



STANDARD OPERATING PROCEDURE

PMO ↔ Product ↔ Engineering

Delivery Process Standard Operating Procedure

Owner	Aman · Product FleetRobo
Date	12 May 2026
Version	1.0 (Draft)
Status	For review — PMO Lead, Product PMs, Eng Lead

INSIDE THIS DOCUMENT

- 1. **Why this SOP exists**
- 2. **Operating principles** — 10 non-negotiables
- 3. **The three phases** — Planning, Dev Build, Rollout
- 4. **The three intake lanes** — A / B / C routing
- 5. **Work classification** — Bug / Small / Dev / Major
- 6. **Track A · P0 incidents** — realistic SLAs
- 7. **Phase 1 · Planning** — Friday Decisions Day
- 8. **Phase 2 · Dev Build** — Tuesday Dev Release
- 9. **Phase 3 · Rollout** — 4 rollout types
- 10. **The tracker** — lane + phase schema
- 11. **Communication protocol**
- 12. **Quality gates & QA's full role**
- 13. **RACI · Metrics · Rollout plan**

DISTRIBUTION	CLASSIFICATION	NEXT REVIEW
Product · PMO · Eng · QA · CS	Internal Use	2026-06-09

A living document. Use it, break it, tell me where it breaks. v1.1 in 4 weeks.

PMO ↔ Product ↔ Engineering Delivery Process

Owner: Aman · Product | FleetRobo **Version:** 1.0 (Draft) **Date:** 2026-05-12 **Audience:** PMO, Product Managers, Engineering Leads, QA, Customer Success, Support **Status:** For review — PMO Lead, Product PMs, QA Lead, Eng Lead

0. Why This SOP Exists

Today, three things are happening at once and stepping on each other:

- Customer-success raises bugs.
- PMO raises enhancement requests on behalf of clients.
- Engineering ships features without release notes, training, or QA-mimicked production data.

The result is predictable — *features get pushed to customers before they are ready*, PMO is implicitly acting as QA, and there is no defensible commitment line back to the client. Customers escalate, the team firefights, and the same loop repeats.

This SOP fixes that by:

1. **Splitting work into three phases** — Planning, Dev Build, Production Rollout — each with its own owner and exit gate.
 2. **Separating Dev Release from Client Release.** Dev finishing is not the same as clients seeing it. Production rollout is its own phase, sized by item type.
 3. **Making decisions explicit** — every ticket gets an Accept / Decline / Park outcome by Friday EOD. No black holes.
 4. **Putting QA + release-notes + training + rollout planning inside the release gate**, not after it.
 5. **One source of truth** (Odo), one weekly Tuesday sync, one Friday decision day.
-

1. Operating Principles

#	Principle	What it means in practice
1	Three intake lanes.	Work enters via Lane A (PM-Routed), Lane B (QA-Routed bug fixes), or Lane C (Incident-Routed POs). The lane decides the path.
2	Three phases, three owners.	Planning (PM), Dev Build (Dev Lead), Production Rollout (PM). Each phase has a clear exit gate.
3	Dev Release ≠ Client Release.	Tuesday Dev Release is internal. Client Release happens after rollout planning, sized by item type.
4	PMO is not QA. QA Lead routes bugs.	PMO does functional validation. QA Lead triages bugs and can route Lane B items directly to Dev without PM bottleneck.
5	PM visibility, not PM bottleneck.	PM sees every Lane B item in the tracker automatically and can pull any item back to Lane A. Visibility is non-blocking.
6	No verbal commitments to clients.	Every external date comes from the tracker's <code>Planned Client Release</code> field.
7	Friday is Decisions Day.	By Friday EOD, every open Lane A ticket has Accept / Decline / Park communicated. Lane B items move on their own QA cadence.
8	Rollout type is picked at planning, not at release.	PM commits to rollout type when Dev picks up the work, not when Dev hands it back.
9	Templates do the lifting.	Every artifact has a template. PM fills blanks, doesn't write from scratch. Low cognitive load is non-negotiable.
10	Capacity guardrail.	50% Core Product, 30% Small Changes, 20% Issues. Anything that breaks this ratio needs Head of Product approval.

2. The Three Phases

Every item that enters the tracker moves through three phases. Each phase has an owner, a duration window, and an exit gate. Time spent in each phase is sized by the item's tier (see Severity Tier System v1).

