Aligning the Organization with the AVPDL

Charles Wilson, Principal Engineer, Cybersecurity Development Lifecycle Practice

11/3/20 10:47:00 AM

Category: security-governance

Tags: security, cybersecurity, autonomous vehicles, certification, ISO 21434, ISO 15288, ISO 26262, ISO 12207, AVPDL, AVCDL

What's in a Name?

Standards give us a way to ensure quality and compatibility. They give practitioners a common language for communicating. They may come from a single company, an industry group (IEEE, SAE), or a national or international standards body (NIST, ISO, UN).

Over time, standards take on lives of their own and can become the foundations of other standards. Sometimes, this leads to the nomenclature of one standard being completely different from that of another when referring to the same thing because the viewpoints of the groups differ.

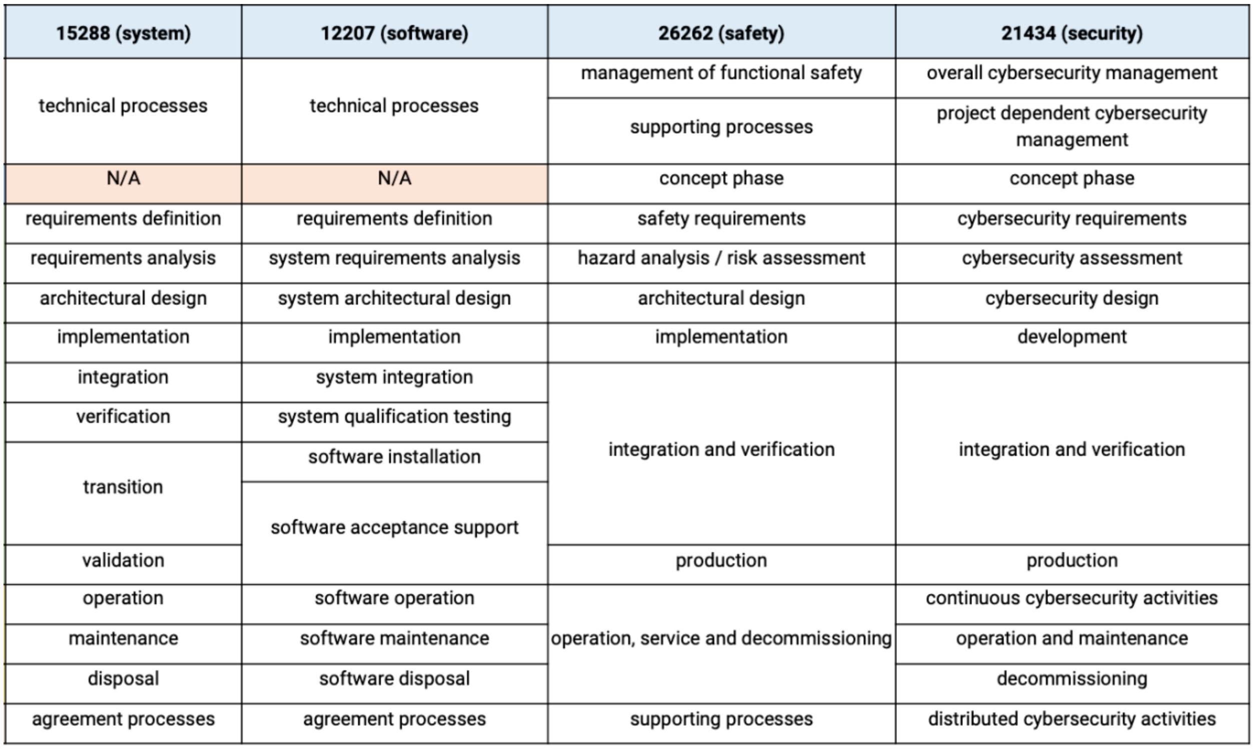
This typically isn't a problem. You don't see many problems between mathematicians and electrical engineers even though one represents complex numbers using **'i'** and the other **'j'**. (Electrical engineers use **'i'** to denote electrical current. This difference bedevils electrical engineering and mathematics students taking courses across disciplines.)

There are times when you need to be able to communicate across multiple groups using disparate, but similar, nomenclature. The Autonomous Vehicle Product Development Lifecycle (**AVPDL**) is intended to establish a nomenclature for communicating about the development lifecycle.

In the autonomous vehicle space, four primary standards govern the product development lifecycle. These are:

Standard	Description
ISO 15288	Systems Development Lifecycle
ISO 12207	Software Development Life Cycle (SDLC)
ISO 26262	Road Vehicles – Functional Safety
ISO 21434	Road Vehicles – Cybersecurity Engineering

Each of these includes phases for governance, planning, requirements, design, implementation, integration, verification, release, operation, maintenance, and decommissioning. They also include provision for interactions with suppliers. The following table shows how each of these standards is organized.



