

DIGITAL PRODUCT

AI SOP Builder Kit

Standard Operating Procedures Built for AI Execution

Logic Impact | \$27 Value | January 2025

Table of Contents

AI SOP Builder Kit

Standard Operating Procedures Built for AI Execution

Introduction: Why SOPs Matter for AI Execution

The AI Execution Gap

Why Traditional SOPs Fall Short for AI

The Compound Effect of Good SOPs

What You'll Build With This Kit

Chapter 1: What Makes a Good SOP for AI

The AI Mindset Shift

The Six Elements of AI-Ready SOPs

Language Patterns That Work

The Context Boundary Problem

Testing Your SOPs

The Iteration Imperative

Chapter 2: The SOP Template

The Master Template

Template Section Deep Dive

Chapter 3: 10 SOP Examples

SOP 1: Lead Response Automation

SOP 2: Content Creation Workflow

SOP 3: Follow-Up Sequence Automation

SOP 4: Calendar Booking Automation

SOP 5: Invoice Sending Automation

SOP 6: Client Onboarding Sequence

SOP 7: Social Media Posting Workflow

SOP 8: Email Sorting and Triage

SOP 9: Call Routing and Qualification

SOP 10: Data Backup and Verification

Chapter 4: How to Train Your AI on SOPs

The Training Process

Training Methods by AI Platform

Prompt Engineering for SOP Execution

Monitoring AI SOP Execution

Common Training Pitfalls

Continuous Improvement

Chapter 5: Maintenance & Version Control

The Cost of Stale SOPs

Version Control System

Review Cycles

Maintenance Triggers

Update Workflow

SOP Deprecation

Documentation Hygiene

Collaboration Practices

Measuring SOP Health

Tools for SOP Management

Appendix: Blank SOP Templates

Template A: Event-Based Automation SOP

Template B: Time-Based Automation SOP

Template C: Human-Initiated SOP

Template D: Decision/Classification SOP

Template E: Multi-System Integration SOP

Conclusion

AI SOP Builder Kit

Standard Operating Procedures Built for AI Execution

Introduction: Why SOPs Matter for AI Execution

The promise of AI in business has always been tantalizing: automate the mundane, free up human creativity, and scale operations without proportional increases in headcount. Yet for many businesses, this promise remains just out of reach. The culprit isn't the technology—it's the lack of clear, structured instructions that AI systems can follow consistently.

This is where Standard Operating Procedures (SOPs) become transformative. Not the dusty binders of corporate bureaucracy, but living, precise documents that bridge the gap between human intent and machine execution.

The AI Execution Gap

Consider this scenario: A business owner asks their AI assistant to "handle customer inquiries." Without explicit instructions, the AI might respond too formally, miss critical follow-up steps, or fail to escalate urgent issues appropriately. The result? Frustrated customers and more work for the human team.

Now imagine the same request with a well-crafted SOP. The AI knows exactly: - Which inquiries to answer immediately - When to escalate to a human - What tone to use for different customer types - How to categorize and tag each interaction - When to trigger follow-up sequences

The difference is night and day.

Why Traditional SOPs Fall Short for AI

Traditional SOPs were written for humans. They rely on context, judgment, and institutional knowledge that AI systems simply don't possess. Phrases like "use your discretion" or "handle appropriately" are meaningless to a language model.

AI-native SOPs are different. They're: - **Explicit**: Every step is clearly defined with no room for interpretation - **Structured**: Consistent formatting that AI can parse and follow - **Conditional**: Clear if/then logic for decision points - **Tool-aware**: Specific about which systems and integrations to use - **Measurable**: Defined outputs that can be verified

The Compound Effect of Good SOPs

When you invest time in creating AI-compatible SOPs, the returns compound:

1. **Consistency:** Every task is executed the same way, every time
2. **Scalability:** Add new AI capabilities without reinventing processes
3. **Training speed:** New AI implementations learn from documented procedures
4. **Quality control:** Clear standards make it easy to audit and improve
5. **Human handoff:** When escalation is needed, the context is clear

What You'll Build With This Kit

By the end of this guide, you'll have:

- A reusable SOP template designed specifically for AI execution
- 10 complete SOP examples covering common business workflows
- A system for training AI assistants on your procedures
- Version control practices to keep SOPs current
- A library of blank templates for rapid SOP creation

Whether you're implementing your first AI assistant or optimizing an existing deployment, this kit provides the framework for reliable, consistent AI execution.

Let's begin.

Chapter 1: What Makes a Good SOP for AI

Creating SOPs for AI requires a fundamental shift in thinking. Unlike human employees who can infer intent and ask clarifying questions, AI systems execute exactly what they're told—nothing more, nothing less. This chapter explores the principles that make SOPs effective for AI execution.

The AI Mindset Shift

When writing for AI, you must become a meticulous instruction designer. Think of it as programming with natural language—every detail matters because the AI has no access to your mental model or the “obvious” context you take for granted.

Key Principle: Explicit Over Implicit

Bad: “Respond to customer emails professionally”

Good: “Respond to customer emails using a friendly but formal tone. Start with ‘Dear [Name],’ end with ‘Best regards,’ and address their specific question in the second paragraph. If the inquiry involves billing, CC the billing team.”

The second version leaves no room for interpretation. It specifies format, tone, structure, and conditional actions.

The Six Elements of AI-Ready SOPs

1. Clear Trigger Conditions

Every SOP needs a precise trigger that tells the AI when to activate. Vague triggers lead to missed opportunities or inappropriate execution.

Effective triggers include: - Specific keywords or phrases - Time-based conditions (“Every Monday at 9 AM”) - Event-based conditions (“When a new lead is added to CRM”) - Data-based conditions (“When invoice is 7 days overdue”)

2. Sequential Steps

AI systems excel at following linear sequences. Break complex workflows into numbered steps that execute in order.

Each step should: - Start with an action verb - Include all necessary details - Specify tools or systems to use - Define success criteria

3. Decision Trees

When decisions are required, frame them as clear if/then statements:

```
IF [condition] THEN [action]
ELSE IF [condition] THEN [action]
ELSE [default action]
```

This structure mirrors programming logic and helps AI systems navigate choices consistently.

4. Tool Specifications

Modern AI doesn't work in isolation—it connects to CRMs, email systems, calendars, and databases. Every SOP must specify: - Which tools to use - What data to input or retrieve - How to format API calls or commands - Error handling for failed connections

5. Output Definitions

Define exactly what constitutes successful completion: - What data should be updated - What notifications should be sent - What records should be created - How to format final deliverables

6. Error Handling

Things go wrong. A good SOP anticipates failures and provides clear escalation paths: - Common error scenarios - Retry logic (how many times, how long to wait) - When to stop and notify a human - How to document the failure

Language Patterns That Work

Certain linguistic structures translate better to AI execution than others:

Use:

- **Imperative verbs:** “Check,” “Create,” “Send,” “Verify”
- **Specific quantities:** “Within 2 hours,” “Maximum 3 attempts”
- **Named entities:** “Gmail inbox,” “HubSpot CRM,” “Slack #general”
- **Boolean logic:** “If status equals ‘pending’ AND priority equals ‘high’”

Avoid:

- **Ambiguous terms:** “Soon,” “Appropriate,” “Handle”
- **Subjective judgments:** “If it looks good,” “When ready”
- **Implicit knowledge:** Assumptions about “normal” procedures
- **Idioms or slang:** “Touch base,” “Circle back”

The Context Boundary Problem

One challenge with AI SOPs is defining the scope of context. AI systems may have access to vast amounts of data but need explicit guidance on what’s relevant to the current task.

Solution: Always specify the data sources the AI should reference: - “Check the customer’s last 3 support tickets in Zendesk” - “Reference the pricing sheet in the shared Drive folder” - “Use the email template from the Templates document”

Testing Your SOPs

Before deploying an SOP to a live AI system, validate it:

1. **Walkthrough test:** Have a human follow the SOP exactly as written
2. **Edge case review:** Identify unusual scenarios and verify they’re covered
3. **Tool integration check:** Confirm all referenced systems are accessible
4. **Output verification:** Ensure expected outputs are clearly defined

Remember: An AI will execute your SOP literally. If there’s a gap or ambiguity, the AI will either make an unpredictable choice or fail entirely.

The Iteration Imperative

Your first version of an SOP won't be perfect. Build in feedback loops: - Review AI execution logs regularly - Note where the AI deviated or struggled - Update the SOP with clarifications - Version your changes for tracking

SOPs are living documents that improve with use.

Chapter 2: The SOP Template

This chapter presents the master template for creating AI-compatible Standard Operating Procedures. Use this structure for every SOP you create, filling in each section with specific details for your workflow.



