Bernd Kudicke and Sven Surrey

Flight Level 2 Design for the Bosch ADAS Integration Platform

Did you know that the first Level-3 autonomous vehicle wasn't approved in the US or China, but in Germany? Level 3 means you can activate the traffic-jam pilot in a Mercedes S-Class, take your hands off the wheel, and comfortably read your emails or newspaper. However, when the system signals, you have to be ready to take control again. Mercedes has since gained approvals in Arizona and California, while Tesla is still stuck at Level 2.

Behind this success is cutting-edge technology - specifically, the ADAS Integration Platform by Bosch Mobility. Launched in 2022 within the ADAS division, this platform is tailor-made for highly automated driving at SAE Level 3. That means vehicle functions that fully take control in certain scenarios, letting the driver sit back and relax with hands off the wheel.

ADAS stands for "Advanced Driver-Assistance System" and refers to all the various assistance features currently found in vehicles—such as lane-keeping assist, parking assist, adaptive cruise control, and more. At the time of our involvement, around 750 colleagues were working within the ADAS Integration Platform to develop a scalable, modular vehicle computer. This computer seamlessly integrates data from multiple technologies—radar, video, ultrasonic sensors, and more—and processes it in a way that enables highly automated driving, even under challenging conditions.

In a pilot project, we had the pleasure of supporting the smart folks from the ADAS Integration Platform as they explored new ways of collaboration across their teams using Flight Levels.

The Organization and Its Goals

When we got involved, one of the key missions was delivering a high-performance computing platform (ACP) for ADAS, specifically designed to handle safety-critical applications. This involved developing, testing, and integrating hardware, hardware drivers, foundational software, and middleware—think of it as a smartphone without any apps installed yet. Those apps came from another unit called "ADAS Control." Around 750 people spread across Germany, Romania, the USA, India, China, and Japan were all pulling together to make the ACP platform a real success.

The clearly stated goal of Systems Engineering (also known as the ACP Framework) for the ACP platform was to rapidly pick up speed, aiming to:

1. outmaneuver competitors with a faster time-to-market, and
2. identify and leverage synergies across multiple projects.

The aim was clear: rapidly and efficiently bring products to life together with automotive industry customers. And that's precisely what the folks from the ACP Framework asked us:
"Can you help us become the fastest players in the market?"

Taking the Shinkansen Through the Jungle

750 people spread across seven countries, with different cultural backgrounds, working in various organizations—how do they even collaborate effectively? How can they jointly develop a complex computer platform?

That’s exactly the first question we tackled. There’s a great saying: "You can't see the forest for the trees.” But in this case, it wasn't even a forest—it was a whole jungle! By talking with different teams, we discovered that, yes, there were indeed some narrow paths connecting them, but you can't pick up speed on jungle trails. What we needed were reliable high-speed connections, clear checkpoints, and a solid timetable. And our job was exactly this: establishing these connections through proper coordination.

How did we tackle this challenge?

1. Our first step: getting crystal clear on the starting situation and mapping out all those jungle paths.
2. Next, we took a deep dive into the dependencies between these paths and captured the workflows to truly understand what was going on.
3. These workflows were then consolidated into value streams, pinpointing exactly what coordination was needed.
4. Super important before jumping into Flight Level systems: running a pilot!
5. Finally, we built the boards iteratively and launched them into real-life operations.

The whole journey took us just over half a year.

Identifying the dependencies

When developing products, dependencies often act like speed bumps that slow everything down. To become the fastest on the market, we had to clearly understand where these dependencies existed. Our goal was simple: gather facts about the current situation and pinpoint exactly where special coordination was needed. To visualize these dependencies, we used a tool called a Design Structure Matrix (DSM).

First, we identified about 30 involved teams and sent them a questionnaire. Each team rated how intensely they collaborated with other teams, using a scale from 1 (not at all) to 5 (very often). This created a matrix, which we then sorted using a principal component analysis. The result: three clearly defined dependency clusters emerged (see Figure 1).

- In the top-left corner, we saw teams collaborating on software and system development.
- Down in the bottom-right corner, teams focused on hardware development had clustered together.
- And in the middle, somewhat surprisingly, were teams responsible for end-of-line testing, operating fairly independently from both software and hardware teams.

But Figure 1 revealed even more: two clear coordination clusters emerged (middle-left)—one around Architecture & Safety, the other focusing on customer-specific hardware/software projects and framework projects. And right there we had it—perfect entry points for kicking off our Flight Levels pilot!

