

Standing Up the Functional Safety Program

A working playbook for the first functional safety hire on an autonomous systems program — twelve steps from standards landscape to production monitoring, with AI/ML assurance and cross-domain robotics risk-rating built in alongside the automotive core. Includes worked examples, decision matrices, ready-to-use templates, and a glossary of every term you'll hear in your first week.

ISO 26262

ISO 21448 SOTIF

ISO/PAS 8800

ISO 13849

UL 4600

ISO/TS 5083

ISO 10218

ISO/TS 15066

IEC 61508



Patrick Salem

Technical Program & Engineering Leader · Autonomous Systems & Robotics

[linkedin.com/in/patricksalem](https://www.linkedin.com/in/patricksalem)

September 2026

YOU GET THE STANDARDS. YOU DON'T GET THE SEQUENCE.

A 12-step starting map for the first functional safety hire.



AUTOMOTIVE TRACK

ISO 26262 • ISO 21448 (SOTIF)

Cybersecurity mapped alongside safety



ROBOTICS TRACK

IEC 61508 • ISO 10218

ISO/TS 15066 for collaborative applications

THE 12-STEP STARTING MAP

1 FRAME THE PROGRAM

1

Scope the system

Define use, boundaries, ODD

2

Map the standards

Applicability and interfaces

3

Set governance

Roles, reviews, independence

4

Plan the lifecycle

Activities and work products

2 ENGINEER THE SAFETY

5

Analyze hazards

Misuse and insufficiencies

6

Classify the risk

ASIL, SOTIF, robot risk

7

Derive requirements

Goals, concepts, criteria

8

Allocate architecture

Systems and interfaces

3 DEFEND THE CASE

9

Plan verification

Methods, evidence, coverage

10

Manage change

Interfaces and cyber risk

11

Build traceability

Claims linked to evidence

12

Compile safety case

Review, challenge, defend

ONE DEFENSIBLE THREAD

HAZARD → REQUIREMENT → EVIDENCE → CLAIM

EACH STEP IS DETAILED IN THE PAGES THAT FOLLOW

Starting map—not a substitute for your program's tailored safety plan.

Table of Contents

Twelve steps, roles and responsibilities, two diagrams, five decision matrices, three ready-to-use templates, and a glossary — page references below.

01	Map the standards landscape	4
02	Stand up the safety organization	6
	Roles & responsibilities (RACI)	6
03	Define the item & scope	7
04	Run HARA & assign ASIL	7
05	Write safety goals & the Functional Safety Concept	10
06	Assess SOTIF exposure	11
07	Build the technical safety concept	13
08	Tailor verification & validation	15
09	Integrate cybersecurity	15
10	Extend safety to suppliers	16
11	Compile the safety case	16
12	Monitor after SOP	18
	How it fits together (document flow & V-model)	18
	Decision matrices	20
	Templates (HARA, ASIL rating, Safety Plan)	22
	Glossary	27

01 Map the standards landscape

Before touching a program, know which document governs which question — you'll be asked to cite this in every design review. The same underlying discipline (IEC 61508) branches differently depending on whether you're on-road, off-road, or on a factory floor. One terminology note that applies throughout this playbook: "L2/L3/L4/L5" refers to **SAE J3016**'s levels of driving automation — it's a taxonomy, not a safety standard, but every standard below assumes you know which level you're building to.

WHAT IT GOVERNS	ON-ROAD / ADAS	OFF-ROAD / ROBOTICS
Base functional safety random HW faults & systematic errors	ISO 26262	IEC 61508 (the base standard both sides derive from)
Safety of the intended function performance limits, nothing actually broken	ISO 21448 (SOTIF)	<i>no dedicated equivalent — argued via conservative PL/SIL selection</i>
AI/ML-based perception & decision components	ISO/PAS 8800	<i>no dedicated standard yet</i>
Risk-rating method for safety-related control systems	ASIL ISO 26262-3, Matrix 1	PL / SIL ISO 13849, IEC 62061
Full-autonomy / ADS safety case no driver or operator fallback	UL 4600 + ISO/TS 5083 (2025)	ISO 3691-4 + ANSI/A3 R15.08, for AMRs
Software updates (OTA)	ISO 24089 + UNECE R156 (regulation)	<i>not covered in this playbook's off-road scope</i>
Cybersecurity	ISO/SAE 21434 + UNECE R155 (regulation)	<i>not covered in this playbook's off-road scope</i>

A quick orientation, not the full list — robot and machine product-level standards (ISO 10218, ANSI/RIA R15.06, ISO/TS 15066, ISO 3691-4) and process-maturity models are named in full below. UNECE R155/R156 are regulations, not standards — the standards satisfy them. See Matrix 4 (Decision Matrices) for how the risk-rating scales themselves compare.

ON-ROAD: AUTOMOTIVE & ADAS

- **ISO 26262:2018** (2nd ed.) — functional safety of electrical/electronic systems (the "known failure" standard: random hardware faults and systematic errors).
- **ISO 21448:2022** (SOTIF) — safety of the intended function: the system does what it's designed to do, but the design itself has performance limits (the ADAS-specific standard).

- **ISO/PAS 8800:2024** — safety for AI-based road-vehicle functions (perception models, ML components) — increasingly required alongside SOTIF for camera/lidar-based stacks. Still a "Publicly Available Specification," not yet a full ISO standard, but already being used as the basis for supplier certification.
- **ISO/SAE 21434:2021** — cybersecurity engineering; safety and security work products must be co-managed, not siloed.
- **ISO/TS 5083:2025** — safety for automated driving systems: design, verification, and validation. New (2025) and the ISO-side counterpart to UL 4600 — it layers on top of ISO 26262/SOTIF/8800 specifically for programs with an actual ADS (L3+), rather than replacing them.
- **UL 4600** (3rd ed.) — consensus safety case standard for fully autonomous systems (L4/L5 road vehicles); use if the program includes an ADS, not just driver-assist.
- **ISO 24089:2023** — software update engineering (the OTA standard); typically paired with UNECE R156 below for programs that update software in the field.
- **UNECE R155 / R156** — not ISO standards but EU/UK/Japan-market type-approval *regulations*: R155 requires a Cybersecurity Management System (CSMS, satisfied in practice via ISO/SAE 21434), R156 requires a Software Update Management System (SUMS, satisfied via ISO 24089). Know the difference between "the standard" and "the regulation that requires it" — an assessor will.

OFF-ROAD: ROBOTICS & AUTONOMOUS MOBILE SYSTEMS

- **IEC 61508** — the base functional safety standard everything above derives from; the right reference when a program spans multiple domains.
- **ISO 10218-1/-2:2025** — safety requirements for industrial robots and robot systems/integration. The 2025 revision folded cobot requirements (formerly ISO/TS 15066) directly into the base standard and added explicit cybersecurity requirements — confirm which edition your supply chain is actually certified to.
- **ISO/TS 15066** — collaborative robot (cobot) safety: force/pressure limits for shared human-robot workspaces. Being absorbed into ISO 10218 (above); still the standalone reference until your program's robots are certified to the 2025 edition.
- **ANSI/RIA R15.06** — the U.S. national safety standard for fixed industrial robot arms, harmonized with ISO 10218. Not the right citation for a mobile robot — see R15.08 below.
- **ANSI/A3 R15.08** (Parts 1–3) — the U.S. standard specifically for industrial mobile robots (AMRs): the robot itself, the system/fleet, and the application/integration. This, not R15.06, is the standard a U.S. AMR program should be citing.
- **ISO 3691-4:2023** — safety for driverless industrial trucks and AMRs — the closest off-road analog to UL 4600, and the international counterpart to R15.08.
- **ISO 13849-1:2023 / IEC 62061** — safety-related control system design for machinery; this is where Performance Level (PL) and machinery SIL come from (see the cross-domain matrix, page reference in Decision Matrices).

