# Title

Good morning, all. I'm Charles Wilson, cybersecurity engineering technical fellow at Motional responsible for cybersecurity development lifecycle practice. Let me begin by thanking the conference session chairs for the opportunity to be today’s first speaker. I hope everyone’s sufficiently caffeinated.

This talk is entitled “**There and Back Again: Building a Cybersecurity Development Lifecycle from Scratch to Comply with ISO/SAE 21434 and UNR 155**.” I realize that's a pretty vague title and it probably leaves a lot of you wondering what I'm going to be speaking about this morning, but I'll try to elaborate as we go.

# Introduction

In the automotive industry, we have a lot on our plate with respect to cybersecurity. When I was hired at motional, about 4 1/2 years ago, I was tasked with making sense of these then not quite complete standards and regulations in order to ensure that our product, which is autonomous vehicle AKA a ROBOTAXI comply with them.

When I was looking at what was out there, I thought, “this won’t take very long. There’s probably a lot of really good work done and it's all well-known and I'll be done in about 6 months or so.” Here I am about 4 1/2 years later and I think I'm starting to get a pretty good handle on it.

So, let's look at the AVCDL, where it came from, what it does, and how it manages to achieve the goals of sound cybersecurity engineering and compliance.

# Standards and Regulations

First let's talk about the standards and regulations.

# Standards and Regulations Ecosystem

Here’s the current stack of standards and regulations that form the ecosystem we live in.

At its foundation, we have ISA 9000 and its kin, which establish our QMS, that thing that allows us to track all of the other stuff we do.

Above that, we have the dual SDLC for systems and software 15288 and 12207 respectively.

And then probably the thing we fixate most on is ISO/SAE 21434, along with its corresponding regulation UN R155, which covers cybersecurity management systems.

Above that we have ISO 24089 and UN R156 dealing with software update management systems. As a note, the harmonization between these two has been completed and should be in the interpretation document for R156 in the next several months.

Sitting on top of all of that, we have R157, which is the auto lane keeping system, or ALKS.

Finally, out in the future we have the Aries proposal for ADS from the EU. The UN is beginning to look at creating the regulation that will probably lean into that work.

But that's not everything that's out there. We don't just take a regulation or a standard and say, “let's create a check list and just do those things and then we’ll have something which is cyber secure. We need to create a set of processes which support each other and will allow us to create a compliant product with sound cybersecurity.

# Reference Sources

With this goal in mind, I went back to some base documents. These included the aforementioned ISO/SAE 21434 and UN R155, and also included things like ISO 26262 since we're doing cybersecurity in the pursuit of safety.

On the other side of the house, is the long-standing work done by Microsoft in the early 2000s on the security development lifecycle. Between these we have the work that has been done by NIST in the area of cybersecurity workforce roles and responsibilities called the NCWF. Also, from NIST is a compendium of best practice document called the SSDF.

I took all of these and I looked at it and looked at them and said it's like, great, there are all these things that we have. After reviewing these, I recognized that there needed to be a way to establish a common basis for talking about the different phases of activities that exist,

# How the Standards Line Up

Here we can see that we've got 15288, 12207, 26262 and 21434; what they talk about; and how they chunk their various activities. You'll notice that although these standards build upon each other, they don't quite line up. Mostly they do, but not sufficiently.

# Lifecycle Phase Harmonization

One of the first things I did was to create this. It’s a phase harmonization that established eight lifecycle phases and two process sets (organization and supplier). The lifecycle phases is where the main focus of the AVCDL was.

You'll notice there are things (shown in purple) that don't quite align neatly. In the end these were divided, but that’s something I felt that could be coped with.

# How Sources Inform the AVCDL

Here we can see what happens when you take all of these different sources and you try to determine how they speak to the product development lifecycle. As you can see, the Microsoft SDL (shown at the bottom) is pretty much one-to-one in terms of how we’re dividing things.

Above the MSSDL, the SSDF from NIST as well as 26262 apply over fairly broad areas.

Then there’s 21434 which is just this nightmare of organization. Way too much caffeine went into the creation of that document in my view.

At this point all the activities were identified and bucketed. Now what?

Since 21434 is the is the base that we're shooting for because this is automotive and this is cybersecurity engineering, the next step was to make sure that I was covering all of the requirements in that standard.

