# Title

I'm Charles Wilson, senior principal engineer at motional responsible for cybersecurity development lifecycle practice.

This presentation will cover background information about the Autonomous Vehicle Cybersecurity Development Lifecycle or AVCDL.

# Introduction

[break]

At the end of August 2021, the international standard covering road vehicle cybersecurity engineering (ISO/SAE 21434) was released.

This technical standard is the basis for international vehicle regulations in the area of cybersecurity.

[break]

What does this mean for vehicle development?

How are we going to achieve compliance with this standard?

I was hired by Motional to look at this very problem.

[break]

The answer to this question is embodied in the Autonomous Vehicle Cybersecurity Development Lifecycle (or AVCDL for short).

Before getting into the details of the AVCDL, let’s look at the various elements that led to its creation.

# Lifecycle Basics

First, let’s cover some lifecycle basics.

[break]

At the root of everything are the policies of the company.

These are the **why**.

Policies are the motivating factor for everything that the company does.

[break]

The second element are the processes.

They're the what, who, and when of the activities that achieve the policies.

[break]

Finally, we have procedures which tell us how a specific process is going to be implemented.

[break]

A lifecycle is the collection of processes that guide how a product is created, used, and eventually disposed of.

# Standards and Regulations Ecosystem

Next, let's look at the standards and regulations ecosystem in which our lifecycle is going to operate.

[break]

Underpinning all of the standards and regulations is ISO 9001.

This is the quality management system (or QMS) and acts as the basis for compliance in any regulated industry.

Within the automotive realm the complementary standard IATF 16949.

[break]

Within our context the next relevant standard is ISO 15288.

This is the system development lifecycle.

It’s not actually a lifecycle, but rather describes what processes need to be undertaken in the context of systems engineering in order to create a product.

The actual implementation of a lifecycle is left to the individual organization.

[break]

Above 15288 is ISO 12207.

This is the software development lifecycle and just like 15288 is not actually a lifecycle.

The processes specified in 12207 are extensions of those specified in the systems development lifecycle and are specialized to software development.

[break]

Next, we have a pair of technical standards specific to the automotive space.

The first ISO/SAE 21434 covers cybersecurity engineering for road vehicles.

The second ISO 24089 covers software update.

These standards are not an extension of either 12207 or 15288, but rather presume that these two standards are in place in the same way that 15288 presumes that ISO 9000 is in place.

[break]

Next, we'll switch from technical standards to regulations.

In this case, these are the UN regulations governing road vehicles.

Specifically, regulation 155 covering the cybersecurity management system (or CSMS) and regulation 156 covering the system's update management system (or SUMS).

These are paired with their underlying technical standards 21434 and 24089, respectively.

[break]

Atop these two regulations sit another series of regulations, all which have dependencies on them.

Here we see, UN R157 (or ALKS) covering auto lane keeping systems.

The expectation is that there is a conformance to R155 and R156.

[break]

Finally, there is a proposal currently under consideration covering automated driving systems (or ADS) which will require conformance to R155 and R156.

# How the Standards Line Up

Before creating a lifecycle, it's important to ensure that the standards that you’ll be working with actually have alignment.

So, let's go through that exercise.

[break]

Here we see the activities specified in ISO 15288, the systems development lifecycle.

We can consider this standard in three chunks:

technical processes,

requirements through disposal,

and agreement processes for interactions with suppliers and customers.

[break]

When we put ISO 12207 next to 15288, we can confirm that it is, as mentioned previously, an extension into the software realm.

Where it differs is that what 15288 refers to as transition and validation activities 12207 refers to as software installation and software acceptance support.

Other than that, the activities differ in name only.

[break]

Let’s now add ISO 26262 which covers functional safety of road vehicles.

You'll note the introduction of the concept phase.

Additionally, the technical processes have been split into management of functional safety and supporting processes.

In the middle, activities are grouped together reflecting the nature of safety within product development.

Specifically, the integration, verification, and transition phases are grouped together as integration and verification.

Validation has become production and the operation, maintenance and disposal phases are group together into a single phase.

[break]

Finally, we'll add ISO/SAE 21434.

