06] Cybersecurity concept
[WP-09-07] Verification report of cybersecurity concept
[WP-15-01] Damage scenarios
[WP-15-02] Identified assets and cybersecurity properties
[WP-15-03] Threat scenarios
[WP-15-04] Impact rating
[WP-15-05] Attack paths
[WP-15-06] Attack feasibility rating
[WP-15-07] Risk value
[WP-15-08] Risk treatment decision per threat scenario

9.3.1 Apply Security Requirements and Risk Information to Design [AVCDL-Design-1]

Owner

Group: Development

NCWF Role: Software Developer

Administration

security	devops	development	risk
R	-	R	-

Determine which security requirements the software's design should meet and determine what security risks the software is likely to face during production operation and how those risks should be mitigated by the software's design. Addressing security requirements and risks during software design instead of later helps to make software development more efficient.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Requirements-2] Requirements Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Detailed functional requirements
Risk	**none**

AVCDL Products

- Design Showing Security Considerations

ISO 21434 Required Work Products

[WP-09-06] Cybersecurity concept
[WP-10-01] Refined cybersecurity specification
[WP-15-02] Identified assets and cybersecurity properties

WP.29 CSMS Requirements

[7.2.2.2(b)] The processes used for the identification of risks to vehicle types. Within these processes, the threats in Annex 5, Part A, and other relevant threats shall be considered.

[7.2.2.2(e)] The processes used for testing the cyber security of a vehicle type.

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052, SC.3.180
4	**none**
5	AU.5.055

9.3.2 Security Design Review [AVCDL-Design-2]

Owner

Group: Security

NCWF Role: Systems Requirements Planner

Administration

security	devops	development	risk
R	-	R	C

Help ensure the software will meet the security requirements and satisfactorily address the identified risk information.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-1] Apply Security Requirements and Risk Information to Design

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Element detailed design
Risk	**none**

AVCDL Products

- Security Design Review Report

ISO 21434 Required Work Products

[WP-06-03] Cybersecurity assessment report
[WP-09-07] Verification report of cybersecurity concept

WP.29 CSMS Requirements

[7.2.2.2(d)] The processes in place to verify that the risks identified are appropriately managed.
[7.2.2.2(e)] The processes used for testing the cyber security of a vehicle type.

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055, SC.5.230

9.3.3 Attack Surface Reduction [AVCDL-Design-3]

Owner

Group: Security

NCWF Role: Security Architect

Administration

security	devops	development	risk
R	-	R	-

Attack surface reduction encompasses shutting off or restricting access to system services, applying the principle of least privilege, and employing layered defenses wherever possible. It is primarily used when dealing with externally supplied elements where access to the design is not provided.

Note: Attack surface analysis is a methodology which trails in maturity when compared with threat modeling. Automated measures in this area will be limited.

Note: The attack surface analysis **AVCDL** work products are generated through application of the threat prioritization plan set out in [AVCDL-Foundation-9] **Threat Prioritization Plan**.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-1] Apply Security Requirements and Risk Information to Design

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Functional OS interface design
Risk	**none**

AVCDL Products

- Attack Surface Analysis Report
- Ranked / Risked Threat Report
- Threat Report

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	RM.2.143
3	AU.3.049, AU.3.052, RM.3.144
4	**none**
5	AU.5.055

9.3.4 Threat Modeling [AVCDL-Design-4]

Owner

Group: Security

NCWF Role: Security Architect

Administration

security	devops	development	risk
R	-	R	R

Threat modeling is an exercise which may be done at any stage of development. It realizes an abstraction of the system as a set of interacting processes managing resources passing data between them. It is on these data flows that automated threat modeling tools reason.

In that same way that security requirements should be considered at multiple levels in order to provide a complete landscape, so to do threat models.

Note: Threat modeling is a team exercise, encompassing program/project managers, developers, and testers, and represents the primary security analysis task performed during the software design stage.

Note: The threat modeling **AVCDL** work products are generated through application of the threat prioritization plan set out in [AVCDL-Foundation-9] **Threat Prioritization Plan**.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Foundation-9] Threat Prioritization Plan
[AVCDL-Design-1] Apply Security Requirements and Risk Information to Design

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Element detailed design
Risk	**none**

AVCDL Products

- Threat Modeling Report
- Ranked / Risked Threat Report
- Threat Report

ISO 21434 Required Work Products

[WP-09-02] Threat analysis and risk assessment
[WP-09-03] Cybersecurity goals
[WP-09-04] Cybersecurity claims
[WP-09-05] Verification report
[WP-15-01] Damage scenarios
[WP-15-03] Threat scenarios
[WP-15-04] Impact rating
[WP-15-05] Attack paths
[WP-15-06] Attack feasibility rating
[WP-15-07] Risk values
[WP-15-08] Risk treatment decision per threat scenario

WP.29 CSMS Requirements

[7.2.2.2(b)] The processes used for the identification of risks to vehicle types. Within these processes, the threats in Annex 5, Part A, and other relevant threats shall be considered.

[7.2.2.2(c)] The processes used for the assessment, categorization and treatment of the risks identified.

CMMC Applicable Practices

Level	Practice
1	**none**
2	RM.2.143
3	AU.3.049, AU.3.052, RM.3.144
4	**none**
5	AU.5.055

9.3.5 Design Gate [AVCDL-Design-5]

Owner

Group: Security

NCWF Role: Secure Software Assessor

Administration

security	devops	development	risk
R	-	R	R

Design phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase. The security advisor assigned to the release must certify that the project team has satisfied security requirements for this phase.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Design-2]	Security Design Review
[AVCDL-Design-3]	Attack Surface Reduction
[AVCDL-Deisgn-4]	Threat Modeling

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	**none**
Risk	**none**

AVCDL Products

- Design Phase Gate

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055

9.4 Implementation Phase

Predecessor: Design Phase

Successor: Verification Phase

The implementation phase of development is based on **MSSDL Implementation Phase** (MSSDL P8-10) and **SSDF Prepare Well-Secured Software** (PW.4-7, 9).

[AVCDL-Implementation-1] Use Approved Tools (MSSDL P8)

Development teams should strive to use the latest version of approved tools to take advantage of new security analysis functionality and protections.

[AVCDL-Implementation-2] Configure Build Process to Improve Security (SSDF PW.6)

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities before testing occurs.

[AVCDL-Implementation-3] Use Secure Settings by Default (SSDF PW.9)

Help improve the security of the software at installation time, which reduces the likelihood of the software being deployed with weak security settings that would put it at greater risk of compromise.

[AVCDL-Implementation-4] Reuse Well-Secured Software (SSDF PW.4)

Reuse of well-secured (verified) software lowers the costs of development, expedites development, and decreases the likelihood of introducing additional security vulnerabilities.

[AVCDL-Implementation-5] Code Securely (SSDF PW.5)

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities during source code creation.

[AVCDL-Implementation-6] Deprecate Unsafe Functions (MSSDL P9)

Project teams should analyze all functions and APIs that will be used in conjunction with a software development project and prohibit those that are determined to be unsafe.

[AVCDL-Implementation-7] Static Analysis (MSSDL P10)

Project teams should perform static analysis of source code.

[AVCDL-Implementation-8] Dynamic Program Analysis (MSSDL P11)

Run-time verification of software programs is necessary to ensure that a program's functionality works as designed.

[AVCDL-Implementation-9] Security Code Review (SSDF PW.7)

The security team and security advisors should augment static analysis with other automated or human review as appropriate.

[AVCDL-Implementation-10] Fuzz Testing (MSSDL P12)

Fuzz testing is a specialized form of dynamic analysis used to induce program failure by deliberately introducing malformed or random data to an application.

[AVCDL-Implementation-11] Implementation Gate (MSSDL P3)

Implementation phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase.

Implementation phase product dependencies are visualized in Figure 11.

ISO 21434 Required Work Products

[WP-10-07] Integration and verification reports

9.4.1 Use Approved Tools [AVCDL-Implementation-1]

Owner

Group: Development

NCWF Role: Software Developer

Administration

security	devops	development	risk
C	C	R	-

Development teams should strive to use the latest version of **approved** tools and components to take advantage of new security analysis functionality and protections. Unapproved tools and components should never be used. The build system should verify that the tools and components currently being used have been approved for use in the creation of this product.

Note: The list of approved tools and components, and their associated security checks, such as compiler and linker options, and warnings were created in the foundation phase.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Foundation-3] Toolchain Support
[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	Component tracking comparison system
Development	**none**
Risk	**none**

AVCDL Products

- List of Tools and Components Used

ISO 21434 Required Work Products

[WP-05-05] Evidence of tool management

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055

9.4.2 Configure Build Process to Improve Security [AVCDL-Implementation-2]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
C	R	C	-

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities before testing occurs.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	Build system
Development	List of adopted secure build settings
Risk	**none**

AVCDL Products

- Build Process Documentation

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	CM.2.061, CM.2.064, CM.2.065, CM.2.066
3	CM.3.068
4	CM.4.073
5	CM.5.074

9.4.3 Use Secure Settings by Default [AVCDL-Implementation-3]

Owner

Group: Security

NCWF Role: Security Architect

Administration

security	devops	development	risk
R	-	R	-

Helps improve the security of the software at installation time, which reduces the likelihood of the software being deployed with weak security settings that would put it at greater risk of compromise.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Element detailed design
Risk	**none**

AVCDL Products

- Secure Settings Document

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	CM.2.061, CM.2.064, CM.2.065, CM.2.066
3	CM.3.068
4	CM.4.073
5	CM.5.074

9.4.4 Reuse Well-Secured Software [AVCDL-Implementation-4]

Owner

Group: Development

NCWF Role: Software Developer

Administration

security	devops	development	risk
C	I	R	-

Lower the costs of software development, expedite software development, and decrease the likelihood of introducing additional security vulnerabilities into the software. These are particularly true for software that implements security functionality, such as cryptographic modules and protocols.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	List of libraries used
Risk	**none**

AVCDL Products

- Component / Version - Product / Version Cross-reference Document

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

none

9.4.5 Code Securely [AVCDL-Implementation-5]

Owner

Group: Development

NCWF Role: Software Developer

Administration

security	devops	development	risk
C	-	R	-

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities during source code creation.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Element implementation
Risk	**none**

AVCDL Products

- Secure Development

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	SC.2.179
3	MP.3.125, SC.3.177, SC.3.181, SC.3.183, SC.3.185, SC.3.186, SC.3.187, SC.3.190, SC.3.191
4	SC.4.197
5	**none**

9.4.6 Deprecate Unsafe Functions [AVCDL-Implementation-6]

Owner

Group: Development

NCWF Role: Software Developer

Administration

security	devops	development	risk
C	-	R	-

Many commonly used functions and APIs are not secure in the face of the current threat environment. Project teams should analyze all functions and APIs that will be used in conjunction with a software development project and prohibit those that are determined to be unsafe.

Note: The list of unsafe functions should have been created in the foundation phase.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	List of deprecated functions in use
Risk	**none**

AVCDL Products

- Currently Used Deprecated Functions Document

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

none

9.4.7 Static Analysis [AVCDL-Implementation-7]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
C	R	C	-

Project teams should perform static analysis of source code.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	• Static analysis infrastructure • Static analysis settings tracking
Development	List of adopted security-related settings
Risk	**none**

AVCDL Products

- Static Analysis Report

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

none

9.4.8 Dynamic Program Analysis [AVCDL-Implementation-8]

Owner

Group: Development

NCWF Role: Software Developer

Administration

security	devops	development	risk
C	-	R	-

Run-time verification of software programs is necessary to ensure that a program's functionality works as designed. This verification task should specify tools that monitor application behavior for memory corruption, user privilege issues, and other critical security problems.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	Dynamic analysis testing infrastructure
Development	List of adopted security-related tools
Risk	**none**

AVCDL Products

- Dynamic Analysis Report

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	**none**
4	**none**
5	SI.5.222

9.4.9 Security Code Review [AVCDL-Implementation-9]

Owner

Group: Security

NCWF Role: Secure Software Assessor

Administration

security	devops	development	risk
R	-	C	-

Static code analysis by itself is generally insufficient to replace a manual code review. The security team and security advisors should be aware of the strengths and weaknesses of static analysis tools and be prepared to augment static analysis tools with other tools or human review as appropriate.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	Code review infrastructure
Development	Element implementation
Risk	**none**

AVCDL Products

- Secure Code Review Summary

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052, CA.3.162
4	**none**
5	AU.5.055

9.4.10 Fuzz Testing [AVCDL-Implementation-10]

Owner

Group: Security

NCWF Role: Vulnerability Assessment Analyst

Administration

security	devops	development	risk
R	C	C	-

Fuzz testing is a specialized form of dynamic analysis used to induce program failure by deliberately introducing malformed or random data to an application. The fuzz testing strategy is derived from the intended use of the application and the functional and design specifications for the application.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Design-5] Design Gate

External Group Product Dependencies

Group	Inputs
Devops	Fuzz testing process infrastructure
Development	Element implementation
Risk	**none**

AVCDL Products

- Fuzz Testing Report

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

[7.2.2.2(e)] The processes used for testing the cyber security of a vehicle type.
[7.2.2.2(f)] The processes used for ensuring that the risk assessment is kept current.
[7.2.2.2(g)] The processes used to monitor for, detect and respond to cyber-attacks, cyber threats and vulnerabilities on vehicle types and 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.

CMMC Applicable Practices

none

9.4.11 Implementation Gate [AVCDL-Implementation-11]

Owner

Group: Security

NCWF Role: Secure Software Assessor

Administration

security	devops	development	risk
R	R	R	-

Implementation phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase. The security advisor assigned to the release must certify that the project team has satisfied security requirements for this phase.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Implementation-1] Use Approved Tools
[AVCDL-Implementation-2] Configure Build Process to Improve Security
[AVCDL-Implementation-3] Use Secure Settings by Default
[AVCDL-Implementation-4] Reuse Well-Secured Software
[AVCDL-Implementation-5] Code Securely
[AVCDL-Implementation-6] Deprecate Unsafe Functions
[AVCDL-Implementation-7] Static Analysis
[AVCDL-Implementation-8] Dynamic Program Analysis
[AVCDL-Implementation-9] Security Code Review
[AVCDL-Implementation-10] Fuzz Testing

External Group Product Dependencies

none

AVCDL Products

- Implementation Phase Gate

ISO 21434 Required Work Products

[WP-10-07] Integration and verification reports

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055

9.5 Verification Phase

Predecessor: Implementation Phase

Successor: Release Phase

The verification phase of development is based on **MSSDL Verification Phase** (MSSDL P11-3) and **SSDF Produce Well-Secured Software** (SSDF PW.8).