# ISO/SAE 21434 – AVCDL Mapping

And that's what you see in this diagram. This is a working document that I use to ensure that I’m covering all the requirements in 21434. Across the top we have all of the activities and what I would term phase requirements within the AVCDL. On the left is the list of 21434 work product requirements.

This is great for me, but this isn't overly helpful in terms of communicating to people.

# AVCDL Framework

So, I created this visualization called the AVCDL framework. It breaks down to three main chunks. There’s the foundation, which is stuff that you create once and then use for every product. This includes training, code protection, release integrity, and plans for incident response, decommissioning, threat prioritization and deployment. The area in green which are the activities you do in the building of each individual product. The ones in the yellow are the post-production aspects (operations and decommissioning).

# Design Deficiencies vs Implementation Defects

When you look at how this breaks down further, we can see that we have two different sets of coverage that we're trying to achieve here. We have the ones that address design deficiencies and we have the ones that address implementation defects.

Now 21434 doesn't require that we do any of the things that are looking for implementation defects. Which are in my view, really important. 21434 focuses on the fact that that you will do a TARA that you do upfront and at the end you check it to make sure that that you're good. You're supposed to keep it fresh.

But here you can see very clearly that there are things we do in the design phase, which are intended to catch issues early and then we have those things which we do in the implementation phase, which are far more tactical. There we're doing things like static analysis, dynamic analysis, fuzz testing, and secure code reviews.

# Distribution of Cybersecurity Activities

You may notice that the distribution of activities has a normal distribution look to it. That's because the implementation phase is where most of the cybersecurity engineering is being done. That's where we're applying the cryptography and ensuring that you don't compromise security through poor software development practices.

# V-model View

Since the AVCDL does not prescribe a particular implementation methodology, an elaboration document was created to help those who use the V-model understand how it relates to the AVCDL.

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

In this example from the document, we can see the threat modeling and attack surface analysis reports 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.

Now there are those individuals who see linear implementation methodologies as at best antiquated and at worst evil. They look at the AVCDL framework and say, “you know that that's waterfall and we’re agile.” My answer to that is that I write in English using a left or right script and the best way to put all this data together in a compressed form is to use this rectilinear format.

# Cyclic View

Even though the AVCDL goes to great pains to explain that it does not prescribe a particular implementation methodology, some people aren’t moved until you can show them the data in a form they’re comfortable with.

This is what the AVCDL framework looks like when you visualize it in a cyclic view.

# Cyclic Feedback

As with the V-model, aspects of cyclic implementation methodology are addressed in an 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.

Although we generally have forward progress cyclically, retrograde events may be forced when issues arise.

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. This portion of the product dependency diagram shows how all the AVCDL’s products can be traced back to fundamentals such as the cybersecurity goals.

In practice we should start with goals and let them drive the various activities which support them.

# Details

As you can imagine, the AVCDL took a lot of time to put together. It also evolved over time. At times it seemed that every answer brought up multiple new questions. Let’s look at some of the patterns that come up repeatedly in the AVCDL.

# Typical Phase Requirement

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

Every phase requirement identified both the responsible group and role within that group. RASIC information is provided for possible participating groups. Training is indicated, as are the requirement’s dependencies, both within the AVCDL and specific needs from other groups. A list of products and their documents is provided. Finally, how the requirement fulfills various standards and regulations.

# 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, 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.

# Document References

Each document in the AVCDL has a full set of references, so it's not just an opinion-based creation I encourage everyone to review these references in order to better understand the reasoning behind the choices made.

# AVCDL Materials

As you might imagine, this sounds like a lot of stuff, and in fact it is. There's a primary document. There’re about 70 secondary documents. There are 10 blog posts that that describe things at a very high level, so that management can get a good overview of things. There are worksheets and templates for various activities to be considered.

# Supply Chain

Here’s the elephant in the room, if you will. Supply chain. Now it's great if you're doing cybersecurity engineering, but in something as complex as a modern vehicle, it’s doubtful that you're going to be the only participant in your supply chain. Early on it became obvious that supply chain needed to be addressed.

# Supplier Selection

How do we do integrate cybersecurity into supplier selection? This question drove the creation of three inputs to the process. Those were a self-reported maturity, a manufacturers disclosure statement, and a mapping between a supplier process set and the AVCDL.

