RCM Facilitation Tool Kit

Structured Prompt Cards for RCM Teams

55 PROMPT CARDS

RELIABILITY MANAGEMENT

www.reliabilitymanagement.co.uk

Version 33

cut

cut

cut

HOW TO USE THESE CARDS

Front Side — How to Facilitate

Main Prompt

The primary question to ask the team.

Instruction

How to guide the discussion and what to capture.

Follow Up

A probing question to deepen or confirm the answer.

TIP

Work through the FRONT to facilitate, then use the BACK to quality-check the team's answer before moving on.

CARD CATEGORIES

55 structured prompts across 7 RCM topics

Context & Scope	In scope / operating states / constraints & goals	10
Functions	What the asset must do and to what performance standard	11
Functional Failures	How each function fails — total, partial or intermittent	3
Failure Modes	Specific causes of each functional failure	14
Failure Effects	What happens when each failure mode occurs	7
Consequences	Safety, environmental and operational impact	2
Tasks		8

cut

cut

cut

HOW TO USE THESE CARDS

Back Side — How to Evaluate Quality

Why This Is Important

The purpose and value of this step.

Red Flags & Common Traps

Warning signs the answer is weak or wrong.

What Good Looks Like

The standard for a complete, quality answer.

TIP

Use the BACK as an audit checklist — if the answer trips a red flag, revisit the question before progressing.

cut

Context & Scope

Main Prompt

What is in scope, and what is out of scope?

Instruction

Confirm the asset / system level, list included tags / functions and list explicit exclusions.

Follow Up

Which specific equipment tags / systems / functions are included, which are explicitly excluded?

cut

cut

Context & Scope

Main Prompt

Which operating states exist (start-up, shutdown, standby, normal, emergency)?

Instruction

Ask the team to enumerate states actually used in this context.

Follow Up

What operating states do you actually use in this context?

What Is in Scope, and What Is Out of Scope?

Why This Is Important

It prevents scope creep, keeps the analysis auditable, and ensures the team assesses the correct asset / system level.

Red Flags & Common Traps

- Scope defined as "everything in the plant" or "we will cover it later".
- No explicit exclusions.
- Mixing scope with solutions.

What Good Looks Like

A written scope statement with included asset tags / systems and explicit exclusions

cut cut cut

Which Operating States Exist (Start-up, Shutdown, Standby, Normal, Emergency)?

Why This Is Important

Different states have different functions, loads, protections, failure exposure and failure consequences.

Red Flags & Common Traps

- Only "normal" state considered.
- Start / stop / emergency ignored.
- No clear entry / exit criteria.
- Treating rare states as irrelevant despite high risk.

What Good Looks Like

Normal operation (including start up and shut down) is the default assumption. If used just in start-up, standby or emergency, this should be stated in the function and described in general terms in the context

cut

Context & Scope

Main Prompt

What is normal operation versus special conditions?

Instruction

Define the "normal envelope" with measurable ranges

Follow Up

Can we define "normal" as measurable ranges (min / target / max) ?

cut

cut

Context & Scope

Main Prompt

What is the main business purpose of this system?

Instruction

Ask "what business outcome does this enable?" then name the KPI(s) affected and the operational impact if unavailable.

Follow Up

If this system is unavailable, what business outcome is lost and which KPI(s) change?

cut

What Is Normal Operation versus Special Conditions?

Why This Is Important

It distinguishes expected variability from true abnormality, avoiding false failure definitions and wrong task selection.

Red Flags & Common Traps

- "Normal" described with adjectives only ("stable / good").
- No measurable envelope.

What Good Looks Like

A measurable "normal envelope" (min / target / max + units)

cut

cut

What Is the Main Business Purpose of This System?

Why This Is Important

It anchors criticality and prioritization to business value, not technical preference.

Red Flags & Common Traps

- Purpose stated as component function ("it pumps").
- No KPI link.
- Ignoring safety-barrier purpose.
- Conflicting stakeholder views not reconciled.

What Good Looks Like

A concise value statement linking the system to business outcomes and KPIs, including impact of unavailability.

cut

Context & Scope

Main Prompt

What are the top problems / pain today (capacity, safety, staffing, spares, outages)?

Instruction

Describe the problems / pain associated with this system

Follow Up

Describe the problems / pain associated with this system

cut

cut

Context & Scope

Main Prompt

What is the operating profile (hours / year, starts, cycling, load changes)?

Instruction

Quantify hours / year, starts / year, ramps and cycling severity; confirm from historian / logs vs operator estimate.

Follow Up

Can we quantify hours / year by state, starts / year, ramp rates, cycling frequency, and typical load swings?

What Are the Top Problems/ Pain Today (Capacity, Safety, Staffing, Spares, Outages)?

Why This Is Important

Problems / pain drive objectives

Red Flags & Common Traps

- Problems / pain not described in the context.

What Good Looks Like

Problems / pain described in the context at a high level

cut cut cut

What Is the Operating Profile (Hours/year, Starts, Cycling, Load Changes)?

Why This Is Important

Duty profile determines wear-out, cycling fatigue, start-related risk, and required task frequencies.

Red Flags & Common Traps

- Using nameplate ratings instead of actual duty.
- Starts / cycling underreported.
- Seasonal / campaign patterns missed.
- Reliance on "typical day" anecdotes.

What Good Looks Like

Quantified duty profile (hours / year by state, starts / year, ramping / cycling severity, load range) with data source noted.