Process maturity still matters on both sides: **Automotive SPICE (ASPICE)** for road-vehicle suppliers — and, where cybersecurity work products are in scope, **Automotive SPICE for Cybersecurity (VDA)** alongside it — or a program's own capability model for robotics suppliers not already in the automotive supply chain.

COMMON PITFALL

Treating ISO 26262 and SOTIF as interchangeable, or assuming compliance with one covers the other. A program can be fully ISO 26262 compliant and still ship a hazardous SOTIF failure mode — they answer different questions.

02 Stand up the safety organization

ISO 26262-2

Functional safety fails without organizational backing before it fails on engineering merit. Get this in place first.

- Get a **Functional Safety Manager (FSM)** formally appointed with authority independent of program schedule/cost pressure — this may be you.
- Write the **Safety Plan**: scope, milestones, work products, tailoring rules, confirmation measures, tool qualification approach.
- Define **independence requirements** by ASIL (who can review/approve their own work vs. who needs a separate reviewer or department).
- Set up a **competency matrix** — ISO 26262 expects documented evidence that people doing ASIL C/D work are qualified to do it.
- Establish a **safety culture** mechanism: anonymous escalation path, "stop the line" authority, lessons-learned loop from field issues.

COMMON PITFALL

Appointing an FSM who reports into the program schedule owner. Independence on an org chart means nothing if the person approving a safety gate also owns the ship date they'd be delaying.

Roles & responsibilities (RACI)

Who's accountable at each step — the question every new hire asks and most safety plans answer only vaguely. R = does the work, A = owns the outcome and signs off, C = consulted before the decision, I = told after.

☒ R — Responsible ☐ A — Accountable ☐ C — Consulted ☐ I — Informed

Step	FSM	Systems Eng.	Safety Eng.	Independent Assessor	Supplier
01 Standards landscape	A	R	R	I	I
02 Safety organization	A/R	C	C	I	I

03 Item & scope	A	R	C	I	C
04 HARA & ASIL	A	C	R	I	I
05 Safety goals & FSC	A	R	R	C	I
06 SOTIF	C	R	A/R	C	C
07 Technical safety concept	I	R	A	C	R
08 Verification & validation	I	C	A	C	R
09 Cybersecurity	C	C	R	I	C
10 Suppliers	A	C	C	I	R
11 Safety case	A	C	R	R	C
12 Monitor after SOP	A	C	R	I	C

Step 06 (SOTIF) and Step 11 (safety case) are the two places accountability most often gets fumbled — SOTIF because it's easy to assume the safety engineer owns it alone (systems engineering has to be equally hands-on), and the safety case because the independent assessor's R here is real review work, not a rubber stamp.

03 Define the item & scope

ISO 26262-3 Cl.5

Every downstream document inherits from this. Get it reviewed and baselined before HARA starts.

- Write the **Item Definition**: functional boundaries, interfaces, operating environment, dependencies on other vehicle or robot cell systems.
- Define the **Operational Design Domain (ODD)** — speed range, road type or workspace layout, weather, lighting, human proximity — this drives ASIL/PL/SIL scope alike.
- Confirm what's **reused vs. new** — inherited safety cases from a prior platform need a gap/impact analysis, not a rewrite.

COMMON PITFALL

Freezing the item definition and never revisiting it. A new sensor, a changed ODD, or a late architecture change silently invalidates the HARA built on top of it — nobody notices until an assessor asks for the traceability.

04 Run HARA & assign ASIL

ISO 26262-3 Cl.6

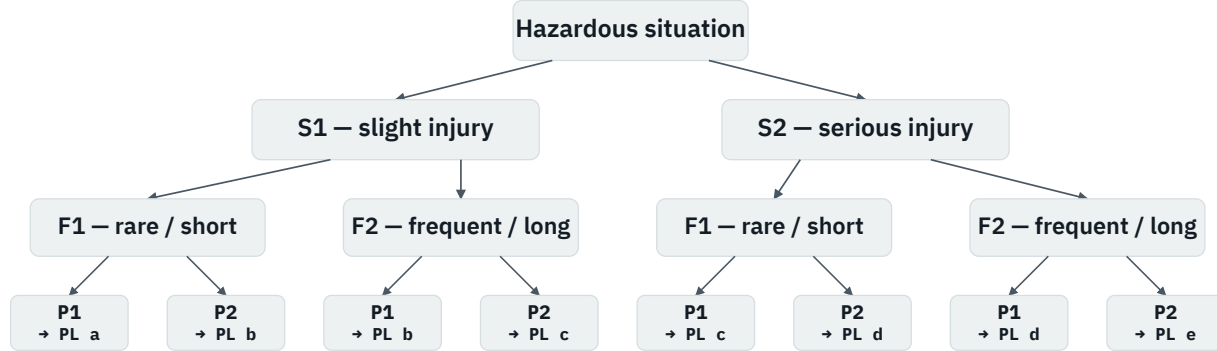
The Hazard Analysis and Risk Assessment is where the classification matrix in this playbook gets used for real. Do this with a cross-functional team (systems, test, field data if available) — not solo.

- Brainstorm **malfunctioning behaviors** per function, then pair each with an operational situation to get a hazardous event.
- Rate each event on **Severity (S0–S3), Exposure (E0–E4), Controllability (C0–C3)** — use Matrix 1 (page reference in the Decision Matrices section) to derive ASIL.
- Write a **Safety Goal** for every hazardous event above QM — one sentence, top-level, ASIL inherited from the hazard.
- Log everything in a **traceability tool** (DOORS, Polarion, Codebeamer) — this becomes the backbone of your entire safety case.

Worked example — Automatic Emergency Braking (AEB)

MALFUNCTION	Unintended full braking at highway speed with no obstacle present.
OPERATIONAL SITUATION	Highway driving, following traffic, high speed, host vehicle ahead of following traffic.
HAZARDOUS EVENT	Following vehicle cannot stop in time and rear-ends the host vehicle after an unexpected hard brake.
SEVERITY	S3 — rear impact at highway closing speed carries life-threatening/fatal potential.
EXPOSURE	E4 — highway following traffic is a high-probability, near-daily operating situation.
CONTROLLABILITY	C3 — the following driver has little to no time to react to a sudden, unannounced stop.
ASIL	D — read directly off Matrix 1 at S3 × E4 × C3.
SAFETY GOAL	"The AEB function shall not induce braking that following traffic cannot reasonably react to." (ASIL D, inherited from the hazard.)

The same hazard-to-rating discipline applies outside road vehicles — only the method and the vocabulary change. A collaborative robot arm, an AMR, or a machine-safety application typically runs this step under **ISO 13849-1** (or IEC 62061 for SIL) instead of ISO 26262, deriving a **Performance Level (PL a–e)** from Severity, Frequency/duration of exposure, and Possibility of avoidance rather than ASIL's S/E/C triple. The underlying question is identical — "how bad, how often, how avoidable" — so a team fluent in HARA can usually pick up a 13849 risk graph in an afternoon.



