AIBOS Uganda: System-Driven Delivery Framework

A Comprehensive Plan for Quality, Communication, and Operational Excellence

> “From a person-dependent organization to a system-driven organization.”

Table of Contents

Section 0 — Executive Summary

Purpose

This document is the operational blueprint for AIBOS Uganda's transformation from a person-dependent delivery model into a system-driven organization. It consolidates all agreed frameworks, processes, standards, and tools into a single reference that any team member — current or new — can use to understand how we work, what is expected, and how to contribute effectively.

The Core Transformation Goal

From	To
Quality depends on specific individuals	Quality is enforced by the system
Communication happens when someone remembers	Communication follows a defined protocol
Decisions live in people's heads	Decisions are documented and accessible
Managers spend time firefighting	Managers monitor, guide, and improve the system
Knowledge is lost when someone is unavailable	Knowledge lives in documentation and tools

What This Document Covers

This framework addresses seven interconnected pillars:

1. **Standards** — SLAs, KPIs, and measurable expectations
2. **Operations** — Daily routines, meetings, and reporting
3. **Quality** — QA layers, ownership, release governance, and priority framework
4. **Communication** — Client and internal communication protocols
5. **Knowledge** — Documentation, Google Drive + docs/ repository, and internal vs. shared knowledge boundaries
6. **People** — Capacity building, skills mapping, hiring, and recognition
7. **Technology** — AI tools and shared team environments for sustainable scale

How to Use This Document

- **New team members:** Read Sections 1–5 first for full context, then refer to specific sections as needed
- **Product Managers:** Sections 5, 6, 7, 8, and 9 are your primary operational reference
- **Developers:** Sections 6, 7, 11.3, and 13 are most relevant to your daily work
- **Leadership:** Sections 2, 3, 4, 14, and 15 provide governance and strategic overview
- **All team members:** Section 12 (Reward System) and Section 10.7 (Talent) apply to everyone's growth and day-to-day experience

Section 1 — Background & Context

1.1 The Challenge: Client Feedback

AIBOS Uganda received structured feedback from client DATAGRID that highlighted recurring process gaps. These gaps are not unique to DATAGRID — they reflect systemic issues across all AIBOS Uganda operations, including internal projects. AIBOS Japan is also considered a client of AIBOS Uganda.

Quality Issues Identified:

- Deliverables submitted at a quality level that would not pass Japanese review standards
- Bugs that would be caught by basic testing are reaching the client
- Over-engineering beyond MVP scope — building features the client did not ask for

Communication Issues Identified:

- Tasks started based on misunderstood requirements
- Slow responses to client questions
- Deadlines not confirmed before work begins
- Task delays communicated on the due date rather than in advance
- Previously identified mistakes being repeated

1.2 The Root Cause: Person-Dependency

The underlying cause of these issues is not a lack of skill or knowledge. The root cause is structural:

> **The organization currently relies too heavily on specific individuals to maintain quality and communication standards. When those individuals are busy, unavailable, or overloaded, standards drop.**

This creates two problems:

- Talented individuals carry an unsustainable personal burden
- The organization cannot scale because performance is tied to people, not systems

1.3 The Goal

```
Current State                          Target State
──────────────────────────────────────────────────
Talented individual carries quality    System enforces quality
Manager fixes every problem            Process prevents problems
Knowledge lives in one person          Knowledge lives in documentation
Communication depends on memory        Communication follows protocol
──────────────────────────────────────────────────
```

1.4 What This Is and Is Not

This IS	This is NOT
A process and systems initiative	Punishment or blame for any individual
A collective effort across all levels	A performance review mechanism
An opportunity to build better systems	A criticism of individual capability
A foundation for AI-enabled growth	A set of rules imposed from the top down

1.5 Guiding Mindset

Principle	Meaning in Practice
Mistakes are opportunities	Every problem found is a chance to build a system that prevents it from recurring
DRY (Don’t Repeat Yourself)	Once a mistake happens, we build a system so it never happens again
Team over individual	“Assignee” means “Owner” — the owner’s job includes asking the team for help
Standards over individuals	If a process works for one person, document it so it works for everyone
Japan-Uganda as one team	Different contexts, same goal — building a company of global standard

Section 2 — Organizational Structure & Roles

2.1 Responsibility Model

There are two types of responsibility in every activity:

Role	Description	Example
Management Owner	Bears ultimate responsibility. Assigns and oversees Execution Owners. Responsible for the system succeeding.	A manager who owns the meeting process — even if someone else facilitates
Execution Owner	Fulfills responsibility within their defined scope. Accountable for their piece of delivery.	A PM who facilitates the standup as the Execution Owner of that meeting

Key principle: The Execution Owner is responsible for doing the work. The Management Owner is responsible for making sure the work gets done — including asking for help, removing blockers, and ensuring quality.

2.2 Initiative Ownership Structure

Role	Person	Scope
Management Owner	Ambrose	Overall ownership of SLA/KPI improvement initiative
Execution Owners	Martin, Joseph, and Project PMs	Implementation and day-to-day execution
Advisor — Development	Prakhar (Head of Development, AIBOS Japan)	Guidance on development process and rule changes
Advisor — Operations	Nate (Head of AIBOS Uganda)	Guidance on Uganda structure and communication matters

2.3 Japan-Uganda Leadership Responsibilities

Leader	Ultimate Responsibility
Prakhar	Bridging the client and development team; approving development process changes
Nate	Overall Uganda structure and operational functioning
Ambrose	Day-to-day management of the SLA/KPI system and delivery standards

2.4 Communication Channels

```
Type of Communication              Channel / Person
──────────────────────────────────────────────────
General questions and discussions  B1 Channel (visible to all)
Communication-related matters      Consult Nate
Development process / rule changes Consult Prakhar
Project-level daily updates        Daily Reports Channel
Client communication               Per project PM, following protocol
──────────────────────────────────────────────────
```

> **Note:** The B1 channel is the default for transparency. Direct messages should be used only for sensitive matters — everything operational should be visible to the team.

Section 3 — SLA Framework

3.1 Overview

Service Level Agreements (SLAs) define the minimum standards of behavior expected from every team member. They are not guidelines — they are commitments. Meeting SLAs consistently is what builds trust with clients and within the team.

3.2 SLA Standards

Area	Standard	Why It Matters
Client Response Time	First response within **15 minutes** during working hours	Eliminates client anxiety about being ignored; builds trust
Deadline Management	Escalate delay risks **immediately upon recognition** — never on the due date	Gives the client and team time to adjust; prevents surprise failures
Requirements & Scope	Confirm understanding **in writing** before starting work; MVP scope first	Eliminates work based on assumptions; prevents over-engineering
Quality Assurance	Two-stage check: **developer self-test → QA Owner verification** before any client delivery	Stops preventable bugs from reaching the client
Meeting Management	**Pre-assign roles**; share decisions and next actions **same day**	Ensures meetings produce outcomes, not just discussions

3.3 SLA Monitoring

- SLA adherence is observed through the daily KPI log (see Section 4)
- Violations are not punitive — they are data points for system improvement
- Recurring violations in the same area trigger a process review

Section 4 — KPI Framework

4.1 KPI Definitions

KPI	What It Measures	Good Result
15-min response violations	Number of times a client message went unacknowledged for more than 15 minutes	0 per week
Delay escalation rate	Percentage of delay risks shared proactively (before the due date)	100%
Pre-delivery bugs found in QA	Number of bugs caught internally before reaching the client	Track trend downward
Post-delivery bugs reported by client	Number of bugs the client found after delivery	0 target
On-time delivery rate	Percentage of tasks delivered on or before the agreed deadline	100% target

4.2 Collection & Review Cadence

KPI	Collected	Reviewed
15-min response violations	Daily (PM logs per project)	Weekly (PDMO meeting)
Delay escalation rate	Daily (PM logs per project)	Weekly (PDMO meeting)
Pre-delivery bugs found in QA	Daily (QA Owner logs)	Weekly (PDMO meeting)
Post-delivery bugs (client reported)	Daily (PM logs)	Weekly (PDMO meeting)
On-time delivery rate	Weekly (PM confirms)	Weekly (PDMO meeting)

4.3 KPI Dashboard

Tool: Google Sheets (stored in the team Google Drive under Operations/KPI Tracking/)

Structure — One Sheet per Project, One Summary Tab:

KPI	Mon	Tue	Wed	Thu	Fri	Weekly Total	Target
15-min response violations							0
Delay risks escalated proactively							100%
Pre-delivery bugs found in QA							Log & trend
Post-delivery bugs (client)							0
On-time delivery							100%

Rules:

- Each PM fills their project's row during the **evening retrospective**

- Ambrose reviews the full summary tab every **Tuesday** in the PDMO meeting
- Japan team (Prakhar/Nate) has view access at all times
- An empty cell by EOD is flagged as a potential gap

Summary Tab Format:

Project	Response SLA	Delay Escalation	Pre-delivery Bugs	Client Bugs	On-time Rate	Overall
Create Project						ON TRACK / WATCH / BLOCKED
EMS						
FOCUST						
AI Phone Agent						

Section 5 — Daily Operational System (The Daily Pulse)

5.1 Overview

The Daily Pulse is the operational heartbeat of AIBOS Uganda. It ensures every working day begins with alignment and ends with validated progress. It is structured as two loops — a morning setup loop and an evening validation loop.

