

Sessions — Standard Operating Procedure (SOP)

Owner: Matt Evans (PM)

Created: 2026-05-11 (Session 3, Prompt Activity 1)

Location: `C:\Users\mevan\OneDrive\Documents\Claude Project\00a_Sessions_SOP.md`

Purpose: Define how a "session" is conducted on this project, end to end. This is the master orientation document for anyone joining the project — in particular the incoming Tech Support person. Read this first; then read the three other root documents it references.

1. What a "Session" Is

A **session** is a single working engagement between the PM (Matt Evans) and the AI Agent (Claude, running inside Claude Cowork). A session has:

- A **sequential number** (Session 1, Session 2, Session 3, ...) that never resets.
- A **start trigger** typed by the PM: `Let's start session N` (where N is the next sequential number).
- A **rest-state** at close — a written summary of where the work stands when the session ends, so the next session can pick up cleanly.

A session contains **one or more workflows**. A workflow is a stream of related work (e.g., Candidate Vetting, App 8 Spec build, CO Backlog execution). Multiple workflows can run in a single session if the PM directs.

Within each session the PM may pre-plan a set of **Prompt Activities** — discrete asks the PM has written up ahead of time. Example: Session 3, Prompt Activity 1 was "Create a Sessions SOP" — which is what produced this document.

2. Cadence

- **Day of week:** Weekdays (Monday through Friday).
 - **Time:** Typically around 9:00 AM US Eastern Time.
 - **Surface:** Claude Cowork (filesystem-anchored desktop application). Web chat is fallback only; the protocols in this project assume Cowork.
-

3. Roles

3.1 Project Manager (PM) — Matt Evans

- Triggers each session by typing `Let's start session N`.
- Sets the active workstream and the day's objective.
- Identifies hiccups in real time and names new communication rules.
- Sole judge of suspension decisions (see `00c_Communication_Contract.md`).

3.2 AI Agent — Claude (via Claude Cowork)

- Verifies folder access at session start (Rule 10 of the Communication Contract).
- Reads `00c_Communication_Contract.md` and `00d_Session_Log.md` directly from the project folder.
- Confirms contract rules locked, summarizes the prior session's rest-state, and directs the first action.
- Executes the workflow under all active rules.
- Appends the session entry to `00d_Session_Log.md` and updates `00c_Communication_Contract.md` if any new rule was established.

3.3 Tech Support Person (incoming)

- Provides technical insight where the PM lacks it (system, tooling, integration, troubleshooting).
- Attends sessions when notified, even when not part of the immediate workflow — presence increases session efficiency and catches issues the PM may miss.
- Reviews this SOP, the Communication Contract, and the most recent Session Log entry before first session.

4. Pre-Session Prep

4.1 PM responsibilities (before the session)

1. **Notify Tech Support at least 48 hours in advance** of an upcoming session. This applies whether Tech Support is part of the day's workflow or not — the PM encourages attendance for efficiency and technical coverage.
2. **Decide the active workstream and session objective.** One workstream is the default focus per session; multi-workstream sessions are allowed but require the PM to sequence them explicitly.
3. **Pre-plan the agenda (optional but recommended).** The PM writes up one or more Prompt Activities ahead of time to keep the session orderly. Each Prompt Activity is a self-contained ask Claude can execute in sequence.

4.2 Tech Support responsibilities (before the session)

- Confirm receipt of the 48-hour notice.
- Block the session time on calendar.
- Have ready: any tooling, credentials, or environment access likely to be needed for the workflow.

5. Session Start Protocol

The opening sequence is fixed. Steps run in this order, every session, Session 3 onward.

1. **PM types:** `Let's start session N.`
2. **Claude verifies folder access** to `C:\Users\mevan\OneDrive\Documents\Claude Project`. If access is missing, Claude triggers the folder-access prompt before any other action (Rule 10).
3. **Claude reads** `00c_Communication_Contract.md` and `00d_Session_Log.md` directly from the project

folder. No file uploads.

4. **Claude reads** `00e_AI_Agent_Log_Maintenance_SOP.md` and runs the continuity check defined there (Section 6 of 00e): verify canonical file paths still resolve; glob for new artifacts since last session; log any path mismatch or duplicate to `00b_Session_Continuity_Log.md` immediately.
5. **Claude confirms** all active contract rules are locked, summarizes the prior session's rest-state, and directs the first action of the new session.
6. **PM directs** the first workflow or hands off to a Prompt Activity from the pre-planned agenda.