It resembles 26262, except in the areas of operation, maintenance and decommissioning.

Those areas more closely align with 15288 and 12207's breakdown.

Additionally, 21434 is much more specific, saying distributed cybersecurity activities rather than agreement or supporting processes.

[long pause]

If you have an organization creating products subject to regulation, you'll need to synchronize your various groups and their activities.

You're going to want them to speak a common language.

You don't want to have cybersecurity talking about development while everyone else is talking about implementation.

How do you enable multiple groups with disparate nomenclatures to communicate, share information, and have synchronization points without confusion?

[break]

We're going to do that by creating a mapping between the processes that the vendor uses and the ones that the customer uses.

# AVPDL – Autonomous Vehicle Product Development Lifecycle

We address this by introducing the Autonomous Vehicle Product Development Lifecycle.

The AVPDL is a simple but powerful construct which lays out a set of generic phases and processes.

It allows groups to use a common language to clearly communicate with other groups when discussing activities in various phases or processes.

[break]

We still have the problem that things don’t quite line up, and we're dividing and grouping various things.

But you'll notice that where we are introducing additional separation or grouping, it's across phases that lend themselves to this type of decomposition.

For instance, the separation of operations and decommissioning is common among three of these standards and should not cause any great difficulty for those in safety who are using 26262.

[break]

You'll note that I said that there were processes and phases within the AVPDL.

The processes are things which are managed primarily at the organizational level rather than by any one group.

Project management, document management and training are examples of organizational processes.

[break]

At the bottom, we see supplier processes, these are also primarily managed at the organizational level but with contributions being made by each of the specific groups.

For instance, within the context of 21434 we have to have a cybersecurity interface agreement to go alongside the functional interface agreement that exists between the supplier and the customer.

[break]

This brings us to the lifecycle phases.

These activities directly pertain to the creation, operation and decommissioning of products requiring the coordinated efforts of multiple groups.

The choice of divisions is based on what the largest similar activity groupings are.

The motivation for the scope of activities in a phase is that you’re either going to need a gating activity because there is a dependency on previous phases that would prevent you from moving to the next phase without having verified that you've taken care of everything that you need to in the current one

Or (in the case of the operation phase) that the activities cannot be otherwise grouped.

And so, what we see here is that green phases are fundamental gating spots for project management where we want to make sure that we’ve completed all activities across all groups before proceeding.

We also want to do this because we have to ensure that we have traceability as required within a regulated industry.

Traceability will be covered in greater detail later in this presentation.

[pause]

The foundation phase doesn't have a gate because its activities are ongoing throughout the entirety of the product lifecycle.

Additionally, once you've created the processes specified in the foundation phase, they can be reused future projects.

Similarly, the operations phase isn't gated because operations are ongoing.

The activities used to handle issue tracking and fleet management will be the same regardless of the product within the operational phase.

All of the same structure and frameworks apply equally.

Lastly, decommissioning isn’t gated because it can occur at any time and has no other phase dependencies. Individual products are sent in for repair, or an entire series may be retired.

# Reference Sources

Now that we've established the phases that we're going to use for this lifecycle,

let's consider some specific sources that we're going to use in order to determine what the activities in each of these phases are going to be.

[break]

First and foremost is ISO/SAE 21434, the road vehicle cybersecurity engineering standard.

This is our base technical standard and it’s going to serve as the most important reference used in the creation of the AVCDL.

[break]

As previously mentioned, ISO 26262, the road vehicle functional safety standard serves as our guide because although we can create an arbitrarily secure product,

what we need to do is have enough security to ensure that the vehicle is going to be operating in a safe way without undue impact on functional requirements.

[break]

Another source from the standard side is UN regulation 155 dealing with cybersecurity management systems.

[break]

In the area of best practices we’ll be utilizing the seminal work The Security Development Lifecycle developed by Microsoft roughly 20 years ago.

[break]

A relatively recent, but very informative work is the SSDF from NIST.

This document, entitled Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework

is a compendium of best practices and has many informative elements which are not collected anywhere else.

[break]

Rounding out our primary reference documents is NIST special publication 800-181, National Initiative for Cybersecurity Education Cybersecurity Workforce Framework (also called the NCWF).

