

## OpsPilot

# Lessons Learned — User Manual

Transferable, Embedded, Verifiable Lessons · ISO 30401 · AI Engineering Co-Pilot



### AI-GENERATED CONTENT · INDEPENDENT VERIFICATION REQUIRED

This manual was produced with AI assistance and is only as good as the information it was given. Every statement, figure, standard reference and conclusion must be independently verified by a competent, suitably qualified person before it is relied upon. It is a draft aid to your judgement — not a finished, authoritative, or certifying document. Professional and legal responsibility for any reliance rests with you and your organisation. See the full Engineering Disclaimer at [opsinnovatech.com/engineering-disclaimer](https://opsinnovatech.com/engineering-disclaimer).

**What this guide covers** — what a real lesson learned is, how the OpsPilot module makes it stick, what to have ready, and the output you receive.

## 1. What is a lesson learned (that actually sticks)?

Most “lessons learned” are awareness notes — “remember to communicate better,” “ensure the team reviews the plan” — and they vanish within six months because nothing in the system changed. A real lesson is transferable, embedded and verifiable: it traces to a documented change in a controlled document, owned by a named person, that makes the mistake harder to repeat. The test is simple — if the only action is “tell people,” it isn't a lesson, it's a reminder, and reminders decay.

OpsPilot builds to *ISO 30401*, the *NASA LLIS lifecycle*, the *US Army After-Action Review structure*, *PMBOK/ISO 21502* and *Toyota's A3/Hansei discipline*. It strips vague verbs (ensure, manage, consider, review), forces every action to name a controlled document and an owner, and pushes up the hierarchy of embedding — a design-standard change beats “communicate to the team” every time.

## 2. What the OpsPilot module does

Role	Responsibility
<b>AI Knowledge Coach (OpsPilot)</b>	Builds transferable, embeddable, verifiable lessons — challenging vague verbs, forcing each action to name a specific controlled document and owner, pushing up the hierarchy of embedding, and insisting on weak-signal analysis and prior-lesson cross-references.
<b>Knowledge Owner (you)</b>	You have first-hand context — what happened and why. You provide the source event, the expected-versus-actual narrative, the systemic gap, the specific document edit that will embed the lesson, and the verification.

## 3. The hierarchy of embedding

Not all actions are equal. The higher up this hierarchy, the more durable the lesson — OpsPilot pushes you up it:

Level	Type of action (strongest first)
Level 1	Change a design standard or specification — the mistake becomes impossible.
Level 2–3	Change a procedure, checklist or controlled template.
Level 4–5	Add a training or competency requirement.
Level 6	“Communicate to the team” — weakest; decays within months.

## 4. What you will be asked — have this ready

---

- The source event and the differential narrative — what was expected versus what actually happened.
- The systemic gap that allowed it.
- The specific controlled-document edit that will embed the lesson, and its owner.
- How the change will be verified as effective.

## 5. What you receive — the output

---

A complete Lessons Learned record (Word): the source event, the expected-versus-actual analysis, the systemic gap, weak-signal analysis, cross-references to prior lessons, and an embedding action that names a controlled document and owner — traceable and verifiable, not awareness-only.

## 6. Worked example (illustrative)

---

A project suffered a costly clash because a vendor's drawing used a different datum convention. The weak lesson: “remind engineers to check datum conventions.” Gone in six months. OpsPilot strips that and pushes for the embedding action: amend the drawing-control procedure to require datum-convention verification at the design-review checkpoint, with the drawing-management lead as owner, and verify by auditing the next ten vendor drawing approvals. Now the lesson lives in a controlled document at a checkpoint that already happens — it's structural, not a memory test. The weak-signal analysis also asks whether earlier near-misses hinted at this, and cross-references any prior datum lessons so the organisation isn't re-learning the same thing.

## 7. Getting the best result

---

- **Embed, don't announce.** If the action is “tell people,” it will decay — name a document to change.
- **Climb the hierarchy.** A design-standard change beats a procedure tweak beats a reminder.
- **Name a document and an owner.** A lesson with no controlled-document home isn't embedded.
- **Verify it worked.** An embedding action without verification is a hope, not a lesson.

---

OpsPilot — AI Engineering Co-Pilot. Learn more at [opsinnovatech.com](https://opsinnovatech.com)