6. During the Session

6.1 Hiccup capture

At any point, the PM may flag a hiccup — a friction event, inefficiency, or miscommunication the PM wants eliminated going forward. The PM may simply name it (e.g., "this is rule 9"), or describe the problem and ask Claude to formalize it. Outcomes:

- If the hiccup points to a **new communication standard**, Claude formalizes it as a new rule and appends it to `00c_Communication_Contract.md`. Rule Change Log gets a new row.
- If the hiccup points to a **continuity gap between sessions** (memory loss, wrong startup prompt, vocabulary mismatch, surface-mismatch), Claude logs it in `00b_Session_Continuity_Log.md` under the appropriate failure mode (A, B, C, or D).

Standing principle: **try not to rework an activity more than once if possible**. Rework loops are friction events and may themselves become rules (Rule 9 already covers this case — no rework loops on substantive PM-supplied data).

6.2 Strikes and suspension

A violation of an **established** rule is a strike. Suspension trigger: 3 strikes in a row, with the PM as sole judge of "in a row" (per `00c_Communication_Contract.md`).

6.3 Compression bias

Per Rule 8, Claude actively looks for ways to compress steps. When a step could plausibly be skipped or merged, Claude flags the skip path alongside the recommended path so the PM can decide in one read.

6.4 PM-Initiated Minor Changes — Outside Change Order Process

Standing practice (established Session 4, Prompt Activity 1, 2026-05-12): Not every change runs through the Change Order process. The PM reserves the right to make minor changes to project assets directly — without filing a Change Order — when routing through CO would slow execution unacceptably.

What may bypass the CO process (PM judgment):

- App-level settings the PM controls in an admin console (e.g., visibility toggles, access scope, plan tier).
- Integration switches and configuration parameters the PM owns directly.
- Minor copy edits to non-deliverable assets.
- Administrative toggles, naming alignment, cosmetic adjustments.

- Other non-functional changes that do not alter scope, requirements, deliverables, or workflow architecture.

What stays in the CO process:

- Substantive functional changes.
- Scope additions or reductions.
- Requirements modifications.
- Anything affecting deliverables, timelines, or workflow architecture.

Logging convention — "PM Notes of Reference": When the PM makes a minor change between sessions, the PM will try to report it in the next session as a "PM Note of Reference." Claude captures the Note in the `Minor_Changes_Log` sheet of `Backlog_Tracker.xlsx` (and mirrors it into the Session Log block at session close). Each Note includes: what changed, where (URL / file path / App ID / asset identifier), the date of the change, who changed it, and the reason. The PM is not bound to report every minor change in real time — the discipline is best-effort circle-back, not a hard log requirement.

Rationale: Forcing every minor administrative change through the Change Order process bogs down execution, which the PM judges unacceptable. The CO machinery is designed for substantive work product changes, not configuration housekeeping. This SOP item protects execution velocity while preserving an audit trail through the session log.

PM is sole judge of what counts as "minor." If Claude believes a PM-named "minor" change is actually substantive, Claude flags the concern once; the PM's call is final.

6.5 PM-Touch-Zero on Logs — AI Agent Owns Log Maintenance

Standing principle (named Session 4, Prompt Activity 3, 2026-05-12): The PM operates against exactly one document — `C:\Users\mevan\OneDrive\Documents\Claude Project\09_Admin_Submissions\Backlog\Super_Agent_Prompts.docx` — his **Single Point of Execution**. Every other artifact in the project (logs, trackers, communication records, scorecards, frameworks, SOPs, continuity log, session log, candidate dashboards) is **maintained by the AI agent**, not the PM.

The PM does not edit, audit, or routinely read project logs. The AI agent is responsible for:

- Catching its own errors via continuity checks at session start (no PM safety net).
- Surfacing conflicts proactively in chat when duplicate or ambiguous artifacts are detected (no silent picks).
- Translating log updates into chat summaries when the PM needs to know — logs alone do not communicate.
- Cleaning up stale or duplicate artifacts at session close (no PM cleanup pass).

