GUIDE B3

TURN8

HOW TO CONFIGURE **AI STUDIO AGENTS** AND **AUTOMATE WORKFLOWS**

'AI agent configuration & workflow automation

WHERE THIS IS USED

- AI Studio agent escalation
- Foundry-as-a-Service engagements
- Corporate Incubators with AI-native ventures

AUDIENCE

- AI Studio Operators
- Venture Technical Leads
- Process Automation Engineers
- Innovation Program Managers

PHASE

Phase Two: Validation & Design → AI Infrastructure Design (Weeks 5–9)

1

EXECUTIVE SUMMARY

AI Studio agents are configurable AI systems that automate research, analysis, customer interaction, and operational workflows inside a venture. This guide teaches teams to map their core venture processes, identify which workflows are candidates for AI automation, configure the appropriate AI agents, and validate that the agents perform reliably before integrating them into the MVP. For corporate ventures deploying AI Studio, this guide is the primary technical operational reference for Phase Two.

2

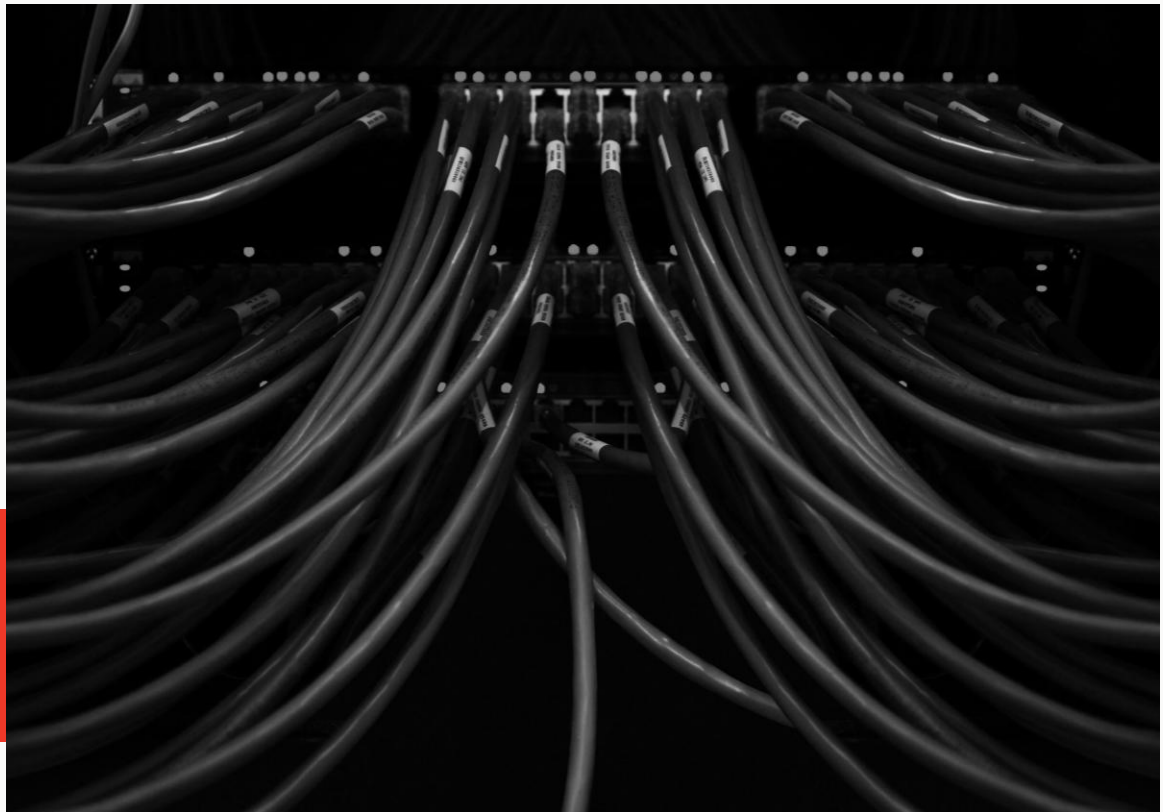
THE CORE PROBLEM

The AI Agent Trap

- Teams deploy AI agents before mapping the processes they are meant to automate — the agent amplifies broken workflows.
- AI agents are configured by technical team members who do not understand the customer journey.
- Agents are not tested with real-world edge cases before deployment — they fail in production.
- Teams over-automate: they use AI for tasks that require human judgment, eroding customer trust.
- No escalation protocol exists: when the agent fails, there is no clear handoff to a human.

