

Flight Level 2 Design für die Bosch ADAS Integration Plattform

Wusstest du, dass das erste Level-3-Fahrzeug nicht in den USA oder China zugelassen wurde, sondern in Deutschland? Ach so, „Level 3“ bedeutet: Wenn du den Staupiloten in einer Mercedes S-Klasse aktivierst, kannst du deine Hände vom Lenkrad nehmen und währenddessen deine Mails oder die Zeitung lesen. Sobald das System ein Signal gibt, musst du mit deiner Aufmerksamkeit wieder ganz beim Fahren sein. Inzwischen hat Mercedes auch Zulassungen in Arizona und Kalifornien, während Tesla lediglich eine Level-2-Zulassung vorweisen kann.

Hinter diesem Erfolg steckt Spitzentechnologie, genauer gesagt die ADAS Integration Plattform von Bosch Mobility. Diese wurde 2022 innerhalb des ADAS-Bereichs gegründet und speziell für das hochautomatisierte Fahren nach SAE Level 3 entwickelt – also für Fahrfunktionen, bei denen das Fahrzeug in bestimmten Situationen vollständig die Steuerung übernimmt und der Fahrer die Hände vom Lenkrad nehmen kann.

ADAS ist das Akronym für „Advanced Driver-Assistance System“, womit sämtliche derzeit in Fahrzeugen verbauten Assistenzsysteme wie Spurhalteassistent, Parkassistent, Abstandsassistent etc. gemeint sind. In der ADAS Integration Plattform arbeiteten zum Zeitpunkt unseres Einsatzes rund 750 Kolleginnen und Kollegen daran, einen skalierbaren, modularen Fahrzeugcomputer zu entwickeln, der Daten aus unterschiedlichen Technologien – Radar, Video, Ultraschall usw. – zusammenführt und so verarbeitet, dass hochautomatisiertes Fahren auch unter anspruchsvollen Bedingungen möglich wird.

Wir durften in einem Pilotprojekt die smarten Leute im Bereich ADAS Integration Plattform dabei unterstützen, mit Flight Levels neue Wege für die Zusammenarbeit zwischen den Teams auszuloten.

Die Organisation und ihr Ziel

Zum Zeitpunkt unseres Engagements bestand eine Mission darin, für ADAS die hochperformante Computer-Plattform zu liefern (ACP), auf der dank des Systemdesigns sicherheitsrelevante Applikationen laufen können. Entwickelt, getestet und integriert wurden dabei die Hardware, die Hardware-Treiber, Basis-Software und die Middleware – wie bei einem Smartphone, das noch keine Apps hat (diese wurden von einer anderen Einheit namens „ADAS Control“ geliefert). Am Er-

folg der ACP-Plattform arbeiteten rund 750 Personen, die in Deutschland, Rumänien, in den USA, in Indien, China und Japan angesiedelt waren.

Erklärtes Ziel des System Engineerings (= ACP-Framework) für die ACP-Plattform war es, so schnell wie möglich Fahrt aufzunehmen, um

1. in puncto Time-to-Market die Konkurrenz auszumönnieren und
2. Synergien zwischen verschiedenen Projekten zu finden und zu nutzen.

Gemeinsam mit den Kunden aus dem Automobilsektor sollten Produkte schnell und effizient umgesetzt werden können. Und genau mit dieser Frage traten die Vertreter:innen des ACP Frameworks an uns heran: „Könnt ihr uns dabei helfen, die Schnellsten am Markt zu werden?“

Mit dem Shinkansen durch den Dschungel

750 Leute in sieben Ländern, mit verschiedenen kulturellen Hintergründen, aus verschiedenen Organisationen – wie arbeiten die überhaupt zusammen? Wie entwickeln sie gemeinsam eine Computer-Plattform?

Wir versuchten als Erstes, uns genau davon ein Bild zu machen. Es gibt das schöne Sprichwort „den Wald vor lauter Bäumen nicht sehen“, doch hier war es kein Wald. Es war ein ganzer Dschungel! Durch die Gespräche mit verschiedenen Teams konnten wir ausmachen, dass es in diesem Dschungel durchaus sehr gezielte Verbindungspfade zwischen den Teams gab, doch es ist kaum möglich, auf Dschungelpfaden Geschwindigkeit aufzunehmen. Was wir hier brauchten, waren verlässliche Hochgeschwindigkeitsverbindungen mit klaren Zwischenstopps und einem Fahrplan. Diese Verbindungen durch die entsprechende Koordination herzustellen, war unsere Aufgabe.