[AVCDL-Verification-1] Penetration Testing (SSDF PW.8)

Penetration testing identifies vulnerabilities before software is released so they can be corrected before release, which prevents exploitation.

[AVCDL-Verification-2] Threat Model Review (MSSDL P13)

The threat models should be reviewed to ensures that any design or implementation changes to the system have been accounted for, and that any new attack vectors created as a result of the changes have been reviewed and mitigated.

[AVCDL-Verification-3] Attack Surface Analysis Review (MSSDL P13)

The attack surface analysis should be reviewed to ensures that any design or implementation changes to the system have been accounted for, and that any new attack vectors created as a result of the changes have been reviewed and mitigated.

[AVCDL-Verification-4] Verification Gate (MSSDL P3)

Verification phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase.

Verification phase product dependencies are visualized in Figure 12.

ISO 21434 Required Work Products

[WP-10-04] Verification report for the refined cybersecurity specification
[WP-10-05] Vulnerability analysis report
[WP-10-07] Integration and verification reports
[WP-11-01] Validation report

9.5.1 Penetration Testing [AVCDL-Verification-1]

Owner

Group: Security

NCWF Role: Vulnerability Assessment Analyst

Administration

security	devops	development	risk
R	C	C	-

Help identify vulnerabilities before software is released so they can be corrected before release, which prevents exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities. Executable code is binaries, directly executed bytecode, directly executed source code, and any other form of code an organization deems as executable.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Implementation-11] Implementation Gate

External Group Product Dependencies

Group	Inputs
Devops	Penetration testing process infrastructure
Development	Operational system
Risk	**none**

AVCDL Products

- Penetration Testing Report
- Ranked / Risked Threat Report
- Threat Report

ISO 21434 Required Work Products

[WP-10-06] Integration and Verification Specification

WP.29 CSMS Requirements

[7.2.2.2(e)] The processes used for testing the cyber security of a vehicle type.

[7.2.2.2(f)] The processes used for ensuring that the risk assessment is kept current.

[7.2.2.2(g)] The processes used to monitor for, detect and respond to cyber-attacks, cyber threats and vulnerabilities on vehicle types and 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.

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	**none**
4	CA.4.164, CA.4.227
5	**none**

9.5.2 Threat Model Review [AVCDL-Verification-2]

Owner

Group: Security

NCWF Role: Security Architect

Administration

security	devops	development	risk
R	-	R	R

The threat models should be reviewed to ensure that any design or implementation changes to the system have been accounted for, and that any new attack vectors created as a result of the changes have been reviewed and mitigated.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Design-4] Threat Modeling
[AVCDL-Implementation-11] Implementation Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Updated element detailed design
Risk	**none**

AVCDL Products

- Updated Threat Model

ISO 21434 Required Work Products

[WP-10-04] Verification report for the refined cybersecurity specification
[WP-10-05] Vulnerability analysis report
[WP-10-06] Integration and Verification Specification

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	RM.2.143
3	RM.3.144
4	**none**
5	**none**

9.5.3 Attack Surface Analysis Review [AVCDL-Verification-3]

Owner

Group: Security

NCWF Role: Security Architect

Administration

security	devops	development	risk
R	-	R	-

The attack surface analysis should be reviewed to ensure that any design or implementation changes to the system have been accounted for, and that any new attack vectors created as a result of the changes have been reviewed and mitigated.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Design-3] Attack Surface Reduction
[AVCDL-Implementation-11] Implementation Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Updated functional OS interface design
Risk	**none**

AVCDL Products

- Updated Attack Surface Analysis

ISO 21434 Required Work Products

[WP-10-04] Verification report for the refined cybersecurity specification
[WP-10-05] Vulnerability analysis report
[WP-10-06] Integration and Verification Specification

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	RM.2.143
3	RM.3.144
4	**none**
5	**none**

9.5.4 Verification Gate [AVCDL-Verification-4]

Owner

Group: Security

NCWF Role: Secure Software Assessor

Administration

security	devops	development	risk
R	-	R	R

Verification phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase. The security advisor assigned to the release must certify that the project team has satisfied security requirements for this phase.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Verification-1] Penetration Testing
[AVCDL-Verification-2] Threat Model Review
[AVCDL-Verification-3] Attack Surface Analysis Review

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Updated element detailed design
Risk	**none**

AVCDL Products

- Verification Phase Gate

ISO 21434 Required Work Products

[WP-10-07] Integration and verification reports
[WP-11-01] Validation report

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055

9.6 Release Phase

Predecessor: Verification Phase

Successor: Operation Phase

The release phase of development is based on **MSSDL Release Phase** (MSSDL P14-6).

[AVCDL-Release-1] Final Security Review (MSSDL P15)

The Final Security Review (FSR) is a deliberate examination of all the security activities performed on a software application prior to release.

[AVCDL-Release-2] Archive (MSSDL P16)

All pertinent information and data must be archived to allow for post-release servicing of the software.

[AVCDL-Release-3] Release Gate (MSSDL P16)

Release phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase.

Release phase product dependencies are visualized in Figure 13.

ISO 21434 Required Work Products

[WP-06-04] Release for post-development report

9.6.1 Final Security Review [AVCDL-Release-1]

Owner

Group: Security

NCWF Role: Secure Software Assessor

Administration

security	devops	development	risk
R	C	C	C

This is a deliberate examination of all the security activities performed on a software application prior to release. The FSR is performed by the security advisor with assistance from the regular development staff and the security and privacy team leads. The FSR is not a "penetrate and patch" exercise, nor is it a chance to perform security activities that were previously ignored or forgotten.

The FSR usually includes an examination of:

- threat models
- exception requests
- tool output
- performance reports

These are compared against the previously determined quality gates or bug bars. Regressions discovered at this stage indicate a failure in the verification phase.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Verification-4] Verification Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Final design documentation
Risk	**none**

AVCDL Products

- Final Security Review Report

ISO 21434 Required Work Products

[WP-06-04] Release for post-development report

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	RM.2.143
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055

9.6.2 Archive [AVCDL-Release-2]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
-	R	C	-

Everything necessary to reproduce and maintain the product must be archived.

This includes:

- specifications
- source code
- binaries
- private symbols
- threat models
- documentation
- emergency response plans
- license and servicing terms for any third-party software
- other data necessary to perform post-release servicing tasks

---

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Release-1] Final Security Review

External Group Product Dependencies

Group	Inputs
Devops	• Artifact storage infrastructure • Artifact tracking system
Development	Final materials for deployment
Risk	**none**

AVCDL Products

- Archive Manifest

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	PE.1.131, PE.1.133, PE.1.134
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055, RE.5.140

9.6.3 Release Gate [AVCDL-Release-3]

Owner

Group: Security

NCWF Role: Secure Software Assessor

Administration

security	devops	development	risk
R	R	R	R

Release phase exit is conditional (formally gated) on completion of all **AVCDL** phase requirements and work products for this phase. The security advisor assigned to the release must certify that the project team has satisfied security requirements.

Training Provided

none

Phase Requirement Dependencies

[AVCDL-Release-2] Archive

External Group Product Dependencies

none

AVCDL Products

- Release Phase Gate

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	**none**
2	**none**
3	AU.3.049, AU.3.052
4	**none**
5	AU.5.055

9.7 Operation Phase

Predecessor: Release Phase

Successor: Requirements Phase or Decommissioning Phase

The operation phase is based on **SSDF Vulnerability Report Practices** (SSDF RV).

[AVCDL-Operation-1] Identify and Confirm Vulnerabilities (SSDF RV.1)

Help ensure vulnerabilities are identified more quickly so they can be remediated more quickly, reducing the window of opportunity for attackers.

[AVCDL-Operation-2] Assess and Prioritize the Remediation (SSDF RV.2)

Help ensure vulnerabilities are remediated as quickly as necessary, reducing the window of opportunity for attackers.

[AVCDL-Operation-3] Root Cause Vulnerabilities (SSDF RV.3)

Help reduce the frequency of vulnerabilities in the future.

[AVCDL-Operation-4] Secure Deployment

Software must be deployed in a secure manner.

Operation phase product dependencies are visualized in Figure 14.

ISO 21434 Required Work Products

[WP-08-03] Cybersecurity event triage
[WP-08-04] Cybersecurity event assessment
[WP-08-05] Vulnerability analysis
[WP-08-06] Evidence of managed vulnerability
[WP-08-X1] Apply incident response protocols
[WP-12-X1] Production control plan implementation
[WP-13-X1] Cybersecurity incident response plan implementation

9.7.1 Identify and Confirm Vulnerabilities [AVCDL-Operation-1]

Owner

Group: Security

NCWF Role: Cyber Defense Incident Responder

Administration

security	devops	development	risk
R	-	C	-

Help ensure vulnerabilities are identified more quickly so they can be remediated more quickly, reducing the window of opportunity for attackers.

Note: The incident response report **AVCDL** work product is generated through application of the threat prioritization plan set out in [AVCDL-Foundation-9] **Threat Prioritization Plan**.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Foundation-7] Incident Response Plan
[AVCDL-Release-3] Release Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Element detailed design
Risk	**none**

AVCDL Products

- Cybersecurity Incident Report

ISO 21434 Required Work Products

[WP-08-03] Cybersecurity event triage
[WP-08-04] Cybersecurity event assessment
[WP-08-06] Evidence of managed vulnerabilities
[WP-08-X1] Apply incident response protocols
[WP-13-X1] Cybersecurity incident response plan implementation

WP.29 CSMS Requirements

[7.2.2.2(g)] The processes used to monitor for, detect and respond to cyber-attacks, cyber threats and vulnerabilities on vehicle types and 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.

[7.2.2.2(h)] The processes used to provide relevant data to support analysis of attempted or successful cyber-attacks.

CMMC Applicable Practices

Level	Practice
1	SI.1.210
2	IR.2.093, IR.2.096
3	IR.3.098, SA.3.169
4	RM4.149, RM.4.150, SA.4.171, SA.4.173, SI.4.221
5	IR.5.102

9.7.2 Assess and Prioritize Remediation [AVCDL-Operation-2]

Owner

Group: Security

NCWF Role: Cyber Defense Forensics Analyst

Administration

security	devops	development	risk
R	-	C	C

Help ensure vulnerabilities are remediated as quickly as necessary, reducing the window of opportunity for attackers.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Foundation-7] Incident Response Plan
[AVCDL-Release-3] Release Gate

External Group Product Dependencies

none

AVCDL Products

- Cybersecurity Incident Report

ISO 21434 Required Work Products

[WP-08-05] Vulnerability analysis
[WP-08-06] Evidence of managed vulnerabilities
[WP-08-X1] Apply incident response protocols
[WP-13-X1] Cybersecurity incident response plan implementation

WP.29 CSMS Requirements

[7.2.2.2(f)] The processes used for ensuring that the risk assessment is kept current.
[7.2.2.2(g)] The processes used to monitor for, detect and respond to cyber-attacks, cyber threats and vulnerabilities on vehicle types and 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.

CMMC Applicable Practices

Level	Practice
1	SI.1.210
2	IR.2.094, RM.2.143
3	**none**
4	**none**
5	**none**

9.7.3 Root Cause Vulnerabilities [AVCDL-Operation-3]

Owner

Group: Security

NCWF Role: Cyber Defense Forensics Analyst

Administration

security	devops	development	risk
R	-	C	-

Help reduce the frequency of vulnerabilities in the future.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Foundation-7] Incident Response Plan
[AVCDL-Release-3] Release Gate

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	Element implementation
Risk	**none**

AVCDL Products

- Cybersecurity Incident Report

ISO 21434 Required Work Products

[WP-08-05] Vulnerability analysis
[WP-08-06] Evidence of managed vulnerabilities
[WP-08-X1] Apply incident response protocols
[WP-13-X1] Cybersecurity incident response plan implementation

WP.29 CSMS Requirements

[7.2.2.2(g)] The processes used to monitor for, detect and respond to cyber-attacks, cyber threats and vulnerabilities on vehicle types and 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.

CMMC Applicable Practices

Level	Practice
1	SI.1.210
2	IR.2.094, RM.2.143
3	**none**
4	**none**
5	**none**

9.7.4 Secure Deployment [AVCDL-Operation-4]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
C	R	C	-

Software must be deployed in a secure manner.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Foundation-10] Deployment Plan
[AVCDL-Release-3] Release Gate

External Group Product Dependencies

Group	Inputs
Devops	• Deployment infrastructure • Deployment process
Development	Materials for deployment
Risk	**none**

AVCDL Products

- Software Deployment Report

ISO 21434 Required Work Products

[WP-12-X1] Production control plan implementation

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	AC.1.001, AC.1.002, AC.1.003. AC.1.004, IA.1.076, IA.1.077, PE.1.131, PE.1.133, PE.1.134
2	AC.2.007, AC.2.008, AC.2.009, AC.2.010, IA.2.081, IA.2.082, MP.2.119, MP.2.120, MP.2.121, SC.2.179
3	AC.3.017, AC.3.018, AC.3.019, AC.3.014, AM.3.036, AU.3.049, AU.3.052, IA.3.083, IA.3.084, IS.3.086
4	AM.4.226
5	AU.5.055

9.8 Decommissioning Phase

Predecessor: Operation Phase

Successor: N/A

Decommissioning is a part of the lifecycle of an item or component and is considered in the concept and product development phases.

Decommissioning is different from end of support. An organization can end support for an item or component, but that item or component can still function as designed in the field. Both decommissioning and end of support present cybersecurity implications, but those implications are considered separately.

Every product release should include a decommissioning plan containing information as to how to properly dispose of the security-related information constrained within the product.

[AVCDL-Decommissioning-1] Apply Decommissioning Protocol

The decommissioning protocol specified in the decommissioning plan should be applied to the system coming out of service.

Decommissioning phase product dependencies are visualized in Figure 15.

ISO 21434 Required Work Products

none

9.8.1 Apply Decommissioning Protocol [AVCDL-Decommissioning-1]

Owner

Group: Devops

NCWF Role: Information Systems Security Developer

Administration

security	devops	development	risk
I	R	-	-

Apply protocols appropriate to ensuring that any and all security-related information has been purged from the system.