5.2 The Daily Pulse Flow

```
MORNING LOOP
──────────────────────────────────────────────────────────────
09:00 | COMPANY STANDUP (6–10 min)
      | All teams. High-level alignment on the day.
      | Focus: priorities, blockers, team spirit
      ▼
09:10 | TEAM PREPARATION MEETING (10 min)
      | Each team internally. Led by the PM.
      | Focus: technical alignment, task clarity, who owns what
      ▼
09:20 | PM → MANAGEMENT OWNER SYNC
      | PMs report urgent blockers to Ambrose.
      | Focus: anything that needs management attention before the day runs
      | Output: Ambrose aware of all critical risks before 09:30
──────────────────────────────────────────────────────────────

WORK DAY
──────────────────────────────────────────────────────────────
      | Teams execute. PMs monitor. Blockers escalated immediately.
      | Client responses follow the 15-min protocol (Section 7).
──────────────────────────────────────────────────────────────

EVENING LOOP
──────────────────────────────────────────────────────────────
16:30 | TEAM PEER REVIEWS (30 min)
      | Led by PM. All team members.
      | Focus: review staging environment, cross-check each other's work
      | QA Owner runs smoke test checklist
      ▼
17:00 | STATE OF PRODUCT CONFIRMATION
      | PM confirms product status for the day
      | Focus: what is done, what is at risk, what needs escalation
      ▼
17:15 | MANAGEMENT "BIG 3" MEETING
      | Ambrose, Martin, Joseph
      | Focus: company-wide daily status, unresolved blockers, decisions
      ▼
17:30 | EOD REPORT TO CHANNEL
      | PM posts standardized daily report
      | Audience: Client + Internal Management
──────────────────────────────────────────────────────────────
```

5.3 Morning Standup Guidelines

- Maximum 10 minutes — no problem-solving in the standup
- Each person answers: What did I do? What will I do today? Any blockers?
- Blockers are noted and addressed in the PM sync that follows

5.4 Team Preparation Meeting Guidelines

- Maximum 10 minutes
- Led by the Project PM

- Technical focus: task ownership, dependencies, environment status
- Output: every team member knows their exact task for the day

5.5 PM → Management Owner Sync

- Brief and focused — only urgent items
- PMs report: anything that could affect client delivery, deadlines, or team capacity
- Ambrose makes immediate decisions on anything critical
- Non-urgent items go to the PDMO weekly meeting

Section 6 — Quality Assurance System

6.1 Philosophy

Quality is not an individual's responsibility — it is the system's output. When a bug reaches the client, the question is not "who wrote the bug?" but "which layer of the QA system failed to catch it?"

The goal is to make it structurally difficult for poor quality to reach the client.

6.2 The 4-Layer QA Filter

If any layer fails, the item goes back to the developer — it does not move forward.

6.3 QA Ownership

Role Definition — QA Owner:

Responsibility	Description
Pre-delivery check	Reviews every deliverable before it goes to the client using the Definition of Done checklist
Bug gate	Anything not ready goes back to the developer — no exceptions
KPI logging	Logs pre-delivery bugs found in the daily KPI sheet
Smoke test lead	Runs the standard checklist after every deployment to staging
Escalation	If a deliverable has critical gaps close to deadline, escalates to PM immediately

Named QA Owners per Project:

Project	QA Owner	Backup QA Owner	Rotation (Every)
Create Project (BP)	To be nominated by team	To be nominated by team	2–3 months
EMS	To be nominated by team	To be nominated by team	2–3 months
FOCUST	To be nominated by team	To be nominated by team	2–3 months
AI Phone Agent	To be nominated by team	To be nominated by team	2–3 months

> **Process:** Each team nominates their QA Owner by **March 1** (aligned with Phase 1 of the Implementation Roadmap). Criteria: not the person who wrote the feature being tested; understands the product well; has time allocated for QA duties alongside delivery work.

> **QA Owner Roles:** Runs the smoke test checklist on every staging deployment; signs off on the DoD checklist; logs pre-delivery bugs in the KPI sheet; escalates to PM when quality is at risk before a deadline. The QA Owner is the team's last line of defence before the client sees any work.

> **See Appendix H for the QA Owner Appointment Form.**

Rotation Policy:

- QA Owner rotates every 2–3 months
- This builds QA competency across the whole team
- The outgoing QA Owner briefs the incoming one before handover

6.4 Definition of Done (DoD) Checklist

Before any feature or task can be marked complete, the following must all be true:

```
DEFINITION OF DONE — CHECKLIST
──────────────────────────────────────────────────
☐  Requirements confirmed in writing before work started
☐  Developer has self-tested the feature
☐  Code has been peer-reviewed
☐  Feature works as expected in the staging environment
☐  No obvious bugs or broken flows in staging
☐  QA Owner has reviewed and signed off
☐  Documentation updated if required (in /docs)
☐  Client notified if delivery is imminent
──────────────────────────────────────────────────
```

A Pull Request cannot be merged to staging unless the DoD checklist is confirmed.

6.5 Two-Environment Standard

Environment	Purpose	Who Accesses It
Development (Dev)	Active feature development and testing	Developers only
Staging	Client acceptance testing; PM and QA validation	PM, QA Owner, Client (where applicable)
Production	Live, stable, client-facing system	Client and end users only

Rule: No feature goes to Production without passing through Staging and receiving confirmation.

6.6 Release Governance

6.7 Project Priority Framework

Managing quality requires knowing what to protect most. Not all features carry the same risk when things change — this framework categorizes features by priority and ensures that any change is assessed for impact before it is applied.

PRIORITY LEVELS

Priority	Label	Definition	Stability Rule
P0 — Critical	Core / Always On	Features that must work at all times. Failure here is unacceptable to the client.	No changes without full QA sign-off and a documented rollback plan.
P1 — High	Core Workflow	Features essential to the main user journey. Bugs here are high-urgency.	Changes require staging validation before any production deployment.
P2 — Standard	Supportive	Secondary features or enhancements. Bugs are tolerable temporarily.	Standard QA process applies.
P3 — Low	Nice-to-Have	Minor features or cosmetic improvements.	Can be deferred without client impact.

> **Rule:** P0 features are never modified without a documented risk assessment and rollback plan. If a P0 feature is at risk of being broken by a change, that change must wait for a safe deployment window.

RISK ASSESSMENT REQUIREMENT

Before any change to a P0 or P1 feature, the PM and QA Owner must complete a brief risk assessment:

```
CHANGE RISK ASSESSMENT
──────────────────────────────────────────────────
Feature Being Changed:              [Name / Module]
Priority Level:                     P0 / P1 / P2 / P3
What is changing:                   [Brief description]
Potential impact on other features: [List affected areas]
Can it be reverted if it fails:     Yes / No
Rollback steps if needed:           [Step-by-step rollback]
Who approves:                       [PM + QA Owner]
Client informed?                    Yes / No / Not required
──────────────────────────────────────────────────
```

ROLLBACK PROTOCOL

Every P0 and P1 deployment must have a documented rollback plan before it is merged to Production.

```
Deployment goes to Production
        │
        ▼
┌──────────────────────────────────────────────┐
│ Smoke test passes within 30 minutes?         │
└──────────────────────────────────────────────┘
        │
    Fails│
        ▼
Immediately execute rollback:
  1. Revert to last stable Production deployment
  2. Notify PM and QA Owner
  3. PM notifies client with brief honest message
  4. Root cause identified before next attempt
```

CLIENT COMMUNICATION ON PRIORITY CHANGES

Any update that affects P0 or P1 features must be communicated to the client **before** deployment, not after.

Template — Planned Change Notification:

> “Hi [Client Name], we are planning to deploy an update to [Feature/Module] on [Date]. This addresses [brief description]. Internal testing is complete and a rollback plan is in place. If you have any concerns, please let us know before [Time]. We will confirm the outcome in our daily report.”