Context & Scope

Main Prompt

What disturbances / upsets must the system tolerate?

Instruction

List credible upsets (process / utility / ambient / grid), their magnitude / duration, and required response.

Follow Up

What credible upsets occur here, how big / long are they, and is the required system response ride-through, degrade, or trip?

cut cut cut

Context & Scope

Main Prompt

What are known recurring issues (bad actors)?

Instruction

Identify the top recurring "bad actors," quantify frequency and impact, and describe evidence (WOs, alarms, trends).

Follow Up

Which problems recur most often, how frequently, with what downtime / cost impact, and what evidence supports that?

cut

What Disturbances/upsets Must the System Tolerate?

Why This Is Important

Upsets define design and operational stressors that drive failure modes and required tolerances.

Red Flags & Common Traps

- Only average conditions considered.
- Credible upsets ignored.
- No magnitude / duration.
- Required response undefined.
- Assuming operators will "handle it.".

What Good Looks Like

Credible upsets mentioned in the operating context if applicable.

cut cut cut

What Are Known Recurring Issues (Bad Actors)?

Why This Is Important

Recurring issues indicate dominant failure mechanisms and provide high-ROI targets for management.

Red Flags & Common Traps

- "Bad actors" listed without evidence.
- Blaming individuals.
- No frequency / impact.
- Confusing symptoms with recurring causes.
- Ignoring repeat offenders in history.

What Good Looks Like

A prioritized list of recurring issues with frequency, downtime / cost impact, triggers, and evidence references.

Context & Scope

Main Prompt

What is the current maintenance strategy, in general terms (PM / PdM intervals, overhauls)?

Instruction

Summarize current PM / PdM / overhaul / failure-finding at headline level and note known gaps or compliance problems.

Follow Up

At headline level, what PM tasks and intervals exist, what PdM methods / frequencies, and what overhaul triggers are used?

cut cut cut

Context & Scope

Main Prompt

Did anything change recently (process, fuel, operators, physical modifications)?

Instruction

Ask for recent MOC-type changes (process / fuel / operators / mods / software) and what parameters shifted (loads / chemistry / setpoints).

Follow Up

What changed over the life of the asset (process / fuel / operators / mods / software) that could shift loads, chemistry, or setpoints?

What Is the Current Maintenance Strategy, in General Terms (Pm/pdm Intervals, Overhauls)?

Why This Is Important

It establishes the baseline against which to assess the new maintenance programme

Red Flags & Common Traps

- Listing tasks without intent ("we do PMs").
- Intervals copied from OEM without context.
- Overhaul philosophy unclear.
- No view on effectiveness / backlog.

What Good Looks Like

A baseline headline summary of the current maintenance strategy including task types with intervals, triggers, and known execution gaps.

cut

cut

cut

Did Anything Change Recently (Process, Fuel, Operators, Physical Modifications)?

Why This Is Important

Recent changes shift failure behavior and invalidate historical assumptions, intervals, and "known good" settings.

Red Flags & Common Traps

- Changes not considered.
- MOC not consulted.
- "Same as always" assumption.
- New fuel / chemistry / software effects overlooked.
- New operators" practices ignored.

What Good Looks Like

Where applicable, changes over the life of the asset described at a high level in the context

Functions

Main Prompt

Which protections / controls are required (interlocks, trips, alarms)?

Instruction

Separate alarms vs trips vs interlocks; ask for setpoints. List as Functions

Follow Up

List alarms vs trips vs interlocks with setpoints as functions

cut

cut

cut

Functions

Main Prompt

What is the function—one sentence: to + verb + performance standard? What do we want it to do?

Instruction

Force a one-sentence "to + verb + standard" statement and read it back until everyone agrees it is measurable and complete.

Follow Up

Can you state the function as: "to + verb + object + performance standard + conditions" in one sentence?

cut

Which Protections/controls Are Required (Interlocks, Trips, Alarms)?

Why This Is Important

Protections change system behavior and consequences; they can mask degradation or create secondary failure paths.

Red Flags & Common Traps

■ Alarms / trips / interlocks not listed as functions.

What Good Looks Like

Functions include Alarms / trips / interlocks and associated performance standards.

cut

cut

What Is the Function—one Sentence: to + Verb + Performance Standard? What Do We Want It to Do?

Why This Is Important

A single precise function statement is the reference for all functional failures and task justifications.

Red Flags & Common Traps

■ Function phrased as component description ("pump"), not outcome.
■ Missing performance standard.
■ Multiple functions in one sentence.
■ Not state / condition specific.

What Good Looks Like

One function sentence in the form "to + verb + object + performance standard + conditions," measurable and agreed by the team.

Functions

Main Prompt

What secondary functions are also required (efficiency, safety, containment, control, comfort, appearance, indication, environment)?

Instruction

Systematically consider whether secondary functions apply (safety / containment / control / environment / indication / efficiency).

Follow Up

Which non-production requirements must also be met (safety, containment, control, indication, environment)?

cut

cut

cut

Functions

Main Prompt

Are there any hidden functions (only needed on in the event of another failure)?

Instruction

The function will often be written "To be capable of…. In the event of….." (or something similar)

Follow Up

If the wording "To be capable of…. In the event of….." is appropriate

What Secondary Functions Are Also Required (Efficiency, Safety, Containment, Control, Comfort, Appearance, Indication

Why This Is Important

Secondary functions often drive safety, regulatory, and quality consequences that add to the task list.