Training Provided

yes

Phase Requirement Dependencies

[AVCDL-Foundation-8] Decommissioning Plan

External Group Product Dependencies

Group	Inputs
Devops	**none**
Development	List of data stored on systems
Risk	**none**

AVCDL Products

- Decommissioning Report

ISO 21434 Required Work Products

none

WP.29 CSMS Requirements

none

CMMC Applicable Practices

Level	Practice
1	AC.1.001, AC.1.002, AC.1.003. AC.1.004, IA.1.076, IA.1.077, MP.1.118, PE.1.131, PE.1.133, PE.1.134
2	AC.2.007, AC.2.008, AC.2.009, AC.2.010, IA.2.081, IA.2.082, SC.2.179
3	AC.3.017, AC.3.018, AC.3.019, AC.3.014, AM.3.036, AU.3.049, AU.3.052, IA.3.083, IA.3.084, IS.3.086, MA.3.115, PE.3.136
4	AM.4.226
5	AU.5.055

10. Requirement Role Assignments

The following table shows the ***AVCDL*** process requirement role assignments.

Requirement	Name	Group	NCWF Title
Foundation-1	Training	Security	Cyber Instructor
Foundation-2	Roles and Responsibilities	Security	Systems Requirements Planner
Foundation-3	Toolchain Support	Devops	Information Systems Security Developer
Foundation-4	Definition of Security Requirements	Security	Systems Requirements Planner
Foundation-5	Protect the Code	Devops	Information Systems Security Developer
Foundation-6	Ensure Release Integrity	Devops	Information Systems Security Developer
Foundation-7	Incident Response Plan	Security	Partner Integration Planner
Foundation-8	Decommissioning Plan	Security	partner integration Planner
Foundation-9	Threat Prioritization Plan	Security	Systems Requirements Planner
Foundation-10	Deployment Plan	Security	Information Systems Security Developer
Requirements-1	Definition of Security Requirements	Security	Security Architect
Requirements-2	Requirements Gate	Security	Secure Software Assessor
Design-1	Take Security Requirements and Risk Information into Account During Software Design	Development	Software Developer
Design-2	Review the Software Design to Verify Compliance with Security Requirements and Risk Information	Security	Systems Requirements Planner
Design-3	Attack Surface Reduction	Security	Security Architect
Design-4	Threat Modeling	Security	Security Architect
Design-5	Design Gate	Security	Secure Software Assessor
Implementation-1	Use Approved Tools	Development	Software Developer
Implementation-2	Configure the Compilation and Build Process to Improve Executable Security	Devops	Information Systems Security Developer
Implementation-3	Configure the Software to Have Secure Settings by Default	Security	Security Architect
Implementation-4	Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality	Development	Software Developer
Implementation-5	Create Source Code Adhering to Secure Coding Practice	Development	Software Developer
Implementation-6	Deprecate Unsafe Functions	Development	Software Developer
Implementation-7	Static Analysis	Devops	Information Systems Security Developer
Implementation-8	Dynamic Program Analysis	Development	Software Developer
Implementation-9	Security Code Review	Security	Secure Software Assessor

Implementation-10	Fuzz Testing	Security	Vulnerability Assessment Analyst
Implementation-11	Implementation Gate	Security	Secure Software Assessor
Verification-1	Penetration Testing	Security	Vulnerability Assessment Analyst
Verification-2	Threat Model Review	Security	Security Architect
Verification-3	Attack Surface Analysis Review	Security	Security Architect
Verification-4	Verification Gate	Security	Secure Software Assessor
Release-1	Final Security Review	Security	Secure Software Assessor
Release-2	Archive	Devops	Information Systems Security Developer
Release-3	Release Gate	Security	Secure Software Assessor
Operation-1	Identify and Confirm Vulnerabilities on an Ongoing Basis	Security	Cyber Defense Incident Responder
Operation-2	Assess and Prioritize the Remediation of all Vulnerabilities	Security	Cyber Defense Forensics Analyst
Operation-3	Analyze Vulnerabilities to Identify Their Root Causes	Security	Cyber Defense Forensics Analyst
Operation-4	Secure Deployment	Devops	Information Systems Security Developer
Decommissioning-1	Decommissioning Protocol	Devops	Information Systems Security Developer

Table 2 - Requirement Role Assignments

11. Groups

This folder contains documents related to the groups responsible for implementation of the ***AVCDL***.

- Devops
- Development
- Security

The following shows the mapping of requirements to Group: (highlight is accountable)

		Title	security	devops	development	risk
Foundation	1	Training	R	I	C	I
	2	Roles and Responsibilities	R	C	C	
	3	Toolchain Support	C	R	C	
	4	Definition of Security Requirements	R	I	I	
	5	Protect the Code	C	R		
	6	Ensure Release Integrity	C	R	C	
	7	Incident Response Plan	R		C	
	8	Decommissioning Plan	R	C	C	
	9	Threat Prioritization Plan	R		I	I
	10	Deployment Plan	C	R	C	
Requi reme	1	Security Requirements Definition	R		I	
	2	Requirements Gate	R		R	
Design	1	Apply Security Requirements and Risk Information to Design	R		R	
	2	Security Design Review	R		R	C
	3	Attack Surface Reduction	R		R	
	4	Threat Modeling	R		R	R
	5	Design Gate	R		R	R
Implementation	1	Use Approved Tools	C	C	R	
	2	Configure the Compilation and Build Process to Improve Executable Security	C	R	C	
	3	Configure the Software to Have Secure Settings by Default	R		R	
	4	Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality	C	I	R	
	5	Create Source Code Adhering to Secure Coding Practice	C		R	
	6	Deprecate Unsafe Functions	C		R	
	7	Static Analysis	C	R	C	
	8	Dynamic Program Analysis	C		R	
	9	Security Code Review	R		C	
	10	Implementation Gate	R	R	R	
	11	Fuzz Testing	R	C	C	
Verification	1	Penetration Testing	R	C	C	
	2	Threat Model Review	R		R	R
	3	Attack Surface Analysis Review	R		R	
	4	Verification Gate	R		R	R
Release	1	Final Security Review	R	C	C	C
	2	Archive		R	C	
	3	Release Gate	R	R	R	R
Operation	1	Identify and Confirm Vulnerabilities on an Ongoing Basis	R		C	
	2	Assess and Prioritize the Remediation of all Vulnerabilities	R		C	C
	3	Analyze Vulnerabilities to Identify Their Root Causes	R		C	
	4	Secure Deployment	C	R	C	
De c	1	Apply Decommissioning Protocol	I	R		

Table 3 – Requirement - Group Mapping

Note: Information regarding RACI is at:
https://en.wikipedia.org/wiki/Responsibility_assignment_matrix

11.1 Groups [Devops]

The following process requirements are the responsibility of devops:

Requirement	Description
Foundation-3	Toolchain Support
Foundation-5	Protect the Code
Foundation-6	Ensure Release Integrity
Foundation-10	Deployment Plan
Implementation-2	Configure the Compilation and Build Process to Improve Executable Security
Implementation-7	Static Analysis
Release-2	Archive
Operation-4	Secure Deployment
Decommissioning-1	Decommissioning Protocol

Table 4 - Devops Requirement Responsibilities

11.2 Groups [Development]

The following process requirements are the responsibility of development:

Requirement	Description
Design-1	Take Security Requirements and Risk Information into Account During Software Design
Implementation-1	Use Approved Tools
Implementation-4	Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality
Implementation-5	Create Source Code Adhering to Secure Coding Practice
Implementation-6	Deprecate Unsafe Functions
Implementation-8	Dynamic Program Analysis

Table 5 - Development Requirement Responsibilities

11.3 Groups [Security]

The following process requirements are the responsibility of security:

Requirement	Description
Foundation-1	Training
Foundation-2	Roles and Responsibilities
Foundation-4	Definition of Security Requirements
Foundation-7	Incident Response Plan
Foundation-8	Decommissioning Plan
Foundation-9	Threat Prioritization Plan
Requirements-1	Definition of Security Requirements
Requirements-2	Requirements Gate
Design-2	Review the Software Design to Verify Compliance with Security Requirements and Risk Information
Design-3	Attack Surface Reduction
Design-4	Threat Modeling
Design-5	Design Gate
Implementation-3	Configure the Software to Have Secure Settings by Default
Implementation-10	Fuzz Testing
Implementation-11	Implementation Gate
Verification-1	Penetration Testing
Verification-2	Threat Model Review
Verification-3	Attack Surface Analysis Review
Verification-4	Verification Gate
Release-1	Final Security Review
Release-3	Release Gate
Operation-1	Identify and Confirm Vulnerabilities on an Ongoing Basis
Operation-2	Assess and Prioritize the Remediation of all Vulnerabilities
Operation-3	Analyze Vulnerabilities to Identify Their Root Causes

Table 6 - Security Requirement Responsibilities

12. NCWF Roles

Note: This material is extracted from the NIST NCWF documentation. It is included here for reference only.

NIST SP 800-181 [***National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework (NCWF)***] provides an exhaustive breakdown of cybersecurity roles and their associated tasks.

The following NCWF roles have been identified as necessary in support of implementation of ***AVCDL***:

- Cyber Defense Forensics Analyst
- Cyber Defense Incident Responder
- Cyber Instructor
- Information Systems Security Developer
- Partner Integration Planner
- Secure Software Assessor
- Security Architect
- Software Developer
- Systems Requirements Planner
- Systems Security Analyst
- Vulnerability Assessment Analyst

Note: There will be tasks and abilities called out for roles in NCWF which are not leveraged. Additionally, there will be areas where there is not a 1-to-1 mapping.

12.1 Cyber Defense Forensics Analyst (IN-FOR-002)

Analyzes digital evidence and investigates computer security incidents to derive useful information in support of system/network vulnerability mitigation.

ID	Task
T0027	Conduct analysis of log files, evidence, and other information to determine best methods for identifying the perpetrator(s) of a network intrusion.
T0036	Confirm what is known about an intrusion and discover new information, if possible, after identifying intrusion via dynamic analysis.
T0048	Create a forensically sound duplicate of the evidence (i.e., forensic image) that ensures the original evidence is not unintentionally modified, to use for data recovery and analysis processes. This includes, but is not limited to, hard drives, floppy diskettes, CDs, PDAs, mobile phones, GPS, and all tape formats.
T0049	Decrypt seized data using technical means.
T0075	Provide technical summary of findings in accordance with established reporting procedures.
T0087	Ensure that chain of custody is followed for all digital media acquired in accordance with the Federal Rules of Evidence.
T0103	Examine recovered data for information of relevance to the issue at hand.
T0113	Identify digital evidence for examination and analysis in such a way as to avoid unintentional alteration.
T0165	Perform dynamic analysis to boot an ‘image’ of a drive (without necessarily having the original drive) to see the intrusion as the user may have seen it, in a native environment.
T0167	Perform file signature analysis.
T0168	Perform hash comparison against established database.
T0172	Perform real-time forensic analysis (e.g., using Helix in conjunction with LiveView).
T0173	Perform timeline analysis.
T0175	Perform real-time cyber defense incident handling (e.g., forensic collections, intrusion correlation and tracking, threat analysis, and direct system remediation) tasks to support deployable Incident Response Teams (IRTs).
T0179	Perform static media analysis.
T0182	Perform tier 1, 2, and 3 malware analysis.
T0190	Prepare digital media for imaging by ensuring data integrity (e.g., write blockers in accordance with standard operating procedures).
T0212	Provide technical assistance on digital evidence matters to appropriate personnel.
T0216	Recognize and accurately report forensic artifacts indicative of a particular operating system.
T0238	Extract data using data carving techniques (e.g., Forensic Tool Kit [FTK], Foremost).
T0240	Capture and analyze network traffic associated with malicious activities using network monitoring tools.

T0241 Use specialized equipment and techniques to catalog, document, extract, collect, package, and preserve digital evidence.

T0253 Conduct cursory binary analysis.

T0279 Serve as technical expert and liaison to law enforcement personnel and explain incident details as required.

T0285 Perform virus scanning on digital media.

T0286 Perform file system forensic analysis.

T0287 Perform static analysis to mount an "image" of a drive (without necessarily having the original drive).

T0288 Perform static malware analysis.

T0289 Utilize deployable forensics toolkit to support operations as necessary.

T0312 Coordinate with intelligence analysts to correlate threat assessment data.

T0396 Process image with appropriate tools depending on analyst's goals.

T0397 Perform Windows registry analysis.

T0398 Perform file and registry monitoring on the running system after identifying intrusion via dynamic analysis.

T0399 Enter media information into tracking database (e.g., Product Tracker Tool) for digital media that has been acquired.

T0400 Correlate incident data and perform cyber defense reporting.

T0401 Maintain deployable cyber defense toolkit (e.g., specialized cyber defense software/hardware) to support Incident Response Team mission.

T0432 Collect and analyze intrusion artifacts (e.g., source code, malware, and system configuration) and use discovered data to enable mitigation of potential cyber defense incidents within the enterprise.

T0532 Review forensic images and other data sources (e.g., volatile data) for recovery of potentially relevant information.

T0546 Write and publish cyber defense recommendations, reports, and white papers on incident findings to appropriate constituencies.

12.2 Cyber Defense Incident Responder (PR-CIR-001)

Investigates, analyzes, and responds to cyber incidents within the network environment or enclave.

ID	Task
T0041	Coordinate and provide expert technical support to enterprise-wide cyber defense technicians to resolve cyber defense incidents.
T0047	Correlate incident data to identify specific vulnerabilities and make recommendations that enable expeditious remediation.
T0161	Perform analysis of log files from a variety of sources (e.g., individual host logs, network traffic logs, firewall logs, and intrusion detection system [IDS] logs) to identify possible threats to network security.
T0163	Perform cyber defense incident triage, to include determining scope, urgency, and potential impact, identifying the specific vulnerability, and making recommendations that enable expeditious remediation.
T0164	Perform cyber defense trend analysis and reporting.
T0170	Perform initial, forensically sound collection of images and inspect to discern possible mitigation/remediation on enterprise systems.
T0175	Perform real-time cyber defense incident handling (e.g., forensic collections, intrusion correlation and tracking, threat analysis, and direct system remediation) tasks to support deployable Incident Response Teams (IRTs).
T0214	Receive and analyze network alerts from various sources within the enterprise and determine possible causes of such alerts.
T0233	Track and document cyber defense incidents from initial detection through final resolution.
T0246	Write and publish cyber defense techniques, guidance, and reports on incident findings to appropriate constituencies.
T0262	Employ approved defense-in-depth principles and practices (e.g., defense-in-multiple places, layered defenses, security robustness).
T0278	Collect intrusion artifacts (e.g., source code, malware, Trojans) and use discovered data to enable mitigation of potential cyber defense incidents within the enterprise.
T0279	Serve as technical expert and liaison to law enforcement personnel and explain incident details as required.
T0312	Coordinate with intelligence analysts to correlate threat assessment data.
T0395	Write and publish after action reviews.
T0503	Monitor external data sources (e.g., cyber defense vendor sites, Computer Emergency Response Teams, Security Focus) to maintain currency of cyber defense threat condition and determine which security issues may have an impact on the enterprise.
T0510	Coordinate incident response functions.