The Master Template

```

# SOP: [TASK NAME]

## Metadata
- **SOP ID**: [Unique identifier, e.g., SOP-001]
- **Version**: [Version number and date]
- **Owner**: [Person responsible for maintenance]
- **AI System**: [Which AI tool executes this SOP]
- **Last Updated**: [Date of last revision]

## Purpose
[Clear, one-paragraph description of what this SOP accomplishes and why it exists]

## Trigger
[Precise conditions that activate this SOP]

**Trigger Type**: [Time-based / Event-based / Manual]
**Trigger Details**: [Specific criteria]

## Prerequisites
- [Data, access, or conditions required before execution]
- [System authentication requirements]
- [Required templates or resources]

## Tools & Systems
| Tool | Purpose | Access Required |
|-----|-----|-----|
| [Tool Name] | [What it's used for] | [Permission level] |

## Procedure

### Step 1: [Action Name]
**Action**: [Specific action to take]
**Input**: [Required data or context]
**Process**: [Detailed description of what to do]
**Output**: [What this step produces]
**Time Limit**: [Maximum time for this step]

### Step 2: [Action Name]
[Continue pattern...]

## Decision Points

### Decision: [Decision Name]
**Condition**: [What to evaluate]
**IF** [condition]: [Action to take]
**ELSE IF** [condition]: [Alternative action]
**ELSE**: [Default action]

## Expected Output
- [Specific deliverable 1]
- [Specific deliverable 2]
- [Data updates made]
- [Notifications sent]

## Quality Check

```

```

- [Verification step 1]
- [Verification step 2]
- [Success criteria]

## Error Handling

### Error: [Error Scenario]
**Symptom**: [How to recognize this error]
**Retry**: [Number of retries and interval]
**Escalation**: [When and how to notify humans]
**Documentation**: [What to log about the failure]

## Security & Compliance
- [Data handling requirements]
- [Privacy considerations]
- [Audit trail requirements]

## Change Log
| Version | Date | Changes | Author |
|-----|-----|-----|-----|
| 1.0 | [Date] | Initial creation | [Name] |

```

Template Section Deep Dive

Metadata

The metadata section provides essential tracking information. The SOP ID creates a unique reference that can be used in logging, error messages, and cross-references between procedures.

Best Practice: Use a consistent naming convention like [CATEGORY]-[NUMBER] (e.g., SALES-001, SUPPORT-015).

Purpose

Write the purpose statement for a new team member who knows nothing about your operation. Explain the business value and the problem this SOP solves.

Example: “This SOP ensures all inbound leads receive a personalized response within 15 minutes during business hours, improving conversion rates by demonstrating responsiveness and capturing interest while it’s high.”

Trigger

The trigger is critical—it’s where automation begins. Be obsessively specific:

Vague: “When a new lead comes in”

Specific: “When a new row is added to the ‘Leads’ table in Airtable with ‘Source’ equal to ‘Website Contact Form’ and ‘Status’ equal to ‘New’”

Prerequisites

List everything that must be true before the SOP can execute successfully. This prevents partial execution and mysterious failures.

Common prerequisites include: - API authentication tokens being valid - Required templates being accessible - Reference data being current - Human approval for certain thresholds

Tools & Systems

The tools table serves multiple purposes: 1. **Documentation:** Clear record of what’s used 2. **Troubleshooting:** Quick reference when integrations fail 3. **Security audit:** Overview of system access 4. **Onboarding:** Guide for setting up new AI systems

Procedure Steps

This is the heart of the SOP. Each step should be: - **Atomic:** One clear action per step - **Measurable:** Observable completion criteria - **Time-boxed:** Maximum duration specified - **Independent:** Minimal dependency on other steps

Step Structure Best Practice:

```
### Step X: [Verb] [Object]
**Action**: [What to do]
**Input**: [Required information]
**Process**:
1. [Sub-step if needed]
2. [Sub-step if needed]
**Output**: [Expected result]
**Validation**: [How to confirm success]
**Time Limit**: [Maximum duration]
```

Decision Points

Complex workflows require branching logic. Present decisions as clear binary or enumerated choices:

```

### Decision: Lead Qualification
**Evaluate**: Lead score and company size

**IF** lead_score >= 80 AND company_size > 50:
  - Route to Enterprise Sales team
  - Mark priority as "High"
  - Schedule demo within 24 hours

**ELSE IF** lead_score >= 60:
  - Route to Standard Sales team
  - Mark priority as "Medium"
  - Send nurture sequence

**ELSE**:
  - Add to long-term nurture campaign
  - Mark priority as "Low"
  - Re-evaluate in 90 days

```

Expected Output

Define success in concrete terms. What should exist after this SOP runs that didn't exist before?

Include: - Records created or updated - Communications sent - Files generated - Notifications triggered - Status changes made

Quality Check

Build verification into the process. Quality checks catch errors before they propagate:

- **Format verification:** Is the output in the right structure?
- **Content verification:** Does it contain required elements?
- **Logic verification:** Are decision outcomes sensible?
- **Integration verification:** Did all systems update correctly?

Error Handling

Every SOP needs a contingency plan. For each potential failure point:

1. **Identify:** What could go wrong?
2. **Detect:** How will the system know it failed?
3. **Respond:** What should happen next?
4. **Escalate:** When does a human need to intervene?
5. **Document:** What gets logged for post-mortem?

Example Error Block:

```
### Error: API Timeout
**Symptom**: No response from CRM API within 30 seconds
**Retry**: 3 attempts with 10-second intervals
**Escalation**: If all retries fail, send Slack alert to #tech-support
**Fallback**: Queue record for manual processing
**Documentation**: Log incident ID, timestamp, and record details
```

Security & Compliance

Document any special handling requirements: - Personal data protection (GDPR, CCPA) - Industry regulations (HIPAA, SOX) - Company policies (data retention, approval workflows) - Access controls and authentication

Change Log

Maintain a history of modifications: - What changed and why - Who made the change - When it was implemented - Impact on related SOPs

Chapter 3: 10 SOP Examples

This chapter provides complete, production-ready SOPs for common business workflows. Use these as starting points for your own implementations, adapting the specifics to your tools and requirements.

SOP 1: Lead Response Automation

Metadata

- **SOP ID:** SALES-001
- **Version:** 1.0
- **Owner:** Sales Operations Manager
- **AI System:** CRM Automation Bot
- **Last Updated:** 2025-01-15

Purpose

Respond to all inbound leads within 15 minutes of form submission with a personalized acknowledgment that sets expectations and begins qualification.

Trigger

Trigger Type: Event-based **Trigger Details:** New record added to “Leads” table in HubSpot CRM with Source = “Website Contact Form”

Prerequisites

- HubSpot API access token valid
- Email template library accessible
- Sales team Slack channel active
- Lead scoring model loaded

Tools & Systems

Tool	Purpose	Access Required
HubSpot CRM	Lead data source	Read/Write
Gmail	Email delivery	Send access
Slack	Internal notifications	Post to #sales
Clearbit	Company enrichment	API access

Procedure

Step 1: Retrieve Lead Data

Action: Fetch complete lead record **Input:** Lead ID from trigger **Process:** Query HubSpot for all contact properties, form submission data, and page history **Output:** Complete lead profile JSON
Time Limit: 5 seconds

Step 2: Enrich Company Data

Action: Lookup company information **Input:** Lead's company domain **Process:** Call Clearbit API for company size, industry, and technographics **Output:** Enrichment data appended to lead record **Time Limit:** 10 seconds

Step 3: Calculate Lead Score

Action: Apply scoring algorithm **Input:** Lead attributes and enrichment data **Process:** - Check job title against target list (+20 if match) - Add company size score (+10 per 100 employees) - Check for budget keywords in message (+15) - Verify work email domain (+5) **Output:** Numeric score (0-100)
Time Limit: 2 seconds

Step 4: Generate Personalized Response

Action: Compose acknowledgment email **Input:** Lead data, score, and enrichment **Process:** - Select template based on score (high/medium/low) - Personalize greeting with first name - Reference specific inquiry from form - Include relevant case study based on industry **Output:** Complete email draft **Time Limit:** 5 seconds

Step 5: Send Email

Action: Deliver response **Input:** Composed email, lead email address **Process:** - Send via authenticated Gmail - BCC sales tracking address - Set "From" to assigned sales rep if available

Output: Email sent confirmation with message ID **Time Limit:** 10 seconds

Step 6: Update CRM

Action: Record interaction **Input:** Lead ID, email sent data, score **Process:** - Update lead status to "Contacted" - Log email activity - Set follow-up date (+3 business days) - Tag with lead score **Output:** Updated CRM record **Time Limit:** 5 seconds

Step 7: Notify Sales Team

Action: Alert relevant personnel **Input:** Lead data and score **Process:** - Post to Slack #sales channel - Tag assigned rep if known - Include lead score and key details **Output:** Slack notification sent **Time Limit:** 5 seconds

Decision Points**Decision: Response Template Selection**

Condition: Calculated lead score

IF score $\geq$ 75: - Use "High-Priority Lead" template - Request immediate manual follow-up - Schedule executive introduction

ELSE IF score $\geq$ 50: - Use "Standard Lead" template - Include calendar booking link - Add to sales sequence

ELSE: - Use "Nurture Lead" template - Add to educational email sequence - Review in 30 days

Expected Output

- Personalized email sent to lead within 15 minutes
- CRM updated with lead score and status
- Sales team notified via Slack
- Follow-up scheduled in 3 business days
- Activity logged in contact timeline

Quality Check

- Verify email personalization includes correct name
- Confirm lead score calculation is reasonable
- Check that calendar link works (if included)
- Validate email sent from correct domain

Error Handling

Error: Enrichment API Failure

Symptom: Clearbit returns 404 or timeout **Retry:** 2 attempts with 5-second delay **Escalation:** Continue without enrichment; log incident **Fallback:** Use basic lead data only

Error: Email Bounce

Symptom: Gmail returns bounce notification **Retry:** None **Escalation:** Alert sales team to check email validity **Fallback:** Update CRM with invalid email flag

Error: CRM Update Failure

Symptom: HubSpot API returns error **Retry:** 3 attempts with exponential backoff **Escalation:** Send urgent alert to #tech-support **Fallback:** Queue update for manual processing

Security & Compliance

- No sensitive data in Slack notifications
- Email opt-in status verified before sending
- GDPR compliance: Include unsubscribe link
- Data retention: Log emails for 7 years

SOP 2: Content Creation Workflow

Metadata

- **SOP ID:** MKTG-001
- **Version:** 1.0

- **Owner:** Content Marketing Manager
- **AI System:** Content Assistant
- **Last Updated:** 2025-01-15

Purpose

Transform content briefs into draft blog posts, social media snippets, and email newsletters following brand guidelines and SEO requirements.