Red Flags & Common Traps

- Secondary functions missing.

What Good Looks Like

A confirmed list of secondary required outcomes (safety, containment, control, indication, environment, efficiency).

Are There Any Hidden Functions (Only Needed on in the Event of Another Failure)?

Why This Is Important

Hidden functions have an additional failure management task option - failure-finding; this can be used to detect hidden failures which might remain undetected until demanded, increasing the risk of a multiple failure

Red Flags & Common Traps

- Hidden functions missed entirely.
- Protection / standby assumed OK.
- No proof testing considered.

What Good Looks Like

Hidden functions explicitly identified "To be capable of…. In the event of….."

cut

Functions

Main Prompt

What tolerance is acceptable—when does it become a functional failure?

Instruction

Define the acceptable deviation band and the exact point where performance crosses into "fails to meet the standard."

Follow Up

What exact deviation (value / time) crosses from acceptable degradation into functional failure?

cut

cut

Functions

Main Prompt

Do standards change by operating state?

Instruction

Check each operating state and ask whether limits / standards differ; if yes, specify state-specific standards and use the most demanding in the analysis

Follow Up

Specify state-specific standards and use the most demanding in the analysis

cut

What Tolerance Is Acceptable—when Does It Become a Functional Failure?

Why This Is Important

Tolerance defines the boundary between acceptable degradation and functional failure, enabling consistent decisions.

Red Flags & Common Traps

- Tolerance not defined, too wide or unrealistically tight.
- Not linked to business / process needs.
- Mixing alarm setpoint with functional failure threshold.

What Good Looks Like

A clearly defined tolerance band where performance crosses into functional failure for each relevant standard.

cut

cut

Do Standards Change by Operating State?

Why This Is Important

Standards can tighten / loosen by state, changing what constitutes failure and what tasks are effective.

Red Flags & Common Traps

- Assuming standards constant across states.

What Good Looks Like

State-specific standards documented wherever they apply

cut

Functions

Main Prompt

Is this truly a function, not a component description or solution?

Instruction

Challenge solution statements ("to provide a pump") and reframe to required outcome with a measurable standard.

Follow Up

Are we describing an outcome (what must be achieved) rather than a component / solution—can you restate as an outcome?

cut

cut

Functions

Main Prompt

What is required during start-up vs steady state vs shutdown?

Instruction

If applicable, consider start-up, steady, shutdown requirements

Follow Up

What must the system achieve in each phase and what standards apply?

Is This Truly a Function, Not a Component Description or Solution?

Why This Is Important

The analysis is driven by what it must do (the function) rather than what it is (the physical component)

Red Flags & Common Traps

- Solutioneering ("install a bigger pump").
- Describing how not what.
- Mixing function with failure management task.
- Drifting into redesign before defining outcomes.

What Good Looks Like

Function statements expressed as outcomes, not components or solutions, with verbs and standards (no "pump / motor" descriptions).

cut cut cut

What Is Required During Start-up vs Steady State vs Shutdown?

Why This Is Important

Transitional states sometimes drive peak stresses and unique failures; requirements must be state-specific.

Red Flags & Common Traps

- Transitional requirements skipped.
- Start-up stressors ignored.
- Shutdown requirements treated as "not relevant".
- No phase-specific standards or timelines.

What Good Looks Like

Phase-based requirements captured for start-up / steady / shutdown, including standards, constraints, and key dependencies per phase.

Functions

Main Prompt

What response time is required (how fast must it work)?

Instruction

Ask "how fast must it respond?" (seconds / minutes) and link this to safety, control stability, and process requirements.

Follow Up

How fast must it respond (seconds / minutes), under what conditions?

cut cut cut

Functions

Main Prompt

What is the minimum acceptable performance (must-meet threshold)?

Instruction

Agree the must-meet floor (minimum) in numbers / units and confirm it is sufficient to meet business / process needs.

Follow Up

What is the must-meet minimum (number + units + time basis) that still satisfies process / safety needs?

What Response Time Is Required (How Fast Must It Work)?

Why This Is Important

Response time defines whether slow performance is a failure and drives monitoring and protection requirements.

Red Flags & Common Traps

- Response time omitted.
- "Fast enough" language.
- Confusing control loop response with protection response.
- Not linking response time to consequence.

What Good Looks Like

A quantified response-time requirement (seconds / minutes) tied to the operating context

What Is the Minimum Acceptable Performance (Must-meet Threshold)?

Why This Is Important

It sets the "must not drop below" criterion to classify functional failures consistently.

Red Flags & Common Traps

- Minimum set to "as low as possible" or arbitrary.
- Not tied to process constraints.
- Confusing setpoint with minimum acceptable.
- No time basis.

What Good Looks Like

A numeric must-meet minimum threshold with units and time basis that still satisfies process and safety requirements.

Functions

Main Prompt

What is the maximum acceptable performance (upper limit)?

Instruction

Agree the must-not-exceed ceiling (maximum) and confirm what damage / instability occurs above it.

Follow Up

What is the must-not-exceed maximum (number + units) and what damage / instability occurs above it?

cut cut cut

Functional Failures

Main Prompt

How does the function fail (fails to … linked to that function)?

Instruction

Convert each deviation into a "fails to …" statement explicitly linked to the function and standard.