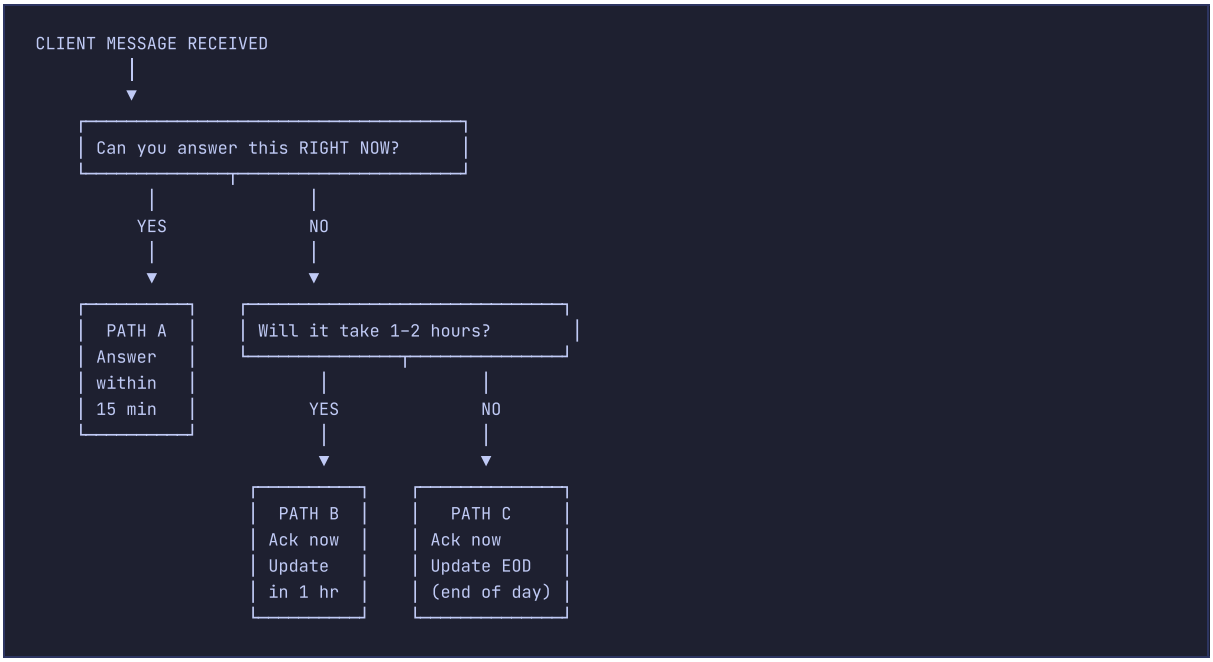
> **See Appendix I for the full Risk Assessment & Rollback Plan template.**

Section 7 — Communication Protocol

7.1 The 15-Minute Response Rule

Every client message received during working hours must receive a first response within **15 minutes**. This does not mean the issue must be solved in 15 minutes — it means the client must know their message has been received and is being handled.

7.2 The Three-Path Decision Tree



7.3 Response Templates (Ready to Use)

Path A — Immediate Response: Direct answer to the client's question, accurate and concise.

Path B — Investigation Acknowledgment:

> "Hi [Client Name], thank you for your message regarding [Topic]. I have flagged this with the team and we are currently reviewing the details. I will get back to you with a status update within the hour. Best regards, [Name]"

Path C — Long-Term Acknowledgment:

> "Hi [Client Name], thank you for reaching out about [Topic]. This requires a deeper review of [Area/Module]. We are looking into it and will provide a full status update in our end-of-day report, or sooner if resolved. Best regards, [Name]"

1-Hour Update (for Path B):

> "Hi [Client Name], a quick update: we have identified that the issue is related to [Area]. We are currently [testing/refining/investigating] the solution. I will confirm the final outcome in our daily report. Best regards, [Name]"

7.4 The 1-Hour Pulse Rule

If a Path B acknowledgment was sent, the Execution Owner must:

- Set a timer after sending the acknowledgment
- Provide a status update within 1 hour — even if the issue is not yet resolved
- Never let a client question go into the next day without a summary in the EOD report

7.5 Risk Escalation Process

```
Risk or delay identified by Developer/PM
         |
         ▼
Immediately notify PM (if not already the PM)
         |
         ▼
PM assesses impact on deadline
         |
         ▼
PM notifies Ambrose (Management Owner) at once
         |
         ▼
Ambrose decides:
  ├── Can the team handle it?  →  Team adjusts, client informed proactively
  └── Needs escalation?        →  Consult Prakhar/Nate same day
         |
         ▼
Client notified with:
  - What the risk is
  - Current impact on timeline
  - What the team is doing to address it
  - When the next update will come
```

> **Rule:** A delay communicated proactively is manageable. A delay communicated on the due date is a failure of the system.

7.6 What "Good Communication" Looks Like

Scenario	Poor Practice	Good Practice
Requirements unclear	Start work, assume	Stop. Confirm in writing before starting
Task will be delayed	Say nothing until due date	Escalate as soon as the risk is recognized
Client asks a technical question	Wait until fully resolved	Acknowledge in 15 min, update in 1 hour
Bug found after delivery	Hope client doesn’t notice	Notify client immediately with resolution plan
Meeting decision made	Forget by next day	Post decisions and next actions in channel same day

Section 8 — Reporting System

8.1 Daily Progress Report

The daily progress report serves **two audiences simultaneously** — the client and internal management. One message, consistent format, sent to the relevant channel at end of day.

Template:

```
Hello Team / Hello [Client Name],

Please find below the daily progress update for [Day, Date].

_____________________________________________________
1. PREVIOUS DAY PROGRESS
_____________________________________________________
- [Completed task or achievement]
- [Another completed item]
- [Any improvement or update made]

_____________________________________________________
2. PLANNED ACTIVITIES (NEXT)
_____________________________________________________
- [Next task / upcoming work]
- [Key priorities for the next period]

_____________________________________________________
3. RISKS / BLOCKERS (if any)
_____________________________________________________
[ ] None at this time
OR
- [Brief description of risk or blocker and current mitigation]

_____________________________________________________
4. SUPPORT NEEDED (if any)
_____________________________________________________
[ ] None
OR
- [Specify what input, approval, or assistance is required]

_____________________________________________________

Best regards,
[Name]
[Role / Team]
```

Submission Rules:

- Sent to the Daily Reports channel by **17:30 every working day**
- CC'd to the client where the project involves direct client communication
- PM is responsible for submission — QA Owner confirms accuracy of section 3

8.2 Weekly PDMO Summary

Produced by Ambrose after each Tuesday PDMO meeting. Shared in the team channel.

Structure:

```
WEEKLY PRODUCT MANAGEMENT UPDATE

Context
──────────────────────────────────────────
[2–3 sentence summary of what this week's
discussions covered and what it reflects.]

Project Health Overview
──────────────────────────────────────────
Project           | Status
──────────────────┼──────────────
Create Project    | ON TRACK / WATCH / BLOCKED
EMS               | ON TRACK / WATCH / BLOCKED
FOCUST            | ON TRACK / WATCH / BLOCKED
AI Phone Agent    | ON TRACK / WATCH / BLOCKED

ON TRACK = performing well   WATCH = minor risks   BLOCKED = needs attention

Project Summaries
──────────────────────────────────────────
[One short paragraph per project covering:
progress, current focus, risks, next steps]

Cross-Team Discussions & Key Decisions
──────────────────────────────────────────
[Any shared learnings, decisions, or
action items from the PDMO discussion]

Closing Note
──────────────────────────────────────────
[Brief encouragement and next steps for the team]
```

8.3 Client Reporting Guidelines

Principle	Detail
Frequency	Daily (via EOD report)
Tone	Professional, honest, confident
On delays	Proactive — notify the moment risk is recognized
On bugs	Transparent — report immediately with resolution plan
On progress	Specific — avoid vague “in progress” without detail
Language	Clear and simple — avoid heavy technical jargon unless the client is technical

Section 9 — Meeting Structure & Schedule

9.1 Overview of the Meeting Ecosystem

```
DAILY                WEEKLY               PERIODIC
─────────────────────────────────────────────────────
Company Standup      Product Meeting      TSS (Technical
(Mon–Fri, 09:00)     (per team, 15 min    Sharing Session)
                     after weekly         (Weekly, Friday)
Team Prep Meeting    team meeting)
(Mon–Fri, 09:10)                          Exploratory
                     PDMO Meeting         Cross-Team Testing
PM–Mgmt Sync         (Tuesday 14:00 EAT)  (During PDMO)
(09:20, as needed)
                     Weekly Tech Check
Peer Review          (per project)
(16:30 daily)

Management
(17:15 daily)
─────────────────────────────────────────────────────
```

9.2 Company Standup

Item	Detail
Duration	6–10 minutes
Frequency	Daily, Monday to Friday
Purpose	Company-wide alignment on daily priorities
Format	Each person: What did I do? What will I do today? Any blockers?
Rule	No problem-solving in the standup — flag and address separately

9.3 Team-Level Product Meetings (15 Minutes)

These sessions happen immediately after each team's regular weekly meeting. They are the foundation of the product management rhythm.

Purpose:

- Align on product priorities for the week
- Complete the weekly product checklist
- Surface blockers, bottlenecks, and missed requirements early

Attendees:

- All team members (mandatory)
- Project Manager (leads)
- Product Managers from other teams (optional — for cross-team learning)

Confirmed Schedule:

Team	Meeting Day	Time
SNS	Monday	After 14:00 team meeting
AI Phone Agent	Tuesday	After 10:30 team meeting
Create Project	Wednesday	After 12:00 team meeting
EMS	Thursday	After 11:00 team meeting
FOCUST	Friday	After 10:00 team meeting

> **Early validation:** The Create Project team's pilot session identified workflow bottlenecks and missed requirements, which were assigned and actioned immediately — directly demonstrating the value of this format.