15288 (system)	12207 (software)	26262 (safety)	21434 (security)
technical processes	technical processes	management of functional safety	overall cybersecurity management
		supporting processes	project dependent cybersecurity management
N/A	N/A	concept phase	concept phase
requirements definition	requirements definition	safety requirements	cybersecurity requirements
requirements analysis	system requirements analysis	hazard analysis / risk assessment	cybersecurity assessment
architectural design	system architectural design	architectural design	cybersecurity design
implementation	implementation	implementation	development
integration	system integration	integration and verification	integration and verification
verification	system qualification testing		
transition	software installation		
	software acceptance support		
validation		production	production
operation	software operation	operation, service and decommissioning	continuous cybersecurity activities
maintenance	software maintenance		operation and maintenance
disposal	software disposal		decommissioning
agreement processes	agreement processes	supporting processes	distributed cybersecurity activities

Unfortunately, they do not agree on what the phases are named or how they are apportioned.

On the face of it, this might not seem like much of an issue. Sadly, since different groups must adhere to different standards, interfacing and synchronization between them is difficult. This is especially true when the issue of gates comes up.

Have Your Tickets Ready

As the showrunners of the product, the product management organization (PMO) uses gates to keep things on track. Gates provide synchronization points needed to keep the various groups properly engaged as development proceeds. At the gate, all groups verify that the phase activities required have been completed.

This might seem like a minor housekeeping issue; and it would be if the various certification processes did not require traceability. As a result, a given component of a system must have completed all the design activities before beginning implementation activities.

Without gates development and safety might have all of their activities completed, but security might not. As a result, development might begin implementation. Unfortunately, if security discovers a design flaw during its review, development might have to undertake a major redesign which could make any implementation unusable.

Speaking the Same Language

This brings us back to the nomenclature. If you have multiple different phases (stages) within the various lifecycles, all with different names, it becomes quite difficult to manage discussions between the various groups.

To solve this problem, the **AVPDL** was created. It provides a common set of names and phases allowing for a common language. This in turn makes the creation, communication, and management of phase gates possible. The following table shows the AVPDL and how it aligns the various standards.

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

There are eight phases and two processes. Why? Because these standards incorporate both development lifecycle and governance elements. Let's look more closely.

Governance

The two governance elements of the **AVPDL** are **organization processes** and **supplier processes**. These deal with the internal and external aspects of orchestrating the product's creation. They include project management, documentation control, policy, supplier interface agreements, and the like. These are all things which are handled at the corporate level and span multiple products.

Lifecycle Phases

The eight lifecycle phases are divided into three distinct groups:

- Foundational
- Intra-developmental
- Post-developmental

Foundational Phases

There is only one phase in the foundational phases, that being the **foundation phase**. This is a phase which cuts across the entire lifecycle. It includes planning for activities that will occur in later phases and preparatory activities whose products are refined as the lifecycle progresses.

Intra-developmental Phases

The five intra-developmental phases are what most people think of when they consider development. They are **requirements**, **design**, **implementation**, **verification**, and **release**. These are fairly well-understood, so I won't address them here.

Post-developmental Phases

The two post-developmental phases are **operation** and **decommissioning**. These are the one's dealing with activities that occur after the product has shipped. The operation phase deals with management of products in the field (deployment, maintenance, use, update, etc.), and product incident response. The decommissioning phase encompasses both product repair (RMA) and decommissioning (removal of product from operational use) activities.