PHASE 1 · PLANNING

Owner: PM

Duration: tier-dependent (15 min to 5 business days)

Exit gate: Friday Decision – Accept / Decline / Park

↓

PHASE 2 · DEV BUILD

Owner: Dev Lead

Duration: Dev-decided (1 sprint, 2 sprints, or per scope)

Exit gate: Tuesday Dev Release – build on internal env

↓

PHASE 3 · PRODUCTION ROLLOUT

Owner: PM

Duration: type-dependent (same day to 3 weeks)

Exit gate: Client Release + comms sent + monitoring active

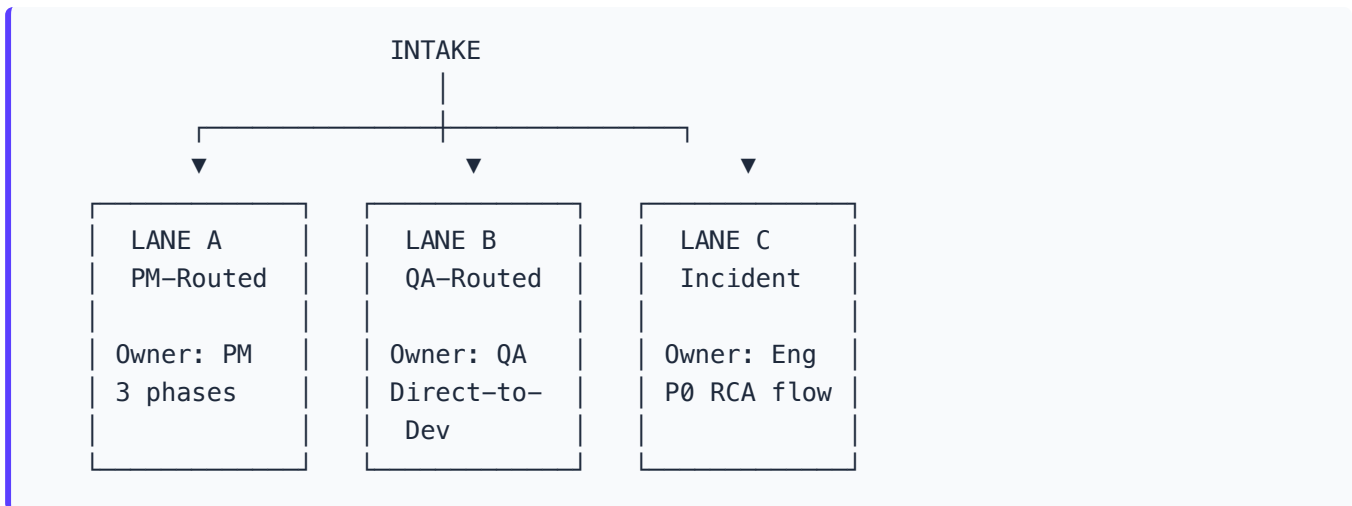
Two distinct rhythms anchor the week:

- **Tuesday · PM ↔ PMO Sync** — weekly status check across all three phases. Visibility, not gating. Discussion can start any day; Tuesday is the minimum cadence. **QA Lead triage meeting happens before** this sync (Tuesday morning, QA picks the slot) so the latest Lane B routing is already visible by 11:00.
- **Friday · Decisions Day** — by EOD, every open Lane A ticket has its outcome communicated to the reporter. Dev sprint scope for the coming sprint is handed to the Dev Lead.

Lanes feed Phases. Lane A items move through all three phases (Planning → Dev Build → Rollout). Lane B items skip Planning (QA Lead triages directly) and enter at Dev Build. Lane C items are P0 incidents with their own response flow (§5).

3. The Three Intake Lanes

Every item that enters the tracker comes through one of three doors. The door determines the path, the owner, and the cognitive load on PM.



3.1 Lane A – PM-Routed

For: Features, enhancements, anything requiring product judgment, anything client-facing where comms matter.

Path: Phase 1 Planning → Phase 2 Dev Build → Phase 3 Production Rollout (the full SOP).

Examples:

- New WhatsApp alerts feature
- Trip Management flow redesign
- All-camera dashboard view
- Hikvision NVR integration
- Pricing structure change
- Anything from Sales or strategic asks

Why PM is in the path: Scope decisions, client comms, sales motion, rollout type, training material. Product judgment needed.

3.2 Lane B – QA-Routed

For: Defects against agreed spec. Things that are clearly broken and clearly fixable without product judgment.

Path: QA Lead triages → directly into Dev sprint → standard Dev Release → Rollout Type 1 or 2.

Owner: QA Lead. They are the triager. They decide:

- Is this actually a bug, or is it expected behaviour PMO is unfamiliar with?
- Is it reproducible? What's the evidence?
- What tier (P1 / P2 / P3)?
- Can it go direct, or does it need PM judgment?

Qualifies for Lane B if ALL of:

1. No product decision needed (the fix is "match the spec")
2. No client comms needed beyond standard release notes
3. Rollout Type 1 or 2 (Direct or Notified) sufficient

4. Single client OR cross-client but uncontroversial
5. Not field-dependent (otherwise Type 4 needed → Lane A)

Examples:

- Button misaligned in dashboard
- Report column sorting reversed
- Typo in error message or label
- API endpoint returning wrong HTTP status code
- Regression caught by QA automation after a Tuesday Dev Release
- UI flow broken on specific browser

Triage SLA (QA Lead):

- P1 Lane B item: triaged within 4 hours
- P2 Lane B item: triaged within 1 business day
- P3 Lane B item: triaged within 3 business days

PM visibility mechanic (non-blocking):

- Every Lane B item is auto-tagged in PM's tracker view
- PM is cc'd on the QA routing decision (tracker notification, not a meeting)
- PM can pull any Lane B item back to Lane A within the triage SLA window
- Default is **QA-routed**. PM stays out unless they choose in.