Trigger

Trigger Type: Manual **Trigger Details:** Content brief moved to “Ready for Creation” status in Notion

Prerequisites

- Content brief approved and complete
- Brand voice guidelines accessible
- SEO keywords provided
- Target publish date confirmed

Tools & Systems

Tool	Purpose	Access Required
Notion	Content brief and drafts	Read/Write
Google Docs	Draft creation	Edit access
SEMrush	SEO optimization	API access
Grammarly	Quality check	Premium access

Procedure

Step 1: Review Content Brief

Action: Extract requirements **Input:** Notion page ID **Process:** - Read brief document - Identify target audience - Extract primary and secondary keywords - Note required CTAs and links - Check content type (blog, newsletter, social) **Output:** Structured content requirements **Time Limit:** 2 minutes

Step 2: Research and Outline

Action: Create content structure **Input:** Topic and keywords **Process:** - Generate SEO-optimized outline - Structure H2 and H3 headers - Plan keyword placement - Identify internal link opportunities
Output: Content outline with word count targets **Time Limit:** 5 minutes

Step 3: Draft Content

Action: Write initial draft **Input:** Outline and brand guidelines **Process:** - Write introduction with hook - Develop each section per outline - Include required examples and data - Add CTAs at appropriate points **Output:** Complete first draft **Time Limit:** 20 minutes

Step 4: SEO Optimization

Action: Optimize for search **Input:** Draft and keyword targets **Process:** - Verify keyword density (1-2%) - Optimize meta description - Add alt text for image placeholders - Check readability score (target: 8th grade) **Output:** SEO-optimized draft **Time Limit:** 5 minutes

Step 5: Brand Voice Check

Action: Align with brand guidelines **Input:** Draft and voice guide **Process:** - Check tone consistency - Verify terminology usage - Ensure CTA language matches brand - Confirm examples are appropriate **Output:** Brand-aligned draft **Time Limit:** 3 minutes

Step 6: Generate Supporting Assets

Action: Create meta content **Input:** Final draft **Process:** - Write social media snippets (3 variations) - Create email newsletter version - Generate excerpt for homepage - Write LinkedIn post summary **Output:** Asset package **Time Limit:** 5 minutes

Step 7: Submit for Review

Action: Queue for approval **Input:** Draft and assets **Process:** - Save to Google Docs - Update Notion status to "Review Pending" - Tag assigned editor - Add to editorial calendar **Output:** Submission confirmation **Time Limit:** 2 minutes

Decision Points

Decision: Content Length

Condition: Content type from brief

IF type = "Thought Leadership": - Target: 2,000-3,000 words - Include original research - Add expert quotes

ELSE IF type = "How-To Guide": - Target: 1,500-2,500 words - Include step-by-step instructions - Add screenshots/diagrams

ELSE IF type = "Newsletter": - Target: 500-800 words - Multiple short sections - Strong CTAs

ELSE: - Target: 1,000-1,500 words - Standard blog format

Expected Output

- Complete draft in Google Docs
- SEO meta description
- 3 social media snippets
- Email newsletter version
- Editorial calendar entry
- Review assignment

Quality Check

- Word count within target range
- Keyword density appropriate
- No grammar errors (Grammarly score 90+)
- All required links included
- CTA placement optimal

Error Handling

Error: Brief Incomplete

Symptom: Missing required fields **Escalation:** Return to brief creator with specific questions

Fallback: Pause until brief is updated

Error: Keyword Conflict

Symptom: Keywords don't match content topic **Escalation:** Flag for content manager review

Fallback: Proceed with closest semantic match

SOP 3: Follow-Up Sequence Automation

Metadata

- **SOP ID:** SALES-002
- **Version:** 1.0
- **Owner:** Sales Operations
- **AI System:** Sequence Bot
- **Last Updated:** 2025-01-15

Purpose

Execute multi-touch follow-up sequences for prospects who haven't responded to initial outreach, with intelligent timing and personalization.

Trigger

Trigger Type: Time-based **Trigger Details:** Daily at 9 AM, check for prospects with no response after 48 hours

Prerequisites

- Prospect in "Attempting Contact" status
- No response received in 48+ hours
- Sequence not already active
- Email deliverability confirmed

Tools & Systems

Tool	Purpose	Access Required
HubSpot	Prospect data	Read/Write
Apollo	Email sequencing	API access
LinkedIn	Social touch	Premium account
Sales Navigator	Prospect research	Access

Procedure

Step 1: Identify Stalled Prospects

Action: Query database **Input:** Current date **Process:** - Find contacts with last email sent > 48 hours ago - Exclude those with responses or meetings booked - Filter by sequence eligibility - Sort by priority score **Output:** List of prospects requiring follow-up **Time Limit:** 5 minutes

Step 2: Research Latest Activity

Action: Gather intelligence **Input:** Prospect list **Process:** - Check LinkedIn for recent posts - Look for company news or funding - Review previous interaction history - Check for mutual connections **Output:** Research notes per prospect **Time Limit:** 2 minutes per prospect

Step 3: Select Sequence Type

Action: Determine approach **Input:** Prospect data and research **Process:** - Analyze prospect seniority - Check industry vertical - Review previous engagement - Match to sequence template **Output:** Assigned sequence type **Time Limit:** 1 minute per prospect

Step 4: Personalize Touch 1

Action: Create first follow-up **Input:** Sequence template and research **Process:** - Reference previous email - Add relevant insight from research - Include new value proposition - Update subject line **Output:** Personalized email draft **Time Limit:** 3 minutes per prospect

Step 5: Schedule Sequence

Action: Set up automation **Input:** Email draft and sequence rules **Process:** - Configure send timing (Day 3, Day 7, Day 14) - Add LinkedIn touch points - Set task reminders for manual steps - Activate sequence **Output:** Active sequence confirmation **Time Limit:** 2 minutes per prospect

Step 6: Update CRM

Action: Record sequence start **Input:** Prospect IDs and sequence details **Process:** - Update contact status to "In Sequence" - Log sequence type and start date - Set expected end date - Add activity note **Output:** Updated CRM records **Time Limit:** 1 minute per prospect

Decision Points

Decision: Sequence Type Selection

Condition: Prospect characteristics

IF seniority = "C-Level" AND company_size > 500: - Use "Executive Premium" sequence - 5 touches over 30 days - Include video messages

ELSE IF industry = "Enterprise SaaS": - Use "Tech Buyer" sequence - Technical case studies - ROI calculator offer

ELSE IF previous_engagement = "Opened Email": - Use "Engaged Nurture" sequence - Shorter intervals - Soft CTAs

ELSE: - Use "Standard Follow-Up" sequence - 4 touches over 21 days - Mix of email and LinkedIn

Decision: Escalation Trigger

Condition: Sequence progress

IF sequence completed AND no response: - Move to "Long-term Nurture" - Check back in 90 days - Add to newsletter only

ELSE IF response received: - Exit sequence - Alert assigned sales rep - Schedule immediate follow-up

Expected Output

- Personalized follow-up emails scheduled
- LinkedIn touch points queued
- CRM updated with sequence status
- Sales rep notifications sent
- Task reminders created

Quality Check

- Each email contains personalization
- No duplicate sequences active
- Timing follows best practices
- Unsubscribe links present

Error Handling

Error: Email Blacklist

Symptom: Bounce or spam folder **Escalation:** Remove from sequence **Fallback:** Try LinkedIn-only sequence

Error: Prospect Changed Jobs

Symptom: Out-of-office indicates departure **Escalation:** Alert sales team **Fallback:** Find new contact at company

SOP 4: Calendar Booking Automation

Metadata

- **SOP ID:** OPS-001
- **Version:** 1.0
- **Owner:** Operations Manager
- **AI System:** Scheduling Assistant
- **Last Updated:** 2025-01-15

Purpose

Process meeting requests, find optimal times, send calendar invites, and prepare pre-meeting briefs for all scheduled calls.