It provides a definitive list of all of the knowledge, skills and activities that each worker in the cybersecurity realm needs in order to effectively fulfill their job.

# How Sources Inform the AVCDL

Now, let's take a look and see how these sources inform the construction of the AVCDL.

[break]

Let's start by taking the AVPDL phases that specify the lifecycle and stretch them out linearly.

Note the legend in the lower-right corner showing that the foundational elements are in blue, the developmental in green, and the post-developmental in yellow, all other elements in white are general information.

[break]

Let's start with the MSSDL.

As you can see, there's a fairly direct mapping between the MSSDL and the phases as described by the AVPDL.

Two deviations are that the requirements phase in the MSSDL specifies activities for both the requirements and the design phases.

Additionally, the release phase speaks to not only the release phase in the AVPDL, but also to the foundation phase.

[break]

The SSDF is a bit less granular, divided into only four topics.

These are protecting the software, preparing the organization, producing well secured software, and respond to vulnerability reports.

This material covers everything except for the product release and decommissioning phases.

[break]

You'll note in the legend that we use a red rounded rectangle for 26262.

There are however no processes that speak directly to cybersecurity activities.

It's more an alignment of materials.

The requirements and design phases are spoken of in the third volume of 26262.

The implementation through verification phases are in volumes four through six.

Finally, the release, operation, and decommissioning phases are covered in volume seven.

[break]

This brings us to ISO/SAE 21434.

As you can see, the mapping is a bit of a mess.

There are multiple phases that are spoken to by the various clauses within 21434.

This is unfortunate, but that's just the way it's structured.

As you can see, there are many things which speak to activities which should be addressed in the foundation phase.

There are also activities which speak to multiple phases.

This makes for the direct usage of 21434 very difficult in terms of an implementation.

In fact, speaks to why it's useful to have a lifecycle as is the assumption by any of these standards, because the lifecycle speaks to the direct set of processes that an organization would actually use, as opposed to the way that these are organized from the standpoint of the standard or a regulatory document.

[break]

And here's a quick view to show you all of the sources combined.

As you can see, the various sources all contribute something differently and they all arrange their information in different manners.

The AVCDL harmonizes all of this information, and in fact, provides something that organizations can implement without having to worry about how any of these sources in particular speak to the material.

# ISO/SAE 21434 Work Products and Requirements

Let's look briefly at how we take one of these standards, specifically ISO/SAE 21434 and use it to inform what we should have in the AVCDL.

Here's a sheet that was created to bring together the 21434 work products and their specific requirements.

Note that there's a column labeled AVCDL phase.

This was used to chunk the information and place the specific requirements into the appropriate AVCDL phase buckets.

# ISO/SAE 21434 – AVCDL Mapping

Using the information from 21434 and other sources, a list of phase requirements for the AVCDL was created, these are shown across the top.

On the left are the 21434 requirements.

What we're able to then do is to show within the field how particular AVCDL phase requirements satisfy particular 21434 requirements.

Additionally, we show where we provide partial support because those requirements are not entirely within the purview of cybersecurity requiring other group or organizational level inputs.

# AVCDL Framework

Here’s the result, the AVCDL framework.

[break]

As you can see, from the AVPDL we have the identified phases for the lifecycle foundation, requirements, design, implementation, verification, release, operation, and decommissioning.

[break]

Within the developmental phases are five gates used to synchronize the AVCDL with the other groups’ lifecycles being used in the implementation of the product.

[break]

As you can see, the foundation phase spans all other phases.

We can see why from the contents

* Training
* roles and responsibilities,
* tool chain support
* overall requirements,
* protection of the code,
* ensuring release integrity,
* incident response plan,
* decommissioning plan,
* threat prioritization plan
* and deployment plan

These are all global level things within the context of cybersecurity and apply to every release of every product.

In fact, having and updating them at this level allows us to have uniformity across the product.

[break]

Within the developmental phases, we find a normal distribution of the processes.

In requirements, we have only one.

We get more in the design.

The bulk of our activities within the implementation.

Verification looks back on the earlier processes.

And finally, we have release, where our only processes are final review and archiving.

