OpsPilot

Defect Elimination — User Manual

Eliminate Recurring Defects Permanently · AI Engineering Co-Pilot

What this guide covers — what defect elimination is, how the OpsPilot module drives a defect to a permanent fix rather than a patch, what to have ready, and the report you receive.

1. What is defect elimination?

Defect elimination is about making a recurring defect impossible — or at least immediately detectable — rather than repeatedly fixing it. The distinction matters: patching treats the symptom and the defect returns; elimination removes the conditions that allow it, so it stops coming back. The discipline pushes hard past “the operator made a mistake” to the systemic reason the mistake was possible.

2. What the OpsPilot module does

Role	Responsibility
🎯 **AI Coach (OpsPilot)**	Pushes you to define the defect precisely, quantify its true impact, find the real root cause (systemic, not operator error), and design a solution that makes the defect impossible or instantly visible — not another band-aid.
👷 **Quality / Operations Lead (you)**	Provide the specific defect details, the occurrence data, and the process context. You validate the root cause and confirm the countermeasure will actually work in your environment.

3. How it works — the guided process

#	Stage	What happens
1	Precise defect definition	Measurable and specific — what, where in the process, since when.
2	Quantify impact	Production, quality and cost consequences.
3	Occurrence pattern	When and how often it appears — the pattern is a clue to the cause.
4	Physical cause analysis	The direct physical mechanism producing the defect.
5	Root cause	The systemic reason — not operator error.
6	Permanent countermeasure	A solution that prevents or instantly detects the defect (error-

#	Stage	What happens
		proofing).
7	Verification plan	How you'll prove the defect is actually gone.
8	Horizontal deployment	Where else the same fix should be applied.

4. What you will be asked — have this ready

- A precise description of the defect — what you're seeing, where it occurs, and when it was first noticed.
- Occurrence and cost data — how often, and what it costs in production, scrap or rework.
- The process context around where the defect appears.
- Any patterns you've noticed — shift, batch, material, machine, season.

Tip — “sometimes the parts are bad” won't get you a fix. “2–3 units per shift, always on line 2, worse after a material changeover” points straight at the cause.

5. What you receive — the output

A defect elimination report (Word) containing the precise defect definition, the quantified impact, the occurrence pattern, the physical and root-cause analysis, the permanent countermeasure, the verification plan, and the horizontal-deployment list.

6. Worked example (illustrative)

A bottling line mislabels roughly three units per shift, always after a product changeover. Defined precisely, the pattern points at the cause: the label reel is changed manually at changeover and the first labels run before the sensor re-registers. The root cause is not “operator error” — it's that nothing forces the sensor re-registration before the line restarts. The permanent countermeasure error-proofs it: an interlock that won't allow the line to start after a reel change until the registration check passes. Verification is simple — zero mislabels across the next twenty changeovers — and the same interlock deploys horizontally to the other three lines.

7. Getting the best result

- **Define it measurably.** A vague defect gets a vague fix.
- **Aim to error-proof.** The best countermeasure makes the defect physically impossible, not just less likely.
- **Reject operator error.** Ask why the process let the error happen — that's the real root.
- **Deploy horizontally.** If it can happen on one line, fix it on all of them.