These are used to derive the requirements and cybersecurity interface agreement. The latter informing the SLA, TARA, and the SBOM.

# Supply Chain Guidance Documents

The supply chain material has an ecosystem of guidance documents to be able to handle all of the questions that come up.

Now you’re probably thinking, “wow, that's a lot. What do we do with all this?”

# Training

There's this thing called training because everyone involved in cybersecurity engineering needs to be both qualified and trained.

# Training Path

In order to do that, we had to establish a training path. Here you see the main training path for the AVCDL. The items in green are the trainings that I have created.

# Supply Chain Training Path

The supply chain has its own training path. It's smaller because the anticipation is that they're doing their own thing, but you need to ensure that they have these bits in common because if you need to do a CIA, you want a consistent way for both participants to talk about it.

# Assessment

Now, this is great, but why should we care? We care because of the need for assessment.

# Evaluating Your Status

The first type of assessment, mentioned earlier is self-measured maturity. We apply it to our own organization in order to be able to show how we’re continually improving. Here's a spreadsheet that list all of the AVCDL phase requirements with a SEI CMM level for each.

# Mapping Your Status

We can visualize it this information with this spider diagram. Around the circle are phase requirements. Over time it can show where we are improving and where deficiencies are.

Remember, if you don't measure it doesn't matter.

# Assessing for Compliance

The second type of assessment is external. The AVCDL is, to my knowledge, the only lifecycle which is fully compliant, in terms of what it can do for cybersecurity, with both 21434 and R155, and also publicly available.

And here are the assessment letters to prove it. I needed two pages for R155 because the list is so long.

Please note that this is an assessment, not a certification, because unlike a certification where you tend to do spot checks, the assessment went line by line in both the base and interpretation documents. The AVCDL also provides the fulfillment documents used to assert compliance.

# Assessment Timeline

Assessment took a long time. We had 19 months of discussions with TÜV SÜD before we were ready to do the 21434 assessment. It took eight months to do the assessment. 14 rounds of back and forth on that. The time required for R55 was shorter. It only took six months because we had already gone through the process and I understood what was going to be needed.

If you intend to get a certification and you think that you can wait until after you’ve finished V&V before starting the process, you're probably not going have a good outcome.

Talk with your certification body’s services group early and often. Make sure that you know where any holes in your processes are.

# Incremental Adoption

“Charles, this is great, but it's just too much. We have suppliers that have no security posture. We can't just dump this on them and tell them to just do it all.”

And so was motivated a way to address this problem, which I call incremental adoption. When you're trying to incrementally adopt, your goal is to achieve security, it's not to achieve compliance. This is because compliance requires a lot more than security does, and you can't just say, well, I'm going to be compliant and then I'll be secure. If you first get yourself secure, compliance becomes a much easier lift.

# Incremental Adoption Spectrum

Here we see an adoption spectrum which takes us from tactical to strategic application. And it chunks it up into five stages.

# Prioritized Phase Requirements

Here's how we decompose the AVCDL phase requirements into those five stages.

The further you progress from tactical to strategic, the greater the interplay and interdependence there'll be between various activities and with other groups within the organization. It's far easier to implement the earlier stage activities and make adjustments to your organization's tooling as you progress.

Implementing the later stage activities and attempting to backfill the earlier stage ones will lead to a lot of churn should changes be required.

# Lessons Learned (title)

Some lessons learned in all of this:

# Lessons Learned

* Learn from the past – there’s a lot of good work out there
* Be systematic and consistent – other groups expect that they can just come to cybersecurity and we need to be ready with consistent answers
* Play well with others – cybersecurity isn’t here to lord over anyone we’re here to support others
* Allow realistic amounts of time – cybersecurity has a cost in both time and resources
* Communicate liberally – no one likes surprises, don’t be the source of them

# Learning More

Now that your brain is full, I’ll leave you with some homework.

# AVCDL on YouTube

First, all those trainings that I mentioned earlier, they're up on YouTube. You can go there and see them.

# AVCDL on GitHub

Second, there is a GitHub site that has all of AVCDL materials (source and distribution forms). It’s all Creative Commons so that we can make sure that that cybersecurity is for and available to everyone and.

# References

Here are two pages of references covering this presentation.

# Questions

And with that, I will open it to questions.