Wie nahmen wir diese Herausforderung in Angriff?

1. Der allererste Schritt war, die Ausgangssituation und alle Verbindungspfade zu erfassen.
2. Als Nächstes sahen wir uns die Abhängigkeiten zwischen diesen Pfaden an und schrieben die Arbeitsabläufe (Workflows) auf, um die Abläufe wirklich zu verstehen.

3. Die Abläufe wurden zu Wertströmen zusammengefasst und dabei wurde untersucht, welche Koordination notwendig war.
4. Ganz wichtig vor der Arbeit mit Flight-Level-Systemen: ein Testlauf!
5. Abschließend wurden die Boards in Iterationen erarbeitet und schließlich im Echtbetrieb eingesetzt.

Dieser Prozess dauerte etwas mehr als ein halbes Jahr.

Die Abhängigkeiten identifizieren

Bei der Entwicklung von Produkten bremsen Abhängigkeiten tendenziell das ganze Geschehen aus. Um die schnellsten am Markt werden zu können, mussten wir uns die vorhandenen Abhängigkeiten deshalb ganz genau ansehen. Für diesen Zweck war uns wichtig, die Ist-Situation so faktenbasiert wie möglich zu erheben, um herauszuarbeiten, an welchen Schnittpunkten eine spezielle Koordination notwendig war. Für die Visualisierung der Abhängigkeiten verwendeten wir daher eine „Design Structure Matrix“ (DSM).

Zunächst machten wir die rund 30 involvierten Teams ausfindig und schickten ihnen einen Fragebogen, auf dem diese jeweils anhand einer Skala von 1 (gar nicht) bis 5 (sehr oft) bewerten konnten, mit welchen anderen Teams sie in welcher Intensität zusammenarbeiteten. Dadurch entstand eine Matrix, die wir mit Hilfe einer „Hauptachsentransformation“ sortierten und damit drei große Abhängigkeitsblöcke sichtbar machten (siehe Abbildung 1).

- Links oben sahen wir die Teams, die an Software und Systemen zusammenarbeiteten.
- Rechts unten gruppierte sich das Hardware-Development.
- In der Mitte befanden sich jene Teams, die End-of-Line-Tests durchführen. Wir hatten nicht erwartet, dass wir diese Teams ziemlich losgelöst von Hard- und Softwareentwicklung finden würden, aber hier war es nun einmal so.

In der Sortierung in Abbildung 1 entdeckten wir noch etwas: zwei Koordinationsblöcke (Mitte links), nämlich für Architecture & Safety und für kundenbezogene Hard- und Softwareprojekte sowie Framework-Projekte. Damit hatten wir unsere Ansatzpunkte für ein Flight-Levels-Pilotprojekt gefunden!

Abbildung der Workflows

Bevor man damit beginnt, Abläufe zu strukturieren und weiterzuentwickeln, muss man sich unbedingt genau ansehen, wie die involvierten Teams aktuell arbeiten bzw. zu arbeiten beabsichtigen. Die Abläufe gleich zu Beginn ohne Rücksicht auf die aktuelle Situation völlig umzuschmeißen, ist selten eine gute Idee. Daher zogen wir die Sammlung der „Dschungelpfade“ aus dem ersten Schritt heran, denn diese sind nichts anderes als die Workflows, die sich durch die Zusammenarbeit und Abhängigkeiten zwischen den einzelnen Teams ergeben. Hätten wir allein auf Basis der Design Structure Matrix damit begonnen, Boards zu bauen, hätte das nach unserer Auffassung aber nicht weit geführt. Daran wären nämlich bis zu zwanzigstufige Koordinationsboards entstanden, die wegen des hohen Verwir-