Follow Up

Can you phrase the functional failure as "fails to [verb] [standard] under [state / conditions]"?

cut

What Is the Maximum Acceptable Performance (Upper Limit)?

Why This Is Important

Upper limits protect against overperformance that creates damage, instability, or safety risk.

Red Flags & Common Traps

- Maximum ignored ("more is better").
- Not recognizing overpressure / overtemp / overspeed risks.
- Missing upper-limit constraints from OEM / protections.

What Good Looks Like

A numeric upper limit with units

cut

cut

How Does the Function Fail (Fails to … Linked to That Function)?

Why This Is Important

Clear functional failure statements enable consistent failure mode mapping and avoid mixing causes and effects.

Red Flags & Common Traps

- Functional failure written as a cause ("seal failure") or symptom ("vibration").
- Not linked to function.
- Inconsistent wording across team.

What Good Looks Like

Functional failures written as "fails to …" statements directly linked to the function and its performance standard.

Functional Failures

Main Prompt

Is it total loss, partial loss, or intermittent loss?

Instruction

Consider all type of functional failure

Follow Up

Is the loss complete, partial, or intermittent?

cut

cut

cut

Functional Failures

Main Prompt

Which performance standard does this refer to?

Instruction

Point to the exact standard parameter (flow, pressure, quality, response time, availability) that is being violated.

Follow Up

Which exact standard parameter is violated (flow, pressure, quality, response time, availability), and what is the limit value?

cut

Is It Total Loss, Partial Loss, or Intermittent Loss?

Why This Is Important

Loss type affects consequences, detectability, and the feasibility of condition-based maintenance.

Red Flags & Common Traps

- Loss type not described (where applicable).

What Good Looks Like

Functional failures listed include total / partial / intermittent (where applicable)

cut

cut

Which Performance Standard Does This Refer To?

Why This Is Important

It ties the deviation to a specific standard, preventing vague failure definitions and wrong thresholds.

Red Flags & Common Traps

- Not tying failure to a specific standard parameter.
- Vague "performance issue".

What Good Looks Like

An explicit reference to the exact performance standard parameter and limit value that is being violated.

Failure Modes

Main Prompt

What could cause this functional failure (root-cause level)?

Instruction

Identify causes at an appropriate level of causation, by asking "why?" such that an action to manage them is identifiable

Follow Up

Ask "Why?

Failure Modes

Main Prompt

Is this a cause or only a symptom?

Instruction

Challenge symptom-only entries ("vibration") and ask "what causes that?" until a controllable root cause is stated.

Follow Up

Is this describing evidence (symptom) or the initiating cause—what mechanism creates the symptom?

What Could Cause This Functional Failure (Root-cause Level)?

Why This Is Important

Root-cause-level causes are needed for actionable failure modes and credible task selection.

Red Flags & Common Traps

- Causes stated at effect level.
- "Wear" without location / part.
- "Vibration" as cause.
- Listing consequences as causes.
- No clearly actionable task.

What Good Looks Like

A set of causes stated at an appropriate level of root-cause, each actionable.

cut

cut

cut

Is This a Cause or Only a Symptom?

Why This Is Important

Symptom-only "causes" lead to circular logic and ineffective tasks that do not control the failure process.

Red Flags & Common Traps

- Stopping at symptom.
- Circular logic.
- No mechanism.
- Inconsistent cause depth between entries.
- Conflating detection with cause.

What Good Looks Like

Causes separated from symptoms, with symptoms captured as effects / indications and causes as initiating mechanisms.

Failure Modes

Main Prompt

Which causes are reasonably likely in this operating context?

Instruction

Filter the cause list for what is reasonably likely under this duty and environment; unlikely failures with high consequences can also be included.

Follow Up

Given this duty and environment, which causes have occurred before or are credible?

cut cut cut

Failure Modes

Main Prompt

Which rare causes are not likely but have severe consequences?

Instruction

Include low-likelihood / high-consequence failure modes in the analysis

Follow Up

Even if unlikely, which failure modes could lead to severe outcomes?

Which Causes Are Reasonably Likely in This Operating Context?

Why This Is Important

Likelihood in context is essential for prioritization and for meeting RCM"s "reasonably likely" criterion.

Red Flags & Common Traps

- Likelihood judged by opinion only.
- Ignoring duty / environment.
- Rare-but-known causes excluded.
- "Never happened here" used incorrectly.

What Good Looks Like

A list of reasonably likely causes for the stated duty / environment

cut

Which Rare Causes Are Not Likely but Have Severe Consequences?

Why This Is Important

Low-likelihood / high-consequence causes may still need to be managed even if infrequent.

Red Flags & Common Traps

- High-consequence scenarios ignored due to rarity.
- Assuming safeguards perfect.

What Good Looks Like

Rare but severe causes included and identified for safety / integrity treatment with scenario description.

Failure Modes

Main Prompt

Which human / organizational causes are plausible?

Instruction

Ask for plausible human / procedural contributors (wrong setpoint, wrong part, wrong method etc etc) and capture barriers / controls.

Follow Up

Which human / procedural errors are plausible here (wrong setpoint, wrong part, wrong method), and what enables them?

cut

cut

cut

Failure Modes

Main Prompt

Which external causes are plausible (contamination, supply issues, ambient, process upset)?

Instruction

Ask for credible external drivers (contamination, supply variability, ambient extremes, process upsets) and how they enter the system.