This gives us a sense of where we're going to be putting the bulk of our cybersecurity activities.

[break]

The last section of the framework contains the post-production phases of operation and decommissioning.

Note that in operation, we're doing a lot of work which is related to operational technology and also deployment.

In decommissioning, we have decommissioning processes because unlike many other groups involved in the vehicle’s lifecycle,

we have cybersecurity relevant materials in the various elements of the vehicle that needs to be addressed if that element is taken out of service, either for full decommissioning or simply for RMA.

# Design Deficiencies vs Implementation Defects

Let's take a moment to consider how the AVCDL addresses our two major concerns within cybersecurity, those being design deficiencies and implementation defects.

[break]

A classic way to approach cybersecurity within a product is by assessing the product after it’s built.

This can be done using traditional penetration testing or earlier on in the process with extensions of what would normally be considered compiler features such as static analysis and dynamic analysis

as well as more sophisticated after-the-fact testing, such as using the analog to unit testing, fuzz testing.

Additionally, you can have security code reviews alongside traditional functional code reviews.

These address implementation defects where the system itself has implemented cybersecurity incorrectly.

[break]

A more sophisticated approach would allow us to address design deficiencies,

where the elements of the system were not designed taking cybersecurity into consideration.

Here we look at things like secure designer reviews, attack surface analysis and threat modeling.

You'll note that there's a duel to these last two processes in the verification phase where we perform a threat modeling review and an attack surface analysis review.

This allows us to ensure that those deficiencies that we identified in the design phase were in fact corrected in the verification phase.

[break]

It's the combination of these two approaches which address both the design efficiencies and the implementation defects that give us the highest level of certainty that we've managed to attain sufficient cybersecurity rigor.

# V-model Overlay

And since this is a lifecycle which works within the context of organizations using the V-model, the AVCDL can in fact show that this model is addressed.

Now having a V on the on the diagram really doesn't get into how this works.

And so, let's look in a more detailed fashion.

# V-model View

Since the AVCDL does not prescribe a particular implementation methodology, an elaboration document has been provided to help those who use this model understand how it relates to the AVCDL.

In this case, that elaboration document is entitled Understanding Verification and Validation in an AVCDL Context.

On the left, is a portion of a diagram showing how particular artifacts, created in earlier stages, are verified in later stages.

In this case, we can look at the threat modeling report and attack surface analysis report created in the design phase and then later reviewed in the verification phase.

Additionally, you can see how particular downstream activities rely on various upstream activities.

# Cyclic View

The AVCDL can also be applied within the context of a cyclic developmental implementation.

If we take that on a large scale, we go from requirements through to release and operation in a cycle anchored by the foundation phase, which applies at all times.

Finally, we have an exit from the cycle when we decommission the product.

Cyclic View - Detail

If we zoom in, we can see how the AVCDL phase requirements overlay onto this cyclic view.

You'll note that the transitions between the five developmental phases and their following phases shows the phase gate that's associated with them.

Cyclic Feedback

The topic of cyclic feedback which is core to a cyclic implementation methodology is described in the Understanding Cybersecurity Risk Freshness in an AVCDL Context elaboration document.

As we can see on the diagram on the left, the implementation phase has feedback which channels through the threat prioritization process and eventually into the issue tracking system.

This feedback can return to the implementation phase directly, or to the design or the requirements phases.

And so, although we generally have forward progress cyclically, we also have retrograde events which may be forced when issues arise.

[pause]

It’s important to note that adoption of a cyclic implementation methodology in a large-scale system subject to regulatory constraints

requires that the system be decomposed to a level of granularity which more readily allows for this type of cyclic feedback.

Traceability

When creating safety-critical cyber-physical systems, we need to be able to establish traceability.

Here's a portion of a diagram showing how traceability is achieved within the AVCDL.

This diagram gives us a way to visualize how traceability can be established for any artifact within the AVCDL.

The large rounded rectangles represent the various AVCDL phases.

Shown are the foundation, requirements, design, and implementation phases.

At the bottom left is a legend.

The color coding indicates the group responsible for each of the various artifacts and activities.

The dependencies are readily apparent.