Simplified for illustration — P is Possibility of avoidance (P1 = possible under specific conditions, P2 = scarcely possible). The full ISO 13849-1 risk graph (Annex A) and its informative notes are the reference; use it, or a tool like SISTEMA, for an actual determination.

Worked example — collaborative robot arm (ISO 13849-1)

HAZARDOUS SITUATION	Robot arm continues motion after an operator crosses into the shared collaborative workspace.
SEVERITY (S)	S2 — contact at operating speed/force can cause serious, potentially irreversible injury.
FREQUENCY (F)	F2 — operators enter the shared workspace frequently during normal cycle operation.
AVOIDANCE (P)	P1 — avoidance possible: the arm runs at reduced collaborative speed and is visible.
PERFORMANCE LEVEL	PLd — read off the graph above at S2 × F2 × P1; the safety function (speed/separation monitoring that stops or slows the arm) must be designed to achieve PLd, typically via a dual-channel, safety-rated encoder architecture with a Category 3 structure.
SAFETY FUNCTION	"The safety-rated monitored stop function shall halt all hazardous motion before the operator can reach a point of contact with the arm." (PLd required.)

The simplified ratings above are for illustration only — an actual Performance Level determination must use the full ISO 13849-1 risk graph (Annex A) or a validated software tool (e.g., SISTEMA), and should account for the specific safety function's architecture (Category B/1/2/3/4) and diagnostic coverage, not just the S/F/P inputs shown here.

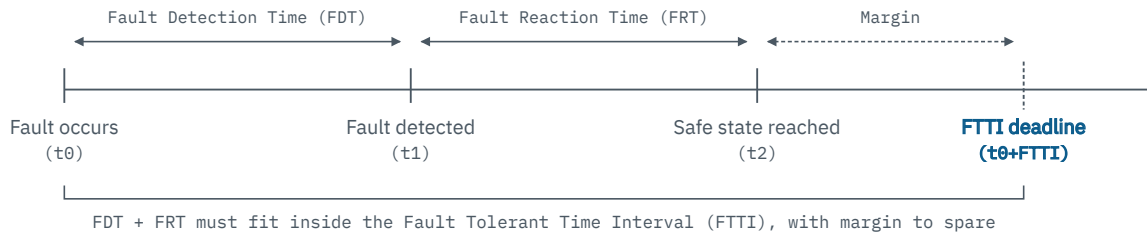
COMMON PITFALL

Running HARA once at concept phase and calling it done. It needs to be re-opened whenever the item definition, architecture, or ODD changes — a stale HARA is worse than no HARA, because it creates false confidence.

05 Write safety goals & the Functional Safety Concept

ISO 26262-3 C1.7

- Derive **Functional Safety Requirements** from each safety goal — allocate them to preliminary architecture (sensing / perception / decision / actuation).
- Specify **safe states** and the fault tolerant time interval for each — this is what your fault handling design has to hit.
- Define warning/degradation strategy — what the driver, operator, or nearby human sees when a fallback triggers.



The budget every fault-handling design has to hit: from the moment a fault occurs (t_0), the system has until the FTTI deadline to detect it and reach a safe state. Spend that budget on detection and reaction, and leave margin — a design that exactly meets FTTI has no tolerance for a slow diagnostic cycle.

COMMON PITFALL

Writing safety goals that are really design requirements in disguise. A safety goal states what must not happen ("unintended acceleration shall not occur"), not how the system prevents it — baking in a specific implementation here locks out better designs later.

06 Assess SOTIF exposure

ISO 21448

This is the step most hardware-background engineers underweight, and the one that matters most for ADAS. ISO 26262 asks "did the part fail?" SOTIF asks "did the system do something unsafe even though nothing failed?" — a perception model missing a pedestrian in glare is a SOTIF problem, not a hardware fault.

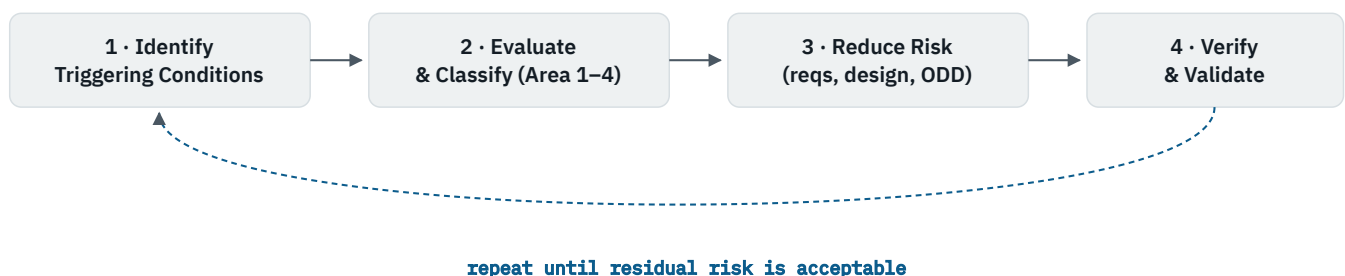
- Identify **triggering conditions** that could expose performance limitations (sensor degradation, edge-case scenarios, adverse weather).
- Classify each scenario using the **Area 1–4 framework** (Matrix 3) and drive Area 3 scenarios down through testing.
- Plan a **scenario-based test strategy** — simulation for coverage breadth, track/fleet or cell/site testing for the scenarios simulation can't validate.

COMMON PITFALL

Treating SOTIF as a single pass instead of the iterative cycle it's designed to be. "We did SOTIF" is a category error — the identify / evaluate / reduce / verify loop (see below) keeps running as long as the vehicle is on the road.

SOTIF DEEP DIVE – THE ISO 21448 CYCLE

Unlike ISO 26262's V-model, ISO 21448 isn't a one-pass decomposition — it's a loop you keep running. Four stages, repeated until residual risk is acceptable:



Each pass through the loop should shrink Area 3 (unknown/unsafe) toward Area 1 (known/safe) — see Matrix 3. A program that runs this once at concept phase and stops has not done SOTIF; it's done a SOTIF-shaped checklist item.

TRIGGERING CONDITIONS – WHAT ACTUALLY SHOWS UP IN STAGE 1

Category	Example	Typical mitigation
Sensor performance limitation	Camera loses contrast in low light or glare; radar multipath near guardrails	Sensor fusion, redundant modality, confidence-weighted blending

Environmental conditions	Heavy rain, fog, or snow occludes the field of view	Degrade gracefully, reduce ODD, add weather-aware confidence thresholds
Unusual object geometry	Atypical vehicle shape, construction equipment, a trailer at an odd angle	Expand training/validation data, add geometric heuristics as a backstop
Algorithm insufficiency	A rare object class or pose the model wasn't trained on (the classic "edge case")	Targeted data collection, scenario mining from fleet data, conservative fallback behavior
Foreseeable misuse	Driver overrides or disables a system in a way that's predictable but not intended	HMI redesign, restrict override conditions, driver monitoring

WORKED EXAMPLE – WALKING A SCENARIO THROUGH THE CYCLE

AEB fails to recognize a cyclist at dusk

TRIGGERING CONDITION	Low ambient light combined with dark, low-contrast clothing reduces the camera's detection confidence below threshold.
INITIAL CLASSIFICATION	Area 3 — unknown/unsafe, discovered through track testing rather than anticipated in advance.
RISK REDUCTION	Add radar/thermal fusion weighting at low ambient light, and lower the confidence threshold for braking specifically in low-light conditions within the validated ODD.
RE-VERIFICATION	Targeted low-light cyclist test matrix (varied clothing contrast, approach angle, closing speed) in addition to the standard regression suite.
FINAL CLASSIFICATION	Area 1 — known/safe, once test results show detection performance meets the acceptance criteria across the matrix. Moves into the regression suite permanently.