Auto-promotion to Lane A (Lane B → Lane A) happens when:

- The fix turns out to need Rollout Type 3 or 4 → promote
- Client comms beyond release notes are needed → promote
- Product judgment surfaces during scoping → promote
- QA Lead is unsure → default to promote (Lane A always works)

3.3 Lane C – Incident-Routed

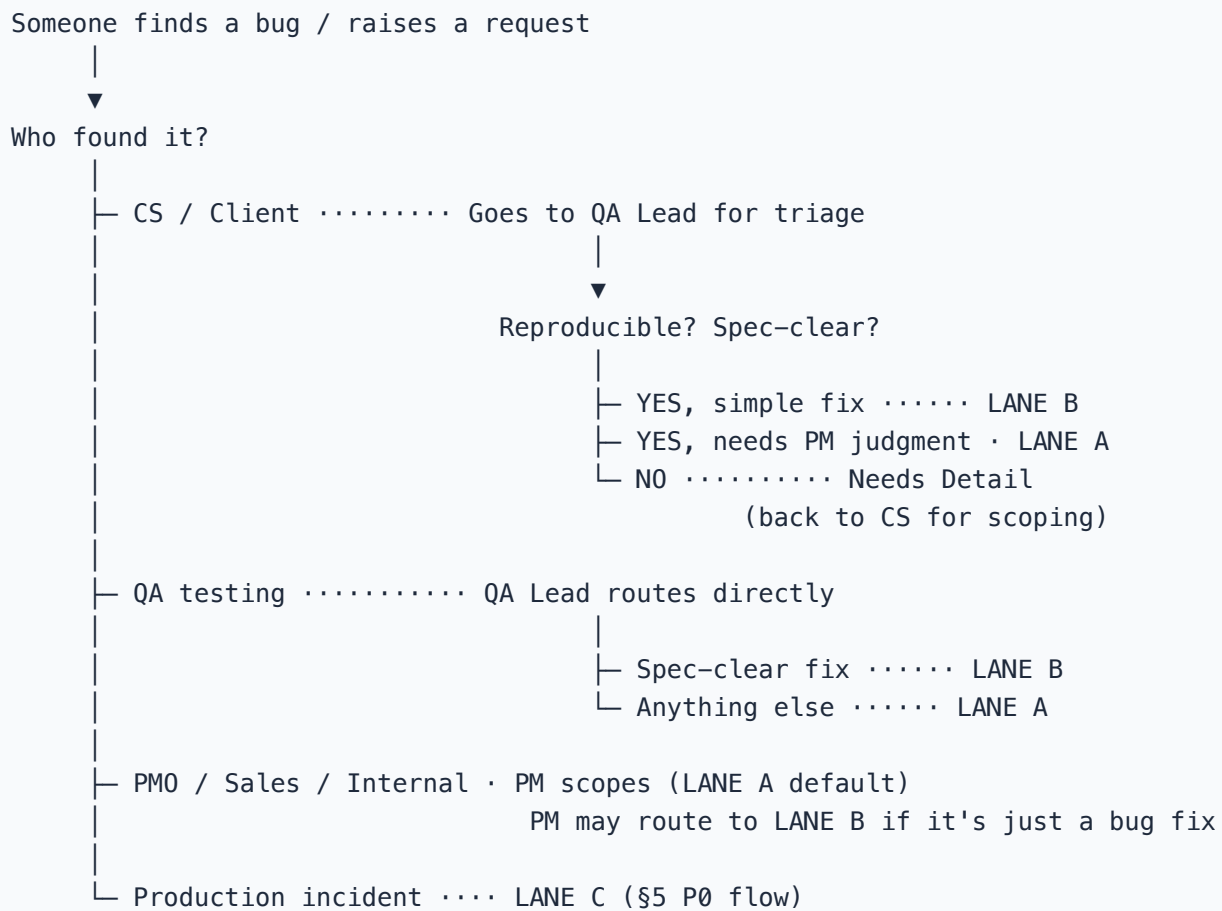
For: P0 incidents only.

Path: Teams `#incidents-bsl` → Eng Lead acknowledges → mitigation → RCA → permanent-fix ticket created in Lane A or Lane B depending on type.

Full flow in §5 (Track A — P0 Incident Response).

3.4 Routing Decision Tree

A single-screen flow for who routes what.



3.5 The "Bug vs Feature" Line

PMO will sometimes raise bugs that are actually feature requests in disguise ("the report should ALSO show X"). QA Lead is empowered to push back.

Hard rule:

- **If it's adding new capability** → it's a feature → Lane A (PM scopes)
- **If it's restoring promised capability** → it's a bug → Lane B (QA routes)

When the line is ambiguous, QA Lead loops in PM via tracker comment. No meeting needed.