9.4 Weekly PDMO Meeting (All Product Managers)

Item	Detail
Title	Weekly PDMO Product Managers Sync
Day & Time	Tuesday at 14:00 EAT
Duration	30–45 minutes
Attendees	All Product Managers + Ambrose
Advisors	Prakhar and/or Nate (as available)

Agenda:

1. Welcome and purpose of the meeting
2. Updates from team product meetings
3. Review of templates and weekly checklist
4. Cross-team dependencies and key issues
5. Team Highlight — one shoutout for SLA/KPI excellence that week
6. Under-the-Radar Contribution — nominated quiet wins
7. Questions, actions, and next steps

9.5 Technical Sharing Sessions (TSS)

Item	Detail
Frequency	Weekly (Friday)
Duration	30 minutes
Purpose	Knowledge sharing, skill development, cross-team learning
Format	One team presents a technical topic; mandatory Q&A from other teams
Output	Summary documented in Google Drive (Operations/TSS Summaries/) and posted in the team channel after the session
Connection	TSS content feeds into external technical blogs (Medium)

Rules for TSS:

- Presenters prepare adequately to fit within 30 minutes
- Every attendee from other teams must ask at least one relevant question
- Key learning is documented within 24 hours of the session

9.6 Full Weekly Meeting Reference

Day	Time	Meeting	Led By	Duration
Monday	09:00	Company Standup	GA	10 min
Monday	After 14:00	SNS Product Meeting	SNS PM	15 min
Tuesday	09:00	Company Standup	GA	10 min
Tuesday	After 10:30	AI Phone Agent Product Meeting	Kevin (PM)	15 min
Tuesday	14:00	PDMO — All PMs Sync	Ambrose	30–45 min
Wednesday	09:00	Company Standup	GA	10 min
Wednesday	After 12:00	Create Project Product Meeting	Ambrose (PM)	15 min
Thursday	09:00	Company Standup	GA	10 min
Thursday	After 11:00	EMS Product Meeting	EMS PM	15 min
Friday	09:00	Company Standup	GA	10 min
Friday	After 10:00	FOCUST Product Meeting	FOCUST PM	15 min
Friday	TBC	Technical Sharing Session	Assigned team	30 min

Section 10 — Capacity Building & Continuity

10.1 The Problem This Solves

When a key person is unavailable, projects should not stall. When a team member is not available, knowledge should not be inaccessible. This section establishes the systems that make AIBOS Uganda resilient.

10.2 The Shadow / Backup System

For every project, there is a **Primary** lead and a **Shadow**. The Shadow is not a passive observer — they are an active participant who stays informed and is ready to take over at any point.

Project	Primary Lead	Shadow / Backup	Shadow Responsibilities
Create Project	To be confirmed	To be confirmed	Attends prep meetings, CC'd on client comms, aware of all blockers
EMS	Martin / Andrew	To be confirmed	Same as above
FOCUST	FOCUST PM	To be confirmed	Same as above
AI Phone Agent	Kevin	To be confirmed	Same as above

Shadow Rules:

- Shadow attends the team's product meeting every week
- Shadow is CC'd on all significant client communication
- If Primary is unavailable, Shadow steps in immediately with no ramp-up delay
- Shadow must be able to give a 5-minute status of the project at any time

10.3 Delegation Framework

Managers should move from doing to overseeing:

```
TODAY (Person-Dependent)          TARGET (System-Dependent)
──────────────────────────────────────────────────────────
Manager writes every report       PM team rotates report writing
Manager monitors every project    KPI dashboard monitors projects
Manager answers every question    Response protocol handles it
Manager catches every bug         QA system catches bugs
Manager holds all knowledge       Documentation holds knowledge
──────────────────────────────────────────────────────────
```

Rotational Reporting:

- Daily reports are not always written by the same person
- PMs rotate the reporting duty among team members
- This ensures every team member understands project status and can communicate it clearly

Manager's New Role Checklist:

Old Activity	New Activity
Writing reports	Reviewing reports for quality
Answering every client question	Ensuring the protocol is followed
Catching bugs personally	Verifying the QA layer caught them
Knowing every detail	Knowing where the details are documented

10.4 Technical Sharing Sessions as Capacity Engine

TSS sessions should actively build team capacity, not just share information passively.

Active TSS Format:

Phase	Duration	Description
Presentation	20 min	Team presents a technical topic, decision, or implementation
Q&A	8 min	Other teams ask questions — minimum 1 question per attendee
Documentation	2 min	Key points saved to Google Drive and posted in the team channel within 24 hours

Cross-pollination rule: The Shadow for a project should lead the Q&A for that project's TSS presentation — they have the most to learn and the most responsibility to stay current.

10.5 The SOP Library (Standard Operating Procedures)

Rule: If a task is done more than twice, it must have a written "How-To" guide.

SOPs are stored in Google Drive under **Operations/SOPs/** for organizational procedures, or in the project's `/docs` folder for project-specific procedures.

Minimum SOP for any recurring task:

```
SOP: [Task Name]
─────────────────────
Purpose: What this task achieves
When to use: Trigger or frequency
Steps:
  1. ...
  2. ...
  3. ...
Owned by: [Role]
Last updated: [Date]
─────────────────────
```

10.6 Team Skills Map

Understanding what each team member does best is critical for task assignment, cross-team assistance, and targeted capacity building. The Skills Map is a living reference that grows as team members develop — it is not a performance evaluation, but a practical tool for PMs and management to match the right people to the right work.

PURPOSE OF THE SKILLS MAP

Use Case	How the Skills Map Helps
Task assignment	PMs assign based on proven strength, reducing delivery risk
Cross-team assistance	When Team A needs support, Team B can identify who has the right skill
Capacity planning	Managers can spot skills gaps before a new project begins
Career development	Team members know what to build next to fill gaps

SKILLS MAP — FRONTEND DEPARTMENT

Team Member	Core Strength	Secondary Strength	Learning Area	Available for Cross-Support
[Name]	[e.g., React / Next.js UI]	[e.g., API integration]	[e.g., Testing]	Yes / Limited / No
[Name]				
[Name]				

Skill categories for Frontend:

- UI component development (React / Next.js)
- State management (Redux, Zustand, Context API)
- API integration and data handling
- Responsive design and accessibility
- Performance optimization
- Testing (Jest, Playwright, Cypress)
- Design implementation (Figma → Code)

SKILLS MAP — BACKEND DEPARTMENT

Team Member	Core Strength	Secondary Strength	Learning Area	Available for Cross-Support
[Name]	[e.g., Node.js / Express]	[e.g., Database design]	[e.g., DevOps]	Yes / Limited / No
[Name]				
[Name]				

Skill categories for Backend:

- API development (REST / GraphQL)
- Database design and optimization (PostgreSQL, MySQL, MongoDB)
- Authentication and security
- Cloud infrastructure and deployment
- Automated testing and CI/CD pipelines
- Microservices and system architecture

- Integration with third-party services and AI/LLM APIs

HOW TO MAINTAIN THE SKILLS MAP

- The Skills Map template lives in Google Drive under **Teams/[Department]/**
- Each PM reviews the map **quarterly** and updates entries after significant project completions
- Team members can self-nominate updated strengths — the PM validates through direct observation
- Skills Map feeds into the TSS calendar — members strong in an area are the natural lead presenter for that topic

> **See Appendix J for the Skills Map completion template.**

10.7 Hiring & Talent Management

THE RIGHT PERSON FOR THE RIGHT TASK

Given the current team size and available resources, the most impactful decision AIBOS Uganda can make is to intentionally match each person to tasks they do well. This is not about limiting growth — it is about building confidence, delivering quality, and giving each individual room to become excellent before expanding their scope.

The benefits are immediate:

- Delivery quality improves because the person best suited to the task owns it
- Developer morale increases because people feel competent and valued, not overwhelmed
- Accountability is clearer when people own work they genuinely understand
- Quality work done confidently feeds directly into the recognition system — reinforcing a positive cycle

MORALE AND TASK ASSIGNMENT

Assignment Style	Impact on Morale
Assigned to area of strength	Confidence, flow, quality output, willingness to take on more
Assigned to unknown areas without support	Stress, slow delivery, self-doubt, quiet disengagement
Assigned with mentorship and a growth plan	Managed challenge, supported learning, long-term capability development

For managers: Before assigning a stretch task (outside someone's confirmed strengths), confirm:

- Is there someone available to mentor them through it?
- Is the deadline realistic for a learn-as-you-go scenario?
- Does the team member know this is a growth assignment, not just another delivery task?

GROWTH PATH AND CAREER DEVELOPMENT

The Reward & Motivation System (Section 12) is designed to reinforce and support individual career development. The link between task assignment, recognition, and growth looks like this:

```
PM assigns work aligned to strength (Skills Map)
        |
Member delivers quality work
        |
Recognition in Tuesday PDMO highlight
        |
Contribution noted in KPI dashboard
        |
Quarterly: TSS Lead Slot or Growth Acknowledgment
        |
Member's contributions documented in Google Drive grow as a track record
        |
Natural growth into new responsibilities over time
```

WHEN HIRING

As the team grows, new hires should be evaluated against the Skills Map — filling documented gaps rather than adding redundant strengths. Each new hire should:

- Be assigned a Shadow mentor from day one
- Have their skill profile added to the relevant department Skills Map
- Receive a clear first-90-days plan aligned to a specific project or workstream

Section 11 — Documentation & Knowledge Management

11.1 Philosophy

> **If it is not documented, it does not exist as a standard. If it is documented, it is assumed to be intentional.**

Documentation is not a chore — it is the mechanism by which individual knowledge becomes organizational knowledge. The DRY principle applies here too: document once, reference everywhere.