THE RELEASE ARGUMENT – HOW TEAMS ACTUALLY SAY "ACCEPTABLE"

Shrinking Area 3 never reaches zero — SOTIF's residual risk has to be argued, not eliminated. Three argument types cover most real safety cases, usually combined rather than used alone:

Argument type	What it claims	Where it's strongest
Exposure-based	Scenario frequency × severity falls below an agreed acceptance threshold	Rare, well-characterized triggering conditions
Comparative / baseline	System performance meets or exceeds a human-driver baseline for the same scenario class	Functions with a direct human-performance analog (e.g., AEB vs. human reaction time)
Field-data confidence	Sufficient test miles or scenario exposure without incident, at a stated statistical confidence level	Mature programs with large validation fleets or simulation libraries

None of these produce a clean "zero risk" answer, and an assessor will expect to see the argument, the data behind it, and its limitations stated explicitly — not just a pass/fail test report.

WHEN THE TRIGGERING CONDITION IS THE MODEL ITSELF – ISO/PAS 8800

SOTIF's identify/evaluate/reduce/verify cycle treats a perception failure as a triggering condition to be found and mitigated. It doesn't, by itself, say anything about **how the ML model was built, trained, and bounded** — and an assessor will ask. **ISO/PAS 8800** (AI/ML safety for road vehicles) fills that gap: it extends ISO 26262's lifecycle and SOTIF's performance-limitation framing with ML-specific assurance activities, so "the model" gets the same traceable rigor as a hand-written software component instead of being treated as a black box with a confusion matrix attached.

Lifecycle phase	ML-specific assurance activity	Typical evidence
Data	Define and document the operational design domain's data distribution ; assess training/validation/test data for coverage, bias, and provenance; version and freeze datasets tied to a model release	Data specification, coverage reports against the ODD, dataset lineage/versioning records
Model	Specify the model's intended function and known limitations (an "insufficiency" statement, not just an accuracy number); select architecture and training process with traceable rationale; manage model versions like any other safety-relevant component	Model function & insufficiency specification, architecture rationale, model version/configuration record
Verification	Verify performance against the documented ODD and insufficiency statement, not just a held-out test split; stress-test against the triggering conditions identified in the SOTIF cycle; quantify and bound performance degradation at the edges of the ODD	Performance verification report mapped to ODD/triggering conditions, robustness/stress-test results
Operation	Monitor for distributional shift between the deployed operating environment and the training/validation data; define retraining and re-verification triggers; feed field data back into the SOTIF cycle and the dataset record	Field-monitoring plan, drift-detection criteria, retraining/re-release trigger definitions

COMMON PITFALL

Treating the ML model as "just another software component" for traceability purposes — tracing a requirement to a module and calling it done. A conventional software unit either meets its spec or doesn't; a model has a *statistical* insufficiency that never fully closes, so the traceability has to connect the requirement to the data, the training process, and a bounded performance claim — not to a single pass/fail test result.

07 Build the technical safety concept

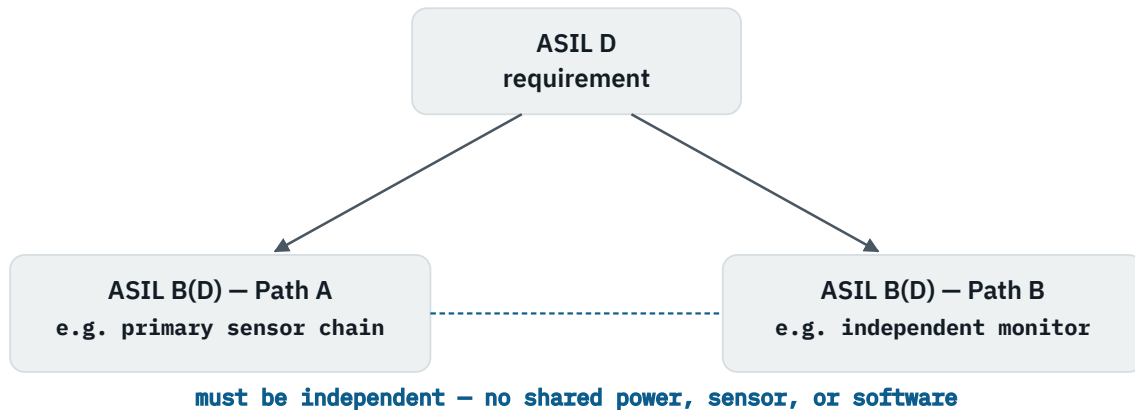
ISO 26262-4/5

- Run **FMEA** (design and process) on each safety-relevant component; run **FTA** top-down from each safety goal to catch gaps FMEA misses.
- For hardware, calculate **FMEDA metrics** — Single-Point Fault Metric (SPFM), Latent Fault Metric (LFM), Probabilistic Metric for Hardware Failures (PMHF) — targets are in Matrix 2.

- Decide where **redundancy or diagnostic coverage** is needed vs. where architectural decomposition (splitting ASIL D into two ASIL B(D) paths) is more cost-effective.

ASIL DECOMPOSITION – THE MECHANICS

Decomposition splits one high-ASIL requirement across two architectural elements so each can be developed to a lower, cheaper ASIL — but only if the two elements genuinely can't fail from the same cause. The "(D)" notation marks an element as decomposed from ASIL D; it still traces back to the original requirement.



If Path A and Path B share a power rail, a sensor, or a software library, a single failure can take out both — the decomposition argument collapses and the requirement is really still ASIL D in disguise.

Original ASIL	Common decomposition options
D	$B(D) + B(D) \cdot C(D) + A(D) \cdot D + QM(D)$
C	$B(C) + A(C) \cdot C + QM(C)$
B	$A(B) + A(B) \cdot B + QM(B)$
A	$A + QM(A)$

Read " $D + QM(D)$ " as: one path is developed to the full original ASIL on its own, and the other carries no safety requirement — valid only when the first path alone can be shown to fully achieve the safety goal independently.

COMMON PITFALL

Decomposing ASIL D into two ASIL B(D) paths that aren't actually independent — a shared power rail, shared sensor, or shared software library undoes the whole argument. Decomposition only reduces rigor if the independence is real and demonstrable, not assumed.

08 Tailor verification & validation

ISO 26262-6/8

- Set **test method rigor by ASIL** — see Matrix 2: black-box vs. requirements-based vs. fault injection vs. formal verification.
- Define **tool confidence levels (TCL)** for any tool whose output isn't independently checked (compilers, static analyzers, simulation tools).
- Build a **V&V traceability matrix** tying every safety requirement to at least one test case and one result.