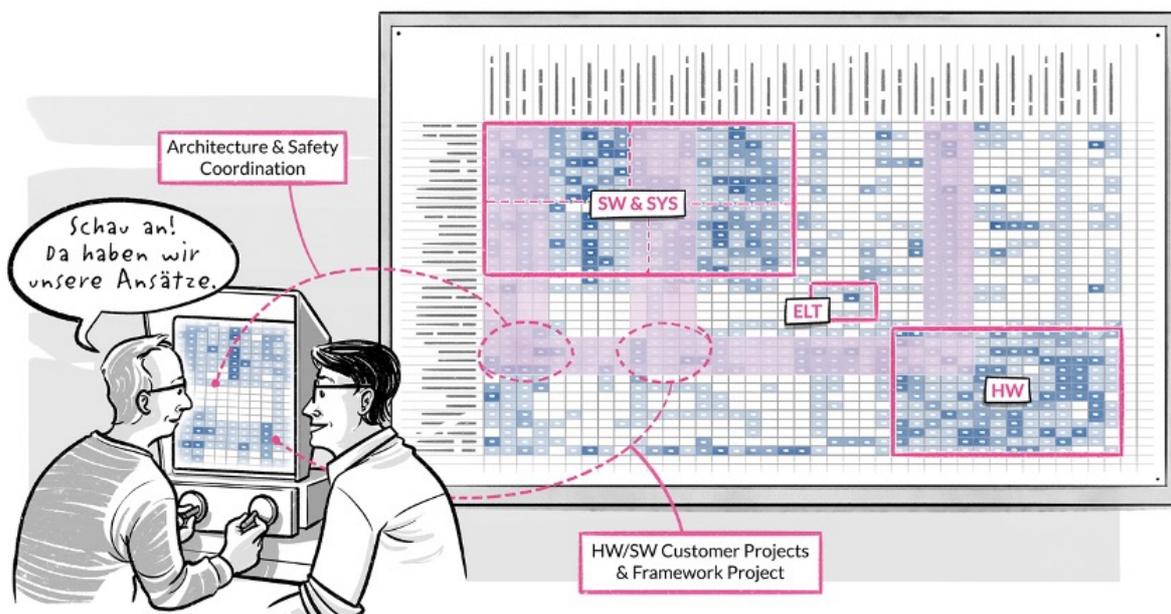


Abbildung 1: Nach der Sortierung zeigten sich drei große Abhängigkeits- sowie zwei Koordinationsblöcke

rungsfaktors und der Kleinteiligkeit nie jemand genutzt hätte. Auch tooltechnisch wäre das nicht abbildbar gewesen.

Wir zogen daher einen Zwischenschritt ein und sahen uns an, ob es in den einzelnen Workflows gleichartige Muster gab, die wir durch intelligente Spaltenbezeichnungen auf maximal zehn Stufen streamlinen konnten. Dazu sammelten wir im ersten Schritt die Flight Item Types, die immer wieder auftraten und sahen uns an, durch welche Trigger die Dschungelpfade eröffnet wurden (siehe Abbildung 2). Wenn zum Beispiel ein „Request for Quotation“ (RFQ) – also ein Angebot – angefordert wurde, war dafür sowohl Expertise notwendig als auch die Nutzung von Synergien innerhalb der Organisation. Sobald wir diese Trigger gesammelt hatten, gingen wir noch einmal ins Detail und folgten dem Dschungelpfad, sprich: Wir konnten so die Workflows (zum Beispiel für RFQs) nachvollziehen und aufschreiben. Mit dieser Information war es dann möglich, die Unterschiede und Ähnlichkeiten zwischen den Workflows herauszufinden.

Insgesamt konnten wir neun zentrale Workflows ableiten, was uns überraschte, denn wir hatten angesichts der erhobenen Abhängigkeiten (siehe Abbildung 1) mit wesentlich mehr gerechnet.

Die Arbeitssystem-Topologie

Im Flight-Levels-Ansatz nutzen wir die sogenannte Arbeitssystem-Topologie, um einen Überblick zu bekommen: Welche Arbeitssysteme gibt es derzeit in der

Organisation auf welchem Flight Level und wie sind diese Systeme miteinander verbunden? Wenn wir dieses Bild zeichnen, können wir erkennen, wo Verbesserungen in den Abläufen erzielt werden könnten.

In einer so großen Organisation ist es notwendig, zwischen Upstream und Downstream zu unterscheiden, also zwischen jenen Bereichen im Arbeitssystem, die strategische Entscheidungen treffen und solchen, in denen darauf basierend die Produkte umgesetzt und geliefert werden. Die Frage ist immer: Wann kann der Downstream mit der Umsetzung beginnen?