4. Work Classification

Every incoming item is classified within **24 hours of capture** into one of four types. Classification is owned by the receiving PM (or CS Lead for inbound bugs).

Type	Definition	Trigger Examples	Tier hint
P0 Incident	Platform on fire. Mass impact, no workaround.	Server down, mass auth fail, data ingestion stopped	P0 (same-day, §5)
Bug	Defect against agreed spec	Button broken, report column wrong	P1 / P2
Small Change	<2 dev-days, no architectural impact	New filter, label change	P2 / P3
Major Feature / Enhancement	>2 dev-days, new module / entity / API	Trip-flow redesign, new dashboard	P1 / P2

Tier is set per the Severity Tier System (4-dimension rubric). Type is operational shorthand for the planning conversation. **If unsure, default to the higher track.** Misclassifying a feature as a small change is the single biggest cause of release-day surprises.

5. Track A (Lane C) – P0 Incident Response

5.1 Definition Gate

P0 means **at least one** of:

- $\geq 1,000$ devices/vehicles offline simultaneously
- Platform-wide login / auth failure
- Data ingestion stopped (Kafka/Java queue jam, log silence)
- Server unreachable for >5 min
- Mass dashboard data inconsistency (>10% trips)
- Customer-facing module fully unusable for any paying client

Anything else is **P1 Bug** and follows §6 (Planning) → §7 (Dev Build) → §8 (Production Rollout) for Lane A items, or QA-triaged Lane B for spec-clear fixes.

5.2 Flow

```

T+0      CS or PMO detects → posts in #incidents-bsl (Teams) with template
T+30 min On-call Eng Lead acknowledges in thread
T+1 hour Severity confirmed (P0 vs P1 demoted) + war-room call if not mitigated
T+2 hours Mitigation work explicitly in motion (Eng confirms direction + ETA)
T+...    Stakeholder updates every 60 min in thread until mitigated
T+24h    RCA document published in tracker (template in §5.4)
T+72h    RCA review meeting with Engineering Leadership + PM + Eng Lead

```

These SLAs are realistic for BSL today (single Eng Lead, no 24/7 rotation). If the Eng Lead is unreachable beyond T+30 min, the incident escalates to Head of Product who pages a backup.

5.3 Incident Post Template (Teams)

```

● [P0] <one-line title>
Detected: <time>
Reporter: <name>
Affected: <client/module/device count>
Symptom: <what user sees>
Last working: <approx time or unknown>
Evidence: <screenshot/log link>

```

5.4 RCA Template (mandatory within 24h)

- **What happened** (timeline with timestamps)
- **Impact** (clients, revenue, support cost)
- **Root cause** (5-whys)
- **Mitigation** (what made it stop)
- **Permanent fix** (Odoo ticket ID + owner + date)
- **Prevention** (test / monitor / alert to add)

5.5 Misuse Guardrail

A P0 channel is worthless if everything becomes P0. If a category triggers ≥ 2 incidents/week or ≥ 10 /month, it stops being an incident and becomes a **systemic backlog item** owned by Engineering (forced into the 15-day cycle as a reliability feature). Tracked on the P0 dashboard.

6. Phase 1 – Planning (Lane A)

Owner: PM **Duration:** tier-dependent (response SLA per Severity Tier System) **Exit gate:** Friday Decisions Day — Accept / Decline / Park communicated to reporter

Note: Phase 1 applies to **Lane A items only**. Lane B items skip Planning — QA Lead triages and routes them directly into Dev Build (see §3.2).

6.1 What Happens in Planning

Planning is the work between "ticket arrived" and "Dev picks it up." It is *not* optional. Skipping it is how we end up with half-scoped builds and missed acceptance criteria.

PM owns this phase. The PM:

1. **Acknowledges receipt** within the tier's Response SLA (P0 = 15 min, P1 = 4h, P2 = 2 days, P3 = 5 days)
2. **Scopes the request** — talks to the reporter, the affected client, CS, or Sales as needed
3. **Drafts acceptance criteria** — what does "done" look like
4. **Identifies dependencies** — other tickets, infra blockers, pilot client needs
5. **Estimates effort** with the Dev Lead (rough sprint sizing, not story points)
6. **Picks the rollout type** (see Production Rollout Playbook v1 — Direct / Notified / Demoed / Piloted)
7. **Decides by Friday** — Accept, Decline, or Park

6.2 The Tuesday PM ↔ PMO Sync

Weekly cadence. Visibility check across all three phases. Items typically discussed:

- New tickets received since last Tuesday — PMO walks PM through what's come in
- Status of tickets in Dev Build (anything slipping?)
- Status of tickets in Production Rollout (any client noise?)
- Anything escalated (client pressure, sales motion, executive ask)

This sync triggers nothing by itself. It's a visibility checkpoint. Discussion can start any day; Tuesday is the minimum cadence.