12.3 Cyber Instructor (OV-TEA-002)

Develops and conducts training or education of personnel within cyber domain.

ID	Task
T0030	Conduct interactive training exercises to create an effective learning environment.
T0073	Develop new or identify existing awareness and training materials that are appropriate for intended audiences.
T0101	Evaluate the effectiveness and comprehensiveness of existing training programs.
T0224	Review training documentation (e.g., Course Content Documents [CCD], lesson plans, student texts, examinations, Schedules of Instruction [SOI], and course descriptions).
T0230	Support the design and execution of exercise scenarios.
T0247	Write instructional materials (e.g., standard operating procedures, production manual) to provide detailed guidance to relevant portion of the workforce.
T0316	Develop or assist in the development of computer-based training modules or classes.
T0317	Develop or assist in the development of course assignments.
T0318	Develop or assist in the development of course evaluations.
T0319	Develop or assist in the development of grading and proficiency standards.
T0320	Assist in the development of individual/collective development, training, and/or remediation plans.
T0321	Develop or assist in the development of learning objectives and goals.
T0322	Develop or assist in the development of on-the-job training materials or programs.
T0323	Develop or assist in the development of written tests for measuring and assessing learner proficiency.
T0352	Conduct learning needs assessments and identify requirements.
T0365	Develop or assist in the development of training policies and protocols for cyber training.
T0367	Develop the goals and objectives for cyber curriculum.
T0381	Present technical information to technical and nontechnical audiences.
T0382	Present data in creative formats.
T0395	Write and publish after action reviews.
T0443	Deliver training courses tailored to the audience and physical/virtual environments.
T0444	Apply concepts, procedures, software, equipment, and/or technology applications to students.
T0450	Design training curriculum and course content based on requirements.
T0451	Participate in development of training curriculum and course content.
T0467	Ensure that training meets the goals and objectives for cybersecurity training, education, or awareness.

T0519 Plan and coordinate the delivery of classroom techniques and formats (e.g., lectures, demonstrations, interactive exercises, multimedia presentations) for the most effective learning environment.

T0520 Plan non-classroom educational techniques and formats (e.g., video courses, mentoring, web-based courses).

T0535 Recommend revisions to curriculum and course content based on feedback from previous training sessions.

T0536 Serve as an internal consultant and advisor in own area of expertise (e.g., technical, copyright, print media, electronic media).

T0926 Develop or assist with the development of privacy training materials and other communications to increase employee understanding of company privacy policies, data handling practices and procedures and legal obligations.

12.4 Information Systems Security Developer (SP-SYS-001)

Designs, develops, tests, and evaluates information system security throughout the systems development life cycle.

ID	Task
T0012	Analyze design constraints, analyze trade-offs and detailed system and security design, and consider life cycle support
T0015	Apply security policies to applications that interface with one another, such as Business-to-Business (B2B) applications
T0018	Assess the effectiveness of cybersecurity measures utilized by system(s)
T0019	Assess threats to and vulnerabilities of computer system(s) to develop a security risk profile
T0021	Build, test, and modify product prototypes using working models or theoretical models
T0032	Conduct Privacy Impact Assessments (PIAs) of the application's security design for the appropriate security controls, which protect the confidentiality and integrity of Personally Identifiable Information (PII)
T0053	Design and develop cybersecurity or cybersecurity-enabled products.
T0055	Design hardware, operating systems, and software applications to adequately address cybersecurity requirements
T0056	Design or integrate appropriate data backup capabilities into overall system designs, and ensure that appropriate technical and procedural processes exist for secure system backups and protected storage of backup data
T0061	Develop and direct system testing and validation procedures and documentation
T0069	Develop detailed security design documentation for component and interface specifications to support system design and development
T0070	Develop Disaster Recovery and Continuity of Operations plans for systems under development and ensure testing prior to systems entering a production environment
T0076	Develop risk mitigation strategies to resolve vulnerabilities and recommend security changes to system or system components as needed
T0078	Develop specific cybersecurity countermeasures and risk mitigation strategies for systems and/or applications
T0105	Identify components or elements, allocate security functions to those elements, and describe the relationships between the elements
T0107	Identify and direct the remediation of technical problems encountered during testing and implementation of new systems (e.g., identify and find workarounds for communication protocols that are not interoperable)

T0109 Identify and prioritize essential system functions or sub-systems required to support essential capabilities or business functions for restoration or recovery after a system failure or during a system recovery event based on overall system requirements for continuity and availability

T0119 Identify, assess, and recommend cybersecurity or cybersecurity-enabled products for use within a system and ensure that recommended products are in compliance with organization's evaluation and validation requirements

T0122 Implement security designs for new or existing system(s)

T0124 Incorporate cybersecurity vulnerability solutions into system designs (e.g., Cybersecurity Vulnerability Alerts)

T0181 Perform risk analysis (e.g., threat, vulnerability, and probability of occurrence) whenever an application or system undergoes a major change

T0201 Provide guidelines for implementing developed systems to customers or installation teams

T0205 Provide input to the Risk Management Framework process activities and related documentation (e.g., system life-cycle support plans, concept of operations, operational procedures, and maintenance training materials).

T0228 Store, retrieve, and manipulate data for analysis of system capabilities and requirements

T0231 Provide support to security/certification test and evaluation activities

T0242 Utilize models and simulations to analyze or predict system performance under different operating conditions

T0269 Design and develop key management functions (as related to cybersecurity)

T0270 Analyze user needs and requirements to plan and conduct system security development

T0271 Develop cybersecurity designs to meet specific operational needs and environmental factors (e.g., access controls, automated applications, networked operations, high integrity and availability requirements, multilevel security/processing of multiple classification levels, and processing Sensitive Compartmented Information)

T0272 Ensure that security design and cybersecurity development activities are properly documented (providing a functional description of security implementation) and updated as necessary

T0304 Implement and integrate system development life cycle (SDLC) methodologies (e.g., IBM Rational Unified Process) into development environment.

T0326 Employ configuration management processes

T0359 Design, implement, test, and evaluate secure interfaces between information systems, physical systems, and/or embedded technologies

T0446 Design, develop, integrate, and update system security measures that provide confidentiality, integrity, availability, authentication, and non-repudiation

T0449 Design to security requirements to ensure requirements are met for all systems and/or applications

T0466 Develop mitigation strategies to address cost, schedule, performance, and security risks

T0509 Perform an information security risk assessment

T0518	Perform security reviews and identify security gaps in architecture.
T0527	Provide input to implementation plans and standard operating procedures as they relate to information systems security
T0541	Trace system requirements to design components and perform gap analysis.
T0544	Verify stability, interoperability, portability, and/or scalability of system architecture

12.5 Partner Integration Planner (CO-OPL-003)

Works to advance cooperation across organizational or national borders between cyber operations partners. Aids the integration of partner cyber teams by providing guidance, resources, and collaboration to develop best practices and facilitate organizational support for achieving objectives in integrated cyber actions.

ID	Task
T0571	Apply expertise in policy and processes to facilitate the development, negotiation, and internal staffing of plans and/or memorandums of agreement.
T0581	Assist and advise interagency partners in identifying and developing best practices for facilitating operational support to achievement of organization objectives.
T0582	Provide expertise to course of action development.
T0601	Collaborate with other team members or partner organizations to develop a diverse program of information materials (e.g., web pages, briefings, print materials).
T0627	Contribute to crisis action planning for cyber operations.
T0629	Contribute to the development, staffing, and coordination of cyber operations policies, performance standards, plans and approval packages with appropriate internal and/or external decision makers.
T0635	Coordinate with intelligence and cyber defense partners to obtain relevant essential information.
T0665	Develop or participate in the development of standards for providing, requesting, and/or obtaining support from external partners to synchronize cyber operations.
T0666	Develop or shape international cyber engagement strategies, policies, and activities to meet organization objectives.
T0669	Develop strategy and processes for partner planning, operations, and capability development.
T0670	Develop, implement, and recommend changes to appropriate planning procedures and policies.
T0671	Develop, maintain, and assess cyber cooperation security agreements with external partners.
T0699	Facilitate interactions between internal and external partner decision makers to synchronize and integrate courses of action in support of objectives.
T0700	Facilitate the sharing of ‘best practices’ and ‘lessons learned’ throughout the cyber operations community.
T0712	Identify and manage security cooperation priorities with external partners.
T0729	Inform external partners of the potential effects of new or revised policy and guidance on cyber operations partnering activities.
T0732	Integrate cyber planning/targeting efforts with other organizations.
T0739	Maintain relationships with internal and external partners involved in cyber planning or related areas.

T0747 Monitor and evaluate integrated cyber operations to identify opportunities to meet organization objectives.

T0759 Contribute to the review and refinement of policy, to include assessments of the consequences of endorsing or not endorsing such policy.

T0760 Provide subject matter expertise to planning teams, coordination groups, and task forces as necessary.

T0763 Conduct long-range, strategic planning efforts with internal and external partners in cyber activities.

T0764 Provide subject matter expertise to planning efforts with internal and external cyber operations partners.

T0766 Propose policy which governs interactions with external coordination groups.

T0772 Prepare for and provide subject matter expertise to exercises.

T0784 Provide cyber focused guidance and advice on intelligence support plan inputs.

T0787 Provide input for the development and refinement of the cyber operations objectives, priorities, strategies, plans, and programs.

T0795 Provide planning support between internal and external partners.

T0817 Serve as a conduit of information from partner teams by identifying subject matter experts who can assist in the investigation of complex or unusual situations.

T0818 Serve as a liaison with external partners.

T0823 Submit or respond to requests for deconfliction of cyber operations.

T0825 Synchronize cyber international engagement activities and associated resource requirements as appropriate.

T0826 Synchronize cyber portions of security cooperation plans.

T0836 Document lessons learned that convey the results of events and/or exercises.

12.6 Secure Software Assessor (SP-DEV-002)

Analyzes the security of new or existing computer applications, software, or specialized utility programs and provides actionable results.

ID	Task
T0013	Apply coding and testing standards, apply security testing tools including “’fuzzing” static-analysis code scanning tools, and conduct code reviews.
T0014	Apply secure code documentation.
T0022	Capture security controls used during the requirements phase to integrate security within the process, to identify key security objectives, and to maximize software security while minimizing disruption to plans and schedules.
T0038	Develop threat model based on customer interviews and requirements.
T0040	Consult with engineering staff to evaluate interface between hardware and software.
T0100	Evaluate factors such as reporting formats required, cost constraints, and need for security restrictions to determine hardware configuration.
T0111	Identify basic common coding flaws at a high level.
T0117	Identify security implications and apply methodologies within centralized and decentralized environments across the enterprise’s computer systems in software development.
T0118	Identify security issues around steady state operation and management of software and incorporate security measures that must be taken when a product reaches its end of life.
T0171	Perform integrated quality assurance testing for security functionality and resiliency attack.
T0181	Perform risk analysis (e.g., threat, vulnerability, and probability of occurrence) whenever an application or system undergoes a major change.
T0217	Address security implications in the software acceptance phase including completion criteria, risk acceptance and documentation, common criteria, and methods of independent testing.
T0228	Store, retrieve, and manipulate data for analysis of system capabilities and requirements.
T0236	Translate security requirements into application design elements including documenting the elements of the software attack surfaces, conducting threat modeling, and defining any specific security criteria.
T0266	Perform penetration testing as required for new or updated applications.
T0311	Consult with customers about software system design and maintenance.
T0324	Direct software programming and development of documentation.
T0337	Supervise and assign work to programmers, designers, technologists and technicians, and other engineering and scientific personnel.
T0424	Analyze and provide information to stakeholders that will support the development of security application or modification of an existing security application.

T0428 Analyze security needs and software requirements to determine feasibility of design within time and cost constraints and security mandates.

T0436 Conduct trial runs of programs and software applications to ensure that the desired information is produced, and instructions and security levels are correct.

T0456 Develop secure software testing and validation procedures.

T0457 Develop system testing and validation procedures, programming, and documentation.

T0516 Perform secure program testing, review, and/or assessment to identify potential flaws in codes and mitigate vulnerabilities.

T0554 Determine and document software patches or the extent of releases that would leave software vulnerable.

12.7 Security Architect (SP-ARC-002)

Ensures that the stakeholder security requirements necessary to protect the organization's mission and business processes are adequately addressed in all aspects of enterprise architecture including reference models, segment and solution architectures, and the resulting systems supporting those missions and business processes.

ID	Task
T0050	Define and prioritize essential system capabilities or business functions required for partial or full system restoration after a catastrophic failure event.
T0051	Define appropriate levels of system availability based on critical system functions and ensure that system requirements identify appropriate disaster recovery and continuity of operations requirements to include any appropriate fail-over/alternate site requirements, backup requirements, and material supportability requirements for system recover/restoration.
T0071	Develop/integrate cybersecurity designs for systems and networks with multilevel security requirements or requirements for the processing of multiple classification levels of data primarily applicable to government organizations (e.g., UNCLASSIFIED, SECRET, and TOP SECRET).
T0082	Document and address organization's information security, cybersecurity architecture, and systems security engineering requirements throughout the acquisition life cycle.
T0084	Employ secure configuration management processes.
T0090	Ensure that acquired or developed system(s) and architecture(s) are consistent with organization's cybersecurity architecture guidelines.
T0108	Identify and prioritize critical business functions in collaboration with organizational stakeholders.
T0177	Perform security reviews, identify gaps in security architecture, and develop a security risk management plan.
T0196	Provide advice on project costs, design concepts, or design changes.
T0203	Provide input on security requirements to be included in statements of work and other appropriate procurement documents.
T0205	Provide input to the Risk Management Framework process activities and related documentation (e.g., system life-cycle support plans, concept of operations, operational procedures, and maintenance training materials).
T0268	Define and document how the implementation of a new system or new interfaces between systems impacts the security posture of the current environment.
T0307	Analyze candidate architectures, allocate security services, and select security mechanisms.
T0314	Develop a system security context, a preliminary system security Concept of Operations (CONOPS) and define baseline system security requirements in accordance with applicable cybersecurity requirements.