11.2 C2-PdMO Templates

The C2-PdMO (Product Design & Management Office) framework provides the standard document lifecycle for all projects.

Document Lifecycle:

```
Step 1: Define product vision        →  product.md
Step 2: Capture requirements         →  requirement-list
Step 3: Map functions                →  function-list
Step 4: Design architecture          →  architecture, tech, infrastructure, ADRs
Step 5: Track execution              →  weekly-tech-check
```

Approval Gates:

Gate	Name	Required Documents
G1	Scope Lock	product doc, requirement-list, function-list
G2	Design Gate	architecture, tech stack, infrastructure, ADRs
G3	Release Gate	All tests pass, design-gate-checklist complete

> **Rule: No code until G2 is passed.**

Template Map:

Phase	Document	Purpose
Planning	product.md	Vision, goals, users, success metrics
Planning	requirement-list	What users need and why
Planning	function-list	What the system does
Design	architecture.md	How the system is structured
Design	tech.md	Approved technology stack
Design	infrastructure.md	Cloud resources and configuration
Design	adrs/	Architecture Decision Records
Execution	weekly-tech-check.md	Health status and blockers
Execution	design-gate-checklist.md	Approval checkpoints

Reference: C2-PdMO Sample Meta on GitHub

11.3 Repository Documentation Standard

Every project repository must maintain a `/docs` folder with the following structure:

```
/docs
├── c2-pdmo/
│   ├── 1-planning/
│   ├── 2-design/
│   └── 3-execution/
│       └── YYYY-MM-DD-weekly.md
└── README.md
```

ADR (Architecture Decision Record) Process:

```
New architectural question arises
        │
        ▼
Create ADR (Proposed) — document the options and rationale
        │
        ▼
Review with team and Prakhar (if needed)
        │
    ┌──────────┐
    │Accepted? │
    └──────────┘
        │
  No ◄──┴──► Yes
  │              │
  Mark rejected    Update architecture.md
  (keep for        Mark ADR as Accepted
  history)         Supersede any prior ADR
```

Rules:

- Every major architecture change starts as an ADR
- `architecture.md` always reflects the current accepted state
- Superseded ADRs are kept for historical reference

11.4 Google Drive & Repository — Knowledge Management System

AIBOS Uganda uses a two-tier knowledge system that separates organizational knowledge from project-level technical documentation.

```
GitHub /docs    →  Technical documentation (C2-PdMO, ADRs, weekly checks)
Google Drive    →  Organizational knowledge (SOPs, templates, guidelines, meeting outputs)
```

Tier 1 — Project Repository (/docs) Already defined in Section 11.3. Each project repository maintains a `/docs` folder containing C2-PdMO planning documents, architecture records, and execution logs. This is the technical record for that specific project.

Tier 2 — Google Drive (Shared Organizational Knowledge) The team Google Drive serves as the central location for all non-code organizational knowledge — templates, SOPs, skills maps, guidelines, KPI tracking sheets, and meeting outputs.

Recommended Google Drive Folder Structure:

```
AIBOS Uganda (Shared Drive)
├── Operations/
│   ├── Templates/          ← All framework templates (Appendices A–M)
│   ├── SOPs/               ← Standard Operating Procedures
│   ├── TSS Summaries/      ← Weekly Technical Sharing Session summaries
│   └── KPI Tracking/       ← Google Sheets KPI dashboards per project
├── Projects/
│   ├── [Project Name]/
│   │   ├── Client Comms/   ← CLIENT-FACING documents
│   │   ├── Internal/       ← INTERNAL project documents
│   │   └── Handover/       ← Shadow handover records
├── Teams/
│   ├── Frontend/           ← Skills Map, department SOPs
│   └── Backend/            ← Skills Map, department SOPs
└── Shared with Clients/    ← Copies of documents shared externally
```

Access Control:

Folder	Access Level
Operations/	All team members (view), PMs + Ambrose (edit)
Projects/[Name]/Client Comms/	PM + Ambrose (edit), client (view on request)
Projects/[Name]/Internal/	Project team (edit), Ambrose (view)
Teams/	All team members (view), PMs + Ambrose (edit)
Shared with Clients/	PM + Ambrose (edit), client (view)

Governance Rules:

- Product team is the primary custodian of the Drive folder structure
- Every template from this framework (Appendices A–M) is stored in Operations/Templates/
- Undocumented practices are not considered standards — if it matters, it goes in Drive or /docs
- New folders should follow the structure above; ad-hoc top-level folders are not permitted

11.5 External Knowledge Sharing

As internal documentation matures, AIBOS Uganda will begin publishing external technical content.

Channel	Purpose	Quality Standard
Medium (initial)	Technical blogs from TSS content	Solves a real problem; not generic AI-generated content
AIBOS Blog (future)	Company-owned technical publishing	Curated, quality-reviewed

Connection to TSS: A TSS session can motivate a blog post, or a blog post can be presented in TSS. Both reinforce learning and external visibility.

Ownership: Product team initiates and manages quality control. Engineering teams are the authors.

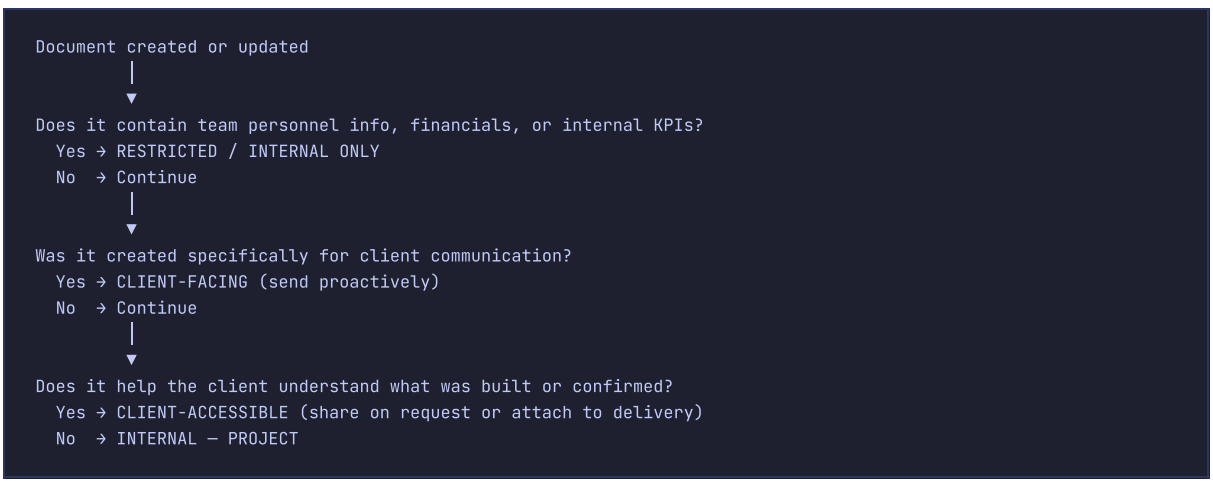
11.6 Internal vs. Shared Knowledge Boundary

Not all documentation is meant for the client. A clear, consistent boundary between what is internal-only and what is shared with the client protects the team, maintains trust, and ensures clients receive the right level of transparency — without unnecessary exposure of internal processes, systems, or discussions.

KNOWLEDGE CLASSIFICATION CATEGORIES

Category	Classification	Description	Examples
Client-Facing	Always shared	Created specifically for client communication	Daily progress reports, release notes, delivery confirmations, requirement confirmations
Client-Accessible	Shared on request	Available to the client if asked, but not proactively sent	Approved technical specs, staging links, test plans, deployment schedules
Internal — Project	Internal only	Used by the project team for execution	ADRs, architecture docs, peer review comments, bug trackers, QA logs, KPI logs
Internal — Organizational	Internal only	Organizational operations, not project-specific	Team performance data, capacity plans, KPI dashboard scores, team health discussions
Restricted	Leadership only	Sensitive to business or personnel	Personnel matters, financial discussions, strategic decisions not yet finalized

DECISION GUIDE — IS THIS SHAREABLE WITH THE CLIENT?