6.3 Friday – Decisions Day

By Friday EOD, every open ticket in Planning has ONE of three outcomes communicated to the reporter:

Decision	What it means	What PM communicates
Accept	Going into upcoming Dev sprint	Sprint date + target Dev Release date + rollout type
Decline	Not happening	Reason + alternative if any
Park	Accepted but not now	Revisit date + dependency / trigger

By Friday EOD, every reporter knows where their ticket stands. This eliminates the ticket-black-hole problem.

Friday is also when **Dev sprint scope is handed to the Dev Lead** for their sprint planning. Dev decides their own sprint length and what fits; PM provides clarity on what's needed and when.

6.4 Definition of Ready (Friday Handoff to Dev)

A ticket cannot enter Dev Build unless it has:

- Tier set per Severity Tier System
- Clear acceptance criteria
- PM owner assigned
- Evidence attached (logs / screenshot / data sample)
- Affected clients identified

- Test data available in QA env (or flagged `field-dependent`)
- Effort estimate confirmed by Dev Lead
- **Rollout type picked** (Direct / Notified / Demoed / Piloted)

Missing any item → ticket stays in Planning, rolls to next Friday.

6.5 Capacity Allocation (per Dev squad, per week)

- 50% — Core Product (major features)
- 30% — Small Changes (PMO-originated)
- 20% — Bugs (CS-originated)

This ratio is non-negotiable without Head of Product sign-off. PM does not push more than capacity allows into a sprint.

7. Phase 2 – Dev Build

Owner: Dev Lead **Duration:** Dev-decided (1 sprint, 2 sprints, or whatever scope demands) **Exit gate:** Tuesday Dev Release — build on internal env, verified by PM (Lane A) or QA Lead (Lane B)

7.1 What Happens in Dev Build

Dev runs their own sprint cycle. PM does not micromanage internal Dev rhythm. PM's job in this phase:

1. **Be available** for clarifications during sprint
2. **Track the target Dev Release date** committed in Planning Phase
3. **Surface slips early** — if Dev flags a date risk, PM updates the tracker and notifies PMO + reporter
4. **Confirm the build** on the internal environment when Dev says it's ready

7.2 Sprint Length

Dev Lead picks sprint length per item. Typical patterns:

- **Bug / Small Change:** 1 sprint (1 week)
- **Standard Feature:** 1–2 sprints
- **Major Feature / Module:** 2–3 sprints OR a dedicated feature track
- **Stability item:** absorbed into existing sprint capacity OR scheduled separately by Eng Leadership

PM controls the "by when," Dev controls the "how many sprints." If Dev says a feature needs 3 sprints but PM has committed a 2-sprint delivery, that conversation happens in Planning Phase — not during the build.

7.3 Tuesday Dev Release

Every Tuesday is the Dev Release window. Items ready for release ship to the internal environment for PM verification. This is **not** the client release — it's the handoff from Dev to PM for rollout planning.

Dev Release Checklist:

- Code merged + deployed to internal env
- QA sign-off on prod-mimic data
- Build matches acceptance criteria from Planning
- Known limitations documented
- Rollback steps confirmed with Eng Lead

PM verifies and signs off. If anything is missing, the item rolls to the next Tuesday Dev Release window.

8. Phase 3 – Production Rollout

Owner: PM **Duration:** type-dependent (same day to 3 weeks) **Exit gate:** Client Release + comms sent + monitoring on

8.1 Rollout Type Drives Everything

The rollout type was picked in Planning Phase. Now it executes. The four types:

Type	Use For	Time from Dev Release to Client	PM Effort
1 · Direct	Single-client bug fix, server-side patch, P0 hotfix	Same day	30 min
2 · Notified	Cross-client bug, visible UX change	1-2 days	1-2 hours
3 · Demoed	New small feature, cross-client UX change, integration	3-5 days	Half day
4 · Piloted	New module, breaking change, field-dependent, major feature	1-3 weeks	1-3 days spread

Full templates, decision flow, and artifact lists are in **Production Rollout Playbook v1**.

8.2 Default Mapping (When PM Doesn't Override)

Tracker Type field	Default Rollout
Bug (single client)	Type 1
Bug (cross-client)	Type 2
Small Change	Type 2 or 3 — PM picks based on visibility
Development / Enhancement	Type 3
Major Feature / New Module	Type 4
P0 Incident hotfix	Type 1

The default exists so PMs don't think hard for routine items. When unsure, bump up one type.

8.3 The Mandatory Artifacts (Cross-Type Floor)

Regardless of type, every release ships with at minimum:

Artifact	Owner	Notes
Release note	PM	1 line for Type 1, paragraph for Type 2, page for Type 3, multi-page for Type 4
CS notification	PM → CS	Always. Even for invisible fixes.
Client comms	CS (content from PM)	Required for Types 2–4. Optional for Type 1 unless client was affected.
Tracker update	PM	Status flows to Released . Actual Release Date filled.

Type 3 adds: internal demo + training material + sales kit if revenue-relevant. Type 4 adds: pilot client agreement + daily check-ins + pilot retro + phased rollout schedule + rollback plan.