Follow Up

Which external factors can drive this (contamination route, supply variability, ambient extremes, process upset) and how do they enter?

cut

Which Human/organizational Causes Are Plausible?

Why This Is Important

Human and procedural factors are common contributors to unreliability and often controllable via standards and training.

Red Flags & Common Traps

- Human factors omitted or politicized.
- Blaming individuals vs system.
- No linkage to procedures / training / controls.
- "Operator error" used as lazy label.
- Also too vague so a task to manage is not clear.

What Good Looks Like

Plausible and clearly defined human / organizational contributors identified with enabling conditions and potential controls / barriers identified.

cut

cut

Which External Causes Are Plausible (Contamination, Supply Issues, Ambient, Process Upset)?

Why This Is Important

External drivers can also contribute to unreliability

Red Flags & Common Traps

- External drivers ignored.
- Contamination routes not defined.
- Supply variability dismissed.
- Ambient extremes not considered.
- Upstream process interactions missed.

What Good Looks Like

Credible external contributors identified with entry path (how it gets in) and boundary / interface ownership documented. (Watch out for analysis creep though)

cut

Failure Modes

Main Prompt

Is the failure mode specific enough so that it means the same thing to everyone?

Instruction

Read the failure mode aloud and test it with "would two teams interpret this identically?"; tighten wording if necessary.

Follow Up

Read it aloud—would two teams interpret it the same way; if not, tighten wording to a single mechanism + location.

cut

cut

Failure Modes

Main Prompt

Are any failure modes duplicates (same meaning, different wording)?

Instruction

De-duplicate by comparing intent and mechanism; merge items that mean the same thing and keep the clearest wording.

Follow Up

Do any entries mean the same thing—can we merge duplicates and keep one clear, standard phrasing?

Is the Failure Mode Specific Enough So That It Means the Same Thing to Everyone?

Why This Is Important

Precision prevents misclassification and ensures different teams implement the same task consistently.

Red Flags & Common Traps

- Failure mode wording ambiguous.
- Multiple mechanisms in one line.
- Missing location / component.
- Jargon that different groups interpret differently.

What Good Looks Like

Failure modes written unambiguously so independent teams interpret them the same way (mechanism + location + condition).

cut

cut

cut

Are Any Failure Modes Duplicates (Same Meaning, Different Wording)?

Why This Is Important

Duplicate modes inflate workload, distort criticality, and hide the true drivers of risk.

Red Flags & Common Traps

- Duplicate modes inflate list.
- "Same thing different words".
- Double-counting in criticality.
- Inconsistent naming conventions across teams.

What Good Looks Like

Duplicate failure modes removed or merged, leaving one standardized phrasing per distinct meaning / mechanism.

cut

Failure Modes

Main Prompt

Is it written without lack of maintenance as the cause?

Instruction

Reject "lack of maintenance" and ask for the specific physical / organizational cause that led to the condition.

Follow Up

Replace "lack of maintenance" with the specific cause: what condition developed, why, and what initiated it?

cut

cut

Failure Modes

Main Prompt

What is the physical mechanism (wear, fatigue, corrosion, erosion, fouling, overheating)?

Instruction

Name the degradation mechanism (wear, fatigue, corrosion, erosion, fouling, overheating) to anchor detection and interval logic.

Follow Up

Which degradation physics applies (wear / fatigue / corrosion / erosion / fouling / overheating), and where does it occur?

Is It Written Without Lack of Maintenance as the Cause?

Why This Is Important

"Lack of maintenance" is not a cause; it blocks root cause control and produces non-actionable outputs.

Red Flags & Common Traps

- "Lack of maintenance" used as root cause.
- Tasks justified to "do more PM".
- No underlying mechanism.
- No controllable initiating cause.

What Good Looks Like

No "lack of maintenance" wording; instead, a specific initiating cause and mechanism is stated.

cut

cut

cut

What Is the Physical Mechanism (Wear, Fatigue, Corrosion, Erosion, Fouling, Overheating)?

Why This Is Important

Mechanism identification links degradation physics to detection methods, intervals, and suitable task types.

Red Flags & Common Traps

- Mechanism guessed without evidence.
- Generic "corrosion" without type / location.
- Mechanism not linked to operating context.
- Ignoring dominant physics.

What Good Looks Like

A named physical mechanism linked to expected evidence and suitable detection method (e.g., wear → vibration / clearance).

Failure Modes

Main Prompt

What is the trigger event / root cause that starts the failure process?

Instruction

Identify the initiating trigger / root cause (event / condition) that starts degradation and capture what prevents or accelerates it.

Follow Up

What event / condition initiates the degradation (start, upset, contamination ingress, thermal cycle), and what prevents it?

cut

cut

cut

Failure Modes

Main Prompt

Can one cause create multiple failure modes (common cause)?

Instruction

Ask "what single cause could hit multiple items at once?" and document system-level common-cause vulnerabilities.

Follow Up

Could a single upstream cause affect multiple components simultaneously—what shared dependency or environment creates that?

cut

What Is the Trigger Event /Root Cause That Starts the Failure Process?

Why This Is Important

The trigger / root cause defines what initiates degradation and what controls or barriers can prevent onset.

Red Flags & Common Traps

- Trigger / root cause not identified.
- Assuming continuous degradation only.
- Missing initiating events (upset / start / ingress).
- No prevention barrier discussion.