Visualizing the workflows

Before you start restructuring or improving any processes, it's absolutely essential to get clear about how teams actually work—or at least intend to work. Flipping everything upside down from the get-go, without regard to current workflows, is rarely a smart move.

That's why we first went back to the "jungle paths" we'd mapped out initially—these were essentially the existing workflows shaped by collaboration and dependencies between teams. If we'd jumped straight from the Design Structure Matrix into building boards, we'd have ended up with ridiculously complicated coordination boards with up to 20 stages. Not only would that have created confusion and overwhelmed everyone, but practically speaking, no tool could handle that level of complexity effectively.

We decided to add an intermediate step: checking if there were repeating patterns in the workflows that we could streamline into a maximum of ten steps using smart column labels.

First, we collected all the recurring Flight Item Types and analyzed the triggers that initiated each jungle path (see Figure 2). For example, a Request for Quotation (RFQ) required both specific expertise and leveraging synergies within the organization. Once we had these triggers identified, we dove deeper into each jungle path, tracing and documenting the workflows

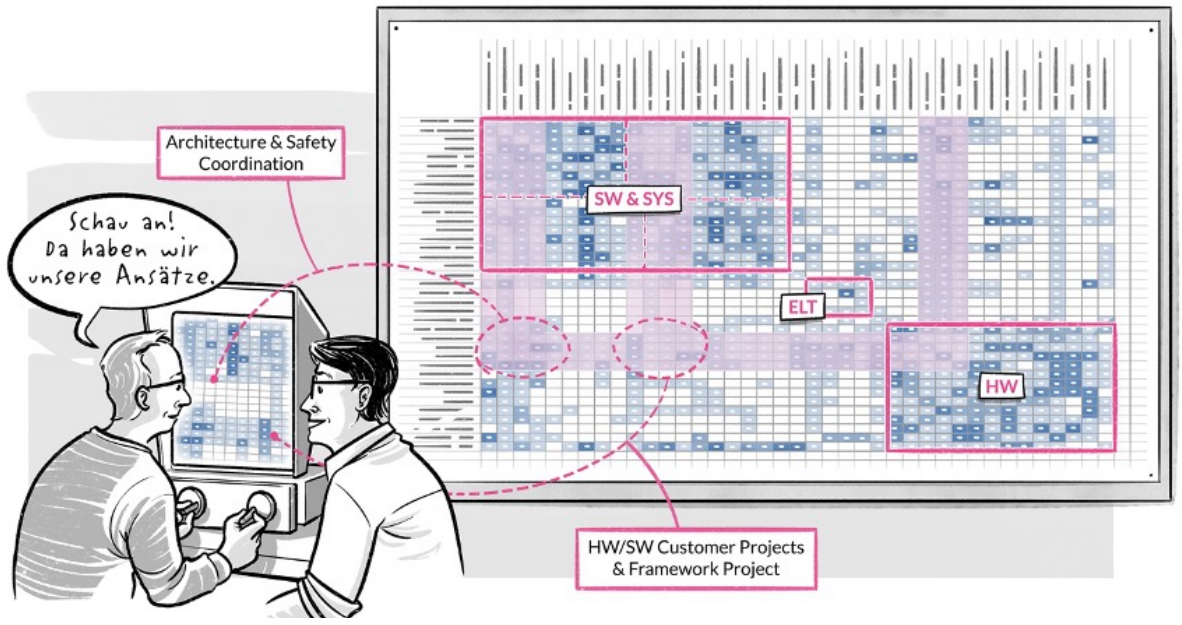

Figure 1: After sorting, three major dependency clusters and two coordination clusters became clearly visible.

step-by-step (like our RFQ example). With this information, we could easily spot similarities and differences between workflows.

In total, we distilled nine core workflows—fewer than we expected, considering the complexity of dependencies we initially found (as shown in Figure 1).

The Work System Topology

In the Flight Levels approach, we use what's called the Work System Topology to get a clear overview of all the work systems currently in place: Which work systems exist at which Flight Level, and how are they interconnected? By mapping this out, we quickly spot areas where improvements in the workflows can make a real difference.

In such a large organization, it's crucial to differentiate between upstream and downstream systems. Upstream involves areas responsible for strategic decisions, whereas downstream systems implement and deliver products based on those decisions. The key question is always: When can downstream actually start delivering?

When working on Work System Topologies for large organizations, we follow a simple rule of thumb: first understand and clearly define the "Point of Commitment"—the exact point at which decisions are made and the green light is given for the work of the next week, month, quarter, or even half-year.