Trigger

Trigger Type: Event-based **Trigger Details:** Email received with keywords “meet,” “schedule,” “calendar,” or “book” OR Calendly booking completed

Prerequisites

- Calendar access (Google/Outlook)
- Meeting type definitions configured
- Buffer time rules established

- Time zone database current

Tools & Systems

Tool	Purpose	Access Required
Google Calendar	Primary scheduling	Read/Write
Calendly	Self-service booking	Admin access
Zoom	Meeting creation	API access
CRM	Contact lookup	Read access

Procedure

Step 1: Parse Request

Action: Extract meeting details **Input:** Email content or Calendly webhook **Process:** - Identify meeting type (intro, demo, follow-up) - Extract preferred times if provided - Determine attendees from email - Note special requirements **Output:** Structured meeting request **Time Limit:** 30 seconds

Step 2: Check Availability

Action: Find open slots **Input:** Requested time range, attendee emails **Process:** - Query primary calendar - Check for conflicts across all attendees - Apply buffer rules (15 min before/after) - Consider time zones **Output:** Available time slots **Time Limit:** 10 seconds

Step 3: Determine Meeting Type

Action: Classify meeting **Input:** Parsed request **Process:** - Match to defined meeting types - Set duration (15/30/45/60 minutes) - Assign meeting template - Select Zoom settings **Output:** Meeting configuration **Time Limit:** 5 seconds

Step 4: Create Calendar Event

Action: Book the meeting **Input:** Time slot, attendees, configuration **Process:** - Create Google Calendar event - Generate Zoom link - Add meeting description template - Include agenda placeholder **Output:** Calendar event created **Time Limit:** 10 seconds

Step 5: Send Invitations

Action: Notify attendees **Input:** Calendar event details **Process:** - Send calendar invites - Include Zoom link and agenda - Add to CRM activity timeline - Send confirmation email **Output:** Invitations sent **Time Limit:** 10 seconds

Step 6: Prepare Meeting Brief

Action: Generate context document **Input:** Attendee information **Process:** - Pull CRM data on attendees - Review previous meeting notes - Check company recent news - Prepare agenda items **Output:** Pre-meeting brief document **Time Limit:** 2 minutes

Step 7: Schedule Reminders

Action: Set up notifications **Input:** Meeting time and attendees **Process:** - 24-hour reminder email - 1-hour Slack notification - Pre-meeting prep task **Output:** Reminders scheduled **Time Limit:** 5 seconds

Decision Points

Decision: Meeting Type

Condition: Keywords and context

IF keywords include “demo” OR “show”: - Type: Product Demo - Duration: 45 minutes - Include: Screen share enabled - Agenda: Discovery + Demo

ELSE IF keywords include “intro” OR “call”: - Type: Intro Call - Duration: 30 minutes - Include: Camera preferred note - Agenda: Get to know each other

ELSE IF keywords include “follow” OR “update”: - Type: Follow-up - Duration: 30 minutes - Include: Previous meeting notes - Agenda: Progress review

ELSE: - Type: General Meeting - Duration: 30 minutes - Standard settings

Decision: Conflict Resolution

Condition: Availability check results

IF no conflicts: - Proceed with booking - Send confirmation

ELSE IF minor conflict (buffer only): - Book with adjusted buffer - Note in confirmation

ELSE IF major conflict: - Send alternative times - Request selection - Hold tentative slot

Expected Output

- Calendar event created with all details
- Zoom meeting generated
- Invitations sent to all attendees
- Pre-meeting brief created
- Reminders scheduled
- CRM activity logged

Quality Check

- Time zones converted correctly
- No double bookings
- Zoom link tested
- Attendee list complete
- Agenda included

Error Handling

Error: No Available Slots

Symptom: Zero availability in requested range **Escalation:** Send alternative time options **Fallback:** Request priority override

Error: Calendar Sync Failure

Symptom: Event not appearing **Retry:** 2 attempts **Escalation:** Manual calendar check

SOP 5: Invoice Sending Automation

Metadata

- **SOP ID:** FIN-001
- **Version:** 1.0
- **Owner:** Finance Manager
- **AI System:** Billing Assistant

• **Last Updated:** 2025-01-15

Purpose

Generate, send, and track invoices for completed work, with automatic follow-up for overdue payments.

Trigger

Trigger Type: Time-based **Trigger Details:** Daily at 8 AM; Also triggered by project status change to “Completed”

Prerequisites

- Project marked complete in system
- Invoice template approved
- Client billing information verified
- Payment terms defined

Tools & Systems

Tool	Purpose	Access Required
QuickBooks	Invoice generation	Full access
Stripe	Payment processing	API access
Gmail	Invoice delivery	Send access
Airtable	Project tracking	Read/Write

Procedure

Step 1: Identify Billable Projects

Action: Find completed work **Input:** Current date, project database **Process:** - Query Airtable for “Completed” projects - Filter for uninvoiced status - Verify deliverables accepted - Check contract terms **Output:** List of projects to invoice **Time Limit:** 2 minutes

Step 2: Calculate Invoice Amount

Action: Determine billing **Input:** Project details **Process:** - Sum approved time entries - Add expenses with receipts - Apply contract rates - Calculate taxes if applicable **Output:** Invoice total and line items **Time Limit:** 1 minute per invoice

Step 3: Generate Invoice

Action: Create invoice document **Input:** Calculated totals, client data **Process:** - Create invoice in QuickBooks - Apply correct template - Add line item descriptions - Set due date (Net 30) **Output:** Draft invoice **Time Limit:** 30 seconds per invoice

Step 4: Review Threshold Check

Action: Flag large invoices **Input:** Invoice amount **Process:** - Compare to threshold (\$5,000) - If over, mark for manager review - If under, auto-approve **Output:** Approval status **Time Limit:** 5 seconds

Step 5: Send Invoice

Action: Deliver to client **Input:** Approved invoice **Process:** - Email PDF to client - CC internal finance team - Include payment instructions - Attach Stripe payment link **Output:** Invoice sent confirmation **Time Limit:** 10 seconds per invoice

Step 6: Update Records

Action: Log invoice **Input:** Sent invoice details **Process:** - Update Airtable with invoice number - Change project status to "Invoiced" - Set payment due date - Create receivable entry **Output:** Updated records **Time Limit:** 10 seconds

Step 7: Schedule Follow-Up

Action: Plan reminders **Input:** Invoice due date **Process:** - Reminder at 7 days before due - Reminder at due date - Escalation at 7 days overdue - Final notice at 30 days overdue **Output:** Reminder sequence scheduled **Time Limit:** 5 seconds

Decision Points**Decision: Payment Terms**

Condition: Client contract

IF client_type = "Enterprise": - Terms: Net 45 - Require purchase order - Send to accounts payable

ELSE IF client_type = "Retainer": - Terms: Net 15 - Auto-charge on file - Invoice monthly

ELSE: - Terms: Net 30 - Standard process

Decision: Overdue Action

Condition: Days overdue

IF days = 7: - Send friendly reminder - Offer payment plan

ELSE IF days = 30: - Send formal notice - Apply late fee - Notify account manager

ELSE IF days = 60: - Escalate to collections - Pause new work - Executive notification

Expected Output

- Invoices generated and sent
- Records updated with invoice numbers
- Payment reminders scheduled
- Receivables tracked
- Client notified

Quality Check

- Invoice amounts match contracts
- Client details correct
- Payment links functional
- PDFs generated properly
- Due dates calculated correctly

Error Handling

Error: Client Payment Failed

Symptom: Stripe decline **Escalation:** Notify client of issue **Fallback:** Offer alternative payment method

Error: Invoice Discrepancy

Symptom: Client questions amount **Escalation:** Hold on follow-ups **Fallback:** Route to account manager

SOP 6: Client Onboarding Sequence

Metadata

- **SOP ID:** CS-001
- **Version:** 1.0
- **Owner:** Customer Success Manager
- **AI System:** Onboarding Coordinator
- **Last Updated:** 2025-01-15

Purpose

Guide new clients through a consistent onboarding experience, ensuring all setup steps are completed and expectations are aligned.

Trigger

Trigger Type: Event-based **Trigger Details:** Contract signed in DocuSign OR deal marked “Closed Won” in CRM

Prerequisites

- Signed contract received
- Primary contact identified
- Service tier confirmed
- Onboarding checklist template ready

Tools & Systems

Tool	Purpose	Access Required
HubSpot	Contact management	Read/Write
Notion	Onboarding docs	Read/Write
Typeform	Intake forms	Admin access
Slack	Team notifications	Post to #onboarding

Procedure

Step 1: Create Onboarding Record

Action: Initialize client setup **Input:** Contract details, CRM data **Process:** - Create Notion onboarding page - Copy checklist template - Set service tier parameters - Assign onboarding specialist **Output:** Onboarding workspace created **Time Limit:** 2 minutes

Step 2: Send Welcome Package

Action: Deliver initial materials **Input:** Client contact info **Process:** - Send welcome email - Include onboarding timeline - Attach getting-started guide - Provide onboarding portal link **Output:** Welcome package delivered **Time Limit:** 5 minutes

Step 3: Send Intake Form

Action: Gather requirements **Input:** Client record **Process:** - Customize Typeform for service tier - Include business goals questions - Add technical requirements section - Request brand assets **Output:** Intake form sent **Time Limit:** 2 minutes

Step 4: Schedule Kickoff Call

Action: Book initial meeting **Input:** Client timezone, preferences **Process:** - Send Calendly link - Request stakeholder list - Block 60 minutes - Prepare kickoff agenda **Output:** Kickoff scheduled **Time Limit:** 3 minutes

Step 5: Prepare Kickoff Brief

Action: Generate meeting prep **Input:** Intake form responses **Process:** - Compile client background - Review submitted materials - Create discussion topics list - Draft project timeline **Output:** Kickoff brief document **Time Limit:** 10 minutes

Step 6: Setup Project Workspace

Action: Create collaboration space **Input:** Service type, team members **Process:** - Create shared Drive folder - Setup project in PM tool - Add client to communication channels - Share relevant templates **Output:** Workspace ready **Time Limit:** 5 minutes

Step 7: Send Pre-Kickoff Reminder

Action: Confirm attendance **Input:** Scheduled meeting time **Process:** - Send 24-hour reminder - Include kickoff agenda - Request any missing materials - Share meeting link **Output:** Reminder sent **Time Limit:** 2 minutes

Step 8: Post-Kickoff Action Items

Action: Document outcomes **Input:** Kickoff meeting notes **Process:** - Send recap email - List confirmed next steps - Assign internal tasks - Update onboarding checklist **Output:** Follow-up sent **Time Limit:** 10 minutes

Decision Points**Decision: Service Tier Path**

Condition: Contract value/services

IF tier = "Enterprise": - Assign dedicated CSM - Weekly check-ins scheduled - Custom setup required - Executive sponsor assigned