T0328 Evaluate security architectures and designs to determine the adequacy of security design and architecture proposed or provided in response to requirements contained in acquisition documents.

T0338 Write detailed functional specifications that document the architecture development process.

T0427 Analyze user needs and requirements to plan architecture.

T0448 Develop enterprise architecture or system components required to meet user needs.

T0473 Document and update as necessary all definition and architecture activities.

T0484 Determine the protection needs (i.e., security controls) for the information system(s) and network(s) and document appropriately.

T0542 Translate proposed capabilities into technical requirements.

T0556 Assess and design security management functions as related to cyberspace.

12.8 Software Developer (SP-DEV-001)

Develops, creates, maintains, and writes/codes new (or modifies existing) computer applications, software, or specialized utility programs.

ID	Task
T0009	Analyze information to determine, recommend, and plan the development of a new application or modification of an existing application.
T0011	Analyze user needs and software requirements to determine feasibility of design within time and cost constraints.
T0013	Apply coding and testing standards, apply security testing tools including "'fuzzing" static-analysis code scanning tools, and conduct code reviews.
T0014	Apply secure code documentation.
T0022	Capture security controls used during the requirements phase to integrate security within the process, to identify key security objectives, and to maximize software security while minimizing disruption to plans and schedules.
T0026	Compile and write documentation of program development and subsequent revisions, inserting comments in the coded instructions so others can understand the program.
T0034	Confer with systems analysts, engineers, programmers, and others to design application and to obtain information on project limitations and capabilities, performance requirements, and interfaces.
T0040	Consult with engineering staff to evaluate interface between hardware and software.
T0046	Correct errors by making appropriate changes and rechecking the program to ensure that desired results are produced.
T0057	Design, develop, and modify software systems, using scientific analysis and mathematical models to predict and measure outcome and consequences of design.
T0077	Develop secure code and error handling.
T0100	Evaluate factors such as reporting formats required, cost constraints, and need for security restrictions to determine hardware configuration.
T0111	Identify basic common coding flaws at a high level.
T0117	Identify security implications and apply methodologies within centralized and decentralized environments across the enterprise's computer systems in software development.
T0118	Identify security issues around steady state operation and management of software and incorporate security measures that must be taken when a product reaches its end of life.
T0171	Perform integrated quality assurance testing for security functionality and resiliency attack.
T0176	Perform secure programming and identify potential flaws in codes to mitigate vulnerabilities.
T0181	Perform risk analysis (e.g., threat, vulnerability, and probability of occurrence) whenever an application or system undergoes a major change.

T0189 Prepare detailed workflow charts and diagrams that describe input, output, and logical operation, and convert them into a series of instructions coded in a computer language.

T0217 Address security implications in the software acceptance phase including completion criteria, risk acceptance and documentation, common criteria, and methods of independent testing.

T0228 Store, retrieve, and manipulate data for analysis of system capabilities and requirements.

T0236 Translate security requirements into application design elements including documenting the elements of the software attack surfaces, conducting threat modeling, and defining any specific security criteria.

T0267 Design countermeasures and mitigations against potential exploitations of programming language weaknesses and vulnerabilities in system and elements.

T0303 Identify and leverage the enterprise-wide version control system while designing and developing secure applications.

T0311 Consult with customers about software system design and maintenance.

T0324 Direct software programming and development of documentation.

T0337 Supervise and assign work to programmers, designers, technologists and technicians, and other engineering and scientific personnel.

T0416 Enable applications with public keying by leveraging existing public key infrastructure (PKI) libraries and incorporating certificate management and encryption functionalities when appropriate.

T0417 Identify and leverage the enterprise-wide security services while designing and developing secure applications (e.g., Enterprise PKI, Federated Identity server, Enterprise Antivirus solution) when appropriate.

T0436 Conduct trial runs of programs and software applications to ensure that the desired information is produced, and instructions and security levels are correct.

T0455 Develop software system testing and validation procedures, programming, and documentation.

T0500 Modify and maintain existing software to correct errors, to adapt it to new hardware, or to upgrade interfaces and improve performance.

T0553 Apply cybersecurity functions (e.g., encryption, access control, and identity management) to reduce exploitation opportunities.

T0554 Determine and document software patches or the extent of releases that would leave software vulnerable.

12.9 Systems Requirements Planner (SP-SRP-001)

Consults with customers to evaluate functional requirements and translate functional requirements into technical solutions.

ID	Task
T0033	Conduct risk analysis, feasibility study, and/or trade-off analysis to develop, document, and refine functional requirements and specifications.
T0039	Consult with customers to evaluate functional requirements.
T0045	Coordinate with systems architects and developers, as needed, to provide oversight in the development of design solutions.
T0052	Define project scope and objectives based on customer requirements.
T0062	Develop and document requirements, capabilities, and constraints for design procedures and processes.
T0127	Integrate and align information security and/or cybersecurity policies to ensure that system analysis meets security requirements.
T0156	Oversee and make recommendations regarding configuration management.
T0174	Perform needs analysis to determine opportunities for new and improved business process solutions.
T0191	Prepare use cases to justify the need for specific information technology (IT) solutions.
T0235	Translate functional requirements into technical solutions.
T0273	Develop and document supply chain risks for critical system elements, as appropriate.
T0300	Develop and document User Experience (UX) requirements including information architecture and user interface requirements.
T0313	Design and document quality standards.
T0325	Document a system’s purpose and preliminary system security concept of operations.
T0334	Ensure that all systems components can be integrated and aligned (e.g., procedures, databases, policies, software, and hardware).
T0454	Define baseline security requirements in accordance with applicable guidelines.
T0463	Develop cost estimates for new or modified system(s).
T0497	Manage the information technology (IT) planning process to ensure that developed solutions meet customer requirements.

12.10 Systems Security Analyst (OM-ANA-001)

Responsible for the analysis and development of the integration, testing, operations, and maintenance of systems security.

ID	Task
T0015	Apply security policies to applications that interface with one another, such as Business-to-Business (B2B) applications.
T0016	Apply security policies to meet security objectives of the system.
T0017	Apply service-oriented security architecture principles to meet organization's confidentiality, integrity, and availability requirements.
T0085	Ensure all systems security operations and maintenance activities are properly documented and updated as necessary.
T0086	Ensure that the application of security patches for commercial products integrated into system design meet the timelines dictated by the management authority for the intended operational environment.
T0088	Ensure that cybersecurity-enabled products or other compensating security control technologies reduce identified risk to an acceptable level.
T0123	Implement specific cybersecurity countermeasures for systems and/or applications.
T0128	Integrate automated capabilities for updating or patching system software where practical and develop processes and procedures for manual updating and patching of system software based on current and projected patch timeline requirements for the operational environment of the system.
T0169	Perform cybersecurity testing of developed applications and/or systems.
T0177	Perform security reviews, identify gaps in security architecture, and develop a security risk management plan.
T0187	Plan and recommend modifications or adjustments based on exercise results or system environment.
T0194	Properly document all systems security implementation, operations, and maintenance activities and update as necessary.
T0202	Provide cybersecurity guidance to leadership.
T0205	Provide input to the Risk Management Framework process activities and related documentation (e.g., system life-cycle support plans, concept of operations, operational procedures, and maintenance training materials).
T0243	Verify and update security documentation reflecting the application/system security design features.
T0309	Assess the effectiveness of security controls.
T0344	Assess all the configuration management (change configuration/release management) processes.
T0462	Develop procedures and test fail-over for system operations transfer to an alternate site based on system availability requirements.
T0469	Analyze and report organizational security posture trends.

T0470 Analyze and report system security posture trends.

T0475 Assess adequate access controls based on principles of least privilege and need-to-know.

T0477 Ensure the execution of disaster recovery and continuity of operations.

T0485 Implement security measures to resolve vulnerabilities, mitigate risks, and recommend security changes to system or system components as needed.

T0489 Implement system security measures in accordance with established procedures to ensure confidentiality, integrity, availability, authentication, and non-repudiation.

T0492 Ensure the integration and implementation of Cross-Domain Solutions (CDS) in a secure environment.

T0499 Mitigate/correct security deficiencies identified during security/certification testing and/or recommend risk acceptance for the appropriate senior leader or authorized representative.

T0504 Assess and monitor cybersecurity related to system implementation and testing practices.

T0508 Verify minimum security requirements are in place for all applications.

T0526 Provides cybersecurity recommendations to leadership based on significant threats and vulnerabilities.

T0545 Work with stakeholders to resolve computer security incidents and vulnerability compliance.

T0548 Provide advice and input for Disaster Recovery, Contingency, and Continuity of Operations Plans.

12.11 Vulnerability Assessment Analyst (PR-VAM-001)

Performs assessments of systems and networks within the NE or enclave and identifies where those systems/networks deviate from acceptable configurations, enclave policy, or local policy. Measures effectiveness of defense-in-depth architecture against known vulnerabilities.

ID	Task
T0010	Analyze organization's cyber defense policies and configurations and evaluate compliance with regulations and organizational directives.
T0028	Conduct and/or support authorized penetration testing on enterprise network assets.
T0138	Maintain deployable cyber defense audit toolkit (e.g., specialized cyber defense software and hardware) to support cyber defense audit missions.
T0142	Maintain knowledge of applicable cyber defense policies, regulations, and compliance documents specifically related to cyber defense auditing.
T0188	Prepare audit reports that identify technical and procedural findings and provide recommended remediation strategies/solutions.
T0252	Conduct required reviews as appropriate within environment (e.g., Technical Surveillance, Countermeasure Reviews [TSCM], TEMPEST countermeasure reviews).
T0549	Perform technical (evaluation of technology) and nontechnical (evaluation of people and operations) risk and vulnerability assessments of relevant technology focus areas (e.g., local computing environment, network and infrastructure, enclave boundary, supporting infrastructure, and applications).
T0550	Make recommendations regarding the selection of cost-effective security controls to mitigate risk (e.g., protection of information, systems and processes).

13. SSDF Background Material

Note: This material is extracted from the NIST SSDF documentation. It is included here for reference only.

The NIST Secure Software Development Framework (**SSDF**) provides a more general approach which calls out numerous practices, and provides references to the applicable standards. SSDF's groups are:

- Prepare the Organization (PO)
- Protect Software (PS)
- Produce Well-Secured Software (PW)
- Respond to Vulnerability Reports (RV)

Within each of these are multiple practices and tasks.

13.1 Prepare the Organization (PO) Practices

Predecessor: N/A

Successor: Protect Software (PS) Practices

Define Security Requirements for Software Development (PO.1)

Ensure security requirements for software development are known at all times so they can be taken into account throughout the SDLC, and duplication of effort can be minimized because the requirements information can be collected once and shared. This includes requirements from internal sources, such as the organization's policies, business objectives, and risk management strategy, and external sources, such as applicable laws and regulations.

Implement Roles and Responsibilities (PO.2)

Ensure everyone inside and outside the organization involved in the SDLC is prepared to perform their SSDF-related roles and responsibilities throughout the SDLC.

Implement a Supporting Toolchain (PO.3)

Use automation to reduce the human effort needed and improve the accuracy, consistency, and comprehensiveness of security practices throughout the SDLC, as well as a way to document and demonstrate use of these practices without significant additional effort or expense.

Define Criteria for Software Security Checks (PO.4)

Help ensure the software resulting from the SDLC meets the organization's expectations by defining criteria for checking the software's security during development.

Note: PO.4 applies to both the foundation and requirements phases

13.2 Protect Software (PS) Practices

Predecessor: Prepare the Organization (PO) Practices

Successor: Produce Well-Secured Software (PW) Practices

Protect All Forms of Code from Unauthorized Access and Tampering (PS.1)

Help prevent unauthorized changes to code, both inadvertent and intentional, which could circumvent or negate the intended security characteristics of the software. For code not intended to be publicly accessible, it helps prevent theft of the software and makes it more difficult for attackers to find vulnerabilities in the software.

Note: PS.1 should be in the foundation phase

Provide a Mechanism for Verifying Software Release Integrity (PS.2)

Help software consumers ensure the software they acquire is legitimate and has not been tampered with.

Note: PS.2 should be in the foundation phase

Archive and Protect Each Software Release (PS.3)

Helps identify, analyze, and eliminate vulnerabilities discovered in the software after release.

Note: PS.3 should be in the foundation phase

13.3 Produce Well-Secured Software (PW) Practices

Predecessor: Protect Software (PS) Practices

Successor: Respond to Vulnerability Reports (RV) Practices

Take Security Requirements and Risk Information into Account During Software Design (PW.1)

Determine which security requirements the software's design should meet and determine what security risks the software is likely to face during production operation and how those risks should be mitigated by the software's design. Addressing security requirements and risks during software design instead of later helps to make software development more efficient.

Note: PW.1 applies to the design phase

Review the Software Design to Verify Compliance with Security Requirements and Risk Information (PW.2)

Help ensure the software will meet the security requirements and satisfactorily address the identified risk information.

Note: PW.2 applies to the design phase

Verify Third-Party Software Compiles with Security Requirements (PW.3)

Reduce the risk associated with using acquired software modules and services, which are potential sources of additional vulnerabilities.

Note: PW.3 applies to the design phase

Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality (PW.4)

Lower the costs of software development, expedite software development, and decrease the likelihood of introducing additional security vulnerabilities into the software. These are particularly true for software that implements security functionality, such as cryptographic modules and protocols.

Note: PW.4 applies to the implementation phase

Create Source Code Adhering to Secure Coding Practice (PW.5)

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities during source code creation.

Note: PW.5 applies to the implementation phase

Configure the Compilation and Build Process to Improve Executable Security (PW.6)

Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities before testing occurs.

Note: PW.6 applies to the implementation phase

Review and/or Analyze Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.7)

Help identify vulnerabilities before software is released so they can be corrected before release, which prevents exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities. Human-readable code is source code and any other form of code an organization deems as human readable.

Note: PW.7 applies to the implementation phase

Test Executable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.8)

Help identify vulnerabilities before software is released so they can be corrected before release, which prevents exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities. Executable code is binaries, directly executed bytecode, directly executed source code, and any other form of code an organization deems as executable.

Note: PW.8 applies to the verification phase

Configure the Software to Have Secure Settings by Default (PW.9)

Help improve the security of the software at installation time, which reduces the likelihood of the software being deployed with weak security settings that would put it at greater risk of compromise.