Consider the design gate in the design phase.

To get to that point, you have to have created the attack surface analysis report, the secure design review report, the threat modeling report, the ranked and risked threat report, and the threat report.

Then, and only then can you say you’re completed the design phase activities.

You'll notice that one line pushes back all the way through to the global security goals and to be able to produce the threat modeling report, you can trace back via connector four which goes back to the threat prioritization plan.

Any artifact created within the AVCDL has similar traceability,

so that nothing is left up to a chance in terms of how it's produced, how it's consumed, and where it's come from.

Typical Phase Requirement

Here we see a typical phase requirement as shown from the AVCDL primary document.

Let's look at the areas that each of these phase requirements will provide.

[break]

First at the top, we see the title.

It's broken down in the following areas.

You'll see the section within the AVCDL.

In this case, 9.3.4, the phase requirements name threat modeling and then a unique identifier which is used both within the AVCDL documentation and also externally in this case, this is AVCDL-Design-4, which says that it's from the design phase and is the fourth phase requirement.

[break]

Next, we have the owner section.

This is broken into two areas, first is the group within the organization driving the requirement, in this case, it's security. (it could also be development, devops, or risk)

Second is the NCWF role. This is the cybersecurity workforce role as defined by NIST.

In this case, it's the security architect.

Both the group and the NCWF role have hyperlinks links which will take you to another section of the document providing summary information regarding the responsibilities of the group and the role.

[break]

The next section is administration.

This is the RACI information.

The four groups of focus within the AVCDL are security, devops, development and risk.

In this case, we see that security, development, and risk share responsibility for this activity.

devops does not have any role within the context of this particular activity and so a dash is used.

[break]

The next section is the description.

This contains general information regarding the activity and its application.

It may also include notes that are applicable to the requirement.

As we see here, the second note has a reference (and link) to AVCDL-Foundation-9, the Threat Prioritization Plan.

[break]

The next section indicates whether training for this particular phase requirement is provided.

In this case, Threat Modeling does in fact provide training.

[break]

The next area covers AVCDL internal dependencies.

These are the predecessor activities that occur within the AVCDL.

In this case, we see that the Threat Prioritization Plan and Applying Security Requirements and Risk Information to Design are both dependencies of this phase requirement.

[break]

The next area lists the external dependencies.

These are dependencies on other groups within the organization.

You'll note that the table shows what inputs devops, development, and risk are expected to provide.

In this case, the devops and the risk groups are not expected to provide anything, but development is expected to provide an element detailed design in order for Threat Modeling to take place.

[break]

The next area lists the products generated by the phase requirement.

These are the products and artifacts that are created as a result of the activity.

These are linked to secondary documents which provide specific information as to the processes necessary to produce the artifacts.

In this case, we see that a Threat Modeling Report, a Ranked / Risked Threat Report, and a Threat Report are the artifacts which are created as a result of this phase requirement’s activities.

[break]

The next area shows the 21434 required work products.

These are the work products specified in 21434 applicable to this phase requirement.

[break]

The next area covers the UN R155 CSMS requirements.

As with the 21434 work products, these are the requirements specifically fulfilled by the phase requirement.

[break]

The next section covers CSMS supplemental information.

These are the R155 requirements which we believe are supported by this phase requirement but for which there is no direct connection.

[break]

Finally, are the CMMC applicable practices.

Each of the five CMMC levels and their applicable best practices are listed.

Process Workflows

Within each of the AVCDL secondary documents, the process’ workflow is decomposed into a series of activities.

In this case, what we're looking at is threat modeling.

Within the threat modeling process, we undertake three individual activities, threat model creation, threat model analysis, and threat candidate triage.

As you can see here, we call out the actors that are participating, the inputs and the outputs.

Additionally, we have optional feedback that may lead to updating the threat model itself.

Once the triage activity is complete, the final artifact is fed into the threat ranking process.

[pause]

The secondary documents that include these workflows have a lot to say about these activities and what motivates them.

They cover information like:

Here's what you do.

These people are the required.

The system design looks like this and these are the factors which drive the activity forward or back.

And lastly, these are the artifacts and reports and how they're structured.