ALWAYS INTERNAL VS. ALWAYS SHARED — QUICK REFERENCE

Always Internal	Always Shared with Client
QA bug logs and pre-delivery test results	Delivery confirmation and release notes
KPI dashboard scores per team	Daily progress report (EOD)
Peer review and code review comments	Requirement confirmations (before work starts)
Architecture Decision Records (ADRs)	Staging environment link for acceptance testing
Team capacity and Skills Map data	Timeline and delivery schedule
Internal escalation and risk records	Delay notifications (proactive)
Shadow assignments and backup plans	Feature change requests (confirmed scope)

TAGGING DOCUMENTS IN GOOGLE DRIVE

When saving documents in Google Drive, include the classification tag in the file name or document description:

- `[CLIENT-FACING]` — place in Projects/[Name]/Client Comms/ or Shared with Clients/
- `[INTERNAL]` — place in Projects/[Name]/Internal/ or Operations/
- `[RESTRICTED]` — place in a leadership-only subfolder with restricted access

This prevents accidental sharing and helps new team members understand what belongs in client communications from day one.

Section 12 — Reward & Motivation System

12.1 Philosophy

> **Systems sustain behavior. Recognition energizes it.**

At AIBOS Uganda's current stage, the most powerful rewards are **recognition and growth opportunity**, not just monetary incentives. The reward system should feel like "this person made the team stronger" — not a competition.

Recognition must be:

- Specific (tied to a real action or KPI outcome)
- Genuine (not generic praise)
- Consistent (not occasional)
- Framed around system contribution (not individual heroics)

12.2 Tier 1 — Weekly Recognition (Zero Cost)

Every Tuesday PDMO meeting includes a **"Team Highlight"** — one specific shoutout for a team or individual that exemplified the SLA/KPI standards that week.

This is posted in the team channel after the meeting.

Examples of strong framing:

- "This week the FOCUST team reported a delay risk on Tuesday — three days before the deadline — giving us time to adjust. That is exactly the culture we are building."
- "AI Phone Agent had zero client-reported bugs this week. The QA layer worked exactly as designed."

12.3 Tier 2 — Monthly KPI-Linked Awards

At the end of each month, review the KPI dashboard and recognize:

Award	Criteria	Recognition
Best Response Rate	Fewest 15-min response violations	Shoutout + noted in monthly summary to Japan
Cleanest Delivery	Lowest post-delivery bugs reported by client	Shoutout + featured in PDMO summary
Best QA Catch	Most pre-delivery bugs caught internally	Shoutout — framed as "protected the team"
Proactive Communicator	Most delay risks escalated early	Shoutout — framed as "showed leadership"
Documentation Champion	Best/most consistent documentation update	Shoutout — connection to Google Drive / docs/ contribution

12.4 Tier 3 — Quarterly Growth Reward

The top performing team member per quarter receives a **TSS Lead Slot** — they choose a topic of their interest and present it to the whole company.

This is a growth opportunity, not just a trophy:

- Builds the individual's profile internally
- Develops their communication and technical articulation skills
- Connects to the external blog initiative — their TSS becomes their first published blog post

If budget allows: A small monetary acknowledgment or team experience (team lunch, etc.) to accompany the TSS slot.

12.5 How to Frame Recognition

Avoid	Use Instead
"You worked very hard"	"You protected the team by escalating early"
"Good job"	"Your QA catch prevented a client issue this week"
"You're the best developer"	"Your documentation means the next person can pick this up easily"

The goal is to make excellence feel systemic and replicable, not exceptional and individual.

12.6 AIBOS Swag & Tangible Recognition

As AIBOS Uganda grows, tangible recognition becomes an important complement to the verbal and written acknowledgment system. Branded swag serves both as a reward and as a visible symbol of belonging to a team that values quality — something people are proud to carry or wear.

PHASE-APPROPRIATE ROLLOUT

Phase	Milestone	Items
Starter	On joining + first 90-day completion	Branded notebook, pen, sticker pack
Recognition	Quarterly top performer or project milestone	AIBOS branded t-shirt or hoodie
Milestone	1-year anniversary or major project launch	Higher-value item (jacket, backpack, or team experience)

Framing swag with meaning: Swag should always be accompanied by a specific reason — it is not a participation trophy. The story is what makes the item memorable:

> "This hoodie goes to the FOCUST team for completing the beta launch on time with zero client-reported bugs. That is the standard we are building this team around."

12.7 Overtime & Extra-Mile Recognition

WHY IT NEEDS ITS OWN CATEGORY

Extra hours put in during a crunch period, a critical release, or an emergency are often invisible in standard KPI-based recognition. Without acknowledgment, repeated overtime builds quiet resentment — even among people who genuinely care about the work.

AIBOS Uganda recognizes extra-mile effort as a distinct and important category of contribution.

RECOGNITION MODEL FOR OVERTIME

Type of Effort	Recognition Approach
Late-night deployment or emergency fix	Named shoutout in next-day morning standup
Critical task completed over the weekend	Acknowledged in the following Tuesday PDMO highlight
Consistently staying late during a sprint	Noted in the monthly summary shared with Japan
Going beyond scope to unblock a teammate	PDMO callout framed as: "This person made it possible for the team to ship"

> **Important balance:** Recognition of overtime is not an invitation to normalize it. The PM who acknowledges the effort should also ask: "What process gap caused this crunch? How do we prevent it next time?" Overtime recognition and process improvement go hand in hand.

COMPENSATORY TIME

Where overtime has been significant and sustained, managers should consider compensatory time (time off in lieu) as a practical acknowledgment. This should be discussed and confirmed with Nate as appropriate for each situation.

12.8 Contribution Recognition System

THE PROBLEM IT SOLVES

In most teams, the most lasting contributions — writing the guide that helped everyone, cleaning up the codebase, documenting the architecture others struggled to understand — happen quietly and go unnoticed. The KPI-based system risks rewarding visible delivery metrics while overlooking equally impactful work that does not show up in a dashboard.

CONTRIBUTION TYPES WORTH RECOGNIZING

Contribution	Why It Matters
Documenting a complex process in Google Drive or the project docs/ folder	Saves every future team member hours of confusion
Writing an SOP the whole team now follows	Makes the organization more resilient
Mentoring a junior member through a difficult task	Multiplies capability across the team
Identifying and fixing a systemic issue proactively	Protects the team before a problem reaches the client
Proposing a process improvement that is adopted	Demonstrates system-thinking and leadership
Leading a TSS session that other teams found valuable	Builds cross-team knowledge and external visibility

THE "UNDER-THE-RADAR" CONTRIBUTION SHOUTOUT

In the weekly Tuesday PDMO summary, a recurring segment: **"This Week's Under-the-Radar Contribution"**

- Any PM or team member can nominate a contribution — not limited to KPI metrics
- One nomination per week is highlighted in the team channel

Sample post:

> "This week's contribution shoutout goes to [Name] for documenting the EMS deployment process in Google Drive. Three team members already referenced it — it paid off immediately."

LONG-TERM: INDIVIDUAL CONTRIBUTION PROFILE

As the documentation library in Google Drive and the project repos grows, each team member will have a **Contribution Profile** — a record of what they have built, documented, and shared. This becomes a career development artifact:

- Referenced during quarterly reviews
- Shared where appropriate with Japan leadership
- Used as the basis for TSS nominations and growth opportunities
- Serves as an internal portfolio for individual progression at AIBOS Uganda

Section 13 — MVP to Production Framework

13.1 The Recurring Challenge

Across multiple AIBOS Uganda projects, the same pattern has emerged: an MVP is built quickly to demonstrate a concept, and almost immediately the client expects it to function as a production-ready system. This has caused:

- Scope creep without proper planning
- Quality gaps because MVP code was not designed for production
- Team stress from rapid, unstructured transitions
- Feature additions without client confirmation (e.g., dark mode, extra features)

13.2 Stage Definitions

Stage	Definition	Quality Expectation	Client Visibility
MVP	Built to validate feasibility or demonstrate a concept	May include shortcuts; not production-grade	Internal / invited preview only
Demo	Presentation-focused; optimized for clarity over completeness	Not assumed production-bound	Stakeholder preview on staging
Beta	Feature-complete but still being validated and stabilized	Near-production quality	Client tests on staging
Production	Stable, tested, and deployed for real users	Meets all QA gates	Live — client and end users

13.3 Transition Guidelines

Before any stage transition, the following must be confirmed:

```
MVP → Demo
──────────────────────────────────────────────
☐  Core concept validated internally
☐  No unconfirmed extra features added
☐  Scope explicitly agreed with client

Demo → Beta
──────────────────────────────────────────────
☐  Requirements fully documented
☐  Architecture reviewed (G2 gate passed)
☐  Client aware this is still a validation stage

Beta → Production
──────────────────────────────────────────────
☐  All QA layers passed
☐  Staging environment stable and client-tested
☐  Client has confirmed readiness
☐  Release governance completed
☐  Production smoke test passed
```

13.4 Environment Flow

```
DEVELOPER
    |
    ▼
DEV environment
(feature built and developer-tested)
    |
    ▼
STAGING environment
(QA Owner reviews; client validates specific features)
    |
Client confirms
    ▼
PRODUCTION environment
(Feature released; monitoring active)
    |
    ▼
EOD report updated
```

13.5 Client Confirmation for Extra Features

> **Before adding any feature beyond the original scope, the PM must confirm with the client:** "We are planning to add [Feature X]. Is this something you would like included?"