What Good Looks Like

A defined initiating trigger (event / condition) and documented barriers that prevent onset or slow progression.

cut

cut

Can One Cause Create Multiple Failure Modes (Common Cause)?

Why This Is Important

Common-cause pathways require system-level controls; treating items independently misses correlated risk.

Red Flags & Common Traps

- Common-cause ignored.
- Treating identical items independently.
- Shared utilities / controls / environment overlooked.
- Correlated failures surprise later.

What Good Looks Like

Common-cause risks identified (shared utility, environment, procedure) and treated as system-level vulnerabilities with higher level management tasks

Failure Modes

Main Prompt

What is the wrong maintenance risk (wrong grease, torque, alignment)?

Instruction

Identify task-induced risk (wrong grease / torque / alignment) and define controls / verification to prevent maintenance-caused failures.

Follow Up

What maintenance actions could create this failure (wrong grease / torque / alignment / procedure) and what checks prevent it?

cut

cut

cut

Failure Modes

Main Prompt

Is the failure mode actionable (a clear task / decision can be assigned)?

Instruction

For each failure mode, force a clear decision path: detect / monitor, restore / replace, redesign, or accept with justification.

Follow Up

Can we assign a clear action / decision: monitor, inspect, restore, replace, redesign, or accept—with criteria?

What Is the Wrong Maintenance Risk (Wrong Grease, Torque, Alignment)?

Why This Is Important

Maintenance-induced failures are a major risk; identifying them prevents harm from the task itself.

Red Flags & Common Traps

- Maintenance-induced failures denied.
- No torque / alignment / grease controls.
- Procedures not verified.
- "We have always done it" mindset.

What Good Looks Like

Maintenance-induced failure risks explicitly listed with prevention controls (tools, procedures, verification checks).

Is the Failure Mode Actionable (a Clear Task/decision Can Be Assigned)?

Why This Is Important

Actionability ensures each failure mode leads to a clear decision, task, or redesign recommendation.

Red Flags & Common Traps

- Failure mode not decision-ready.
- No clear task option.
- Mixing multiple causes.
- Tasks assigned that cannot be executed.
- No ownership.

What Good Looks Like

Each failure mode leads to a clear decision path (on-condition, scheduled restoration or discard, failure-finding, redesign, run-to-failure) with owner.

Failure Effects

Main Prompt

What happens first at the asset—what would we observe?

Instruction

Capture the first observable at the asset (sound, vibration, leak, trend change, quality shift) and where it is detected.

Follow Up

What is the very first observable change at the asset (leak, noise, vibration, trend shift), and where is it detected?

cut cut cut

Failure Effects

Main Prompt

What protections activate, and what is the resulting state?

Instruction

Confirm which protections actuate, what trips, and what the end state is (shutdown, degraded, alarm-only, fail-safe).

Follow Up

Which protection acts first, what trips, and what is the final state (shutdown / degraded / alarm-only / fail-safe)?

cut

What Happens First at the Asset—what Would We Observe?

Why This Is Important

First observable effects enable practical detection points and help define P–F intervals.

Red Flags & Common Traps

- First observable not defined.
- Jumping to secondary effects.
- Relying on late alarms.
- Missing early indicators.
- Observation point unclear.

What Good Looks Like

The first observable effect at the asset is stated with where / how it is detected (operator sense, instrument tag, test).

cut

cut

What Protections Activate, and What Is the Resulting State?

Why This Is Important

Protection response determines consequence severity, downtime, and whether failure becomes evident.

Red Flags & Common Traps

- Protection response assumed.
- No validation of end state.
- Ignoring nuisance trips.
- Bypassed protections not acknowledged.
- Unclear reset conditions.

What Good Looks Like

Protection behavior described as a sequence (what triggers, what acts, final state) and whether it masks evidence.

Failure Effects

Main Prompt

What is the effect on system and plant output?

Instruction

Translate the failure into system and plant impact (capacity, efficiency, quality, emissions, availability) with rough magnitude.

Follow Up

What is the quantifiable impact on system / plant output (MW, t / h, quality, emissions, availability) and typical magnitude?

cut cut cut

Failure Effects

Main Prompt

What secondary damage can occur if operation continues?

Instruction

Ask "if we keep running, what breaks next?" and document escalation paths and secondary damage.

Follow Up

If we keep running, what fails next—what escalation path and secondary damage mechanism exists?

What Is the Effect on System and Plant Output?

Why This Is Important

It connects failures to production impact and validates prioritization and business case.

Red Flags & Common Traps

- Impact described qualitatively only.
- No link to throughput / quality / emissions.
- Confusing local effect with plant effect.
- No magnitude estimate.

What Good Looks Like

The impact on system / plant output expressed in operational terms and, where possible, quantified (capacity, quality, emissions).

cut

cut

cut

What Secondary Damage Can Occur If Operation Continues?

Why This Is Important

Secondary damage can dominate cost / risk and often justifies faster detection or immediate shutdown logic.

Red Flags & Common Traps

- Secondary damage ignored.
- "Run to failure" assumed safe.
- Escalation paths not mapped.
- Overlooking collateral damage and cleanup time.

What Good Looks Like

Secondary damage pathways stated (what fails next, escalation mechanism) with implication for response urgency.

cut

cut

Failure Effects

Main Prompt

What is required to restore function (reset, repair, replace, clean, vent)?