Operational SOP for AI agents: `C:\Users\mevan\OneDrive\Documents\Claude Project\00e_AI_Agent_Log_Maintenance_SOP.md`. The AI Agent SOP defines the canonical-file registry, the log inventory, update triggers, the session-start continuity check, the reconciliation rule for duplicates, and the cleanup discipline. AI agents read it at every session start, immediately after `00c_Communication_Contract.md` and `00d_Session_Log.md`.

Why this was named: Session 4 surfaced two backlogs at different paths (the active `Backlog_Tracker.xlsx` and a stale `Backlog.xlsx`). The Session 3 prep document had pointed Session 4 Claude at the stale file. PM does not read the prep doc, did not catch the error, and reasonably observed that AI agents are maintaining logs the PM never touches — and continuity between sessions on those logs is unreliable. The AI Agent SOP exists to eliminate this class of failure.

6.6 Base44 Trouble Ticket Follow-Up

Standing practice (added Session 5, Prompt Activity 1, 2026-05-14): Base44.app trouble tickets have twice been auto-closed by base44 support after roughly two weeks because the PM did not follow up — the PM is not aware of the procedure for responding to and closing a trouble ticket, and that process is currently broken on this project.

Standing mechanism. Whenever a base44.app trouble ticket is opened, or surfaces in a session, Claude immediately creates a row in the `Action_Items` sheet of `Backlog_Tracker.xlsx` assigned to the PM (Owner: Matt) — the task is to monitor, respond to, and drive the ticket to a proper close-out before base44 auto-closes it. Claude surfaces the task in chat at the same time it logs it (logs do not communicate — Section 6.5).

Open gap. The actual base44 follow-up and communication procedure for trouble tickets is not yet documented. The next time a ticket is live, Claude captures the procedure — where the PM responds, what base44 expects, the response window — so the `Action_Items` task carries concrete steps instead of a generic reminder.

6.7 NFR-PM Role — AI Agent as Project Manager for Non-Functional Requirements

Standing practice (added Session 5, Prompt Activity 5, 2026-05-14): On top of normal session work, the AI agent acts as the project manager for the project's non-functional requirements — privacy, security, beta consent, and data handling. The PM is kept out of technical detail unless an actual decision is required.

Standing duties:

- **Trigger-check every product change.** For any product change, the AI agent checks whether it touches data collection, storage, access, privacy-notice language, beta consent, deletion, security, support, vendors, analytics, or Base44 deployment. If it does, the agent updates the Risk Register and writes a plain-English change request.
- **Maintain the Risk Register** — `C:\Users\mevan\OneDrive\Documents\Claude Project\11_Privacy_NFR_workstream\07-Financial Forecasting App - Risk Register and Compliance Tracker.xlsx`, three sheets (Risk Register, Beta Readiness Checklist, Data Map), kept current as decisions are made.
- **Apply the default risk-reduction rule** — collect less data, store less centrally, avoid highly sensitive identifiers (tax ID, SSN, bank credentials), make beta consent explicit, give users clear export/deletion options.
- **Route change requests through the CO process** — when a change triggers, it becomes a CO or sub-action in the Backlog Tracker, flagged as NFR/privacy. One pipeline, not a parallel track.

Reference materials: the PM Privacy/Security Roadmap and the AI-agent operating instruction, both in `C:\Users\mevan\OneDrive\Documents\Claude Project\11_Privacy_NFR_workstream\`.

Why this was named: Session 5 surfaced that privacy / security / beta-compliance work is a standing workstream, not a one-off task. CO-031 was repurposed into a privacy-first beta-readiness CO (decision D-040) and blocked pending legal review (Triumph Law); the Risk Register is the live holding framework while that CO is blocked.

7. Session Close Protocol

At the end of every session, Claude performs these steps and saves directly to the project folder (Rule 10 — no PM copy step required):