8.4 The PMO Functional Validation Step

Before client release (for Types 2–4), PMO does **functional validation** — end-to-end check from a user's perspective. This is *not* bug-discovery (QA already did that on prod-mimic env). PMO is checking: "Does this work the way a real client will use it?"

If PMO finds a bug, it goes back to Dev as a P1 (new ticket, new Planning Phase). The current rollout does **not** proceed until the bug is fixed and the build re-released.

8.5 Post-Rollout Stabilisation

For Types 3 and 4, a stabilisation window follows client release:

- **Type 3:** 5 business days of active monitoring, daily tracker check by PM
- **Type 4:** 1 week stabilisation + post-rollout review (PM + Eng + CS)

Any issue raised in stabilisation is treated as a P1 by default.

8.6 The PMO Walk-Through (For Types 3 + 4)

Before client release, PM hosts a 30-min walk-through with PMO + CS. Purpose: PMO and CS see the feature *before* clients do, ask their questions, and align on talking points. This is the moment that converts PMO from "QA" to "informed stakeholder."

9. The Tracker (Odoo)

9.1 Single Source of Truth

All work — Issues, Small Changes, Major Features — lives in **one Odoo project** with **type filters**, not separate sheets. Excel is interim only until Odoo migration is complete (target: this week).

9.2 Mandatory Fields

Grouped by phase. PM doesn't fill all at once — fields populate as the ticket moves through phases.

Always (set at intake):

Field	Description
Ticket ID	Auto
Title	One-line problem statement
Type	Bug / Small Change / Development / Major Feature / P0 Incident
Product	LPA / TMS / VT / FleetRobo Core / WMS / Other
Reporter Dept	PMO / CS / Sales / Internal
Affected Client(s)	List
Tier	P0 / P1 / P2 / P3 / Stability / Needs Detail (per Severity Tier System)
Severity Notes	1-2 lines of reasoning for the tier
Evidence	Link to logs/screenshot/recording
PMO/OPS Remark	PMO-only notes
Hot Flag	🔥 / blank

Intake routing (set by QA Lead or PM at first triage):

Field	Description
Lane	A (PM-Routed) / B (QA-Routed) / C (Incident)
QA Triage Decision	Lane B / Promote to Lane A / Needs Detail / Not a bug
QA Owner Role	Set when Lane = B

Phase 1 · Planning (set by PM before Friday Decision — Lane A only):

Field	Description
PM Owner Role	Role label, not individual name
PM Handover Date	When PMO handed off to PM
Blast Radius	High / Medium / Low + count
Workaround	None / Manual / Easy
Time Started	Date or Chronic
Rollout Type	1 Direct / 2 Notified / 3 Demoed / 4 Piloted
Decision Outcome	Accept / Decline / Park (set by Friday EOD)
Decision Notes	Reason if Decline, revisit date if Park
Planned Dev Release Date	Target Tuesday Dev Release
Planned Client Release Date	The only date shared externally

Phase 2 · Dev Build (set by Dev Lead):

Field	Description
Dev Handover Date	When PM handed off to Dev
Sprint Estimate	1 sprint / 2 sprints / per-scope
Dev Release Date	Actual Tuesday the build hit internal env

Phase 3 · Rollout (set by PM):

Field	Description
Rollout Status	Awaiting / Demoed / In Pilot / Phased / Released / Stabilising
Actual Client Release Date	Filled when clients see it
Post-Rollout Notes	Issues, escalations, retro link (Type 3+4)

Status flow (always visible):

Lane A: Open → Triage → In Planning → Decision Made →
 In Dev → In QA → Dev Released →
 Rollout Planning → In Rollout → Released → Stabilising → Closed

Lane B: Open → QA Triage → In Dev → In QA → Dev Released →
 Rollout (Type 1 or 2) → Released → Closed

Lane C: Open → Acknowledged → Mitigating → Mitigated → RCA → Closed
 (+ permanent-fix ticket created in Lane A or B)

9.3 Views

- **By Lane** — A (PM queue) / B (QA queue) / C (incidents)
- **By Phase** — Planning / Dev Build / Rollout / Closed
- **By Tier** — P0, P1, P2, P3, Stability, Needs Detail
- **By Product** — LPA, TMS, VT, FleetRobo Core, Cross-product
- **By PM** — Each PM's owned tickets (Lane A + Lane B items they're cc'd on)
- **By QA Lead** — All Lane B items in triage or routing
- **By Week** — What hits Dev Release this Tuesday / Client Release this week / In stabilisation
- **Friday Decisions Pending** — every Lane A ticket still awaiting Accept / Decline / Park

9.4 Access

Read + write for: All PMs, PMO, CS Leads, Eng Leads, QA, Support Leads. Comment-only: Sales (for visibility on what clients can expect). Read-only: Engineering Leadership + Head of Product (for tracking, not ticket-grinding).