ELSE IF tier = "Professional": - Shared CSM - Bi-weekly check-ins - Standard setup - Account manager assigned

ELSE: - Self-service onboarding - Monthly check-ins - Template setup - Support team assigned

Decision: Escalation Needs

Condition: Onboarding progress

IF intake_form not complete after 5 days: - Send follow-up - Offer assistance - Alert CSM

ELSE IF kickoff missed: - Reschedule immediately - Check for blockers - Notify sales rep

ELSE IF no activity after 14 days: - Executive outreach - Risk flag raised - Retention team alerted

Expected Output

- Onboarding workspace created
- Welcome email sent
- Intake form delivered
- Kickoff call scheduled
- Project workspace ready
- All stakeholders notified

Quality Check

- All template fields populated
- Correct service tier applied
- Intake form customized
- Meeting scheduled in correct timezone
- Access permissions verified

Error Handling

Error: Client Unresponsive

Symptom: No reply after 3 attempts **Escalation:** Sales rep notification **Fallback:** Pause sequence, manual review

Error: Incomplete Intake

Symptom: Form submitted with gaps **Escalation:** Request clarification **Fallback:** Proceed with available info

SOP 7: Social Media Posting Workflow

Metadata

- **SOP ID:** MKTG-002
- **Version:** 1.0
- **Owner:** Social Media Manager
- **AI System:** Social Media Assistant
- **Last Updated:** 2025-01-15

Purpose

Schedule, publish, and monitor social media content across platforms with consistent branding and optimal timing.

Trigger

Trigger Type: Time-based **Trigger Details:** Daily at scheduled post times (9 AM, 1 PM, 4 PM EST); Also triggered by “approve” status in content calendar

Prerequisites

- Content approved in calendar
- Creative assets finalized
- Platform accounts authenticated
- Hashtag sets updated

Tools & Systems

Tool	Purpose	Access Required
Buffer	Scheduling	Full access
Canva	Creative assets	Brand kit access
Notion	Content calendar	Read/Write
Brand24	Monitoring	API access

Procedure

Step 1: Review Approved Content

Action: Check calendar **Input:** Current date **Process:** - Query Notion for approved posts - Filter by scheduled time - Verify assets attached - Check platform assignments **Output:** List of posts to publish **Time Limit:** 5 minutes

Step 2: Verify Assets

Action: Check creative **Input:** Post requirements **Process:** - Confirm image/video files ready - Verify dimensions per platform - Check branding elements - Test alt text availability **Output:** Asset status confirmed **Time Limit:** 2 minutes per post

Step 3: Optimize for Platform

Action: Format content **Input:** Master post content **Process:** - Adjust character counts per platform - Modify hashtags for each network - Resize images if needed - Adapt tone if necessary **Output:** Platform-specific versions **Time Limit:** 3 minutes per post

Step 4: Schedule Posts

Action: Queue for publishing **Input:** Formatted content **Process:** - Upload to Buffer - Set exact publish time - Add to appropriate queue - Enable engagement tracking **Output:** Posts scheduled **Time Limit:** 2 minutes per post

Step 5: Publish Immediate Content

Action: Post real-time content **Input:** Time-sensitive posts **Process:** - Verify approval status - Post to live accounts - Capture post URLs - Save screenshots **Output:** Content published **Time Limit:** 1 minute per post

Step 6: Monitor Initial Engagement

Action: Watch for activity **Input:** Published post URLs **Process:** - Check first 30 minutes - Note any errors - Flag high-engagement posts - Alert for negative comments **Output:** Initial monitoring complete **Time Limit:** Ongoing

Step 7: Update Calendar Status

Action: Mark as published **Input:** Post confirmation **Process:** - Update Notion status - Add published URLs - Note actual publish time - Move to “Published” view **Output:** Calendar updated **Time Limit:** 30 seconds per post

Decision Points

Decision: Platform Selection

Condition: Content type

IF content = “Industry Article”: - LinkedIn (primary) - Twitter (thread) - Facebook (optional)

ELSE IF content = “Visual/Creative”: - Instagram (primary) - Pinterest (if applicable) - LinkedIn (cross-post)

ELSE IF content = “Quick Update”: - Twitter (primary) - LinkedIn (cross-post)

Decision: Engagement Response

Condition: Comment type

IF comment = “Question”: - Draft answer - Route for approval if complex - Respond within 2 hours

ELSE IF comment = “Complaint”: - Flag for immediate review - Draft empathetic response - Escalate to CS team

ELSE IF comment = “Positive”: - Like and respond - Thank the user - Consider for testimonial

Expected Output

- Content published on schedule
- All platforms updated
- Engagement monitored
- Calendar statuses current
- URLs captured

Quality Check

- Links working
- Images displaying correctly

- Copy error-free
- Hashtags relevant
- Times correct

Error Handling

Error: Asset Missing

Symptom: Image not found **Escalation:** Alert content team **Fallback:** Use text-only version

Error: Post Failed

Symptom: API error **Retry:** 2 attempts **Escalation:** Manual posting

SOP 8: Email Sorting and Triage

Metadata

- **SOP ID:** OPS-002
- **Version:** 1.0
- **Owner:** Operations Manager
- **AI System:** Email Assistant
- **Last Updated:** 2025-01-15

Purpose

Automatically categorize, prioritize, and route incoming emails to appropriate team members with initial context.

Trigger

Trigger Type: Event-based **Trigger Details:** New email arrives in monitored inbox; Runs every 5 minutes

Prerequisites

- Gmail API access
- Categorization rules defined
- Team availability calendar
- Response templates library

Tools & Systems

Tool	Purpose	Access Required
Gmail	Email source	Full access
OpenAI	Categorization	API access
Slack	Routing	Post to channels
Airtable	Logging	Read/Write

Procedure

Step 1: Fetch New Emails

Action: Check inbox **Input:** Monitored email address **Process:** - Query Gmail API for unread - Limit to last 100 messages - Check for duplicates - Extract metadata **Output:** List of new emails **Time Limit:** 10 seconds

Step 2: Extract Content

Action: Parse email data **Input:** Raw email data **Process:** - Extract subject, body, sender - Check for attachments - Parse thread history - Identify original date **Output:** Structured email data **Time Limit:** 2 seconds per email

Step 3: Categorize Intent

Action: Classify email type **Input:** Email content **Process:** - Run through categorization model - Identify: Sales, Support, Partnership, Spam, etc. - Score confidence level - Check against whitelist/blacklist **Output:** Category and confidence score **Time Limit:** 5 seconds per email

Step 4: Assess Priority

Action: Determine urgency **Input:** Category + content analysis **Process:** - Check for urgency keywords - Review sender importance - Analyze sentiment - Check business hours timing **Output:** Priority level (High/Medium/Low) **Time Limit:** 3 seconds per email

Step 5: Route to Owner

Action: Assign to team member **Input:** Category and priority **Process:** - Map category to team member - Check availability status - Apply round-robin if shared - Note special handling rules **Output:** Assigned owner **Time Limit:** 2 seconds per email

Step 6: Draft Initial Response

Action: Create reply draft **Input:** Email content + templates **Process:** - Select appropriate template - Customize with email context - Add personalization - Include next steps **Output:** Response draft **Time Limit:** 10 seconds per email

Step 7: Send Notification

Action: Alert assignee **Input:** Assignment details **Process:** - Send Slack DM to owner - Include email summary - Attach response draft - Set SLA timer **Output:** Notification sent **Time Limit:** 2 seconds

Step 8: Update Email Status

Action: Mark and file **Input:** Processed email **Process:** - Apply label in Gmail - Mark as read - Archive if spam - Log to Airtable **Output:** Inbox updated **Time Limit:** 2 seconds per email

Decision Points**Decision: Category Assignment**

Condition: Email content analysis

IF keywords include "pricing," "demo," "interested": - Category: Sales Lead - Route to: Sales team - SLA: 1 hour

ELSE IF keywords include "help," "issue," "broken": - Category: Support - Route to: Support queue - SLA: 4 hours

ELSE IF keywords include "partner," "collaboration": - Category: Partnership - Route to: Business development - SLA: 24 hours

ELSE IF sender_domain = known_client: - Category: Client Communication - Route to: Account manager - SLA: 2 hours

ELSE: - Category: General - Route to: Operations - SLA: 24 hours

Decision: Auto-Response Eligibility

Condition: Category and confidence

IF category = "FAQ" AND confidence > 0.9: - Send auto-response - Include helpful links - Mark as resolved

ELSE IF category = "Out of Office": - File without notification - Update sender status

ELSE: - Require human response - Queue for review

Expected Output

- All emails categorized
- Priority levels assigned
- Team members notified
- Draft responses ready
- Inbox organized
- Activity logged

Quality Check

- Categorization accurate
- Routing correct
- Draft responses appropriate
- No emails missed
- SLA timers set

Error Handling

Error: Categorization Uncertain

Symptom: Confidence < 0.7 **Escalation:** Route to operations for manual triage **Fallback:** Default to "General" category

Error: No Owner Available

Symptom: All assignees out of office **Escalation:** Route to backup queue **Fallback:** Notify manager

SOP 9: Call Routing and Qualification

Metadata

- **SOP ID:** SALES-003
- **Version:** 1.0
- **Owner:** Sales Operations
- **AI System:** Voice Assistant
- **Last Updated:** 2025-01-15

Purpose

Answer inbound calls, qualify callers, route to appropriate departments, and capture lead information for follow-up.