1. **Append a new session block** to the top of `00d_session_Log.md` (newest first). The block contains:
 - Rest-state at session close (the orientation block for the next session)
 - Established-rule violations this session (count and table)
 - What got done
 - Files created or updated
 - Contract / rules changes
 - Workflows touched
 - PM directives at session start
 2. **Update** `00c_Communication_Contract.md` if any new rule was established. Append the rule to Active Rules and add a row to the Rule Change Log.
 3. **Update** `00b_session_Continuity_Log.md` if any continuity-loss event occurred that fits one of the four failure modes.
 4. **Update the PM Story** (`C:\Users\mevan\OneDrive\Documents\Claude Project\Vena_PM_Story\`) when the session produced lessons worth capturing. Standing practice (added Session 3, Prompt Activity 3). Default behavior:
 - Append distilled lessons to `03_Lessons_by_Theme.md` under the appropriate theme. Each lesson includes a distilled statement, an anchor in the project, and `[PM VOICE]` prompts for later reflection.
 - Append narrative material to `01_The_story.md` if a session produced a material milestone or texture worth preserving.
 - Append pivots to `02_Decision_Moments.md` if a key decision occurred.
 - Periodic (not every session): add dual-register reflection to `04_For_Future_Me_and_Other_PMs.md`.
 - Some sessions will produce little or no PM Story material — that is fine. The discipline is to ask "did anything land here that future-me or another PM would learn from?" and only write when the answer is yes.
 5. **Update the header counts** at the top of the Session Log (latest session, total sessions logged, cumulative strikes last 3 sessions).
-

8. Reference Documents (Project Root)

All four reference documents live at the project root: `C:\Users\mevan\OneDrive\Documents\Claude Project\`.

8.1 `00b_Session_Continuity_Log.md` — Session Continuity Log

Tracks every instance where session-to-session continuity loss causes friction, confusion, or rework. Each entry is classified into one of four failure modes:

- **Mode A — Structural memory loss** (NOT eliminable on web chat; mitigated by Cowork or Project Knowledge).
- **Mode B — Wrong-protocol routing** (ELIMINABLE; fix is one labeled startup prompt per workstream).
- **Mode C — Vocabulary / documentation gap** (ELIMINABLE; fix is a glossary or inline definitions).
- **Mode D — Surface-mismatch assumption** (ELIMINABLE on user side; fix is a surface tag on every protocol).

When used: Claude logs an entry here whenever a continuity-loss event occurs in a session that goes beyond the expected upload protocol. Reviewed monthly or every 10 entries, whichever comes first. Diagnostic patterns and process changes are recorded in dedicated sections of the file.

8.2 `00c_Communication_Contract.md` — Communication Contract

The standing communication rules between the PM and Claude. Currently 10 rules in force (Rules 1 through 10). Governs **how** Claude communicates, independent of what the work is.

Key rules to know at a glance:

- Rule 1 — Complete file paths in every action request.
- Rule 2 — No options menu; Claude picks the workflow-aligned path and directs.
- Rule 3 — Every Claude response ends with an explicit "Next step for you:" line.
- Rule 4 — Literal UI language that matches what the PM actually clicks or types.
- Rule 5 — Plain English only; no jargon or consulting-speak.
- Rule 6 — Drafts reviewed inline in chat, not by opening a file.
- Rule 7 — Paste-ready text in fenced code blocks for one-click copy.
- Rule 8 — Always compress; flag skip paths alongside recommended paths.
- Rule 9 — No rework loops on data the PM has already supplied.
- Rule 10 — Folder access established at session start, before any workflow.

When used: Read at the start of every session (step 3 of the Session Start Protocol). Appended whenever a new rule is established mid-session.

8.3 00d_Session_Log.md — Session Log

Day-to-day continuity. Replaces the multi-file orientation Claude used to do at session start. Newest session entry at the top. Each entry contains the rest-state at close (read this first for the next session), violations, what got done, files touched, rules changes, workflows, and PM directives.

When used: Read at the start of every session (step 3 of the Session Start Protocol). Appended at the end of every session.

8.4 00e_AI_Agent_Log_Maintenance_SOP.md — AI Agent Log Maintenance SOP

SOP for AI agents (not for the PM) defining how project logs are maintained between sessions. Contains: the PM-touch-zero principle, the single point of execution (Super_Agent_Prompts.docx), the canonical file registry (single authoritative path per artifact), the log inventory and update triggers, the session-start continuity check, the reconciliation rule for duplicate/ambiguous artifacts, the versioning discipline for the Backlog Tracker, and the cleanup discipline.

When used: Read at the start of every session (step 4 of the Session Start Protocol). Appended whenever a new artifact, log, or pattern emerges that affects the canonical registry or maintenance discipline.