COMMON PITFALL

Treating a green test suite as proof of safety rather than one piece of evidence in an argument. A high pass rate with no traceability back to specific safety requirements won't survive a functional safety assessment.

09 Integrate cybersecurity

ISO/SAE 21434

- Run **TARA** (Threat Analysis and Risk Assessment) alongside HARA — a security vulnerability that enables an unsafe state is a joint safety/security finding, not either team's alone.
- Align **Cybersecurity Assurance Levels (CAL, 1–4)** with ASIL where a threat path could cause a safety-relevant malfunction — CAL is set from impact severity and how reachable the attack vector is (network-accessible rates higher than physical-only access), and it's informative in ISO 21434 but increasingly contractual in supplier agreements.
- Establish a joint **safety-security review gate** so neither team ships a change the other hasn't seen.
- If the program needs EU, UK, or Japan type approval, know that **ISO/SAE 21434 compliance is how you satisfy UNECE R155** (the regulation that actually requires a Cybersecurity Management System) — the standard and the regulation are not interchangeable terms, and an assessor will expect both cited correctly.

For suppliers, **Automotive SPICE for Cybersecurity (VDA)** increasingly travels alongside standard ASPICE assessments — confirm whether your program's supplier agreements require it before the first audit, not after.

COMMON PITFALL

Running TARA and HARA as separate workstreams that never talk to each other. A security vulnerability that can trigger a safety-relevant malfunction is a joint finding — siloed teams typically catch it only at integration, when it's expensive to fix.

10 Extend safety to suppliers

ISO 26262-8 Cl.5

This is where a CDRL/PPAP background is a direct advantage — treat it as an extension of deliverable management, not a separate discipline.

- Issue a **Development Interface Agreement (DIA)** per supplier defining split of safety responsibilities and confirmation measures.
- Require **safety case evidence as a CDRL line item** — DFMEA, PFMEA, FMEDA results, ASIL-appropriate test reports — gated at PPAP.
- Audit supplier **process capability (ASPICE level)** against the ASIL of the parts they're delivering, not just cost/quality/delivery.

COMMON PITFALL

Accepting a supplier's FMEDA numbers without auditing the fault model behind them. A diagnostic coverage claim is only as good as the assumptions it was calculated against — ask to see the fault list, not just the percentage.

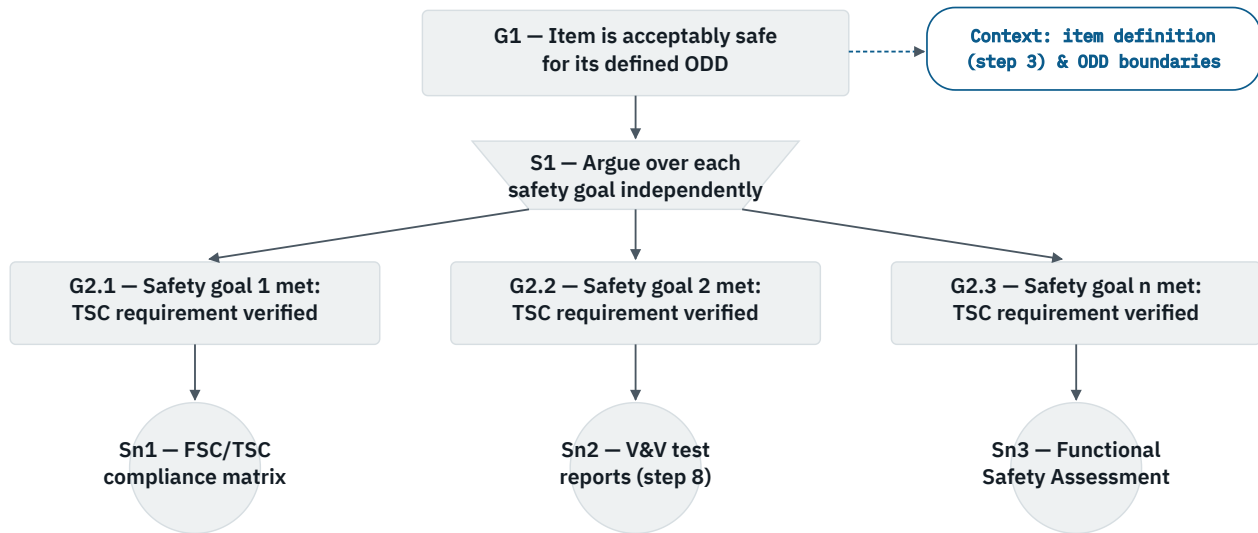
11 Compile the safety case

ISO 26262-2 Cl.6

- Assemble the **Safety Case** — the argument, with evidence references, that each safety goal is achieved.
- Run **Confirmation Reviews** (safety plan, safety case, safety validation) and a **Functional Safety Assessment** for ASIL C/D items, performed with organizational independence from the development team.
- Track open items to closure before SOP — an assessment with unresolved findings is not a pass.

STRUCTURING THE ARGUMENT — GSN, LITE

You don't need a GSN tool to benefit from GSN (Goal Structuring Notation) thinking. The shape is always the same: a top goal, a strategy for how you're arguing it, sub-goals that strategy breaks into, and evidence that actually closes each sub-goal. A safety case that skips straight from "system is safe" to a pile of test reports is missing the argument — the strategy and sub-goals are what tell a reviewer **why** that evidence is sufficient.



Rectangles are goals (claims to be proven), the parallelogram is a strategy (how the argument is decomposed), circles are evidence (what actually closes a goal), and the rounded box is context. The pattern repeats one level deeper for real programs — G2.1 typically decomposes further into a sub-goal per hazard — but even this one layer forces the question a bare evidence list never asks: does this evidence actually close this goal, or just sit near it?

BEYOND ISO 26262 — UL 4600 AND ISO/TS 5083 FOR L4/L5 AND AMRS

The GSN pattern above holds regardless of autonomy level, but a fully autonomous system (no driver to fall back on — an L4/L5 vehicle, a robotaxi fleet, an AMR) has to close goals that ISO 26262 doesn't ask about at all, because they only exist once there's no human in the control loop. **UL 4600** is the fully-elaborated version of this GSN pattern for that case — a consensus safety case standard, widely used and explicitly GSN-based. **ISO/TS 5083:2025** is the newer ISO-side counterpart, covering design, verification, and validation for automated driving systems specifically — the two overlap in intent but aren't identical, and a mature L4 program will typically need to show it has satisfied both, not pick one. Neither replaces ISO 26262/SOTIF; both add goal categories a road-vehicle-only safety case leaves empty:

- **ODD validation evidence** — a goal (and evidence) that the operational design domain was actually characterized and tested against, not just assumed.
- **Fallback / minimal risk condition (MRC) strategy** — what the system does, and how it's argued safe, at the moment it can no longer continue its primary task (pull over, stop in lane, request remote assistance).
- **HMI & interaction evidence** — for occupants, remote operators, and other road users or nearby humans who need to understand what the autonomous system is doing or about to do.
- **Data recording & triage process** — a goal that incidents and near-misses are captured, triaged, and actually fed back into the safety case, not just logged.
- **OTA update argument** — evidence that a software update can't silently invalidate the safety case it was supposed to improve. This has its own standard, **ISO 24089** (software update engineering), and for EU/UK/Japan type approval its own regulation, **UNECE R156**, which requires a Software Update Management System — treat it as a real work product, not a bullet to wave at.