Wenn wir Arbeitssystem-Topologien für so große Organisationen erarbeiten, folgen wir daher einer Faustregel: Modelliere und verstehe zuerst den „Point of Commitment“, also jenen Punkt, an dem die Entscheidung für die Umsetzung getroffen und das Go für die Arbeit des nächsten halben Jahres, der nächsten Woche, des nächsten Monats oder Quartals gegeben wird. Dieser Point of Commitment ist der Dreh- und Angelpunkt, der Up- und Downstream miteinander verbindet und sollte daher ziemlich stabil sein.

Abbildung 3 zeigt im Downstream auf Flight Level 1 die Teams (gelb) sowie die Koordinationsebenen auf Flight Level 2. Auf der linken Seite siehst du im Upstream die ADAS Strategie, die über das Portfolio Management mit dem Downstream in Verbindung steht. Das heißt: Das Portfolio Management ist hier der Point of Commitment, an dem entschieden wird, was die Einheiten im Downstream wann tun sollen.

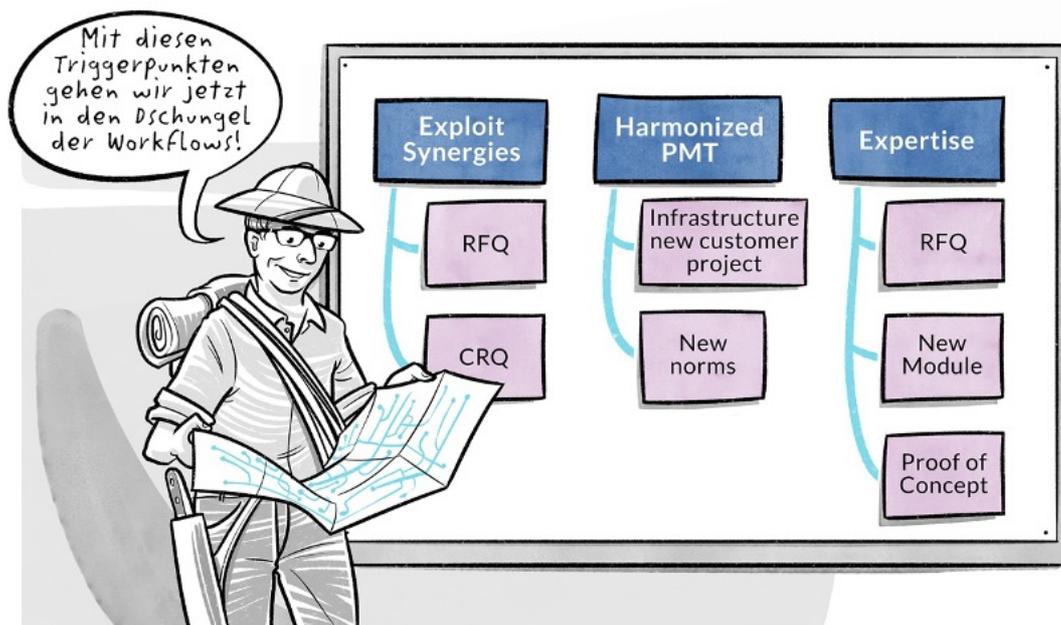


Abbildung 2: Identifizierung der Trigger und Workflows

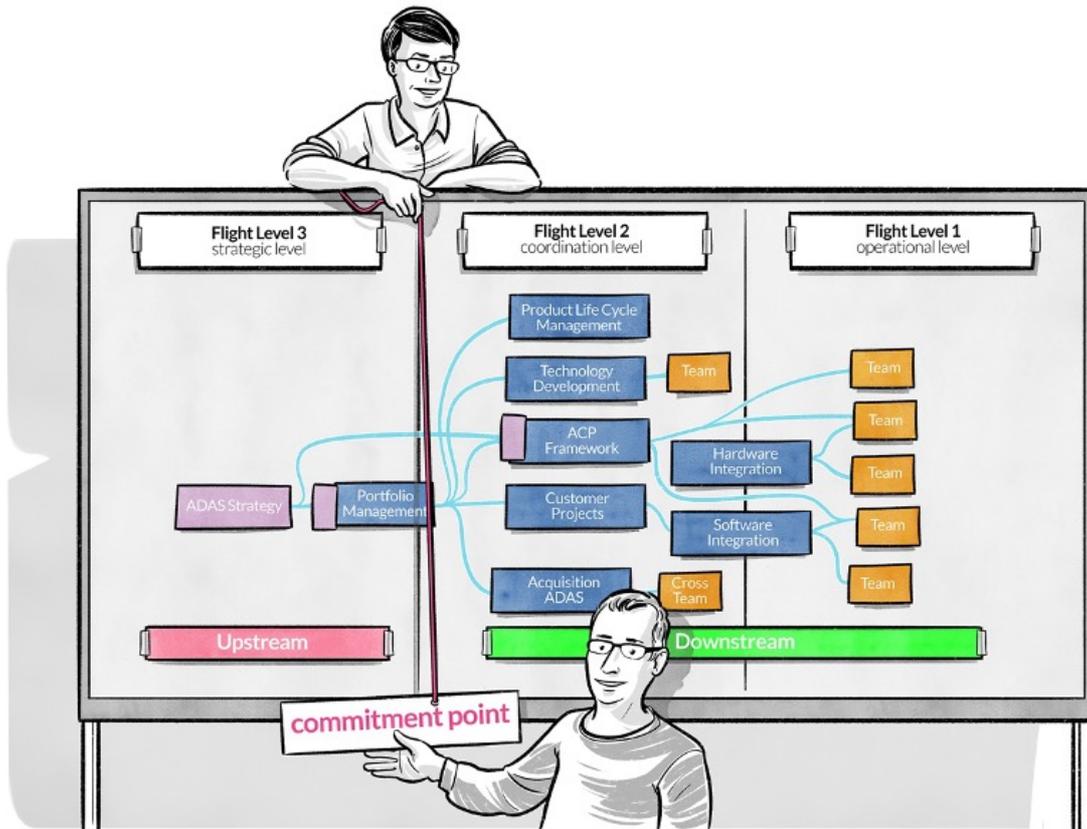


Abbildung 3: Up- und Downstream sowie der Point of Commitment

Warum ist dieser Punkt so wichtig? Es ging der Organisation um Geschwindigkeit und Effizienz, und beides entsteht durch klare Entscheidungswege. Würden sämtliche Einheiten im Downstream selbst entscheiden, was sie wann machen, würde die meiste Zeit mit Diskussionen verlorengelassen. In diesem Fall wäre es nicht mehr möglich, den Fokus für das Gesamtsystem zu halten und es zu steuern.