Trigger

Trigger Type: Event-based **Trigger Details:** Incoming call to main business line

Prerequisites

- Phone system API access
- Team availability status
- Call routing rules configured
- CRM integration active

Tools & Systems

Tool	Purpose	Access Required
Twilio	Call handling	Full access
Airtable	Lead capture	Read/Write
Slack	Team alerts	Post access
Calendly	Meeting booking	API access

Procedure

Step 1: Answer Call

Action: Initial greeting **Input:** Incoming call data **Process:** - Answer within 3 rings - Play greeting message - Identify business and offering - Ask how to direct call **Output:** Caller engaged **Time Limit:** 15 seconds

Step 2: Capture Caller Info

Action: Gather identification **Input:** Caller's introduction **Process:** - Request name and company - Confirm pronunciation - Ask for callback number - Note call purpose **Output:** Basic caller profile **Time Limit:** 30 seconds

Step 3: Qualify Intent

Action: Understand needs **Input:** Caller's explanation **Process:** - Ask qualifying questions - Identify urgency level - Determine service interest - Check existing account status **Output:** Qualification data **Time Limit:** 2 minutes

Step 4: Check Existing Records

Action: Lookup caller **Input:** Name and company **Process:** - Search CRM for existing contact - Review previous interactions - Check open opportunities - Note account status **Output:** Contact history **Time Limit:** 5 seconds

Step 5: Route Call

Action: Connect to right person **Input:** Qualification + availability **Process:** - Check department availability - Review agent skills match - Place caller on hold briefly - Connect or offer callback
Output: Call transferred **Time Limit:** 30 seconds

Step 6: Log Call Details

Action: Record interaction **Input:** Call metadata **Process:** - Create CRM activity - Tag call source - Log qualification notes - Set follow-up if needed **Output:** Call logged **Time Limit:** 10 seconds

Step 7: Send Summary

Action: Alert recipient **Input:** Call details **Process:** - Send Slack notification - Include caller context - Attach CRM link - Note urgency level **Output:** Team notified **Time Limit:** 5 seconds

Decision Points**Decision: Routing Destination**

Condition: Caller qualification

IF intent = "Sales Inquiry" AND company_size > 100: - Route to: Enterprise sales - Priority: High - Warm transfer with context

ELSE IF intent = "Support Issue": - Route to: Support team - Priority: Based on severity - Create ticket number

ELSE IF intent = "Billing Question": - Route to: Billing department - Priority: Medium - Verify account first

ELSE IF no_agent_available: - Offer: Callback scheduling - Collect: Best time to call - Send: Calendly link via SMS

Decision: Escalation Protocol

Condition: Call characteristics

IF caller = "VIP Client": - Bypass queue - Alert account manager immediately - Priority override

ELSE IF issue = "Service Outage": - Escalate to on-call engineer - Create incident report - Notify operations team

ELSE IF legal_mention = true: - Route to legal/compliance - Record call (if not already) - Alert management

Expected Output

- Call answered promptly
- Caller qualified
- Routed to appropriate party
- Details logged in CRM
- Team notified
- Follow-up scheduled if needed

Quality Check

- Greeting professional
- Information captured accurately
- Routing appropriate
- Hold time minimal
- Notes complete

Error Handling

Error: All Lines Busy

Symptom: No agent available **Escalation:** Offer callback option **Fallback:** Send to voicemail with promise of callback

Error: CRM Lookup Failed

Symptom: Cannot find caller **Escalation:** Create new contact record **Fallback:** Treat as new lead

SOP 10: Data Backup and Verification

Metadata

- **SOP ID:** IT-001

- **Version:** 1.0
- **Owner:** IT Manager
- **AI System:** Backup Monitor
- **Last Updated:** 2025-01-15

Purpose

Execute daily automated backups of critical business data, verify backup integrity, and alert on failures or anomalies.

Trigger

Trigger Type: Time-based **Trigger Details:** Daily at 2 AM local time (low-traffic period)

Prerequisites

- Backup storage accessible
- API credentials valid
- Bandwidth available
- Previous backup completed

Tools & Systems

Tool	Purpose	Access Required
AWS S3	Backup storage	Write access
Airtable	Data source	Read access
Google Drive	File backup	API access
Datadog	Monitoring	API access

Procedure

Step 1: Pre-Backup Check

Action: Verify system status **Input:** Current time, system metrics **Process:** - Check storage capacity - Verify network stability - Confirm no active maintenance - Review previous backup logs **Output:** Go/No-go decision **Time Limit:** 2 minutes

Step 2: Database Backup

Action: Export critical databases **Input:** Database connection strings **Process:** - Export Airtable bases - Download as CSV/JSON - Compress data - Encrypt sensitive fields **Output:** Database archives **Time Limit:** 15 minutes

Step 3: File Backup

Action: Copy file assets **Input:** File storage locations **Process:** - Sync Google Drive folders - Copy shared documents - Archive email attachments - Backup configuration files **Output:** File archives **Time Limit:** 30 minutes

Step 4: Upload to Storage

Action: Transfer to cloud **Input:** Local archives **Process:** - Connect to AWS S3 - Upload with multipart transfer - Set storage class (IA/Glacier) - Verify upload completion **Output:** Cloud backups created **Time Limit:** 20 minutes

Step 5: Verify Integrity

Action: Check backup validity **Input:** Uploaded files **Process:** - Calculate checksums - Compare file sizes - Spot-check random samples - Verify encryption **Output:** Integrity report **Time Limit:** 10 minutes

Step 6: Test Restoration

Action: Validate recoverability **Input:** Recent backup sample **Process:** - Restore test file - Verify content accuracy - Check for corruption - Document restore time **Output:** Restoration verified **Time Limit:** 10 minutes

Step 7: Cleanup Old Backups

Action: Apply retention policy **Input:** Backup inventory **Process:** - List backups older than 30 days - Move to Glacier for cost savings - Delete backups past 1 year - Log retention actions **Output:** Storage optimized **Time Limit:** 5 minutes

Step 8: Generate Report

Action: Document status **Input:** All backup metrics **Process:** - Compile backup sizes - Calculate duration - Note any errors - Create status summary **Output:** Backup report **Time Limit:** 2 minutes

Decision Points

Decision: Backup Scope

Condition: Day of week

IF day = "Sunday": - Type: Full backup - Include: All data, historical - Duration: Up to 2 hours

ELSE: - Type: Incremental - Include: Changed data only - Duration: Up to 30 minutes

Decision: Failure Response

Condition: Error type

IF error = "Storage Full": - Immediate: Alert IT manager - Action: Cleanup old backups - Retry: After cleanup

ELSE IF error = "Network Timeout": - Immediate: Log error - Action: Retry 3 times - Escalate: If still failing

ELSE IF error = "Data Corruption": - Immediate: Stop backup - Action: Quarantine affected data - Escalate: Urgent notification

Expected Output

- Complete backup stored in S3
- Integrity verified
- Old backups archived
- Status report generated
- Team notified of success/failure

Quality Check

- All critical data backed up
- Checksums match
- Test restore successful
- No errors in logs
- Storage costs optimized

Error Handling

Error: Backup Incomplete

Symptom: Partial data transfer **Escalation:** Immediate alert **Fallback:** Retry from checkpoint

Error: Verification Failed

Symptom: Checksum mismatch **Escalation:** Quarantine backup **Fallback:** Use previous valid backup

Chapter 4: How to Train Your AI on SOPs

Creating excellent SOPs is only half the battle. The other half is training your AI systems to follow them effectively. This chapter covers the practical steps for onboarding AI assistants to your procedures.

The Training Process

Training AI on SOPs follows a structured approach:

Phase 1: Context Loading

Before your AI can execute an SOP, it needs the full context:

1. **SOP Ingestion:** Feed the complete SOP document to the AI
2. **Metadata Explanation:** Ensure the AI understands the metadata fields
3. **Tool Mapping:** Connect SOP tools to actual API integrations
4. **Test Environment:** Use sandbox systems for initial training

Implementation: Most modern AI platforms allow document upload or knowledge base creation. Upload your SOPs in markdown format and tag them appropriately.

Phase 2: Step-by-Step Walkthrough

Don't expect the AI to execute a full SOP immediately. Start with step-by-step training:

1. **Single Step Execution:** Have the AI execute just Step 1
2. **Validation:** Verify the output matches expectations
3. **Correction:** Provide feedback on any errors
4. **Repetition:** Run the same step multiple times with variations
5. **Progression:** Add Step 2, then Step 3, and so on

Example Training Dialog:

```
User: Execute Step 1 of SOP SALES-001 with test lead ID 12345
AI: [Executes data retrieval]
User: The data is missing the 'company_size' field. In this case,
      you should check the enrichment data.
AI: [Adjusts approach]
User: Good. Now try again with a different lead ID.
```

Phase 3: Decision Tree Training

Decision points are where AI systems often struggle. Create explicit training scenarios:

1. **Boundary Cases:** Test exactly at threshold values
2. **Ambiguous Inputs:** Provide unclear data and see how the AI handles it
3. **Error Scenarios:** Simulate failure conditions
4. **Multi-Condition:** Test complex IF/ELSE chains

Training Exercise Template:

```
Scenario: Lead scoring decision
Input: Lead with score = 74 (threshold is 75)
Expected: Route to Standard Sales (not Enterprise)
Rationale: Scores must MEET threshold, not approach it
```

Phase 4: Tool Integration Testing

AI SOPs often fail at the integration points. Rigorous testing includes:

1. **API Authentication:** Verify tokens and credentials
2. **Rate Limiting:** Test behavior when APIs throttle
3. **Data Formatting:** Confirm inputs match API expectations
4. **Error Responses:** Validate handling of 4xx/5xx responses
5. **Webhook Reliability:** Test event-driven triggers

Integration Checklist: - [] All API credentials validated - [] Sandbox environment configured - [] Rate limits documented - [] Error codes mapped to SOP responses - [] Webhook endpoints tested and secured

Phase 5: Full Workflow Testing

Once individual components work, test complete workflows:

1. **Happy Path:** Standard execution with ideal inputs

2. **Common Variations:** Typical edge cases and alternatives
3. **Stress Testing:** High volume or rapid sequential execution
4. **Failure Recovery:** Mid-workflow error and continuation

Training Methods by AI Platform

Different AI systems require different training approaches:

Custom GPTs (ChatGPT)

- Upload SOPs to Knowledge section
- Create custom instructions referencing SOP format
- Use conversation examples to demonstrate execution
- Enable Code Interpreter for data processing steps

Claude Projects

- Add SOP documents to project knowledge
- Create system prompt emphasizing SOP structure
- Use artifacts for multi-step workflows
- Leverage tool use for integrations

Microsoft Copilot

- Store SOPs in SharePoint/OneDrive
- Create Copilot agents with SOP grounding
- Use Power Automate for integration steps
- Configure action plugins for tool connections

Custom AI Implementations

- Fine-tune on SOP corpus
- Implement RAG (Retrieval-Augmented Generation)
- Create structured output schemas
- Build evaluation pipelines

Prompt Engineering for SOP Execution

The prompts you use to trigger SOP execution matter significantly:

Bad Prompt:

“Handle this lead”

Good Prompt:

“Execute SOP SALES-001 (Lead Response Automation) for lead ID 78910. Follow all steps through completion. If any error occurs, apply the Error Handling section exactly as documented. Report completion status and any outputs generated.”

Effective SOP Execution Prompts Include:

- **SOP ID:** Explicit reference to procedure
- **Input data:** Complete context
- **Completion criteria:** How to know when done
- **Error protocol:** How to handle failures
- **Output format:** Expected structure

Monitoring AI SOP Execution

Once trained, continuous monitoring ensures ongoing accuracy:

Logging Requirements

Every SOP execution should log: - SOP ID and version - Start and end timestamps - Input parameters - Step-by-step execution trace - Outputs generated - Errors encountered - Final status

Quality Metrics

Track these KPIs: - **Success Rate:** % of executions completing without error - **Accuracy Score:** % of outputs meeting quality standards - **Time Variance:** Actual vs. expected execution time - **Escalation Rate:** How often humans need to intervene

Feedback Loops

Create mechanisms for human feedback: 1. **Output Review**: Spot-check AI results 2. **Error Reporting**: Easy way to flag mistakes 3. **Suggestion Collection**: Capture improvement ideas 4. **Regular Audits**: Monthly review of execution logs

Common Training Pitfalls

1. Insufficient Context

Problem: AI lacks access to referenced templates or historical data **Solution**: Include all dependencies in the training environment

2. Over-Generalization

Problem: AI applies patterns from one SOP to inappropriate situations **Solution**: Train on explicit decision boundaries

3. Tool Confusion

Problem: AI uses wrong tool or wrong API endpoint **Solution**: Map tools explicitly and validate in sandbox

4. Implicit Knowledge

Problem: SOP assumes knowledge not provided to AI **Solution**: Review SOPs for implicit assumptions and make explicit

5. Version Mismatch

Problem: AI trained on old SOP version, executed with new **Solution**: Implement version checking and update protocols

Continuous Improvement

SOP training isn't one-time—it's ongoing:

Weekly:

- Review error logs
- Identify recurring issues
- Update SOPs with clarifications

Monthly:

- Analyze execution metrics
- Conduct spot-check audits
- Retrain on updated procedures

Quarterly:

- Full SOP review and updates
- Comprehensive retraining
- Process optimization

Chapter 5: Maintenance & Version Control

SOPs are living documents that require ongoing care. This chapter covers the systems and practices for keeping your SOPs current, accurate, and aligned with evolving business needs.

The Cost of Stale SOPs

Outdated SOPs are worse than no SOPs—they create false confidence and systematic errors. Common symptoms include:

- AI executing deprecated procedures
- Integration failures due to API changes
- Incorrect outputs based on old rules
- Security vulnerabilities from outdated practices
- Compliance gaps from missed regulation updates

Version Control System

Treat SOPs like code. Use a formal version control system:

Version Numbering

Use semantic versioning (MAJOR.MINOR.PATCH): - **MAJOR**: Fundamental process changes - **MINOR**: New steps or significant modifications - **PATCH**: Corrections, clarifications, formatting

Example: SALES-001 v2.3.1

Version Metadata

Every version should include: - Version number - Release date - Author - Change summary - Impact assessment - Rollback instructions

Change Documentation

Maintain a change log for each SOP:

Version	Date	Changes	Author	Reviewer
2.3.0	2025-01-10	Added Step 8 for post-kickoff	Sarah	Mike
2.2.1	2025-01-05	Fixed typo in error handling	Sarah	-
2.2.0	2024-12-20	New CRM integration	Sarah	Mike
2.1.0	2024-12-01	Updated escalation rules	Sarah	Mike
2.0.0	2024-11-15	Complete rewrite	Sarah	Jane

Review Cycles

Establish regular review schedules:

Continuous Monitoring

- AI execution logs (daily)
- Error reports (real-time)
- Integration failures (immediate)

Weekly Review

- Spot-check recent executions
- Review flagged issues
- Quick clarifications

Monthly Review

- Full metrics analysis
- Stakeholder feedback
- Minor updates and patches

Quarterly Review

- Comprehensive SOP audit
- Process improvement workshops
- Major version updates
- Training material refresh

Annual Review

- Strategic alignment check
- Complete SOP inventory
- Retirement of obsolete procedures
- Architecture redesign if needed

Maintenance Triggers

Beyond scheduled reviews, certain events should trigger immediate SOP updates:

Tool Changes

- API version updates
- New feature releases
- Deprecation notices
- Authentication changes

Business Changes

- New service offerings
- Process reorganization
- Staff role changes
- Compliance updates

Performance Issues

- Error rate increases
- Execution time degradation
- Quality score declines
- User complaints

Update Workflow

Follow a structured process for SOP changes:

1. Change Request

Document the proposed change: - SOP ID and section - Current problem - Proposed solution - Expected impact - Requester and date

2. Impact Assessment

Evaluate the change: - AI systems affected - Integration dependencies - Training requirements - Risk level

3. Draft Update

Create the new version: - Edit SOP content - Update version number - Document changes - Add change log entry

4. Review

Have stakeholders validate: - Technical accuracy - Business alignment - Clarity and completeness - Safety and compliance

5. Test

Validate in safe environment: - Sandbox execution - Edge case testing - Integration verification - Rollback testing

6. Deploy

Release to production: - Update AI knowledge bases - Notify affected users - Archive old version - Monitor initial executions

7. Train

Ensure AI systems adapt: - Update training materials - Run execution drills - Monitor accuracy - Gather feedback

SOP Deprecation

Sometimes SOPs need to be retired:

Deprecation Criteria

- Process no longer exists
- Superseded by better procedure
- Tool integrations obsolete
- Consistently poor performance

Deprecation Process

1. Mark SOP as deprecated in metadata
2. Set end-of-life date (minimum 30 days out)
3. Notify all stakeholders
4. Migrate AI systems to replacement SOP
5. Archive (don't delete) old version
6. Update cross-references

Documentation Hygiene

Keep your SOP library organized:

Naming Conventions

- SOP-[CATEGORY]-[NUMBER]: SALES-001
- Descriptive titles
- Consistent formatting

File Organization

```
/SOPs
  /Active
    /Sales
    /Marketing
    /Operations
    /Finance
  /Archive
  /Templates
  /Changelog
```

Searchability

- Tag SOPs by category, tool, and department
- Include keywords in descriptions
- Maintain master index
- Cross-reference related SOPs

Collaboration Practices

SOPs often span multiple teams:

Ownership Model

- Each SOP has a single owner
- Owners responsible for updates
- Cross-functional reviews required
- Escalation path defined

Communication

- Change notifications to affected teams
- Slack channel for SOP discussions
- Monthly SOP newsletter
- Quarterly all-hands review

Conflict Resolution

When teams disagree on SOP content: 1. Document both perspectives 2. Test both approaches 3. Measure outcomes 4. Escalate to leadership if needed 5. Update SOP with decision rationale

Measuring SOP Health

Track these metrics:

Usage Metrics

- Number of AI executions per SOP

- Execution frequency trends
- Coverage across business areas

Quality Metrics

- Success rate by SOP
- Error types and frequency
- Human intervention rate
- Training time for new AI systems

Maintenance Metrics

- Average time between updates
- Review compliance
- Version drift (how many old versions in use)

Tools for SOP Management

Recommended tools for SOP lifecycle:

Documentation

- Notion: Collaborative editing
- Git: Version control
- Confluence: Enterprise docs
- Markdown: Portable format

Automation

- Zapier: Integration triggers
- Make: Complex workflows
- n8n: Self-hosted automation
- Custom: Platform-native tools

Monitoring

- Datadog: Infrastructure monitoring

- Airtable: Process tracking
 - Notion: Review scheduling
 - Slack: Alert notifications
-

Appendix: Blank SOP Templates

Use these templates as starting points for creating your own SOPs. Copy the appropriate template and fill in each section with your specific requirements.