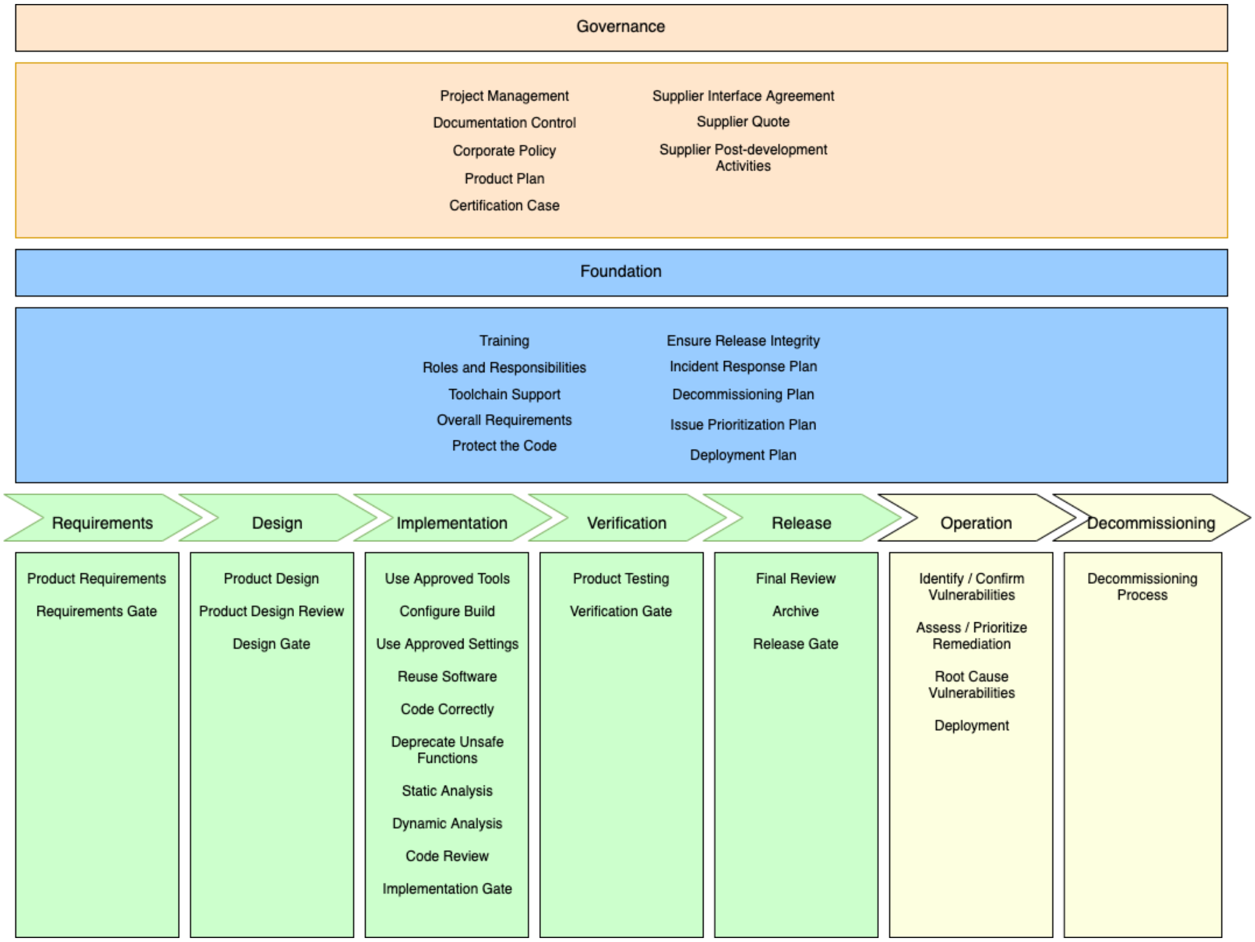
Phase Gates

In order to synchronize the development activities, we need inter-phase gates. But we don't need them between all the phases. Governance processes aren't gated, as they are presumed to be in place before the project starts. Document control systems, legal departments, build systems, source code control, and data privacy policies are all "create once and reuse" elements.

Additionally, once the product has been released, there are no gates required. This leaves the intra-developmental phases. Here we gate exit from each phase.

AVPDL-compliant Development Lifecycle Example

We can visualize an example development lifecycle which adopts the **AVPDL** as follows:



Here we see the governance, foundation, intra-developmental, and post-developmental processes and phases. The various processes and phase requirements shown are for example only. Your actual requirements will depend upon the needs of your organization's individual groups.

Attaining Alignment

Now that we've established a framework for alignment, how do we use it? There are two steps to achieving alignment with the **AVPDL**.

Step 1 – Phase Requirements

The first step in the process of **AVPDL** alignment is the creation / documentation of phase requirements specific to each group within the organization. These most likely already exist but are not necessarily documented either formally or in a way which follows the **AVPDL** structure.

Step 2 – Phase Gates

The second step toward **AVPDL** alignment is the determination and documentation of phase gate requirements. This is the element of the **AVPDL** allowing for coordinated project management leading to certification of a safety-critical product. Once all groups provide this information, the product management team will be able to efficiently orchestrate the progress of the product through its lifecycle.

Cybersecurity Implications

From the standpoint of cybersecurity, the **AVPDL** provides a mechanism to communicate our needs to other teams and coordinate our efforts with the organization as a whole. To support this, we have created a cybersecurity-specific, certification-capable development lifecycle aligning to the **AVPDL**. It's called the Autonomous Vehicle Cybersecurity Development Lifecycle (**AVCDL**). It will be the subject of future posts.

References

1. **ISO 26262 Road Vehicles – Functional Safety**
   https://en.wikipedia.org/wiki/ISO_26262
2. **ISO/SAE DIS 21434 Road Vehicles – Cybersecurity Engineering**
   https://www.iso.org/standard/70918.html
3. **ISO/IEC/IEEE 15288 Systems and Software Engineering – System Life Cycle Processes**
   https://en.wikipedia.org/wiki/ISO/IEC_15288
4. **ISO/IEC 12207 Systems and Software Engineering – Software Life Cycle Processes**
   https://en.wikipedia.org/wiki/ISO/IEC_12207