This Point of Commitment is the critical link connecting upstream and downstream systems, so it needs to be rock-solid.

In Figure 3, you'll see the downstream side with teams (yellow) on Flight Level 1 and their coordinating levels on Flight Level 2. On the left side, in the upstream, we have the ADAS strategy connected to the downstream via Portfolio Management. In other words: the Portfolio Management acts as the Point of Commitment, clearly defining what downstream units should deliver and when.

Why is this Point of Commitment so critical? The organization was focused on speed and efficiency—both of which rely on clear decision-making pathways. If every downstream unit were to independently decide what to work on and when, most of their time would be wasted in endless discussions. In that scenario, it becomes nearly impossible to steer the overall system effectively and maintain focus.

By clearly defining the Point of Commitment, everyone knew exactly where and how coordination happens at the team level and at the product and service level. Even by simply simulating standard cases, we revealed hidden, multidimensional interactions within workflows—issues that, especially in special cases, would inevitably cause costly delays and decision conflicts.

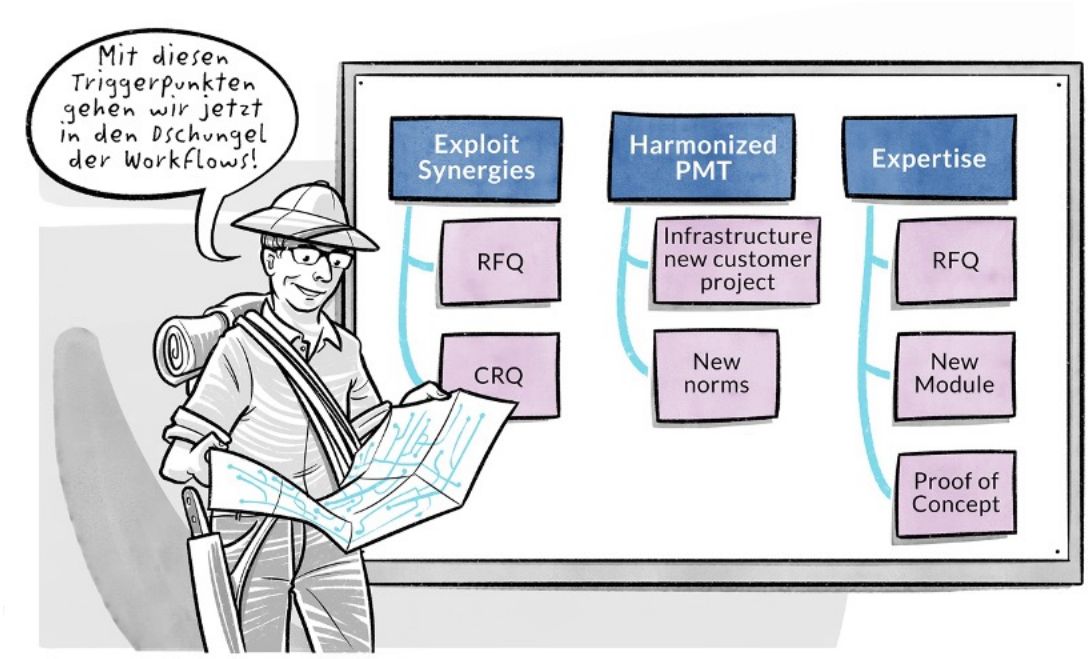

Figure 2: Identifying triggers and workflows

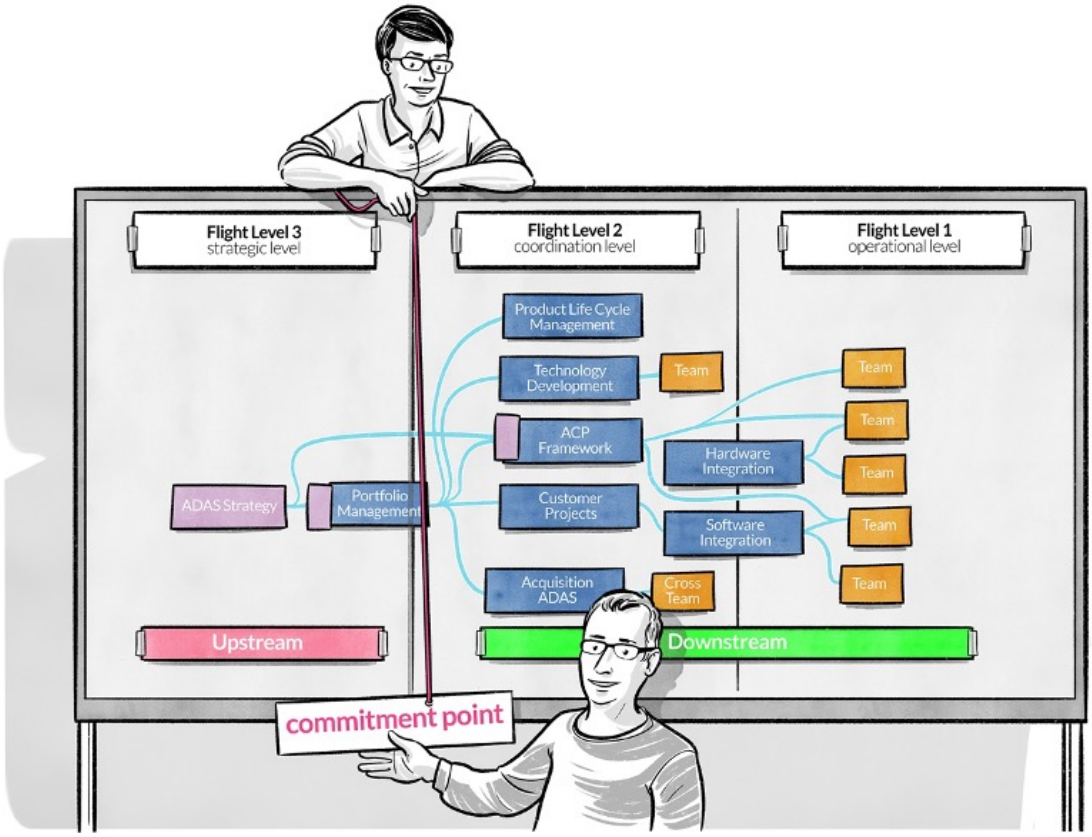

Figure 3: Up- and Downstream and Point of Commitment

Simulations

The Work System Topology laid the groundwork for mapping our Flight Routes. What such a route looked like after several iterations is shown in Figure 4. On the left side, the green boxes represent triggers—things that kick off work. These triggers can also pop up elsewhere, but not all are shown here.