10. Communication Protocol

10.1 Internal Cadence

Sync	When	Who	Purpose
Daily Dev standup	10:00 IST (15 min)	Eng squad	Blockers only
QA Lead Triage	Tue morning (QA picks slot, before PM↔PMO sync)	QA Lead + Dev Lead	Triage Lane B intake, route bugs, decide Lane B vs Lane A. PM + PMO optional observers.
PM ↔ PMO Weekly Sync	Tue 11:00 IST (45 min)	PM + PMO Lead	Status across all three phases. Lane B routing visible from QA triage. Visibility, not gating.
Friday Decisions Day	Fri 14:00–16:00 IST	PM	Communicate Accept / Decline / Park for every open Lane A Planning ticket. Hand sprint scope to Dev Lead.
Dev Sprint Planning	Friday or per Dev Lead's call	Dev Lead + Dev squad	Internal sprint planning based on PM-handed scope + QA-routed Lane B items
Tuesday Dev Release	Every Tue	Dev Lead + PM + QA Lead	Build hits internal env. PM verifies Lane A. QA Lead verifies Lane B. Internal demo if Type 3+4.
Bi-weekly Feature Sync	Alt Mon 11:00 IST (60 min)	PM + Eng Lead + Engineering Leadership	Major-feature progress, Stability Track review
Monthly Product Review	Last Wed of month (90 min)	Full product family	Strategic + metrics

10.2 External Communication (Client-Facing)

- **Date commitments:** Only **Planned Release Date** from Odoo is shared.
- **Slip alerts:** If **Planned Release Date** will slip, PMO sends client comms *within 24 hours of slip-detection* — never on release day. Template in §10.3.
- **Release announcements:** PM publishes release notes; PMO forwards to client with feature highlights and training resources.

10.3 Slip Communication Template

```

Subject: Update on <feature/fix> delivery – <client name>

Hi <client>,

Following up on <ticket ref>. We had committed delivery by <original date>.

We're tracking to deliver by <revised date> instead.

Reason: <one-line, honest, no jargon>
What we're doing about it: <one-line>
Impact on you: <one-line or "none">

I'll send the next update by <date> regardless of status.

– <PMO contact>

```

11. Quality Gates & QA's Full Role

11.1 Current State Problem

PMO is being handed features to test functionally, but PMO is *not* a QA function. And QA has been treated as a passive sign-off step, not as a function that owns bug routing and quality across the platform. Both gaps need closing.

11.2 QA Lead's Four Functions

#	Function	What it means
1	Originator	Files bugs caught by manual testing, regression suite, automation. Triage bugs raised by CS. Routes to Lane A (PM) or Lane B (direct to Dev).
2	Validator	Sign-off on Tuesday Dev Release. Functional validation on prod-mimic env before client release. Distinct from PMO validation (QA = matches spec; PMO = feels right for the user).
3	Quality Owner	Maintains regression test suite. Owns prod-mimic env health. Reports regression rate as a KPI. Flags structural patterns to the Stability Track.
4	Post-Release Monitor	Owns the stabilisation window for Types 3 + 4 releases. Surfaces post-release bugs back into the system. Decides which need to interrupt the next sprint vs roll to next cycle.

11.3 The Quality Pipeline

Stage	Owner	What is verified
Unit + integration tests	Dev	Code correctness
QA on prod-mimic env (in build by Infra team)	QA Lead	Functional correctness, regression, performance — against agreed spec
PMO functional validation	PMO	End-to-end flow as a <i>user</i> would experience it (not bug-hunting)
Pilot client (Type 4 only)	PM + CS	Real-world data behaviour
Broad release	PM + PMO	Rollout to all clients
Post-release stabilisation	QA Lead + PM	Monitoring + bug surfacing in the 5–7 day window after release

PMO entering the loop only **after** QA sign-off is the hard rule. If QA isn't ready by Day 10 of a Type 4 cycle, the release **pauses**.

11.4 QA Lead's Authority

- **Lane B routing decision** — QA Lead decides if a bug goes Lane B (direct to Dev) or Lane A (needs PM scoping). PM can pull back, but QA's default routing is the starting point.
- **Tier veto on Lane B** — if QA Lead believes a Lane B item is mis-tiered (e.g. reporter says P1 but blast radius is single-user), QA Lead can down-tier with documented reason. Mirrors Eng Lead's P0 veto authority.
- **"Not a bug" verdict** — QA Lead can close items as "not a bug" (expected behaviour, reporter misunderstanding, no repro). Closed items are visible to PMO; PMO can escalate if disagreement.
- **Promote to Stability Track** — if QA Lead sees a regression pattern (same module bugged 3+ times in 30 days), QA Lead can flag it as Stability Track work, same as PMO can for P0 patterns.

11.5 Prod-Like Test Environment

The unblocker for everything in this section. It's already on the Infra team's roadmap. **Product tracks this weekly** until live, because every other quality gate depends on it.

Until the prod-mimic env is live, QA Lead operates on the staging environment with documented data gaps. Field-dependent items are flagged at intake and routed to pilot clients (Type 4).