Template A: Event-Based Automation SOP

```

# SOP: [TASK NAME]

## Metadata
- **SOP ID**: [XXX-###]
- **Version**: [X.X.X]
- **Owner**: [Name]
- **AI System**: [System name]
- **Last Updated**: [YYYY-MM-DD]

## Purpose
[One paragraph explaining what this accomplishes]

## Trigger
**Trigger Type**: Event-based
**Trigger Details**: [Specific event that starts this SOP]
**Event Source**: [System generating the event]

## Prerequisites
- [Required condition 1]
- [Required condition 2]
- [Required access/tool]

## Tools & Systems
| Tool | Purpose | Access Required |
|-----|-----|-----|
| [Tool] | [Use] | [Level] |

## Procedure

### Step 1: [Action]
**Action**: [What to do]
**Input**: [What you need]
**Process**: [How to do it]
**Output**: [What you produce]
**Time Limit**: [Max time]

[Repeat for each step]

### Decision Points

#### Decision: [Name]
**Condition**: [What to check]
**IF** [condition]: [Action]
**ELSE**: [Default action]

## Expected Output
- [Deliverable 1]
- [Deliverable 2]

## Error Handling

### Error: [Scenario]
**Symptom**: [How to detect]
**Retry**: [How many times]
**Escalation**: [Who to notify]

```

****Fallback**:** [Alternative action]

Change Log

Version	Date	Changes	Author
-----	-----	-----	-----
1.0	[Date]	Initial creation	[Name]

Template B: Time-Based Automation SOP

```

# SOP: [TASK NAME]

## Metadata
- **SOP ID**: [XXX-###]
- **Version**: [X.X.X]
- **Owner**: [Name]
- **AI System**: [System name]
- **Last Updated**: [YYYY-MM-DD]

## Purpose
[One paragraph explaining purpose]

## Trigger
**Trigger Type**: Time-based
**Schedule**: [When this runs: daily, weekly, specific time]
**Timezone**: [Applicable timezone]
**Exceptions**: [When NOT to run]

## Prerequisites
- [Required condition 1]
- [Required condition 2]

## Tools & Systems
| Tool | Purpose | Access Required |
|-----|-----|-----|
| [Tool] | [Use] | [Level] |

## Procedure

### Step 1: [Action]
**Action**: [What to do]
**Input**: [What you need]
**Process**: [How to do it]
**Output**: [What you produce]
**Time Limit**: [Max time]

[Repeat for each step]

## Expected Output
- [Deliverable 1]
- [Deliverable 2]

## Error Handling

### Error: [Scenario]
**Symptom**: [How to detect]
**Retry**: [Retry logic]
**Escalation**: [Who to notify]

## Change Log
| Version | Date | Changes | Author |
|-----|-----|-----|-----|
| 1.0 | [Date] | Initial creation | [Name] |

```



Template C: Human-Initiated SOP

```

# SOP: [TASK NAME]

## Metadata
- **SOP ID**: [XXX-###]
- **Version**: [X.X.X]
- **Owner**: [Name]
- **AI System**: [System name]
- **Last Updated**: [YYYY-MM-DD]

## Purpose
[One paragraph explaining purpose]

## Trigger
**Trigger Type**: Manual
**Initiated By**: [Who starts this]
**Input Required**: [What they need to provide]
**Command/Phrase**: [How to trigger]

## Prerequisites
- [Required condition 1]
- [Required condition 2]

## Tools & Systems
| Tool | Purpose | Access Required |
|-----|-----|-----|
| [Tool] | [Use] | [Level] |

## Input Parameters
| Parameter | Type | Required | Description |
|-----|-----|-----|-----|
| [Param 1] | [Text/Number/etc] | Yes/No | [What it is] |

## Procedure

### Step 1: [Action]
**Action**: [What to do]
**Input**: [What you need]
**Process**: [How to do it]
**Output**: [What you produce]
**Time Limit**: [Max time]

[Repeat for each step]

## Expected Output
- [Deliverable 1]
- [Deliverable 2]

## Confirmation Message
**Success**: [What to say when done]
**Failure**: [What to say when failed]

## Error Handling

### Error: [Scenario]
**Symptom**: [How to detect]

```

****Response**:** [What to do]

Change Log

Version	Date	Changes	Author
-----	-----	-----	-----
1.0	[Date]	Initial creation	[Name]

Template D: Decision/Classification SOP

```

# SOP: [TASK NAME]

## Metadata
- **SOP ID**: [XXX-###]
- **Version**: [X.X.X]
- **Owner**: [Name]
- **AI System**: [System name]
- **Last Updated**: [YYYY-MM-DD]

## Purpose
[One paragraph explaining classification/decision purpose]

## Trigger
**Trigger Type**: [Event/Manual/Time]
**Trigger Details**: [Specific condition]

## Classification Categories
| Category | Criteria | Action | Priority |
|-----|-----|-----|-----|
| [Cat 1] | [When to apply] | [What happens] | [Level] |
| [Cat 2] | [When to apply] | [What happens] | [Level] |
| [Cat 3] | [When to apply] | [What happens] | [Level] |

## Decision Tree

### Primary Decision: [Name]
**Evaluate**: [What to check first]

**IF** [condition A]:
- [Action set A]
- **Sub-decision**: [Nested check]
  - **IF** [sub-condition]: [Sub-action]
  - **ELSE**: [Default]

**ELSE IF** [condition B]:
- [Action set B]

**ELSE**:
- [Default action]

## Tools & Systems
| Tool | Purpose | Access Required |
|-----|-----|-----|
| [Tool] | [Use] | [Level] |

## Expected Output
- [Classification result]
- [Routing destination]
- [Priority assignment]

## Confidence Thresholds
- **High Confidence** (>90%): Auto-execute
- **Medium Confidence** (70-90%): Execute with notification
- **Low Confidence** (<70%): Queue for human review

```

```
## Error Handling

### Error: Uncertain Classification
**Symptom**: Confidence below threshold
**Action**: Route to human reviewer
**Notification**: Alert assigned team

## Change Log
| Version | Date | Changes | Author |
|-----|-----|-----|-----|
| 1.0 | [Date] | Initial creation | [Name] |
```

Template E: Multi-System Integration SOP

```
# SOP: [TASK NAME]

## Metadata
- **SOP ID**: [XXX-###]
- **Version**: [X.X.X]
- **Owner**: [Name]
- **AI System**: [System name]
- **Last Updated**: [YYYY-MM-DD]

## Purpose
[One paragraph explaining cross-system workflow]

## System Architecture
```

[System A] → [Data Transform] → [System B] → [Action] → [System C]

```

## Trigger
**Trigger Type**: [Event/Manual/Time]
**Trigger Details**: [Specific condition]

## Prerequisites
- [System A] authentication valid
- [System B] API accessible
- [System C] write permissions confirmed
- Data mapping current

## Tools & Systems
| System | Role | API Version | Endpoint |
|-----|-----|-----|-----|
| [System A] | Source | [Version] | [URL] |
| [System B] | Processor | [Version] | [URL] |
| [System C] | Destination | [Version] | [URL] |

## Data Flow

### Stage 1: Data Extraction
**From**: [System A]
**Endpoint**: [API endpoint]
**Method**: [GET/POST/etc]
**Parameters**: [Required inputs]
**Expected Response**: [Data structure]

### Stage 2: Data Transformation
**Action**: [What to transform]
**Mapping**:
| Source Field | Target Field | Transform |
|-----|-----|-----|
| [Field A] | [Field X] | [Direct/Mapped] |
| [Field B] | [Field Y] | [Format change] |

### Stage 3: Data Loading
**To**: [System C]
**Endpoint**: [API endpoint]
**Method**: [GET/POST/etc]
**Payload**: [Transformed data]
**Validation**: [Success criteria]

## Error Handling by System

### System A Errors
**Error**: [Specific error]
**Response**: [What to do]
**Retry**: [Logic]

### System B Errors
**Error**: [Specific error]
**Response**: [What to do]
**Retry**: [Logic]

### System C Errors

```

```
**Error**: [Specific error]
**Response**: [What to do]
**Retry**: [Logic]

## Rollback Procedure
**If failure at Stage 1**: [No action needed]
**If failure at Stage 2**: [Clean up temporary data]
**If failure at Stage 3**: [Reverse changes in System C if possible]

## Monitoring
- **Success Metric**: [What to track]
- **Alert Threshold**: [When to notify]
- **Dashboard**: [Where to view]

## Change Log
| Version | Date | Changes | Author |
|-----|-----|-----|-----|
| 1.0 | [Date] | Initial creation | [Name] |
```

Conclusion

You now have everything you need to create, implement, and maintain AI-compatible Standard Operating Procedures. The AI SOP Builder Kit provides:

- **A proven template structure** for consistent, executable documentation
- **10 complete examples** covering common business workflows
- **Training methodologies** to onboard AI systems effectively
- **Maintenance practices** to keep procedures current
- **Blank templates** for rapid SOP creation

The key to success is treating SOPs as living infrastructure. Like code, they require testing, versioning, and regular updates. But the investment pays dividends in consistency, scalability, and the ability to deploy AI systems with confidence.

Start with one process. Write the SOP. Train your AI. Measure the results. Iterate and improve. Then expand to the next workflow.

The businesses that master AI-compatible SOPs will be the ones that scale efficiently in the AI era. You're now equipped to be among them.

Document Information - Title: AI SOP Builder Kit - Version: 1.0 - Created: January 2025 - Author: Logic Impact - Pages: [Generated]

End of Document