# Permit System Self-Check Toolkit (V1)

## A pressure test for permit-to-work systems

Permit-to-work systems rarely fail in audits.
They fail when pressure enters the system, during routine work, overlapping activity, shift change and restart.

This toolkit is designed to test **how a permit system behaves under real operating conditions**, not how it appears on paper.

### What this is

- A short self-check to surface weak signals
- A prompt for better pre-job and pre-restart discussion
- A way to see where control is assumed rather than verified

### What this is not

- An audit
- A checklist
- A training course
- A scoring system

### How to use it

- Use on **one real job**
- Mixed group: operations, maintenance, issuer, contractor (if applicable)
- Work through the questions together
- Stop when discussion becomes uncomfortable, that's usually the exposure
- Capture **only what needs to change**

If this becomes paperwork, it has failed.



Published by **PermitToWorkGuide**

## Scope & pre-job clarity

**Aim:** Confirm the permit reflects the work as it will actually happen.

1. Does the permit describe **how** the work will be done, not just the outcome?
2. Would someone unfamiliar with the job understand:
    - access route
    - sequence
    - temporary states created?
3. What part of this task is most likely to change once work starts?
4. Are RAMS/method statements specific to this job, or reused by default?
5. What explicitly prevents 'same job, same assessment' thinking?

**Key signal:** If scope is vague, the permit is already behind the work.

**Red flags**

- Descriptions like "fix", "remove", "repair", "replace" with no access or sequence
- "We always do it like this"
- RAMS present but clearly generic or untouched
- Scope agreed verbally but not reflected on the permit

**Immediate fixes (if this section fails)**

- Require one extra sentence on every permit: **"Work method summary: access + key steps + major exposure."**
- Introduce one rule: **Complex work requires a pre-walkdown before the day of execution.**

  

## Isolation & energy control

**Isolation failures rarely come from missing steps. They come from assumed certainty.**

**Aim:** Test whether isolation is proven, not assumed.

1. How is isolation **physically verified** — not just confirmed on paper?
2. Are all forms of residual energy (pressure, heat, stored motion, fluids) explicitly checked, not assumed?
3. Is there a known minimum isolation standard for common hazards?
4. What happens when minimum isolation cannot be met?
5. Are temporary states (bypasses, carry-over isolations) formally controlled and re-verified at each shift or restart?
6. Are personal locks used, and are names visible?

**Key signal:** Paper certainty often replaces physical proof under pressure.

**Red flags**

- “It’s isolated because the sheet says so”
- Interlocks used as primary control

**Immediate fix**

- Add one required statement on isolation verification: **“Isolation proven by: (test method) / (who witnessed) / (time).”**

## Issuer behaviour & workface control

  

**Aim:** Detect drift caused by inconsistency and absence.

1. Do different issuers apply different standards?
2. Do people seek out the “easy” signature?
3. Do experienced workers receive less challenge because they are experienced?
4. Is the permit **displayed at the job**, not carried or left elsewhere?
5. Does the issuer or area authority visit the workface during the job?

**Key signal:** If standards depend on who is issuing, the system is already undermined.

**Red flags**

- “Go to him, he’ll sign it”
- Issuers never visit the workface
- Different standards by shift/issuer
- Jobs completed without permit, then backfilled

**Immediate fixes**

- Agree three non-negotiables every issuer enforces:
    1. permit displayed at job
    2. physical isolation verification
    3. workface check at least once per shift (for defined risk work)
- Introduce one behavioural standard: **No displayed permit = no work.**

  Published by **PermitToWorkGuide**

# SIMOPS, handover & Restart

**Aim:** Identify risk created between permits, shifts, and phases.

These failures occur at boundaries, between tasks, shifts, and phases.

## SIMOPS

1. Is there a shared view of overlapping work in the area, not just individual permits?
2. Is SIMOPS coordination explicitly owned, or assumed to happen "by default"?

## Handover

1. Does handover capture intent, or just status?
2. Is it clear what was difficult or nearly didn't work, not just what was completed?

## Restart

1. Is handback treated as a decision point or an admin step?
2. What new energy is being introduced at restart, and what could that energise unexpectedly?

**Key signal:** Risk often re-enters the system when scrutiny drops.

### Red flags

- SIMOPS managed "permit-by-permit"
- Handover notes are minimal
- Restart rushed once date slips
- Handback sections frequently incomplete

### Immediate fixes

- Create a daily SIMOPS snapshot (even simple): **area + tasks + key interactions + coordination owner.**
- Add one handover prompt: **"What nearly didn't work today?"**
- Treat restart as a phase with one question: **"What new energy is being introduced, and what could that energise unexpectedly?"**

  

# Pressure & vulnerability snapshot

**Aim**: Convert insight into one meaningful change.

> Do not score. Do not average. Do not justify.

## Where is pressure entering the system, and what does it weaken?

- Production / downtime → verification reduced
- Permit volume → challenge softened
- Staffing → shortcuts accepted
- Contractor scheduling → handover degraded
- Restart date pressure → premature sign-off

## Quick vulnerability snapshot

Rate each area as:

- ☐ **Stable**
- ☐ **Fragile**
- ☐ **Exposed**

Areas:

- scope clarity
- isolation integrity
- issuer consistency
- workface control
- SIMOPS coordination
- handover quality
- restart readiness

## What “good” looks like (without theatre)

A permit system is not strong because it is compliant.

It is strong when:

- it holds under time pressure
- it surfaces uncertainty rather than hiding it
- it prevents routine drift
- it maintains control during restart
- it makes SIMOPS interactions visible
- it ensures responsibility is explicit

## What to capture (and what to ignore)

**Capture only:**

- Top 3 exposures
- One immediate change
- One system issue to escalate

**Do not record:**

- Explanations
- Root causes
- Individual names
- Justifications for why something is “acceptable”

## Final Note

Permit systems should not be judged by how well they stop work.

They should be judged by how safely they allow work to begin, and begin again, under pressure.