Document References

As you might imagine each of the documents within the AVCDL compresses a lot of information into a small space.

[break]

Each of the documents provides a section labeled references which point to the materials used in the creation of that document, as well as additional sources of information which elaborate on the subject matter.

AVCDL Documents Available

Now, let's do a quick summary of the AVCDL documents that are available.

In the lower left-hand corner, you'll notice the URL for the AVCDL GitHub repository.

AVCDL Documents Available – Primary Document

Of course, the AVCDL primary document is available.

AVCDL Documents Available – Reference Material

To go along with the primary document is a set of working materials which were used both in the construction of the primary document and the secondary documents.

Now, these take the form of Excel spreadsheets typically, but may also be just reference documents which condense information, not used in the documents themselves.

AVCDL Documents Available – Secondary Documents

The secondary documents cover each of the AVCDL phase requirement products.

AVCDL Documents Available – Supplier Materials

Since supply chain cybersecurity is of such importance a set of documents has been written specifically to address such areas as

* the Cybersecurity Interface Agreement,
* the manufacturer's disclosure statement,
* the cybersecurity maturity of the processes of the supplier organizations
* as well as how to map established processes into AVCDL processes.

Additionally, there are a set of templates and forms to be completed in the establishment of agreements and the gathering of evaluation information.

AVCDL Documents Available – Introductory Blog Posts

Finally, there are a set of blog posts and elaboration documents.

The blog posts provide a gentle introduction to the AVCDL

These are generally targeted at management and decision makers.

The elaboration documents provide information on general subjects such as

* software bill of materials,
* open source
* and how various processes such as risk freshness and the overall supply chain fit into an AVCDL context.

TÜV SÜD Assessment Timeline

One of the strengths of the AVCDL is that it has undergone third party review and assessment from its inception.

it was intended that the AVCDL be able to provide support for organizations attaining certification under various technical and regulatory standards.

[break]

This is the timeline of the AVCDL assessment.

[break]

Work on the AVCDL began in January of 2020.

[break]

In May of that year, discussions began with TÜV SÜD with the intent of them performing an assessment.

[break]

19 months later in November of 2021 the assessment began.

[break]

The assessment for 21434 took eight months.

During that time feedback was provided by TÜV and the AVCDL updated to reflect that feedback.

[break]

In June of 2022, the AVCDL was assessed as being compliant with ISO/SAE 21434 excluding those elements which are the sole responsibility of the organization, such things as project management, quality management systems.

[break]

In August of that same year, it was determined that 26262 did not have sufficient cybersecurity related material to be able to establish a statement of compliance.

[break]

Work then progressed to prepare for the UN R155 assessment which began in January of 2023.

[break]

In June of 2023 after addressing TÜV’s feedback, the AVCDL was assessed as compliant with R155.

Again, excluding those elements which are the responsibility of the organization.

And additionally, excluding those elements which are the sole responsibility of the OEM these are very specific responsibilities such as reporting to the regulator tracking registrations of the vehicles, things which all fall outside of the realm of product cybersecurity.

[pause]

It's worthwhile noting that if you're an organization which has not done a 21434 or UN R155 certification,

that you seriously consider the amount of time that such a review is going to require and ensure that this is built into any product release schedule you have.

AVCDL on GitHub

All AVCDL materials, both in source and distribution forms, are available on our GitHub site, as shown here.

Because of the size of the repository, it's recommended that you either clone the repository or download a ZIP archive of it, if you're not familiar with using git.

Instructions for downloading a ZIP archive are linked to on the repository’s front page.

Training Path

The idea of 1000 pages of documentation and 100 documents may seem daunting.

And so, there's a training path that has been established that allows you to look at particular functions that you'd like to be taking part in and determining which are the specific secondary documents which are going to feed into those activities.

For instance, if you wanted to learn about secure decommissioning, the recommended path would be to start with the AVCDL overview training, then take both training on handling PII and secure deployment, and finally taking the secure decommissioning training.

References

Here are references to the source material used in the creation of this presentation.

They'll also be included in the video description.

Additionally, this presentation’s source material will be provided on the AVCDL GitHub repository.