Note: PW.9 applies to the implementation phase

13.4 Respond to Vulnerability Reports (RV) Practices

Predecessor: Produce Well-Secured Software (PW) Practices

Successor: N/A

Identify and Confirm Vulnerabilities on an Ongoing Basis (RV.1)

Help ensure vulnerabilities are identified more quickly so they can be remediated more quickly, reducing the window of opportunity for attackers.

Assess and Prioritize the Remediation of all Vulnerabilities (RV.2)

Help ensure vulnerabilities are remediated as quickly as necessary, reducing the window of opportunity for attackers.

Analyze Vulnerabilities to Identify Their Root Causes (RV.3)

Help reduce the frequency of vulnerabilities in the future.

14. MSSDL Background Material

Note: This material is extracted from the Microsoft SDL documentation. It is included here for reference only.

The archetype for the SDL is the Microsoft SDL (**MSSDL**). It is visualized as follow:

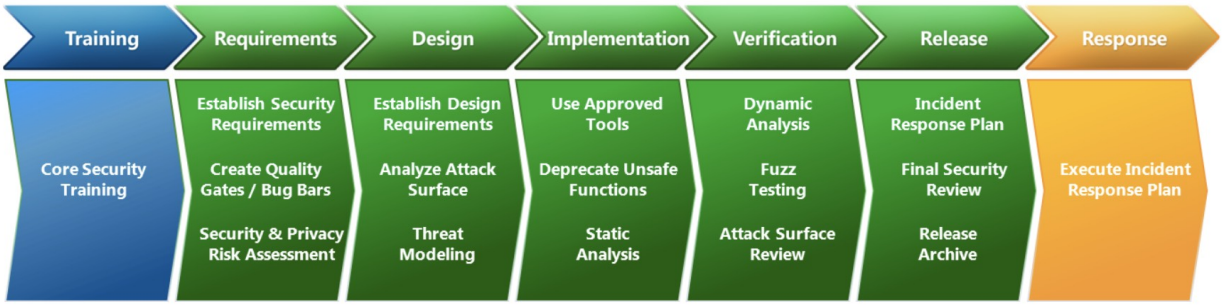

Figure 2: The Microsoft Security Development Lifecycle - Simplified

Figure 7 - MS SDL Lifecycle

The seven phases are:

- Training
- Requirements
- Design
- Implementation
- Verification
- Release
- Response

These are performed in a cyclic fashion where phases 2 through 6 are performed in epicycles (sprints) within the overall cycle (releases).

15. ISO/SAE 21434 Background Material

Note: Summary material regarding ISO/SAE 21434 can be found external to this document.

Road Vehicles - Cybersecurity Engineering is intended to address the cybersecurity aspects of electrical and electronic (E/E) systems within road vehicles. Its goal is to enable organizations to:

- define cybersecurity policies and processes
- manage cybersecurity risk
- foster a cybersecurity culture

The following clauses are within the scope of the ***AVCDL***:

- Continuous Cybersecurity Activities (8)
- Concept (9)
- Product Development (10)
- Cybersecurity Validation (11)
- Production (12)
- Operations and Maintenance (13)
- End of Cybersecurity Support and Decommissioning (14)

The following clauses are not within the scope of the ***AVCDL***:

- Cybersecurity Management
  - Organizational Cybersecurity Management (5)
  - Project Dependent Cybersecurity Management (6)
- Distributed Cybersecurity Activities (7)
- Threat Analysis and Risk Assessments Methods (15)

16. Reference Documents

This section contains a description of the reference documents relating to the ***AVCDL***.

16.1 Standards

1. **Cybersecurity Maturity Model Certification (CMMC)**
   https://www.acq.osd.mil/cmmc/docs/CMMC_ModelMain_V1.02_20200318.pdf
2. **Systems and software engineering - Software life cycle processes**
   https://en.wikipedia.org/wiki/ISO/IEC_12207
3. **Systems and software engineering - System life cycle processes**
   https://en.wikipedia.org/wiki/ISO/IEC_15288
4. **Secure Software Development for Autonomous Vehicles**
   https://www.sae.org/standards/content/iso/sae21434.d1/
5. **Systems Security Engineering - Capability Maturity Model (SSE-CMM)**
   https://www.iso.org/standard/44716.html
6. **Microsoft Security Development Lifecycle (SDL) - simplified implementation**
   http://download.microsoft.com/download/F/7/D/F7D6B14F-0149-4FE8-A00F-0B9858404D85/Simplified Implementation of the SDL.doc
7. **NHTSA Cybersecurity Best Practices for the Safety of Modern Vehicles**
   https://www.nhtsa.gov/staticfiles/nvs/pdf/812333_CybersecurityForModernVehicles.pdf
8. **Guidelines for the Creation of Interoperable Software Identification (SWID) Tags**
   https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8060.pdf
9. **Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations**
   https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-171r2.pdf
10. **NICE Cybersecurity Workforce Framework (NCWF)**
    https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-181r1.pdf
11. **Secure Software Development Framework (SSDF)**
    https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04232020.pdf
12. **Static Analysis Results Interchange Format (SARIF)**
    https://docs.oasis-open.org/sarif/sarif/v2.0/csprd02/sarif-v2.0-csprd02.pdf
13. **Systems Engineering Capability Maturity Model (CMM)**
    https://resources.sei.cmu.edu/asset_files/TechnicalReport/1993_005_001_16211.pdf
14. **Software Package Data Exchange (SPDX®) Specification**
    https://spdx.dev/wp-content/uploads/sites/41/2020/08/SPDX-specification-2-2.pdf
15. **Proposal for a Recommendation on Cyber Security**
    https://unece.org/DAM/trans/doc/2019/wp29grva/ECE-TRANS-WP29-GRVA-2019-02e.pdf

16.2 Secondary Documents

All secondary documents as named below are located relative to this file in the following folder:

```
./reference_documents/secondary_documents
```

General Information

- Ranked / Risked Threat Report [PDF]
- Threat Report [PDF]
- Understanding the Phase Product Dependencies Graph [PDF]
- Understanding Workflow Graphs [PDF]

Foundation Phase

- Code Protection Plan [PDF]
- Cybersecurity Monitoring Plan [PDF]
- Cybersecurity Requirements Catalog [PDF]
- Decommissioning Plan [PDF]
- Deployment Plan [PDF]
- Global Security Goals [PDF]
- Global Security Requirements [PDF]
- Incident Response Plan [PDF]
- List of Approved Tools and Components [PDF]
- Release Integrity Plan [PDF]
- Security Requirements Taxonomy [PDF]
- System to Track Training Participation [PDF]
- Threat Prioritization Plan [PDF]
- Training Catalog [PDF]

Requirements Phase

- Product-level Security Goals [PDF]
- Product-level Security Requirements [PDF]
- Requirements Phase Gate [PDF]

Design Phase

- Attack Surface Analysis Report [PDF]
- Design Phase Gate [PDF]
- Design Showing Security Considerations [PDF]
- Security Design Review Report [PDF]
- Threat Modeling Report [PDF]

Implementation Phase

- Build Process Documentation [PDF]
- Component / Version - Product / Version Cross-reference Document [PDF]
- Currently Used Deprecated Functions Document [PDF]
- Dynamic Analysis Report [PDF]
- Fuzz Testing Report [PDF]
- Implementation Phase Gate [PDF]
- List of Tools and Components Used [PDF]
- Secure Code Review Summary [PDF]
- Secure Development [PDF]
- Secure Settings Document [PDF]
- Static Analysis Report [PDF]

Verification Phase

- Penetration Testing Report [PDF]
- Updated Attack Surface Analysis [PDF]
- Updated Threat Model [PDF]
- Verification Phase Gate [PDF]

Release Phase

- Archive Manifest [PDF]
- Final Security Review Report [PDF]
- Release Phase Gate [PDF]

Operation Phase

- Cybersecurity Incident Report [PDF]
- Software Deployment Report [PDF]

Decommissioning Phase

- Decommissioning Report [PDF]

16.3 Working Material

All working material spreadsheets as named below are located relative to this file in the following folder:

```
./reference_documents/working_material
```

- AVCDL CMMC [XSLX]
- AVCDL mappings [XSLX]
- AVCDL roles and responsibilities [XSLX]

17. Continuous Improvement Progress Summary Example

The following table shows a possible representation for tracking the maturity of the ***AVCDL*** implementation:

AVCDL phase	requirement	name	implementation	CMM level
Foundation	1	Training	partially implemented	1
	2	Roles and Responsibilities	partially implemented	1
	3	Toolchain Support	partially implemented	1
	4	Definition of Security Requirements	partially implemented	1
	5	Protect the Code	implemented	2
	6	Ensure Release Integrity	not implemented	0
	7	Incident Response Plan	partially implemented	1
	8	Decommissioning Plan	not implemented	0
	9	Threat Prioritization Plan	not implemented	0
	10	Deployment Plan	not implemented	0
Requi	1	Definition of Security Requirements	partially implemented	1
	2	Requirements Gate	partially implemented	1
Design	1	Consider Security and Risk During Design	partially implemented	1
	2	Design Review	partially implemented	1
	3	Attack Surface Reduction	not implemented	0
	4	Threat Modeling	partially implemented	1
	5	Design Gate	not implemented	0
Implementation	1	Use Approved Tools	implemented	2
	2	Configure Process for Security	partially implemented	1
	3	Configure Secure Settings by Default	partially implemented	1
	4	Reuse Well-Secured Software	implemented	2
	5	Use Secure Coding Practice	partially implemented	1
	6	Deprecate Unsafe Functions	partially implemented	1
	7	Static Analysis	implemented	2
	8	Dynamic Program Analysis	partially implemented	1
	9	Security Code Review	partially implemented	1
	10	Fuzz Testing	partially implemented	1
	11	Implementation Gate	not implemented	0
Verification	1	Penetration Testing	implemented	1
	2	Threat Model Review	not implemented	0
	3	Attack Surface Analysis Review	not implemented	0
	4	Verification Gate	not implemented	0
Release	1	Final Security Review	not implemented	0
	2	Archive	partially implemented	1
	3	Release Gate	not implemented	0
Operation	1	Identify and Confirm Vulnerabilities	partially implemented	1
	2	Assess and Prioritize the Remediation	not implemented	0
	3	Root Cause Vulnerabilities	partially implemented	1
	4	Secure Deployment	not implemented	0
De	1	Apply Decommissioning Protocol	not implemented	0

Table 7 - Maturity Tracking Example

18. AVCDL Product Dependencies

The following diagrams shows the dependencies between the products from the AVCDL process requirements. These are presented as a dependency graph (reversed) using ISO standard flowchart symbols.

For more information regarding the interpretation of these diagrams, refer to the **Understanding the Phase Product Dependencies Graph** secondary document.