If your program has a safety driver or a human who can take over, most of this is out of scope — note it as such and move on. If it doesn't, treat UL 4600's and ISO/TS 5083's goal structures as a checklist for what G2.1 in the diagram above needs to decompose into before anyone calls the safety case complete.

COMMON PITFALL

Writing the safety case at the end as a documentation exercise. A safety case assembled after the fact almost always surfaces gaps — missing evidence, broken traceability — too late in the schedule to fix cheaply. Build it incrementally as each step closes.

12 Monitor after SOP

ISO 26262-7 / 21448

- Stand up **field monitoring** for safety-relevant fault codes and, for SOTIF, real-world scenario capture from the fleet.
- Define the **field issue triage path**: when does a field finding trigger re-opening HARA vs. a service action vs. a software update?
- Feed field data back into **Area 3→1 reclassification** — this is how your SOTIF case matures over the vehicle's life, not just at launch.

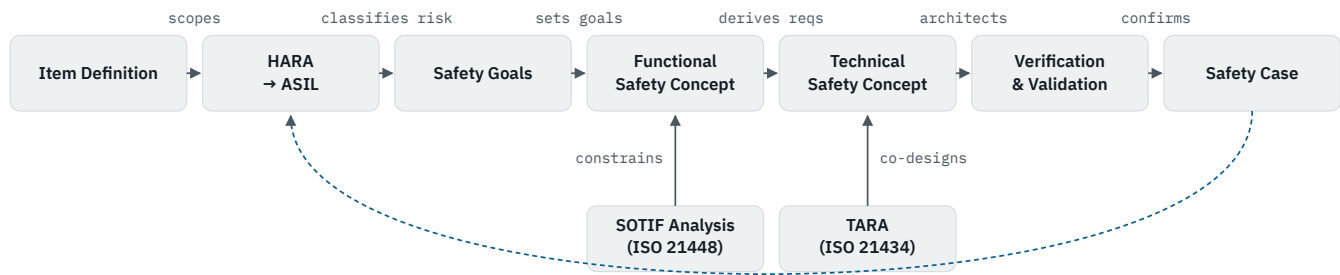
COMMON PITFALL

Treating field monitoring as a warranty or quality function instead of feeding it back into the HARA and SOTIF case. A pattern that looks like "customer complaints" can be an unaddressed hazard — the safety case doesn't end at SOP.

Where to start Monday morning: confirm the FSM appointment and safety plan exist (step 2), pull the current item definition (step 3), and find out whether a HARA has ever been run on this program. If not, that's your first deliverable — everything else depends on it.

How it fits together

The twelve steps read as a checklist, but the documents they produce chain into each other — and, after launch, the chain loops back on itself.

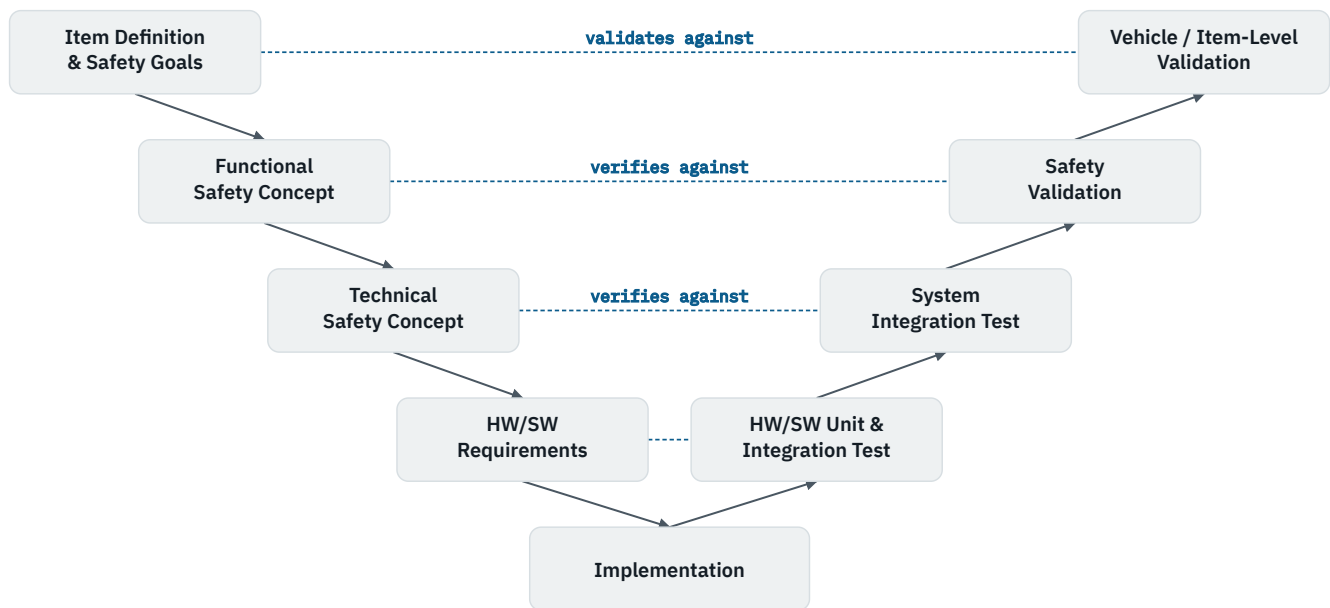


step 12 · field data can reopen the HARA

Each box's output feeds the next: Item Definition scopes the HARA, which classifies risk into Safety Goals, which cascade into the Functional and Technical Safety Concepts, which get verified and rolled into the Safety Case. SOTIF and cybersecurity (TARA) analysis feed into that same chain rather than running separately. After SOP, field monitoring closes the loop — real-world findings can reopen the HARA (step 12), which is why this is a cycle, not a one-time checklist.

THE OTHER STANDARD VIEW: THE V-MODEL

The diagram above shows how documents chain together over time. The V-model (ISO 26262's own reference picture) shows the same lifecycle a different way: decomposition down the left side, integration and test back up the right — and it's what most assessors have in their head when they ask "what verifies this?"



Left side: decompose each safety goal down to HW/SW requirements before a line of code or a circuit is built. Right side: prove, level by level, that what got built actually satisfies the level across from it — a unit test verifies a requirement, a safety validation run validates against the original safety goal. A V-model with no dashed lines — tests that exist but don't trace back to a specific level on the left — is a V-model in name only.

Decision matrices

Five matrices you'll reference constantly: ASIL determination from HARA ratings, the rigor each ASIL requires, SOTIF's known/unknown risk framework, how automotive risk classes map to the terms used in robotics and general machinery, and which work products each ASIL actually requires.

☐ QM — no safety requirement
 ☐ ASIL A
 ☐ ASIL B
 ☐ ASIL C
 ☐ ASIL D — highest rigor

MATRIX 1 — ASIL DETERMINATION (ISO 26262-3, SEVERITY × EXPOSURE × CONTROLLABILITY)

Severity	Exposure	C1 (simple)	C2 (normal)	C3 (difficult)
S1 — light/moderate injury	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2 — severe/life-threatening (survivable)	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3 — life-threatening/fatal	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

E1 = very low probability of exposure · E4 = high probability (occurs in most drives). C1 = drivers can typically control the situation · C3 = difficult or impossible to control.