Durch die eindeutige Benennung des Point of Commitment war nun klar, wo und wie die Koordination auf Teamebene stattfindet und wo sie für die Produkt- und Service-Ebene passiert. Schon allein durch die Simulation von Standardfällen wurde sichtbar, wo in den Workflows schwierige, multidimensionale Interaktionsmuster versteckt waren, die erst recht in Spezialfällen zu kostspieligen Verzögerungen und Entscheidungskonflikten führen würden.

Simulationen

Die Arbeitssystem-Topologie lieferte uns die Grundlage für die Abbildung der Flight Routes. Wie eine solche Route nach einigen Iterationen aussah, zeigt Abbildung 4. Die grünen Kästchen auf der linken Seite sind die Trigger, die Arbeit auslösen. Diese Trigger können auch an anderen Stellen auftreten, die hier nicht alle abgebildet sind. An dieser Flight Route fällt auf, dass sie sehr teamlastig ist und den strategischen Teil ganz ausspart

– eben, weil wir schnelle Verbindungen in der Umsetzung schaffen wollten. Diese Version war das Ergebnis einer iterativen Optimierung und Simulation gemeinsam mit allen Beteiligten, denn die vorhergehenden Flight-Route-Varianten hätten dem Anspruch nach Schnelligkeit und klaren Entscheidungspunkten nicht genügt.