What stands out about this Flight Route is that it's strongly team-focused, deliberately excluding the strategic layer. Why? Because our goal was speed—creating fast, reliable paths for implementation. This version was the outcome of iterative optimization and simulations, collaboratively developed by everyone involved. Earlier iterations didn't quite meet our need for clarity and quick decisions.

The real value of this step was that everyone involved now had a clear picture of the end-to-end process. They could see exactly which workflows would fly—and which wouldn't. That's precisely the point of simulations: workflows aren't quietly sketched out by someone isolated in a back room; they're shaped by the very people who know exactly how work moves through the system or have clear expectations about it.

And this is exactly where the magic happens: People start discussing their actual work, their deliverables, and customer expectations. These conversations don't just reveal different viewpoints about the same reality; they also expose potential roadblocks and dead-ends. We specifically ask about these "toxic paths", because they're easier to avoid once explicitly identified.

Yes, it's an iterative process and, admittedly, takes some time—but it's absolutely essential for truly understanding workflows and optimizing them effectively.

Board Design

The Flight Routes essentially highlighted work results. But that also means something happens between these results—actual tasks, specific things getting done. For example, to move from the outcome "Resource availability feedback" to "Realistic plan proposal," the timeline needs updating. And right here, hidden between the outcomes, is the actual work: the activity "Update timeline."

Later, these activities become columns on the boards, at least in early drafts. Using these activities, we created initial board prototypes with a digital tool, giving