MATRIX 2 — ASIL → REQUIRED RIGOR

ASIL	Review independence	HW targets (SPFM / LFM / PMHF)	Software method rigor	Confirmation measures
QM	Standard quality process	Not applicable	Company coding standard	None (QM process)

A	Different person	No quantitative target — qualitative argument	Structured methods, static analysis	Safety plan review
B	Different person, may share team	≥90% / ≥60% / <100 FIT	Semi-formal notation for architecture	Safety plan + safety case review
C	Different team / organizational separation	≥97% / ≥80% / <100 FIT	Semi-formal + fault injection testing	+ Functional safety audit
D	Independent department or external assessor	≥99% / ≥90% / <10 FIT	Formal methods where feasible, full MC/DC coverage	+ Functional Safety Assessment before SOP

HW metrics are informative ISO 26262-5 targets, defined only from ASIL B upward — ASIL A has no quantitative SPFM/LFM/PMHF target in the standard, only a qualitative hardware safety argument. Confirm your program's tailored values in the safety plan.

MATRIX 3 — SOTIF RISK ACCEPTANCE (ISO 21448 AREAS)

Area 1 — Known / Safe

Behavior is understood and acceptable. No design action; keep in regression test suite.

Area 2 — Known / Unsafe

Hazardous behavior already identified. Must be reduced through requirements, design change, or ODD restriction before release.

Area 3 — Unknown / Unsafe

Discovered via testing, simulation, or field data. Priority target: drive toward Area 1 through added test coverage and design mitigation.

Area 4 — Unknown / Safe

Cannot be fully evaluated in advance. Managed through continuous field monitoring and scenario capture, not upfront elimination.

The SOTIF release argument is essentially: shrink Area 3 through verification until residual risk is acceptable, and instrument the fleet to keep catching what Area 4 hides.

MATRIX 4 — RISK CLASSIFICATION ACROSS DOMAINS

Domain	Governing standard	Risk class	Scale	What drives the class
Automotive / ADAS	ISO 26262	ASIL	QM, A, B, C, D	Severity × Exposure × Controllability (Matrix 1)
Machinery & robot arms	ISO 13849	Performance Level (PL)	a, b, c, d, e	Severity, frequency/exposure, and possibility of avoidance
Process & general industry	IEC 61508 / 62061	Safety Integrity Level (SIL)	SIL 1–4	Probability of dangerous failure on demand (PFD/PFH)
Collaborative robots	ISO/TS 15066	Force/pressure limits	Body-region thresholds	Biomechanical limits for the specific contact type

There's no exact numeric conversion between ASIL, PL, and SIL — they're built on different risk models. As a rough rule of thumb for conversation with a cross-domain team: ASIL D and SIL 3 sit in a similar rigor band; ASIL B/C and PL d are a common comparison point. Always requalify against the governing standard rather than translating numbers directly.

MATRIX 5 — WORK PRODUCTS REQUIRED BY ASIL

Work product	QM	A	B	C	D
Safety plan	—	Required	Required	Required	Required
HARA report	—	Required	Required	Required	Required
Safety case	—	Recommended	Required	Required	Required
DFMEA / PFMEA	Good practice	Recommended	Required	Required	Required
FMEDA / HW metrics (SPFM, LFM, PMHF)	—	—	Required	Required	Required
Confirmation reviews	—	Recommended	Required	Required	Required
Functional safety audit	—	—	—	Required	Required
Functional Safety Assessment	—	—	Recommended	Required	Required (independent assessor)

"Required" reflects ISO 26262's baseline expectation, not a universal mandate — your safety plan is what actually governs tailoring on a given program, and the standard itself permits rationale-based tailoring for A/B. Treat "—" as "not mandated by the standard," not "skip this if you have the budget."

Templates

Three ready-to-use documents, pulled straight from the steps above: a HARA workshop worksheet, a per-hazard ASIL rating justification sheet, and a Safety Plan outline. Each starts on its own page — print it, photocopy it, or paste the structure into your own tool.

TEMPLATE 1 OF 3

HARA Worksheet

When to use it: in the cross-functional HARA workshop (step 4) — systems, test, and field-data representatives in the room, not filled out solo at a desk.

How to fill it in: brainstorm one malfunctioning behavior at a time, pair it with a plausible operational situation, and write the resulting hazardous event in plain language. Rate Severity / Exposure / Controllability using the definitions behind Matrix 1 (Decision Matrices section), then look up the resulting ASIL in that same matrix and carry it into the Safety Goal column. Keep entries short here in the room — expand them into full sentences when you write the formal HARA report afterward.

#	Malfunctioning behavior	Operational situation	Hazardous event	S	E	C	ASIL	Safety goal
Ex	Unintended full braking, no obstacle present	Highway, following traffic, high speed	Following vehicle can't stop in time, rear-ends host	S3	E4	C3	D	AEB shall not induce braking that following traffic can't reasonably react to
1								
2								
3								
4								
5								
6								
7								
8								

Run this for every function in the item definition before calling the HARA complete — a HARA that only covers the functions someone thought to worry about is the single most common gap an assessor finds.

TEMPLATE 2 OF 3

ASIL Rating Justification Sheet

When to use it: one sheet per hazardous event from the HARA Worksheet — " $S3 \times E4 \times C3 = D$ " tells an assessor the answer; this sheet tells them why each input was chosen, which is what gets challenged in a Functional Safety Assessment.

How to fill it in: check one box per rating and write the evidence behind it — a data source, a standard's example table, or an explicit engineering judgment call. An unchecked rationale line is the first thing a reviewer will flag.

PROGRAM / ITEM

HAZARDOUS EVENT ID

DATE

PREPARED BY

HAZARDOUS EVENT (FROM HARA WORKSHEET)

OPERATIONAL SITUATION

Severity

☐ S0 – none ☐ S1 – light/moderate ☐ S2 – severe (survivable)

☐ S3 – life-threatening/fatal

RATIONALE (INJURY SEVERITY BASIS – E.G. AIS SCALE, TEST/FIELD DATA, ENGINEERING JUDGMENT)

Exposure

☐ E1 – very low ☐ E2 – low ☐ E3 – medium ☐ E4 – high

RATIONALE (FREQUENCY/DURATION BASIS – E.G. % OF OPERATING TIME, DRIVE/USAGE DATA)

Controllability

☐ C0 – controllable in general ☐ C1 – simple ☐ C2 – normal

☐ C3 – difficult or uncontrollable

RATIONALE (WHO/WHAT IS EXPECTED TO RESPOND, AND THEIR REALISTIC CAPABILITY TO DO SO)

Resulting ASIL (cross-check against Matrix 1)

☐ QM ☐ A ☐ B ☐ C ☐ D

SAFETY GOAL

DECOMPOSITION CONSIDERED? (Y/N)

Role	Name	Signature	Date
Prepared by			
Independent reviewer (required for ASIL C/D — see RACI)			

Safety Plan — Outline Template

When to use it: start it in step 2, the day the FSM is appointed — don't wait for a complete picture. It's a living document: update it whenever a tailoring decision, milestone, or work product list changes.