Der Outcome dieses Schritts war, dass alle involvierten Personen die Ende-zu-Ende-Prozesse nun ziemlich gut verstanden und durch den Blick auf das Ganze genau sahen, welche Abläufe funktionieren würden und welche nicht. Haargenau das ist der Sinn der Simulation: Die Abläufe werden nicht im stillen Kämmerlein von einer Person aufgezeichnet und bestimmt, sondern es arbeiten genau jene Leute mit, die wissen, wie die Arbeit durch das System fließt oder bestimmte Erwartungen daran haben. Hier passiert nämlich die Magie: Es entstehen Diskussionen, weil die Menschen über ihre Arbeit sprechen und über das, was sie den Kunden liefern wollen. Durch diese Diskussionen werden nicht nur unterschiedliche Sichtweisen auf ein und dieselbe Gegebenheit deutlich, sondern auch Probleme und Wege, die nicht den gewünschten Erfolg bringen würden. Wir fragen dabei immer gezielt nach solchen „toxischen Pfaden“, die sich leichter vermeiden lassen, wenn sie einmal explizit beschrieben wurden. Es ist ein iterativer Prozess, der zugegebenermaßen einige Zeit

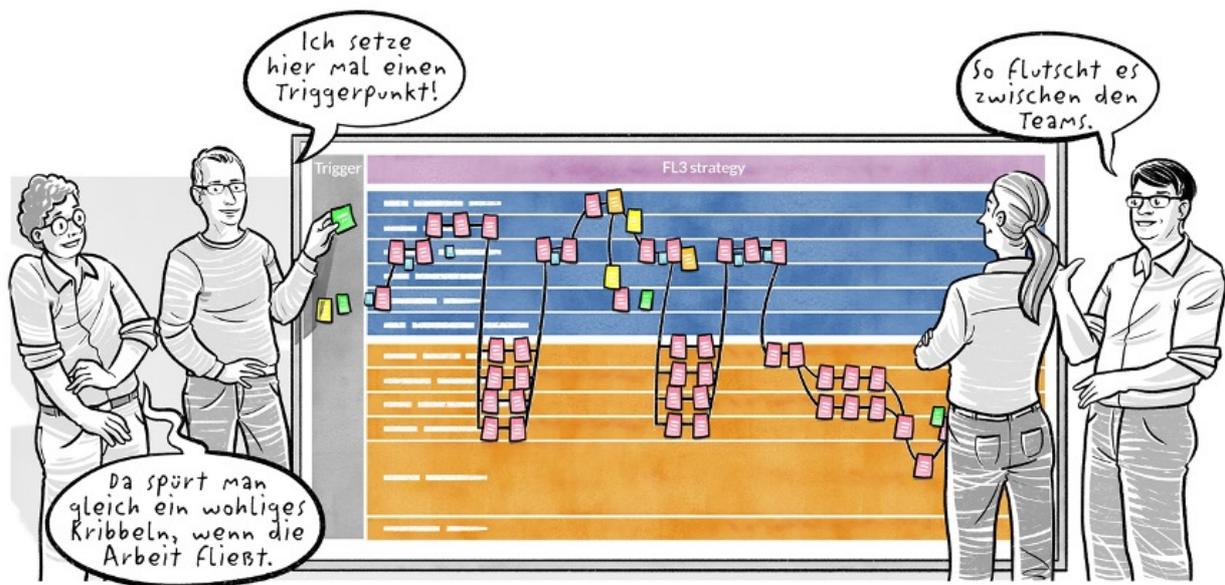


Abbildung 4: Beispiel für eine Flight Route des ACP Framework

braucht, aber essenziell ist, um die Abläufe zu verstehen und auf dieser Basis optimieren zu können.

Das Board Design

Was auf den Flight Routes sichtbar wurde, waren im Grunde Arbeitsergebnisse. Das heißt aber auch, dass zwischen diesen Ergebnissen etwas passiert, es wird etwas Bestimmtes getan. Um zum Beispiel vom Ergebnis „Resource availability feedback“ zum Ergebnis „Realistic plan proposal“ zu gelangen, muss es ein Update der Timeline geben. Hier „versteckt“ sich sozusagen die Arbeit. Die Aktivität „Update timeline“ wird später zu einer Spalte am Board werden, zumindest in den ersten Entwürfen. Aus diesen Aktivitäten bastelten wir in einem Tool erste Board-Prototypen, über die wir mit den beteiligten Personen weiter diskutieren konnten.