Instruction

Describe restoration at a very high level but do not list excessive detail

Follow Up

What is required, in general terms, to rectify the failure?

cut

Failure Effects

Main Prompt

What typical delays occur (diagnosis, isolation, parts, restart)?

Instruction

Identify the dominant delays (diagnosis, access, isolation, parts, permits, restart) to inform repair time. Take a typical worst case

Follow Up

Where does time get lost—diagnosis, access, isolation, permits, parts lead time, restart sequence?" Take a typical worst case

cut

What Is Required to Restore Function (Reset, Repair, Replace, Clean, Vent)?

Why This Is Important

Restoration requirements drive maintainability assumptions, spares strategy, and outage planning.

Red Flags & Common Traps

- Restoration unclear.
- Excessive detail of restoration described.

What Good Looks Like

Restoration listed at a high level and in general terms

cut

cut

What Typical Delays Occur (Diagnosis, Isolation, Parts, Restart)?

Why This Is Important

Delays influence repair time.

Red Flags & Common Traps

- Delays blamed on "bad luck".
- Logistics / access / permits not captured.
- Lead times unknown.
- Restart sequence complexity ignored.

What Good Looks Like

The main delay drivers identified and ranked (diagnosis, access, isolation, parts lead time, restart sequence). Take a typical worst case

Failure Effects

Main Prompt

What alarms / trips / indications do we expect if the failure occurs?

Instruction

Validate expected indications (alarms / trips / trends)

Follow Up

Which alarms / trips / trends indicate it?

cut

cut

cut

Consequences

Main Prompt

Is there an intolerable risk of health, safety or environmental consequences (ie that a known environmental standard or regulation would be breached)?

Instruction

Explicitly consider if there is an intolerable risk of someone being injured or killed or any environmental standard could be breached; if yes, categorise as a safety / environmental consequence

Follow Up

Explicitly consider if there is an intolerable risk of someone being injured or killed or any environmental standard could be breached

What Alarms/trips/indications Do We Expect If the Failure Occurs?

Why This Is Important

RCM depends on reliable evidence; false or missing indications undermine condition-based strategies.

Red Flags & Common Traps

- Indications not considered in failure effects.

What Good Looks Like

Expected indications listed in failure effects.

Is There an Intolerable Risk of Health, Safety or Environmental Consequences (ie That a Known Environmental Standard or…

Why This Is Important

It is the decisive screen for safety / environmental consequences and drives mandatory risk reduction.

Red Flags & Common Traps

- HSE / environmental breach treated as "manageable".
- Standards not specified.
- Reliance on PPE as primary control.

What Good Looks Like

A clear determination of intolerable HSE / environmental risk

cut

Consequences

Main Prompt

Would this failure be evident if it occurs on its own?

Instruction

Classify as evident or hidden when it occurs on its own

Follow Up

If this happens with no other failures, would the loss of function be evident under normal circumstances?

cut

cut

Tasks

Main Prompt

For condition based tasks - Is there a clear P–F interval, and roughly how long is it?

Instruction

Estimate the P–F interval from history / physics and confirm it is long enough to support practical inspection frequency.

Follow Up

What is the P–F interval, and is it long enough for practical inspections?

Would This Failure Be Evident If It Occurs on Its Own?

Why This Is Important

Evident vs hidden determines which failure management tasks are technically feasible

Red Flags & Common Traps

- Hidden vs evident misclassified - failures are often classified as hidden because we do not know the cause of failure.
- This is incorrect.
- The question asked is would the loss of function be evident?.

What Good Looks Like

A classification as evident or hidden when occurring on its own.

cut cut

cut

For Condition Based Tasks - Is There a Clear P–F Interval, and Roughly How Long Is It?

Why This Is Important

A defined P–F interval is required to justify on-condition tasks and set inspection frequency.

Red Flags & Common Traps

- P–F interval guessed with no basis.
- Too short for practical intervals.
- Mixing P–F with MTBF.
- Assuming "continuous monitoring" exists when it does not.

What Good Looks Like

A P–F interval estimate supported by evidence / physics and judged sufficient to allow a practical inspection frequency.

Tasks

Main Prompt

Which on-condition task would detect it in time, at a practical interval?

Instruction

Choose a detection method that will find it early enough and set an interval that is executable with current constraints.

Follow Up

Which detection method finds it early enough, and what interval is achievable with staffing / access constraints?

Tasks

Main Prompt

Have you considered the most effective and efficient of the range of condition based tasks (human senses, performance monitoring, product quality monitoring, Condition monitoring (dynamic, particle, physical, chemical, thermal, electrical))?

Instruction

Compare condition-task options (human senses, performance, quality, vibration / particles / chemical / thermal / electrical) and pick the best for the context.

Follow Up

Compare the candidate methods—human senses, performance, quality, vibration, oil / particle, thermal, electrical—which is the best for the context.?

Which On-condition Task Would Detect It in Time, at a Practical Interval?

Why This Is Important

The task must detect the condition early enough to act before functional failure at a workable interval.

Red Flags & Common Traps

- On-condition task chosen that detects too late.
- Interval unrealistic.
- Detection point not accessible.
- No response time to act before failure.

What Good Looks Like

A selected on-condition task that detects early enough, with interval justified and executable given access / staff constraints.

Have You Considered the Most Effective and Efficient of the Range of Condition Based Tasks (Human Senses, Performance

Why This Is Important

Selecting the most efficient detection method reduces workload while maintaining risk control.