12. Roles & Responsibilities (RACI)

Activity	PMO	PM	Dev Lead	QA Lead	CS
Capture incoming bug	A	I	I	R	R
Capture enhancement request	R	A	I	I	C
Lane routing (A vs B)	C	C	I	A/R	C
Classify tier (P0/P1/P2/P3)	R	A (Lane A)	C	A (Lane B)	C
Detail requirements (Lane A)	I	A/R	C	I	C
Bug reproduction + scoping (Lane B)	I	I	C	A/R	C
Estimate	I	C	A/R	C	I
Code + unit test	I	C	A/R	C	I
QA on prod-mimic env	I	I	C	A/R	I
Regression suite ownership	I	I	C	A/R	I
PMO functional validation	A/R	C	I	C	I
Release notes	C	A/R (Lane A)	I	A/R (Lane B)	C
Training material	C	A/R	I	I	R
Client communication	A/R	C	I	I	R
Post-release stabilisation monitoring	C	C	C	A/R	C
RCA (post-P0)	C	A	R	R	C

Legend: R=Responsible, A=Accountable, C=Consulted, I=Informed.

13. Metrics (How We Know This Is Working)

Reviewed in the Monthly Product Review. Grouped by what they measure.

Process discipline:

Metric	Target	Why it matters
% of releases shipped with full artifacts (notes + training + comms)	≥ 95%	The reason features look low-value to clients today
% of Planned Client Release Date met (no slip)	≥ 85%	Trust with clients
% of rollout types assigned at Planning (not at release)	≥ 90%	Discipline marker — no scrambling
Friday Decisions Day compliance	100% of open Lane A tickets resolved by EOD	No black holes
Avg cycle time — Bug (Lane B)	≤ 7 days	QA routing working
Avg cycle time — Major Feature (Lane A)	≤ 15 days	Sprint discipline

Lane health (the new dimension):

Metric	Target	Why it matters
% of bugs routed Lane B (vs Lane A)	≥ 60%	PM cognitive load reduction working
QA Triage SLA hit rate	≥ 90%	QA Lead not a bottleneck
Lane B → Lane A promotions per month	< 15% of Lane B intake	Routing accuracy
PM pull-backs from Lane B to Lane A	< 5%	PM trust in QA routing

Reliability:

Metric	Target	Why it matters
P0 incidents per month	Trending down	Reliability
Repeat P0 categories	0 (each repeat = Stability Track)	We're learning
Regression rate (post-release bugs in stabilisation)	< 10% of releases	QA suite working
Client escalations on released features	Trending down	Quality gate working
PMO QA-discovery rate	Trending to 0	PMO not used as QA

14. SOP Rollout Plan

Week	Action	Owner
W1 (2026-05-12 → 2026-05-16)	Circulate this SOP. Walk PMO Lead + QA Lead + Product PMs through it. Define P0 categories.	Product
W1	Replicate current Excel tracker in Odoo with new column schema (§9.2).	PMO Lead
W1	Backfill Lane field on all open tickets (default Lane A for existing items).	PMs + QA Lead
W2	First QA Lead triage meeting + first PM↔PMO Tuesday sync under new format.	QA Lead + PM Team
W2	First Friday Decisions Day cycle.	PMs
W3	First Lane B item shipped (QA-routed, direct to Dev, Type 1 or 2 rollout).	QA Lead + Dev Lead
W3	First major-feature release under Type 4 cycle with full artifacts.	PMs
W4	First monthly review using metrics in §13.	Product + Engineering Leadership
W6	SOP v1.1 — incorporate first month's learnings.	Product

15. Open Items (To Resolve Before Lock)

- P0 category list** — owned by PMO Lead + Product, due **2026-05-13 EOD**.
- QA Lead triage cadence** — QA Lead picks the Tuesday morning slot for the weekly triage meeting. Confirm by W1 EOD.
- Regression suite ownership** — QA Lead to inventory existing automated tests and identify gaps. Due W2.
- Odoo schema migration** — who builds the replica in Odoo? Owner + ETA needed.
- Prod-like test env timeline** — confirm with Infra team; gate the QA step on this.
- Capacity ratio (50/30/20)** — confirm with Eng Lead this is achievable next sprint.
- Pilot-client volunteer for Type 4 rollouts** — CS to identify a low-risk friendly client per product.
- Tooling for incident channel** — Teams group exists for TMS, needs VT + LPA + cross-product equivalents.

Appendix A – Companion Documents

- **Severity Tier System v1** — the 4-dimension rubric, P0/P1/P2/P3/Stability tiers, classification flow
- **Production Rollout Playbook v1** — the four rollout types (Direct / Notified / Demoed / Piloted) with templates
- **PMO Live Tracker** — 28 items mapped to this schema, tiered per the rubric (see [/live-tracker/](#))

Appendix B – Inspirations / References

- **Atlassian's incident handover model** (P0 RCA discipline)
- **Spotify squad model** (capacity ratio idea)
- **Stripe's "release notes as a product"** (treating notes as a real artifact)

This SOP is a living document. It will be wrong in places. Use it, break it, tell me where it breaks. v1.1 in 4 weeks.

— Aman Product | FleetRobo