# Title

I'm Charles Wilson, Staff Product Cybersecurity Architect at Torc Robotics.

This presentation is an introduction of the AVCDL for non-cybersecurity individuals.

# Introduction

The AVCDL is a cybersecurity development lifecycle designed to address the needs of safety-critical cyber-physical systems such as those in the automotive industry where in recent years we've seen the introduction of increasingly sophisticated computer-based systems.

This has led to the creation of a number technical and regulatory standards intended to address the cybersecurity concerns these systems present.

Most notably are ISO/SAE 21434 related to cybersecurity engineering for road vehicles and UN R155 regarding road vehicle cybersecurity.

Let's take a moment to look at the greater cybersecurity standards and regulations ecosystem.

# Standards and Regulations Ecosystem

Here's a view of the cybersecurity standards and regulations ecosystem.

There's a lot going on here.

In blue, we have technical standards.

These relate to what needs to happen from an engineering standpoint.

In white are activities that we need to do during the creation of the products.

Shown here are United Nations regulations.

Finally, in gray we have future regulation currently under development.

Starting from the bottom, we have the ISO 9000 series which is basically the QMS and related document management system elements.

Atop that, we have two standards that related to systems and software engineering.

And finally, we get to the cybersecurity blocks within the automotive space, dealing with cybersecurity engineering and software update.

These two have corresponding UN regulations for the cybersecurity management system and the software update management system.

Later regulations are dependent on the earlier ones.

And so, the one for the auto lane keeping system, R157 requires that you conform to R155 and R156.

And the same is true with the proposed European Union proposal, which will probably be rolled into the UN regulations in the future for autonomous driving systems.

This proliferation of standards and regulations is the problem that the AVCDL is trying to deal with and avoid.

The AVCDL provides a set of manageable, comprehensible activities from the standpoint of development, project management and safety that fulfill all the various requirements and regulations without the need for getting into the weeds of the specifics of ISO, SAE, or UN language.

Let's take a step back now and ask ourselves what is a lifecycle?

# Lifecycle Basics

To understand a lifecycle, it's important to first organize the space that the lifecycle occupies.

We start with policy because policy tells us why we do things.

Policies motivate our actions.

The next level down are processes.

These are the what, who, and when that allow us to implement those policies.

Finally, we have procedures. These are the how.

Procedures are very specific. They take a generic process and provide concrete steps as to how to implement it.

[break]

A lifecycle is the collection of processes and can be applied across multiple different domains and allows us to have dependencies that go across the entirety of the supply chain.

Note that procedures are not part of the lifecycle, because they are specific and unique to each organization implementing the processes.

# AVCDL Framework

Here's an overview diagram of the AVCDL framework itself.

It has roughly 70 processes spread across eight phases.

[break]

These are **foundation**, **requirements**, **design**, **implementation**, **verification**, **release**, **operation**, and **decommissioning**.

They’re collected into three groups.

[break]

The first, the foundation phase contains those processes we do across the entirety of the product line.

We established these processes once and apply to every project.

For instance, when you're talking about roles and responsibilities, those on a product-by-product basis.

The same is true for the overall requirements, tool chain support, or incident response.

[break]

The next group contain the inter-developmental phases.

These are the build-the-product phases. They follow the same general pattern as a standard development lifecycle where we have **requirements**, **design**, **implementation**, **verification** and **release** phases.

[break]

Finally, the post-developmental phases include **operation** and **decommissioning**.

This is where we're handling the product once it’s been released.

And it's also where we handle retiring the product, whether permanently or for RMA. We have to consider the cybersecurity standpoint, including the organizational IP and cryptographic materials in the device. This is regardless of whether we need to send it back to a supplier for repair or retire it from service.

[break]

Taken together, these phases and processes represent the totality of the product lifecycle from a cybersecurity perspective.

# Design Deficiencies vs Implementation Defects

One might look at the AVCDL and think that there are a lot of activities there.

Do we really need all these activities to ensure cybersecurity for the product?

That's a good question.

Let's look at two areas that we think about in a realm known as defense-in-depth that we base our choice for cybersecurity activities on.

These are **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 **penetration testing** or earlier on in the development process with extensions to what would normally be considered compiler features such as **static** and **dynamic analysis**, as well as more sophisticated after-the-fact testing, such as using the analog to **unit testing** that being **fuzz testing**.

Additionally, you can have **secure code reviews** alongside of traditional **functional code reviews**.

All these activities address implementation defects.

These are where the system itself has implemented cybersecurity related elements incorrectly.

[break]

A more sophisticated approach allows us to address design deficiencies where the elements of the system were not designed taking cybersecurity into consideration.

Here, we use activities like **security design reviews**, **attack surface analysis** and **threat modeling**.