This applies even when the feature seems helpful or proactive. A feature built without confirmation is a feature the client may not want to pay for.

13.6 Project Intake Process

Every new client project or major scope change must pass through the Product team before implementation begins.

```
Client submits new project / request
         |
         ▼
Product team receives and reviews
(Ambrose + relevant PM)
         |
         ▼
Product team assesses:
  - What is the scope?
  - What phase does it start at? (MVP/Beta/Production)
  - What team / resources are needed?
  - What is the delivery timeline?
         |
         ▼
Product team briefs the implementation team
         |
         ▼
G1 gate: Requirements documented
         |
         ▼
Implementation begins
```

Section 14 — Implementation Roadmap

14.1 Overview

14.2 Phase 1 — Foundation (Feb 23 – Mar 15)

Goal: Prepare the system. Build awareness. Assign ownership.

Action	Owner	Target Date
Confirm Ambrose as Management Owner	Prakhar / Nate	Feb 23
Share and review this framework document with Martin and Joseph	Ambrose	Feb 27
Nominate QA Owners for each project	Each team	Mar 1
Set up KPI tracking sheet (Google Sheets)	Ambrose	Mar 1
Confirm shadow assignments per project	Ambrose + PMs	Mar 7
Brief full team on daily pulse and SLA expectations	Ambrose + PMs	Mar 10–14
Set up Google Drive folder structure (Operations/, Projects/, Teams/)	Product team	Mar 14

14.3 Phase 2 — Trial Operation (Mar 16 – Mar 31)

Goal: Run the system. Identify friction. Adjust.

What Goes Live	Notes
Daily Pulse (standup → prep → sync → peer review → EOD report)	Full cadence from day one
KPI daily logging begins	Each PM fills their row every evening
15-min response protocol	All teams begin tracking
QA Owner role activated	DoD checklist used for every delivery
Weekly PDMO meeting running	Tuesday 14:00 EAT
Team product meetings running	Per confirmed schedule

Review points: Ambrose reviews what is working and what feels "clunky" — adjustments made before April.

14.4 Phase 3 — Proving Period (Apr 1 – Apr 30)

Goal: Confirm adoption. Measure outcomes. Build confidence.

Metric	Target
15-min response violations	Trending toward zero
Delay risks escalated proactively	80%+ of the time
Post-delivery bugs reported by client	Significant reduction from baseline
KPI sheet filled consistently	90%+ completion rate
Team able to operate without Ambrose present for a day	Yes

At the end of April, a review meeting with Prakhar and Nate confirms whether the system is functioning and what the next phase looks like.

14.5 Phase 4 — AI Tools & Shared Environment (May)

Goal: Once the operational foundation is stable, introduce AI tools and a shared team environment to accelerate delivery and reduce cost.

Areas to roll out (see Section 15 for full detail):

- Claude Code for automated test generation and code review support
- AI-assisted documentation drafting for Google Drive and project docs/ entries
- Automated KPI reporting from existing data sources
- Shared team VM / server environment setup (1 shared account model)

Prerequisites before Phase 4 begins:

- Phase 3 Proving Period (April) confirms system is functioning
- KPI sheet at 90%+ fill rate consistently
- QA Owner roles active and rotational handover documented
- At least one full TSS cycle completed

> **Important:** AI tools are introduced to accelerate a working system — not to fix a broken one. The April proving period must confirm system stability before this phase begins.

Section 15 — AI Tools & Shared Team Environment

15.1 Philosophy

> **AI tools are introduced to accelerate a working system — not to fix a broken one.**

The AI integration plan for AIBOS Uganda is deliberately staged. The operational foundation (Sections 3–14) must be stable and adopted before AI tools are layered on top. When that foundation is in place, AI can compress timelines, reduce repetitive work, and allow the team to focus on higher-value decisions.

15.2 AI Tools for Managers and Teams

The following areas are planned for AI tool integration (see Section 14.5 — Phase 4):

Area	AI Application	Benefit
Code review	AI-assisted code review (e.g., Claude Code) to supplement the QA layer	Faster catch of standard errors; developers get immediate feedback
Test generation	Automated unit and integration test generation from code	Reduces manual test writing; improves regression coverage
KPI reporting	Automated reporting from existing data sources	Reduces time spent manually compiling weekly reports
Documentation	AI-assisted drafting of Google Drive documents, SOPs, and project docs/ entries	Faster documentation; encourages consistent formatting
Requirement analysis	AI review of requirement documents for gaps or ambiguities	Catches missing edge cases before development starts

TOOLS IN SCOPE (PHASE 4 — MAY ONWARDS)

- **Claude Code** — AI coding assistant for development and automated test generation
- **AI-assisted review** — Supplement to peer code review; identifies common issues quickly
- **Automated KPI dashboarding** — Scripts or integrations that pull data from existing channels into the dashboard automatically

> **Important:** Managers should be trained to use AI tools to review, guide, and audit — not to replace judgment. AI output is always reviewed by a human before any client-facing use.

15.3 Shared Team Environment

As AIBOS Uganda grows, access to AI tools and development infrastructure should be structured for cost efficiency and team-wide access — not fragmented across individual personal accounts.

THE SHARED VM / SERVER WORKSPACE MODEL

Rather than each team member maintaining their own separate accounts and local environments, AIBOS Uganda will move toward a **shared team server environment**:

```
SHARED TEAM WORKSPACE
──────────────────────────────────────────────────
1 Virtual Machine / Server  →  Accessible by whole team
1 Shared Account (per tool) →  Used by all team members
Centralized environment     →  Consistent setup, no "works on my machine"
──────────────────────────────────────────────────
```

WHY THIS MODEL

Challenge (Individual Accounts)	Solution (Shared Environment)
Each developer maintains a separate AI tool subscription	One shared account reduces cost significantly
Environment inconsistencies cause “works on my machine” bugs	One server environment means everyone runs the same setup
Onboarding a new team member requires full environment setup	New members connect to existing server — ready in minutes
AI tool costs scale with headcount	Shared model scales at a fraction of individual-account costs

IMPLEMENTATION NOTES

- The shared VM should be set up as part of the **Phase 4 rollout (May)**
- Access control: each team member has their own login to the VM, but shares the AI tool account credentials via a secure, centrally managed method (e.g., team password manager)
- The VM environment setup should be documented in Google Drive (Operations/SOPs/) as the standard development environment going forward
- As the team grows and usage demands increase, this model can be upgraded (more compute, additional accounts) — but the shared foundation keeps early-stage costs manageable

COST EFFICIENCY OVER TIME

```
Small Team (Today)
  1 shared account — cost fixed regardless of team size
          |
          ▼
Growing Team (6–12 months)
  Same account, more users — still 1 subscription cost
          |
          ▼
Scaled Team (12+ months)
  Evaluate: 1 upgraded tier vs. multiple accounts
  Decision informed by actual usage data
```

> **Note:** This model works best for internal development and AI tools. Client-facing systems, production accounts, and individual authentication for billing-critical services should remain separate for security and accountability.

Appendices

Appendix A — Daily Report Template

```
Hello Team / Hello [Client Name],

Please find below the daily progress update for [Day, Date].

1. PREVIOUS DAY PROGRESS
- [Item 1]
- [Item 2]

2. PLANNED ACTIVITIES (NEXT)
- [Item 1]
- [Item 2]

3. RISKS / BLOCKERS (if any)
[ ] None at this time
OR: [Description and mitigation]

4. SUPPORT NEEDED (if any)
[ ] None
OR: [What is needed, from whom, by when]

Best regards,
[Name] | [Role / Team]
```

Appendix B — 15-Minute Response Templates

Path B — Investigation Acknowledgment:

> "Hi [Client Name], thank you for your message regarding [Topic]. I have flagged this with the team and we are currently reviewing the details. I will get back to you with a status update within the hour. Best regards, [Name]"

Path C — Long-Term Acknowledgment:

> "Hi [Client Name], thank you for reaching out about [Topic]. This requires a deeper review of [Area]. We are looking into it and will provide a full update in our end-of-day report, or sooner if resolved. Best regards, [Name]"

1-Hour Update:

> "Hi [Client Name], a quick update: we have identified the issue is related to [Area]. We are currently [testing/investigating/refining] the solution. I will confirm the final outcome in our daily report. Best regards, [Name]"

Appendix C — Definition of Done Checklist

```
DEFINITION OF DONE
──────────────────────────────────────────
☐  Requirements confirmed in writing before work started
☐  Developer has self-tested the feature
☐  Code has been peer-reviewed
☐  Feature works correctly on staging
☐  No obvious bugs or broken flows in staging
☐  QA Owner has reviewed and signed off
☐  Documentation updated if required
☐  Client notified if delivery is imminent
──────────────────────────────────────────
QA Owner sign-off: ____________________
Date: ____________________
```

Appendix D — KPI Dashboard Template

KPI	Mon	Tue	Wed	Thu	Fri	Weekly Total	Target
15-min response violations							0
Delay risks escalated proactively							100%
Pre-delivery bugs found in QA							Log trend
Post-delivery bugs (client reported)							0
On-time delivery							100%