How to use this outline: each numbered section below is a required section of the actual Safety Plan document, with a one-line note on what belongs in it. Write the real plan in your own document tool, using this as the skeleton — the two tables (work products, milestones) can be copied in directly.

PROGRAM / ITEM

PLAN VERSION & DATE

FSM NAME

APPLICABLE ASIL RANGE

1 Scope & applicability

Which lifecycle phases, variants, and sites does this plan cover? Note any tailoring rationale for ASIL A/B work products here — ISO 26262 permits tailoring, but only with documented rationale, not silent omission.

2 Organization & independence

Name the FSM and confirm the reporting line sits outside program schedule/cost authority (step 2 pitfall). Reference the RACI table rather than re-deriving it here.

3 Lifecycle activities & work products

List the work products this program will actually produce against Matrix 5 — mark each Required or Recommended, and name an owner.

Work product	Required?	Owner	Target milestone

4 Milestones & gate reviews

Confirmation Reviews, Functional Safety Audit, and Functional Safety Assessment timing — tie each to a program milestone, not a calendar date that can silently slip.

Milestone	Gate / confirmation measure	Target date

Safety Plan — Outline Template (continued)

5 Tool qualification (TCL) approach

List any tool whose output isn't independently checked downstream (compilers, static analyzers, simulation tools) and its assigned Tool Confidence Level.

6 Change management

Define what triggers re-opening the HARA (step 12) versus a standard change request — this is the line that keeps the safety case from silently going stale.

7 Supplier management

Reference each supplier's Development Interface Agreement (step 10) and the CDRL/PPAP gates where safety evidence is required.

8 Field monitoring & feedback

State the field issue triage path (step 12) and how a field finding feeds back into HARA reclassification, not just a service bulletin.

9 Approval & sign-off

The plan itself is a confirmation-review artifact — it needs sign-off before the program treats it as governing.

Role	Name	Signature	Date
Functional Safety Manager			
Program Manager			
Independent Assessor (ASIL C/D)			

Glossary

The acronyms you'll hear in your first week, spelled out once so you're not nodding along.

Term	Meaning
ASIL	Automotive Safety Integrity Level — the risk class (QM, A, B, C, D) a hazard is assigned in HARA; drives how much rigor everything downstream needs.
HARA	Hazard Analysis and Risk Assessment — the process that finds hazardous events and rates them to produce an ASIL.
ODD	Operational Design Domain — the conditions (roads or workspace, speeds, weather, lighting, human proximity) a function is actually designed to handle.
SOTIF	Safety Of The Intended Function (ISO 21448) — safety risk from a system's performance limits, not from anything breaking.
FMEA	Failure Mode and Effects Analysis — bottom-up: for each part, what happens if it fails this way?
FTA	Fault Tree Analysis — top-down: for this hazard, what combination of faults could cause it?
FMEDA	Failure Mode, Effects and Diagnostic Analysis — quantifies the hardware metrics below (SPFM, LFM, PMHF).
SPFM	Single-Point Fault Metric — the share of hardware faults that are either safe or caught by a diagnostic.
LFM	Latent Fault Metric — the share of "silent" faults (no immediate effect) that still get caught eventually.
PMHF	Probabilistic Metric for random Hardware Failures — the predicted failure rate target for the whole item, in FIT (failures per 10 ⁹ hours).
FSC / TSC	Functional Safety Concept / Technical Safety Concept — FSC states safety requirements at the system level; TSC implements them in the actual architecture.
TARA	Threat Analysis and Risk Assessment (ISO 21434) — HARA's cybersecurity counterpart.
CAL	Cybersecurity Assurance Level (1–4) — security's version of ASIL, set from impact severity and how reachable the attack vector is; informative in ISO 21434 but often contractual with suppliers.
TCL	Tool Confidence Level — how much you have to trust (or independently check) a development or test tool's output.
FSM	Functional Safety Manager — the role with organizational authority over the safety program; may be you.
DIA	Development Interface Agreement — the contract splitting safety responsibilities between you and a supplier.
CDRL / PPAP	Contract Data Requirements List / Production Part Approval Process — where safety evidence gets required and gated from suppliers.
ASPICE	Automotive SPICE — the process maturity model assessors use to judge whether your (or a supplier's) engineering process can be trusted.
PL	Performance Level (ISO 13849) — the robotics/machinery equivalent of ASIL, rated a through e.

SIL	Safety Integrity Level (IEC 61508/62061) — the general-industry risk class, rated 1 through 4, based on probability of dangerous failure.
AMR	Autonomous Mobile Robot — a self-navigating robot (e.g. warehouse or factory) governed by ISO 3691-4, the closest off-road analog to UL 4600.
Cobot	Collaborative robot — designed to share workspace with people; force and pressure limits come from ISO/TS 15066.
RACI	Responsible, Accountable, Consulted, Informed — the matrix mapping each safety deliverable to the role that owns it vs. merely reviews or receives it.
FTTI	Fault Tolerant Time Interval — the minimum time from a fault occurring to a possible hazardous event, if no safety mechanism acts. Sets the budget that FDT + FRT must fit inside.
FDT	Fault Detection Time — how long the system takes to detect a fault has occurred, measured from the fault itself.
FRT	Fault Reaction Time — how long the system takes to reach a safe state after detecting the fault. FDT + FRT must be less than FTTI, with margin.
ASIL decomposition	Splitting one high-ASIL requirement across two or more independent elements (e.g. "ASIL D" → "ASIL B(D) + ASIL B(D)") so each element can be developed to a lower ASIL — valid only if the elements are genuinely independent (no shared power, sensor, or software fault).
GSN	Goal Structuring Notation — a graphical way to lay out a safety argument as goals, strategies, and the evidence that actually closes each goal, rather than a flat list of evidence.
ISO/PAS 8800	AI/ML safety for road vehicles — extends ISO 26262/SOTIF with data, model, verification, and operation assurance activities specific to machine-learning components.
UL 4600	Safety case standard for fully autonomous products (no driver fallback) — adds goal categories ISO 26262 doesn't cover: ODD validation, MRC strategy, HMI evidence, data triage, and OTA update safety.
MRC	Minimal Risk Condition — the safe state (e.g. stop in lane, pull over) a fully autonomous system reaches when it can no longer continue its primary task.
ISO/TS 5083	Safety for automated driving systems: design, verification, and validation (2025) — the ISO-side counterpart to UL 4600 for L3+ systems with an actual ADS; layers on top of ISO 26262/SOTIF rather than replacing them.
ISO 24089	Software update engineering — the standard typically used to satisfy UNECE R156's Software Update Management System (SUMS) requirement.
UNECE R155 / R156	EU/UK/Japan type-approval regulations, not standards — R155 requires a Cybersecurity Management System (satisfied via ISO/SAE 21434); R156 requires a Software Update Management System (satisfied via ISO 24089).
SAE J3016	The taxonomy behind "L2/L3/L4/L5" — levels of driving automation, from no automation (L0) to full automation with no ODD limit (L5). A taxonomy, not a safety standard.
ANSI/A3 R15.08	U.S. standard for industrial mobile robots (AMRs), in three parts (robot, system/fleet, application) — distinct from R15.06, which covers fixed industrial robot arms.

Reference playbook — validate ASIL targets, HW metric thresholds, and tailoring decisions against your program's own safety plan and OEM-specific requirements before use in a formal safety case.