You'll notice that there're duals of these last two activities in the verification phase where we perform a **threat modeling review** and an **attack surface analysis review**.

This allows us to ensure that the deficiencies identified in the design phase were in fact addressed by the time we get to the verification phase.

[break]

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

# Implementation Methodologies

Let's take a moment to talk about implementation methodologies.

Some people look at this rectilinear representation of the AVCDL framework and say, well that that's a waterfall implementation.

In reality, this representation was chosen because it is the most compact way to present the information showing the different phases, their processes, and how they are grouped together.

It's not a statement about implementation methodology.

In fact, the AVCDL is implementation methodology neutral.

This is an important point because in any highly complex system, such as an autonomous vehicle, there are dozens of components, produced by multiple suppliers, each with their own development processes and implementation methodology. It’s critical to be able to align cybersecurity activities regardless of the particular implementation methodology being used.

Next, we'll look at two specific methodologies which the AVCDL addresses directly.

# V-model Implementation Methodology

The first methodology is most heavily used by safety groups.

It’s the V-model implementation methodology.

We see it realized in ASPICE and the like. ISO 26262 also leans into this model.

An explanation of how the AVCDL accommodates the V-model is explained in an elaboration document titled **Understanding Verification and Validation in an AVCDL Context**.

# Cyclic Implementation Methodology

The second implementation methodology of note is one which has become a favorite of development, that is the cyclic implementation methodology.

This methodology has actually been around for quite some time. It’s short-cycled and iterative in approach. Two specific realizations are spiral and agile.

What we see here is how the AVCDL framework could be visualized in such an implementation.

There's an elaboration document, **Understanding Cybersecurity Risk Freshness in an AVCDL Context**, which covers the implications of the cyclic implementation methodology and the feedback loops that go with it.

Process Workflows

An important element provided by all AVCDL processes is a workflow diagram like the one shown here.

From the standpoint of project management, this is a useful and important source of information as it enables them to know what individuals and resources will be required, what activities will be undertaken, what outputs, and what reports will be generated by the process.

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

Within this process, we undertake three distinct activities, **threat model creation**, **threat model analysis**, and **threat candidate triage**.

As you can see, we call out the actors, inputs, outputs, and reports.

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 a common threat ranking process.

The documents that include these workflows provide detailed explanations about these activities and what motivates them.

They also provide information about the artifacts and reports, and how they're structured.

All this information is critical to being able to plan and schedule these activities, and also to coordinate the dependencies between groups and for the process overall.

AVCDL Documents

As you might imagine covering an entire cybersecurity development lifecycle requires a lot of explanation.

The AVCDL doesn't simply provide a single huge tome to do so.

It's broken up into many different pieces allowing for both incremental adoption and for focused learning.

The primary document covers the basics of the AVCDL, how it was constructed, what the various goals of the AVCDL are, the requirements for various cybersecurity roles referenced, as well as how each of the phases decomposes into a set of processes.

[break]

A set of secondary and elaboration documents covers each of these processes in detail and provides additional material that help understanding the processes and specific concepts that are used within the AVCDL.

[break]

There's also a set of templates and working materials to assist cybersecurity SMEs in performing the various activities necessary to implement the lifecycle.

[break]

Additionally, it's important to consider that the supply chain is a critical part of cybersecurity engineering.

[break]

And so, there's a set of documents dedicated strictly to the supply chain, including creation of the cybersecurity interface agreement and a set of templates to assist in that process.

[break]

Finally, there is a set of introductory blog posts created to give individuals who don't work in cybersecurity an understanding of various concepts that cybersecurity relies upon to accomplish its mission.

Assessed for Compliance

The AVCDL has been assessed for compliance with both ISO/SAE 21434 and UN R155.

This assessment was done by TÜV SÜD and you can see here the letters of assessment which are included in the AVCDL document set.

This was a line-by-line assessment of the AVCDL to ensure that if used that all the requirements of both 21434 and R155 would be met.

Having knowledge that the AVCDL satisfies both the standard and regulation, gives any organization a leg up when it comes product certification. Knowing that the path they're going to take has been validated greatly improves the likelihood of success.

AVCDL on GitHub

All the documents in the AVCDL and the working materials used to construct them are available on the GitHub site shown here.

And it's recommended that if you're going to access them, that you either clone the repository or download a ZIP archive of it.

There are instructions for downloading the ZIP archive if you're not familiar with using GIT.

AVCDL on YouTube

A set of training videos have been created to cover various aspects of the AVCDL that follow the training path found in the AVCDL primary document.

It's important to note that all of the AVCDL materials with the exception of the introductory blog posts are intended for product cybersecurity engineering practitioners.

References

Finally, here are references used in the construction of this presentation.