Red Flags & Common Traps

- Selecting the fanciest technology not best ROI.
- Ignoring human senses / performance / quality monitoring.
- Overloading PdM program.
- No selection criteria.

What Good Looks Like

Selection of the most effective / efficient method for the mechanism and context.

Tasks

Main Prompt

If the failure has safety / environmental consequences is the task worth doing (ie does the task reduce the probability of failure to a tolerable level)?

Instruction

For safety / environment, test whether the task actually reduces probability to tolerable—not just adds activity—then document rationale.

Follow Up

Does this task measurably reduce failure probability to a tolerable level for HSE risk—what is the risk reduction argument?

cut

cut

cut

Tasks

Main Prompt

If the failure has economic consequences is the task worth doing (ie over a period of time, is the cost of doing the task less than the cost of the failure it is meant to manage)?

Instruction

For economic impact, compare task lifecycle cost versus expected failure cost avoided; keep only cost-effective tasks.

Follow Up

Over time, is task cost lower than expected failure cost avoided—can we estimate both with rough numbers?

If the Failure Has Safety/ Environmental Consequences Is the Task Worth Doing (ie Does the Task Reduce the Probability of…

Why This Is Important

For HSE consequences, tasks must reduce probability to a tolerable level, not just "do something."

Red Flags & Common Traps

- Safety task justified without risk reduction logic.
- Task does not reduce probability.
- Treating "inspection" as a magic control.
- Tolerable risk not considered.

What Good Looks Like

For HSE / environmental consequences, a strong argument that the task reduces probability to tolerable (not just activity).

cut cut

cut

If the Failure Has Economic Consequences Is the Task Worth Doing (ie Over a Period of Time, Is the Cost of Doing the Task…

Why This Is Important

For economic consequences, tasks must be cost-effective over time versus the failure they prevent.

Red Flags & Common Traps

- Economic task added without cost comparison.
- Ignoring labor / outage costs.
- Counting "avoided failure" without probability.
- Task becomes busywork.

What Good Looks Like

For economic consequences, a lifecycle cost comparison showing task cost < expected failure cost avoided over time.

Tasks

Main Prompt

For hidden failures, is a failure finding task technically feasible (ie it is possible to check if the item has failed, is it practical to do the task at the required interval)?

Instruction

For hidden failures, confirm proof testing is possible and practical at the needed interval

Follow Up

Can we practically test for failure at the needed interval (access, shutdown, time, coverage), and what is the proof-test method?

cut

cut

cut

Tasks

Main Prompt

For scheduled restoration or discard, is there an age at which there is a rapid increase in the conditional probability of failure, and do most failures occur after this age?

Instruction

For age-based actions, require wear-out (conditional probability increase) before assigning restoration / discard intervals.

Follow Up

Would there be wear-out (failure rate increases after age X); if yes, what age?

For Hidden Failures, Is a Failure Finding Task Technically Feasible (ie It Is Possible to Check If the Item Has Failed, Is It…

Why This Is Important

Hidden failures may require feasible proof testing; if not feasible, redesign or redundancy decisions may follow.

Red Flags & Common Traps

- Failure-finding technically possible but impractical.
- Proof test causes outages.
- Coverage incomplete.
- Interval not achievable.
- Tests introduce new risks.

What Good Looks Like

For hidden failures, proof testing shown to be technically possible and practical at the required interval

cut cut cut

For Scheduled Restoration or Discard, Is There an Age at Which There Is a Rapid Increase in the Conditional Probability of…

Why This Is Important

Age-related patterns justify scheduled restoration / discard; without them, age-based tasks add cost without benefit.

Red Flags & Common Traps

- Age-based restoration assigned without consideration of wear-out.
- Confusing increasing consequence with increasing probability.
- Using OEM intervals blindly.

What Good Looks Like

For age-based tasks, consideration of wear-out (increasing conditional failure probability after age X) with chosen age justified.

cut

Tasks

Main Prompt

Might a one-time change be technically feasible and worth doing? (Physical redesign, training, spares, procedural)?

Instruction

Consider one-time changes (redesign, training, spares, procedures) when cyclic maintenance tasks are ineffective or uneconomic.

Follow Up

Could a one-time change eliminate the failure mechanism (redesign / training / spares / procedure), and is it cheaper than the alternatives?

cut

cut

RELIABILITY MANAGEMENT

Structured prompt cards for RCM facilitation teams

www.reliabilitymanagement.co.uk

RCM FACILITATION TOOL KIT

55 Cards · 7 Categories · 6 Sections per Card

Might a One-time Change Be Technically Feasible and Worth Doing? (Physical Redesign, Training, Spares, Procedural)?

Why This Is Important

One-time changes can eliminate or materially reduce risk at lower lifecycle cost than recurring maintenance.

Red Flags & Common Traps

■ One-time change redesign / training / spares / procedure not considered.

What Good Looks Like

One-time changes considered, evaluated and recorded. They are appropriate for the failure modes they manage

cut

cut

cut

WHAT'S INSIDE

55 Structured Prompt Cards

7 RCM Categories Covered

6 Sections per Card

∞ Reusable Across Projects

CATEGORIES

Context & Scope	10
Functions	11
Functional Failures	3
Failure Modes	14
Failure Effects	7
Consequences	2
Tasks	8