Das Board Design passierte also parallel zur Simulation, weil wir sofort die Aktivitäten formulierten, die von einem Zwischenergebnis zum nächsten führen. Uns ist aufgefallen, dass wir dabei auf einiges achten müssen:

- Die Aktivitäten sollten so formuliert werden, dass sich darunter mehrere Flight Item Types subsummieren lassen.
- Bei der Benennung der Aktivitäten ist meistens etwas Modellierung notwendig, weil zum Beispiel unterschiedliche Ingenieursdisziplinen dieselbe Arbeit unterschiedlich benennen. Deshalb muss ein wenig herumprobiert werden, bis eine gemeinsame Sprache gefunden ist.
- Das deutet schon darauf hin, dass das Board Design ebenfalls ein iterativer Prozess ist. Die involvierten

Personen machen einen ersten Entwurf, probieren das Board mit Use Cases aus der Vergangenheit aus, mit Use Cases für die Zukunft, und kommen dann irgendwann zu einer Schlussfolgerung: Entweder ist das Board gut genug, um damit zu starten, oder es muss noch weiter modifiziert werden.

Nachdem wir ein „vorläufig endgültiges“ Board Design hatten, luden wir die Teams zu einer Simulation mit einem physischen Board ein, um den gesamten Ablauf mehrmals durchzuspielen. Abschließend fragten wir die teilnehmenden Personen nach ihren Eindrücken. Die Antworten zeigten uns, wie wichtig Simulationen sind, bevor man mit einer Flight-Levels-Systemarchitektur in den Echtbetrieb geht: „Durch diesen Workshop haben wir jetzt ein gemeinsames Verständnis“, „Wir können mit diesem Board für das Management transparent machen, welche Teile der Organisation die größte Arbeitslast tragen“, „Wir verstehen jetzt, wie unsere Abläufe in Zukunft funktionieren sollten und wie wir sie visualisieren können“, „Wir können damit transparent zeigen, was die Koordinationsebene macht“. Das finale, „echte“ Board bildeten wir in Confluence mit JIRA-Tickets ab.

Was haben wir erreicht?

Natürlich standen wir in diesem gesamten Prozess auch Herausforderungen gegenüber, auf die wir keinen Einfluss hatten. 2023 gab es zwei große Veränderungen in der Organisation, die unsere anfänglichen Pläne, Flight Levels über das ACP Framework hinaus einzusetzen, über den Haufen warfen. Dennoch haben wir viel erreicht:

- Wir konnten die Wertströme und die wichtigsten Koordinationspunkte identifizieren.
- Wir haben einen Prototyp für die Systemarchitektur entwickelt und konnten das Potenzial des Konzepts belegen.

Trotz der Veränderungen war es wichtig aufzuzeigen, welche Möglichkeiten es gibt, um in Zukunft mit Flight Levels noch schneller zu werden. Das möchten wir dir auch mitgeben: Gerade in großen Organisationen wird es immer wieder plötzliche seismische Verwerfungen geben, die niemand vorhersehen kann. Wichtig ist das, was man dennoch auf diesem Weg lernt: Die Menschen im Bereich ADAS und wir wissen durch die zahlreichen Iterationen jetzt auf jeden Fall, wie der Shinkansen durch den Dschungel fahren kann. Wir haben gelernt, welche Workshops man mit welchen Gruppen machen kann und welche nicht. Also egal, wie es läuft, ob du ein Flight Level Design völlig umsetzen kannst oder nicht: Du wirst auf jeden Fall immens viel lernen.



Bernd Kudicke war einer der ersten Flight Level Coaches bei Bosch und unterstützt die Organisationsgestaltung und -transformation des Unternehmens. Als Flight Level Coach und Viable-System-Model-Coach verbindet und optimiert er die verschiedenen Workflow-Systeme, um sie zu

Arbeitssystemtopologien mit maßgeschneiderten Entscheidungsstrukturen und Kommunikationskanälen zu machen. Am Flight Level Thinking Modell schätzt er die starke Co-Creation mit seinen Kunden, um exzellente Lösungen für organisatorische Schmerzpunkte zu schaffen.



Sven Surrey zählt zu den Pionieren von Agile@Bosch und ist heute einer der Agile Enterprise Coaches bei Bosch Digital. Bis Oktober 2023 war er für die Organisationsentwicklung des Produktbereichs ADAS Compute verantwortlich und war somit bei Bosch auch einer der Ersten mit

Erfahrungen in der agilen Hardwareentwicklung. Seit 2018 beschäftigt sich Sven verstärkt mit der Transformation traditioneller Linienorganisationen in agile Führungssysteme, die strategische Veränderungen antizipieren und die Agilität des Unternehmens fördern. Dafür nutzt er das Viable-System-Model und seit 2019 auch Flight Levels.