8.5 00f_Change_Order_Submission_SOP.md — Change Order Submission SOP

PM-facing SOP defining the interface for submitting requirements as Change Orders. Four-step model (name → AI drafts inline → PM responds submit/edits/skip → AI commits). Lists what the PM provides (required vs. optional), what happens after submission (Super Agent loop), Status value definitions, and special cases (privacy show-stoppers, conceptual requirements, big-picture UI/UX framing, minor PM-initiated changes that bypass the CO process).

When used: Read by the PM when he has a requirement to submit and needs a refresher on what to provide. Read by AI agents when drafting new CO entries to ensure the submission protocol is honored.

8.6 00d_Session_Startup_Prompt.md — Session Startup Prompt (legacy reference)

Originally a paste-in opening prompt for web-chat sessions. Largely superseded by Rule 10 (folder access) — Cowork sessions no longer require file attachments because Claude reads the project root directly. Retained for reference and for any fallback web-chat session.

Note: This file currently references "5 rules" and a file-attach workflow that predates Rules 6-10 and the folder-access protocol. Update is a candidate cleanup item when the PM elects.

9. Glossary

Term	Definition
Session	A single working engagement triggered by <code>Let's start session N.</code>

Term	Definition
Workflow	A stream of related work executed within a session (e.g., Candidate Vetting).
Workstream	Synonym for workflow at the project-portfolio level (e.g., App 8 Spec build, Candidate Vetting, CO Backlog, PM Story, Salvage Tracker).
Prompt Activity	A self-contained, pre-written ask in the PM's agenda for a session.
Hiccup	A friction event, inefficiency, or miscommunication the PM flags for elimination.
Rule	A formalized communication standard captured in 00c_Communication_Contract.md.
Strike	A violation of an established rule. Three in a row triggers suspension review.
Rest-state	The written summary at session close describing where the work stands for the next session.
Toll Gate	A project-level milestone gating subsequent phases (currently three identified).
Surface	The environment where a session runs — Cowork (filesystem-anchored desktop) or web chat.

10. Quick Reference — Where Things Live

Item	Path
Project root	C:\Users\mevan\OneDrive\Documents\Claude Project\
Sessions SOP (this file)	C:\Users\mevan\OneDrive\Documents\Claude Project\00a_Sessions_SOP.md
Session Continuity Log	C:\Users\mevan\OneDrive\Documents\Claude Project\00b_Session_Continuity_Log.md
Communication Contract	C:\Users\mevan\OneDrive\Documents\Claude Project\00c_Communication_Contract.md
Session Log	C:\Users\mevan\OneDrive\Documents\Claude Project\00d_Session_Log.md
AI Agent Log Maintenance SOP	C:\Users\mevan\OneDrive\Documents\Claude Project\00e_AI_Agent_Log_Maintenance_SOP.md
Change Order Submission SOP	C:\Users\mevan\OneDrive\Documents\Claude Project\00f_Change_Order_Submission_SOP.md
PM Single Point of Execution	C:\Users\mevan\OneDrive\Documents\Claude Project\09_Admin_Submissions_Backlog\Super_Agent_Prompts.docx
Active Backlog Tracker	C:\Users\mevan\OneDrive\Documents\Claude Project\09_Admin_Submissions_Backlog\Backlog_Tracker.xlsx

Item	Path
Session Startup Prompt (legacy)	C:\Users\mevan\OneDrive\Documents\Claude Project\00d_Session_Startup_Prompt.md

11. Onboarding Checklist — New Tech Support Person

1. Read this SOP top to bottom.
2. Read `00c_Communication_Contract.md` — all 10 active rules.
3. Read the most recent session block in `00d_Session_Log.md` (newest at the top).
4. Skim `00b_Session_Continuity_Log.md` to understand the failure-mode taxonomy.
5. Confirm with the PM the date and time of the first session you will attend.
6. Block the session time on calendar; confirm 48 hours ahead.
7. Arrive on time, observe, ask questions in the moment when a technical issue arises, and flag any inefficiency you spot — the PM treats those as hiccups and may convert them into rules.

12. Maintenance

- This SOP is updated when sessions evolve in structure (new roles, new artifacts, new cadence, new protocols).
- Updates are appended in place; the file is saved directly to project root.
- Material changes are noted in `00d_Session_Log.md` for the session in which they occurred.