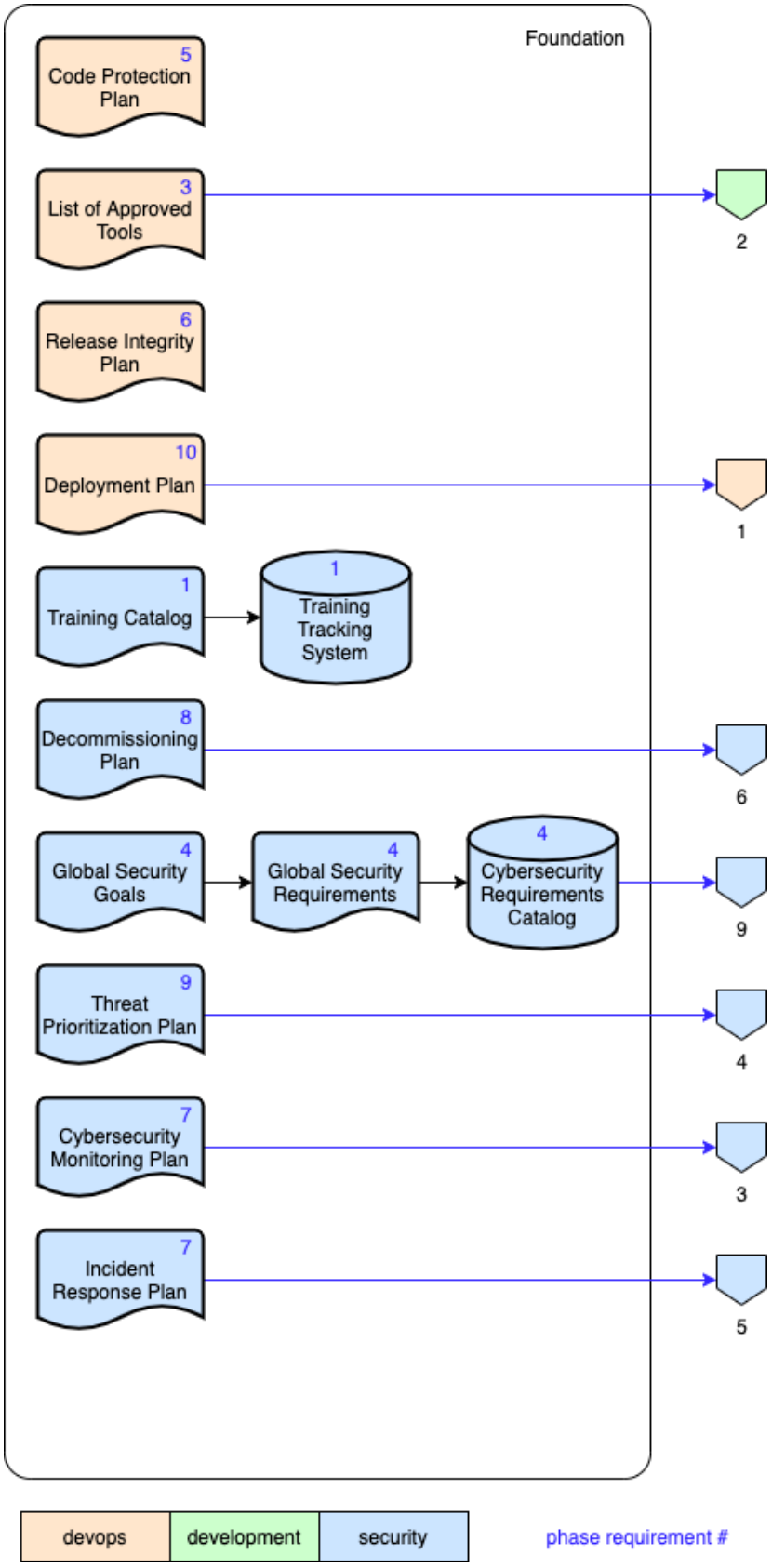

Figure 8 - AVCDL product dependencies – foundation phase

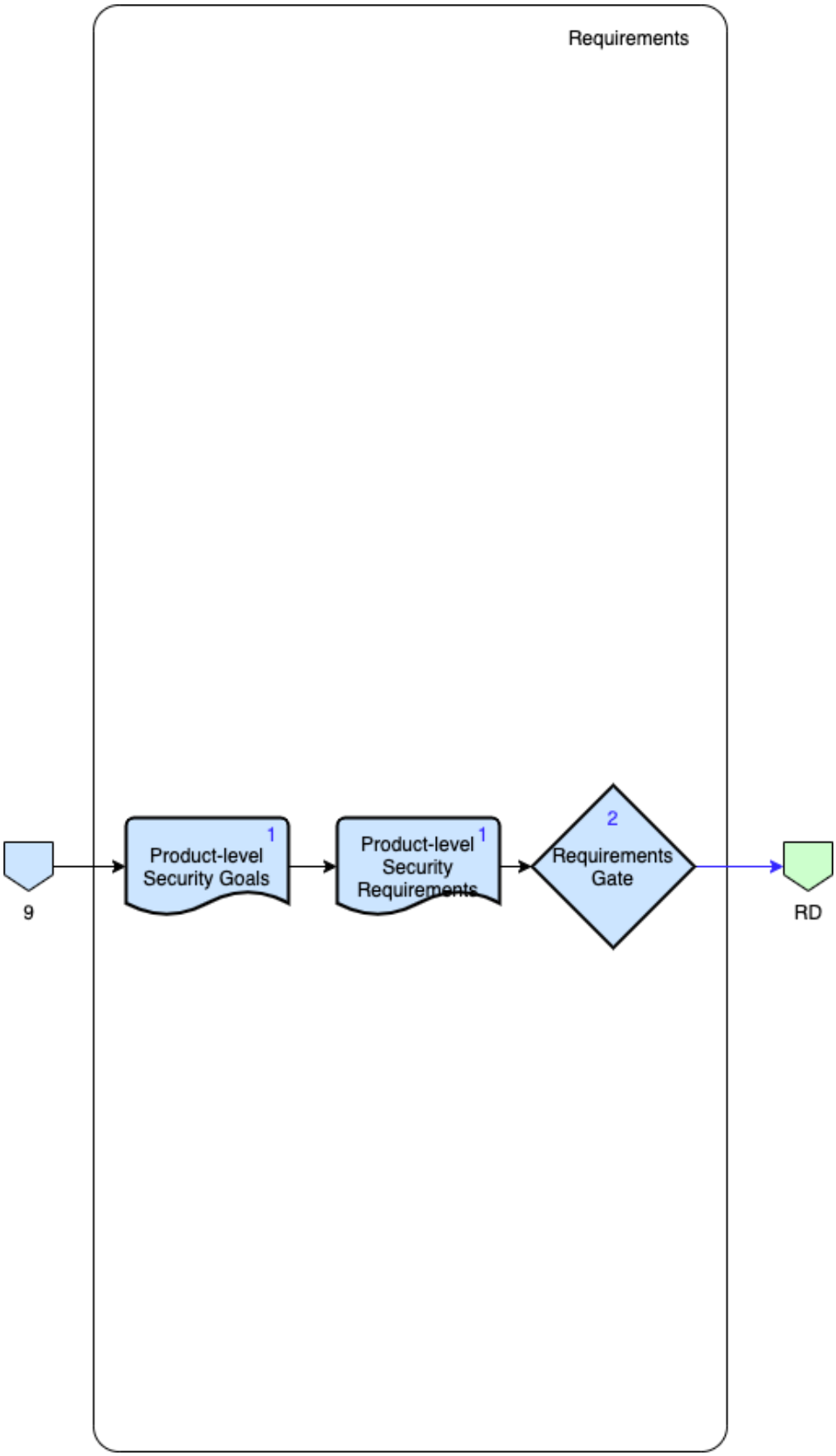

Figure 9 - AVCDL product dependencies – requirements phase

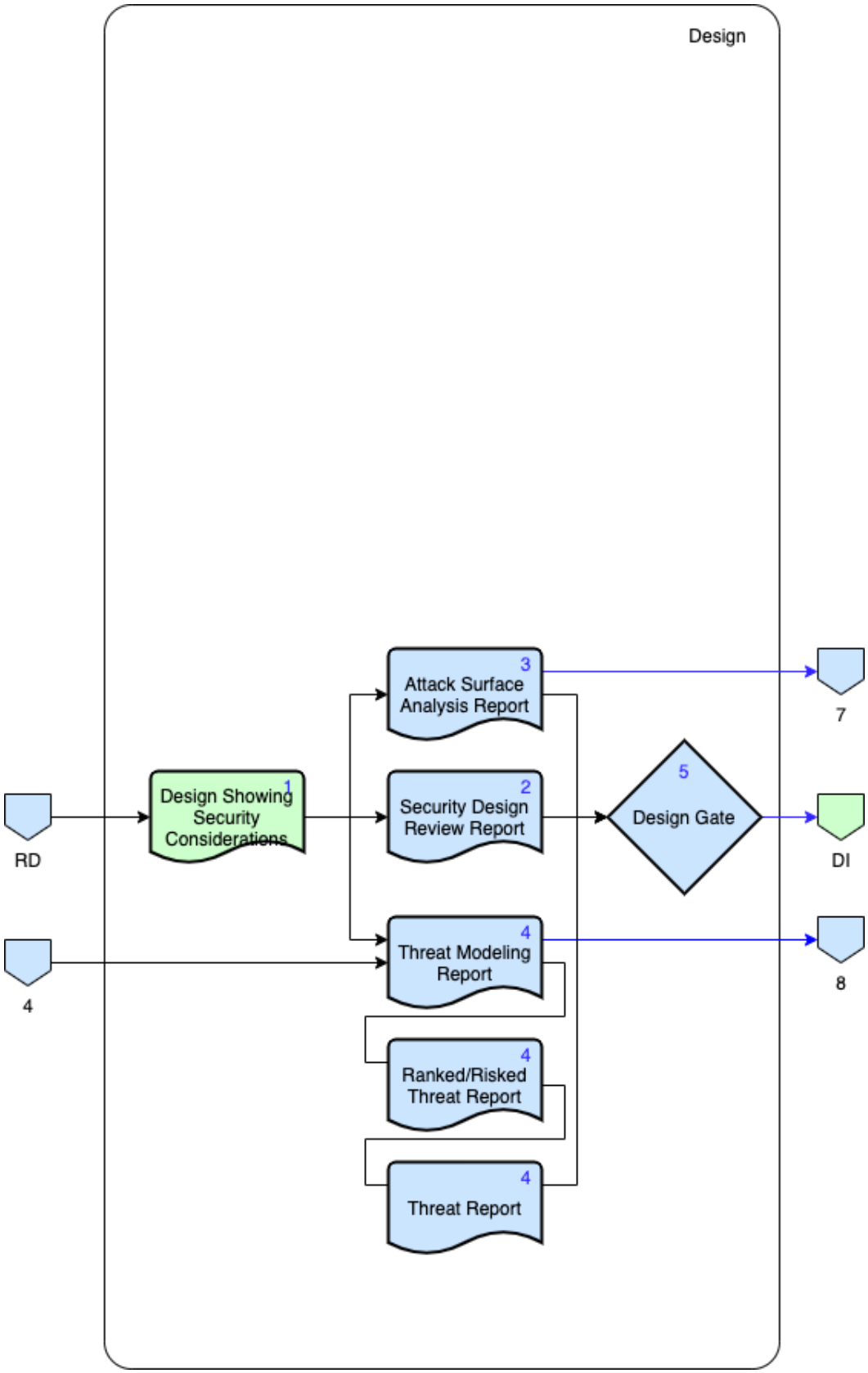

Figure 10 - AVCDL product dependencies – design phase

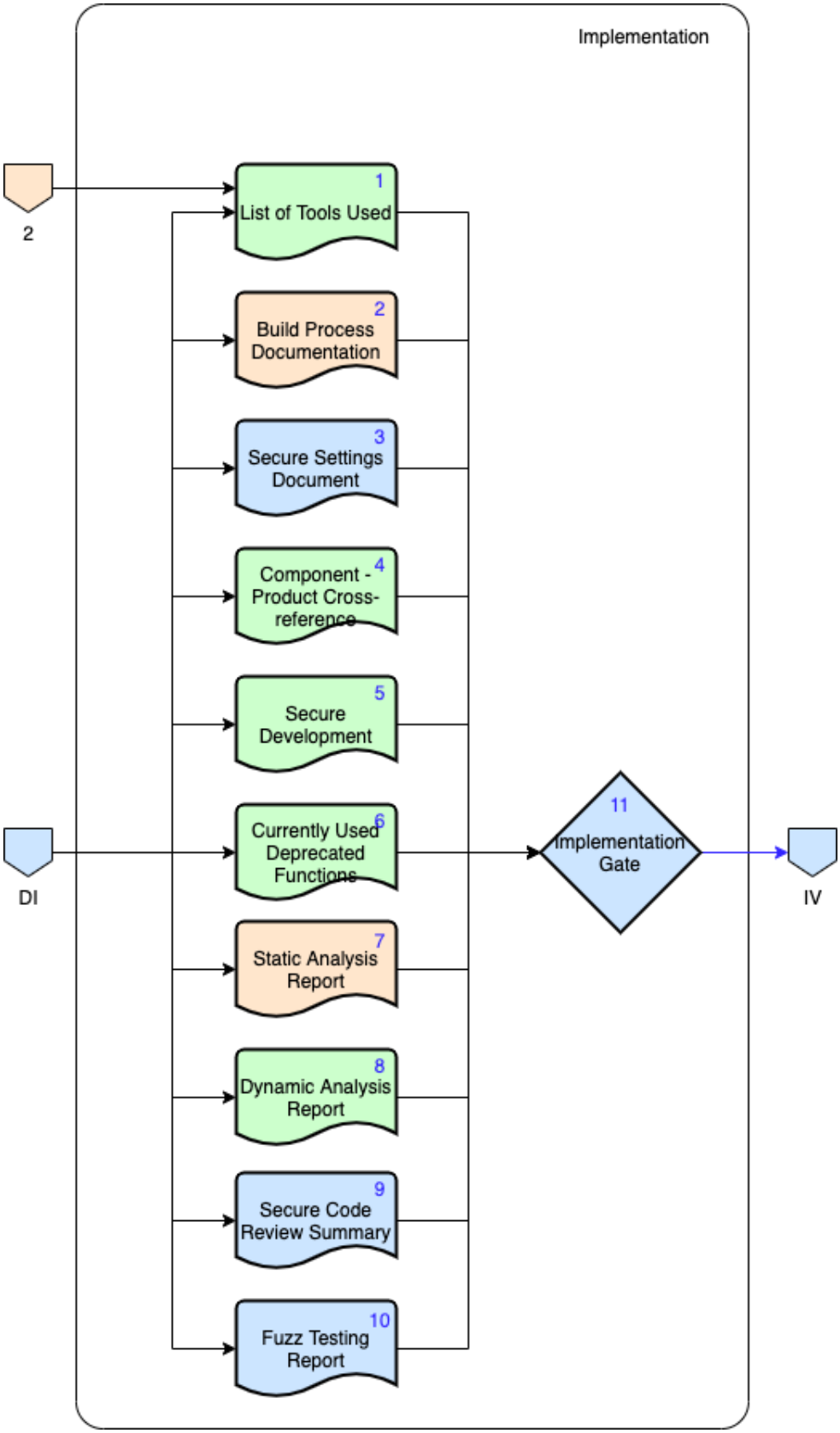

Figure 11 - AVCDL product dependencies – implementation phase

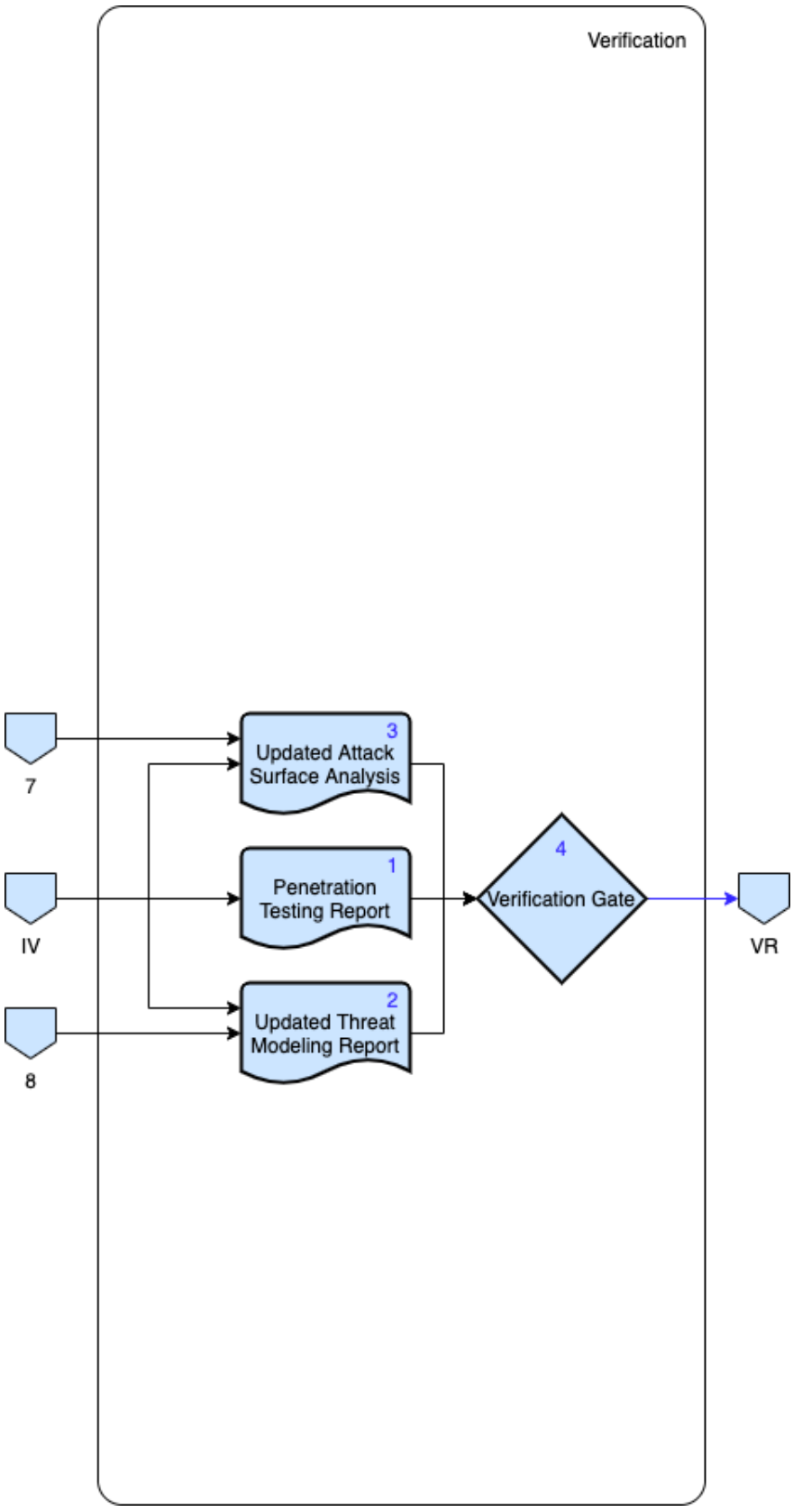

Figure 12 - AVCDL product dependencies – verification phase

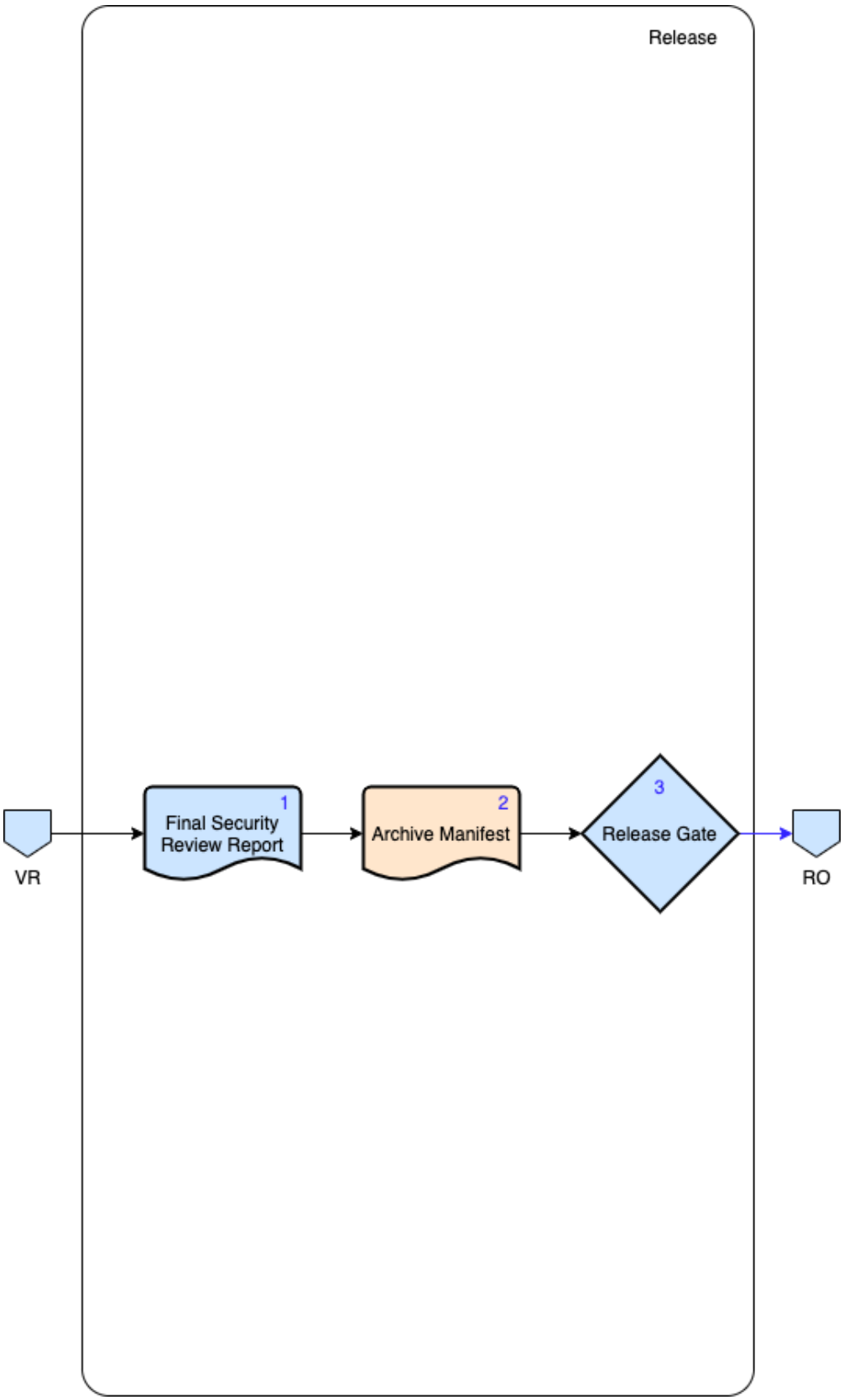