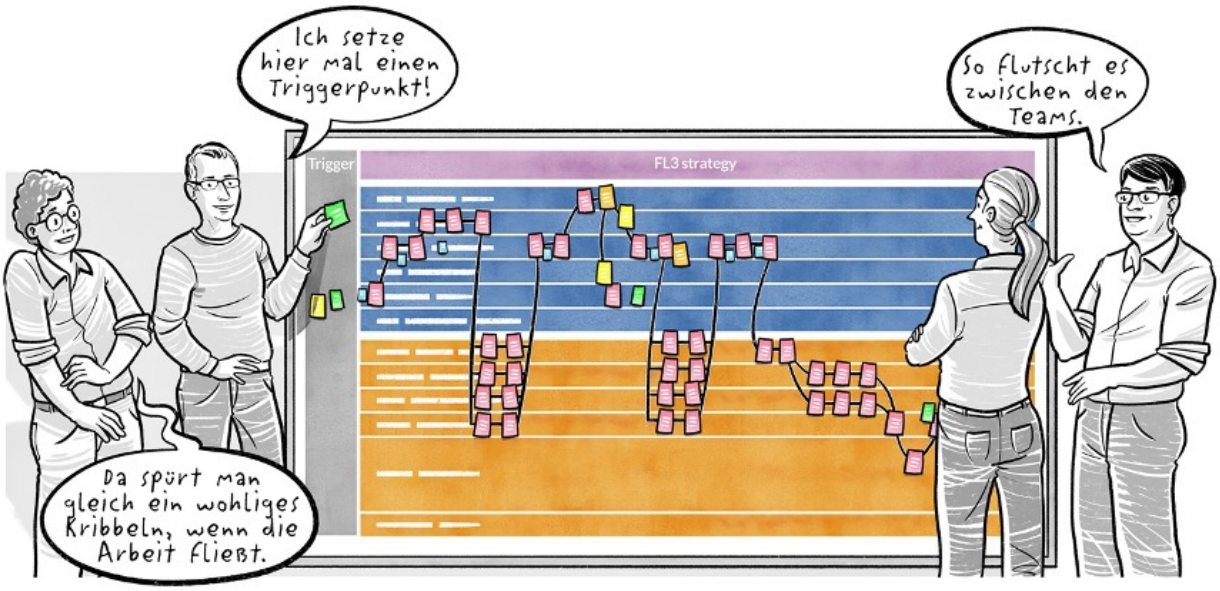

Figure 4: Example of a Flight Route for the ACP Framework

everyone involved something tangible to discuss and further refine.

So, the Board Design was happening in parallel with the simulations because we immediately defined the activities leading from one interim result to the next. And during this step, we learned a few key things:

- Activities should be phrased broadly enough to cover multiple Flight Item Types.
- Naming activities usually requires some modeling because, let's face it, different engineering disciplines often call the same tasks by different names. So, we had to experiment a bit until everyone found a common language.
- This highlights that board design is itself an iterative process. Participants create a first draft, test the board using past use cases, try it with future scenarios, and eventually reach a conclusion: either the board is "good enough" to launch, or it needs further tweaking.

After arriving at a "preliminary final" board design, we invited the teams to a simulation session with a physical board to run through the entire workflow multiple times. Afterwards, we asked everyone involved about their impressions, and their feedback confirmed just how valuable simulations are before launching a Flight Levels system architecture into real-world operations:

- “Through this workshop, we’ve developed a shared understanding.”
- “This board lets us clearly show management which parts of the organization carry the heaviest workload.”
- “We now understand exactly how our workflows should function going forward, and how we can visualize them.”
- “It clearly demonstrates what happens at the coordination level.”

The final, "real" board was then implemented digitally using Confluence with JIRA tickets.

What have we achieved?

Of course, throughout this entire process, we faced some challenges that were out of our control. In 2023, two major organizational shifts threw our original plans—to roll out Flight Levels beyond the ACP Framework—completely off course. Despite these hurdles, we still accomplished a lot:

- We clearly identified the value streams and key coordination points.
- We developed a prototype for the system architecture, successfully demonstrating the potential of the concept.

Despite all the changes, it was crucial for us to clearly show what's possible—and how organizations can leverage Flight Levels to become even faster in the future. That’s something we want you to remember, too: especially in large organizations, there will always be sudden seismic shifts nobody can anticipate. But what truly matters is what you learn along the way.
Thanks to numerous iterations, both we and the people from ADAS now know exactly how the Shinkansen can speed smoothly through the jungle. We discovered which workshops are effective for which groups—and which ones aren't. So, whether you fully implement your Flight Levels design or not, one thing’s certain:
You’ll learn a heck of a lot along the way!

Bernd Kudicke *was one of Bosch's first Flight Levels coaches and actively supports the organization's improvement and transformation initiatives. As both a Flight Levels coach and Viable System Model coach, he connects and optimizes workflow systems, shaping them into clear Work System Topologies with tailored decision-making structures and communication channels. Bernd especially values the strong co-creation aspect of Flight Levels thinking, which allows him and his clients to develop excellent solutions for organizational pain points together.*

Sven Surrey *is one of the pioneers behind Agile@Bosch and is currently an Agile Enterprise Coach at Bosch Digital. Until October 2023, he led organizational development for the ADAS Compute product area, making him one of Bosch's first experts in agile hardware development. Since 2018, Sven has increasingly focused on transforming traditional hierarchical organizations into agile leadership systems that anticipate strategic shifts and proactively adapt to them.*