Appendix E — Weekly PDMO Agenda Template

```
WEEKLY PDMO PRODUCT MANAGERS SYNC
Tuesday | 14:00 EAT

1. Welcome and purpose of the meeting
2. Updates from team product meetings
3. Review of templates and weekly checklist
4. Cross-team dependencies and key issues
5. Team Highlight — one shoutout for SLA/KPI excellence that week
6. Under-the-Radar Contribution — nominated quiet wins
7. Questions, actions, and next steps
```

Appendix F — Meeting Schedule Reference

Day	Meeting	Led By	Duration
Mon	Company Standup	GA	10 min
Mon	SNS Product Meeting	SNS PM	15 min (after 14:00)
Tue	Company Standup	GA	10 min
Tue	AI Phone Agent Product Meeting	Kevin (PM)	15 min (after 10:30)
Tue	PDMO All-PMs Sync	Ambrose	30–45 min (14:00)
Wed	Company Standup	GA	10 min
Wed	Create Project Product Meeting	Ambrose	15 min (after 12:00)
Thu	Company Standup	GA	10 min
Thu	EMS Product Meeting	EMS PM	15 min (after 11:00)
Fri	Company Standup	GA	10 min
Fri	FOCUST Product Meeting	FOCUST PM	15 min (after 10:00)
Fri	Technical Sharing Session	Assigned team	30 min

Appendix G — C2-PdMO Document Reference Map

Charter Requirement	Template Location	Status
Architecture Governance	2-design/architecture.md, tech.md, adrs/	Covered
Traceability (REQ→FUNC→COMP)	Throughout planning and design phases	Covered
Feasibility & Quality Audit	3-execution/design-gate-checklist.md	Covered
Weekly Architecture Review	3-execution/weekly-tech-check.md	Covered
Design Gate (Approval)	3-execution/design-gate-checklist.md	Covered
Weekly Tech Check Format	3-execution/weekly-tech-check.md	Covered
Design Approval Request	1-planning/requirement-list, function-list	Covered

Appendix H — QA Owner Appointment Form

Deadline: March 1 (Phase 1 — Foundation)

```
QA OWNER APPOINTMENT
─────────────────────────────────────────────────────────

Project:              _____________________________
QA Owner (Primary):   _____________________________
QA Owner (Backup):    _____________________________
Appointed By (PM):    _____________________________
Appointment Date:     _____________________________
Rotation Schedule:    Every 2-3 months (next: ________)

QA Owner Responsibilities (confirm understanding):
  ☐  Run smoke test checklist on every staging deployment
  ☐  Sign off on the Definition of Done checklist
  ☐  Log all pre-delivery bugs in the KPI sheet
  ☐  Escalate to PM immediately if quality risk is detected before deadline
  ☐  Brief incoming QA Owner before rotation handover

QA Owner Signature: ____________________  Date: _______
PM Signature:       ____________________  Date: _______
─────────────────────────────────────────────────────────
```

Appendix I — Risk Assessment & Rollback Plan Template

```
CHANGE RISK ASSESSMENT & ROLLBACK PLAN
─────────────────────────────────────────────────────────

Project:                    _____________________________
Feature / Module Changed:   _____________________________
Priority Level:             P0 / P1 / P2 / P3
Change Description:         _____________________________
Planned Deployment Date:    _____________________________

RISK ASSESSMENT
─────────────────────────────────────────────────────────

What could go wrong?        _____________________________
Which features are affected? _____________________________
Is this reversible?         Yes / No
Risk level:                 Low / Medium / High

ROLLBACK PLAN
─────────────────────────────────────────────────────────

Step 1:   Revert to last stable deployment
Step 2:   Notify PM and QA Owner immediately
Step 3:   PM notifies client — [template below]
Step 4:   Root cause investigation begins
Step 5:   Fix confirmed and tested before re-deployment

CLIENT COMMUNICATION TEMPLATE (Rollback)
─────────────────────────────────────────────────────────

"Hi [Client], we encountered an issue with the recent
update to [Module]. We have reverted to the previous
stable version to protect your service. The issue is
under investigation and we will confirm a resolution
timeline in our next update. We apologize for any
inconvenience. [Name]"

APPROVAL
─────────────────────────────────────────────────────────

PM Sign-off:        ____________________  Date: _______
QA Owner Sign-off:  ____________________  Date: _______
Client Notified?    Yes / No              Date: _______
─────────────────────────────────────────────────────────
```

Appendix J — Skills Map Template

To be filled per department and stored in Google Drive under **Teams/[Department]/Skills Map**. Updated quarterly.

```
DEPARTMENT SKILLS MAP
_____________________________________________________

Department:     Frontend / Backend
Last Updated:   _____________________________
Updated By:     _____________________________

SKILL CATEGORIES (mark each member's level):
  ★★★ = Strong (can lead and teach)
  ★★  = Confident (delivers independently)
  ★   = Learning (needs some support)
  —   = Not yet explored

MEMBER SKILL GRID
_____________________|_______|_______|_______|_______|_______|_____
Team Member          | Skill | Skill | Skill | Skill | Skill | Cross-Team
                     |   1   |   2   |   3   |   4   |   5   | Available?
_____________________|_______|_______|_______|_______|_______|_____
[Name]               |       |       |       |       |       | Yes/No
[Name]               |       |       |       |       |       |
[Name]               |       |       |       |       |       |
_____________________|_______|_______|_______|_______|_______|_____

NOTES / DEVELOPMENT AREAS:
[Name]: Targeting [skill] by [quarter]
[Name]: Mentor assigned for [skill] — [Mentor Name]
_____________________________________________________
```

Appendix K — Stakeholder Communication Matrix

Stakeholder	Communication Type	Channel	Frequency	Owner
DATAGRID / Client	Daily progress update	Client channel / email	Daily (17:30)	Project PM
DATAGRID / Client	Delay or risk notification	Direct message + channel	Immediately on risk recognition	Project PM
DATAGRID / Client	Planned change notification	Client channel	Before P0/P1 deployments	PM + QA Owner
AIBOS Japan (Prakhar)	Weekly status summary	PDMO summary document	Weekly (after Tuesday PDMO)	Ambrose
AIBOS Japan (Nate)	Operational matters	Direct consultation	As needed	Ambrose
Internal team	Daily Pulse (EOD report)	Daily Reports channel	Daily (17:30)	PM per project
Internal team	Weekly PDMO highlights	Team channel	Weekly (Friday)	Ambrose
Internal team	TSS session summary	Google Drive + team channel	Within 24hrs of session	Presenting team

Appendix L — Shadow / Backup Handover Template

Used when a Primary lead is unavailable and the Shadow steps in.

```
SHADOW HANDOVER BRIEFING
──────────────────────────────────────────────────────────

Project:                _____________________________
Primary Lead:           _____________________________
Shadow (Stepping In):   _____________________________
Handover Date:          _____________________________
Expected Duration:      _____________________________

CURRENT STATUS
──────────────────────────────────────────────────────────

Active tasks:           _____________________________
Pending client items:   _____________________________
Known blockers:         _____________________________
Next deadline:          _____________________________

CLIENT CONTACTS
──────────────────────────────────────────────────────────

Primary contact:        _____________________________
Preferred channel:      _____________________________
Tone/context notes:     _____________________________

KEY LINKS
──────────────────────────────────────────────────────────

Project board:          _____________________________
KPI sheet row:          _____________________________
Repository:             _____________________________
Staging URL:            _____________________________

PRIMARY LEAD NOTES:
[Any additional context the Shadow needs to know]

Shadow confirms readiness:  ____________________  Date: _______
──────────────────────────────────────────────────────────
```

Appendix M — Weekly Product Checklist Template

Used by each PM during the weekly 15-minute team product meeting.

```
WEEKLY PRODUCT CHECKLIST
──────────────────────────────────────────────────────────

Project:        _____________________________
Week of:        _____________________________
PM:             _____________________________
Meeting Date:   _____________________________

DELIVERY CHECK
──────────────────────────────────────────────────────────
☐  All tasks from last week: completed / carried over / escalated?
☐  Any task delivered without client confirmation of requirements?
☐  Any deliverable that went to client without QA sign-off?
☐  Any deadline missed this week? (If yes: documented and communicated)

QUALITY CHECK
──────────────────────────────────────────────────────────
☐  QA Owner performed smoke test after last deployment
☐  No P0 or P1 feature broken in current staging
☐  KPI sheet filled every day this week

COMMUNICATION CHECK
──────────────────────────────────────────────────────────
☐  All client messages responded to within 15 minutes
☐  No delay risks communicated late (on due date)
☐  EOD report sent every working day

PLANNING CHECK
──────────────────────────────────────────────────────────
☐  Next week's priorities agreed by team
☐  Any blockers that need management attention?
☐  Shadow up to date on project status?

NOTES / ACTIONS FROM THIS MEETING:
[Action] → [Owner] → [Target Date]
──────────────────────────────────────────────────────────
```

Version: 2.0 Last Updated: February 2026 Document Owner: Ambrose Alanda For questions or updates: B1 Channel