Figure 13 - AVCDL product dependencies – release phase

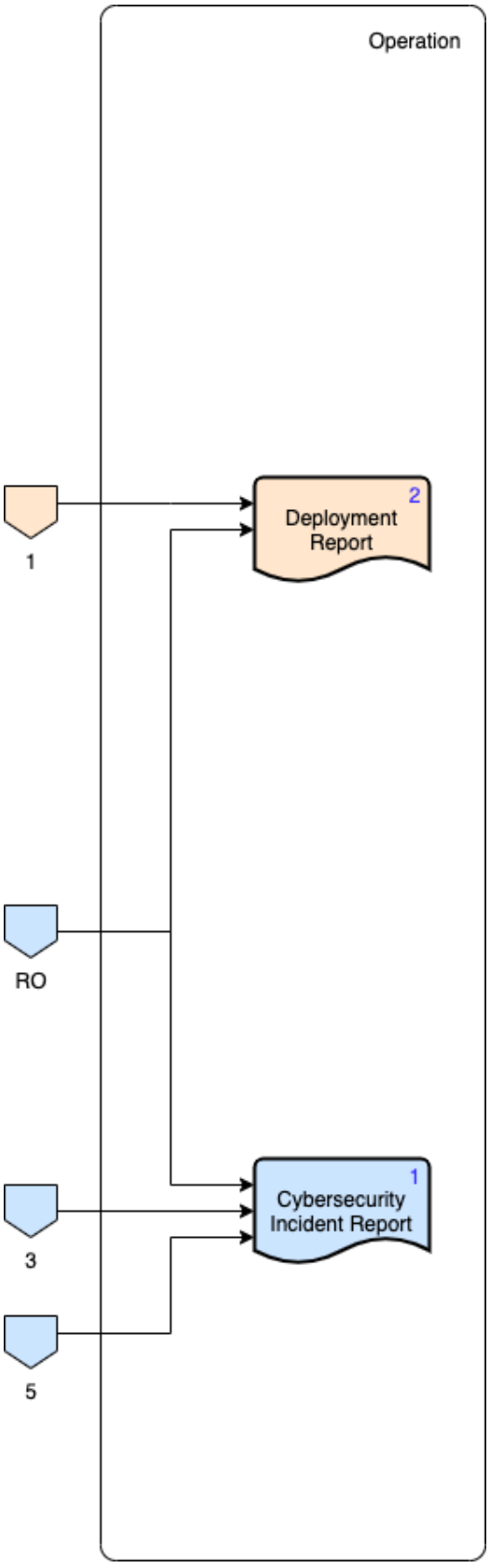

Figure 14 - AVCDL product dependencies – operation phase

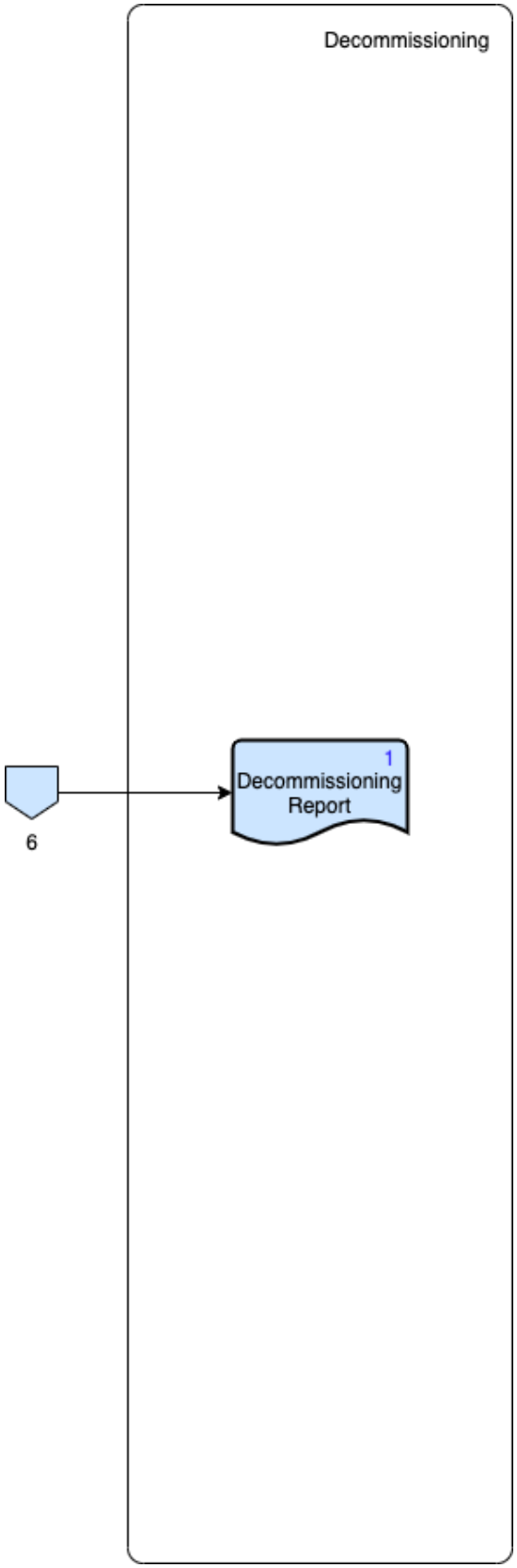

Figure 15 - AVCDL product dependencies – decommissioning phase

19. AVCDL Training Path

The following figure shows the dependencies between the training associated with the ***AVCDL*** phase requirements. It is presented as a visual reference for assumed prior knowledge.

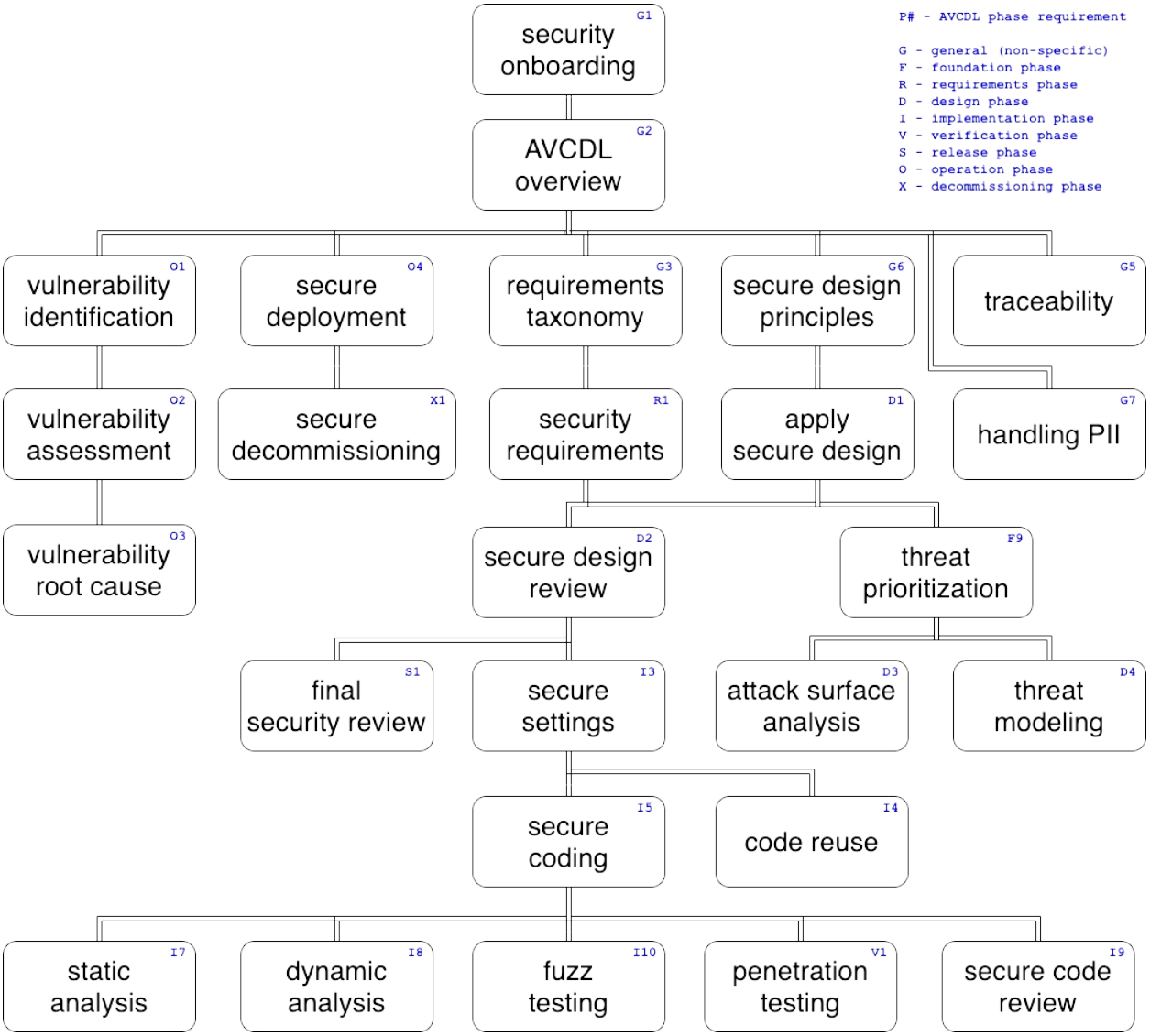

Figure 16 - AVCDL Training Path