In GCC corporate and government contexts:

AI-generated outputs used in customer-facing workflows may be subject to data localisation requirements and sector-specific AI governance standards. Verify with your legal and compliance teams before deploying customer-facing agents. Internal operations agents typically carry lower compliance risk and are a good starting point.



PREREQUISITES

- Completed Guide B1: MVP Specification with a clear core use case
- A documented process map of the venture's core customer workflow (even a rough flowchart is sufficient)
- Access to TURN8 AI Studio platform or equivalent (n8n, Make, Zapier + LLM integration, or custom API)
- API keys or platform credentials for required integrations (CRM, data sources, communication tools)
- A defined escalation protocol specifying the conditions under which the AI agent should hand off to a human reviewer.

4

EXPECTED OUTPUT/ SUCCESS CRITERIA

You Have Succeeded When:

- ☑ A Process Map exists showing all core venture workflows with AI-automation candidates marked
- ☑ At least 2 AI agents configured and tested: one for customer-facing tasks, one for internal operations
- ☑ Each agent has a documented system prompt, input/output specification, and escalation protocol
- ☑ Agent performance validated against a test set of 50+ real-world inputs with >90% accuracy
- ☑ Agents integrated into the MVP prototype and validated in Guide B2 sessions

STEP-BY-STEP INSTRUCTIONS

STEP 1 MAP YOUR CORE WORKFLOWS

1.1 List every task your team and customers perform in the core use case (from Guide B1 MVP Spec).

1.2 For each task, rate it on two dimensions: Frequency (How often?) and Complexity (How much human judgment?). Plot on a 2x2 matrix.

1.3 High Frequency + Low Complexity quadrant = prime candidates for AI automation. Mark these.

1.4 High Complexity or High Stakes tasks = require human judgment. These are escalation points, not automation candidates.

STEP 2 DESIGN YOUR AI AGENT ARCHITECTURE

2.1 For each automation candidate, design the agent using this specification format:

FIELD	DESCRIPTION
Agent Name	Clear, functional name (e.g., 'Lead Qualification Agent', 'Insight Synthesis Agent')
Trigger	What event activates this agent? (New form submission, scheduled time, API call)
Input	What data does the agent receive? (Format, source, validation rules)
System Prompt	The instructions that define the agent's role, constraints, and output format
Output	What does the agent produce? (Text, structured JSON, action, notification)
Escalation Rule	Under what conditions does the agent flag for human review?
Success Metric	How do you measure whether the agent is performing correctly?

STEP 3 WRITE EFFECTIVE SYSTEM PROMPTS

3.1 The system prompt is the most critical component of any AI agent. Use this structure:

SYSTEM PROMPT TEMPLATE

```
You are [AGENT ROLE] for [VENTURE NAME]. Your job is to [PRIMARY
TASK]. You serve [TARGET USER]. Always [KEY BEHAVIOR 1]. Never
[KEY CONSTRAINT 1].When [EDGE CASE CONDITION], respond with
[SPECIFIC RESPONSE].Output your response in [FORMAT: plain text /
JSON / structured list].If you are uncertain or the input does not
match expected patterns, respond with:"ESCALATE: [reason]".
```

STEP 4 BUILD AND TEST THE AGENT

4.1 Configure the agent in your AI Studio platform. Connect inputs and outputs using the workflow tool (n8n, Make, or equivalent).

4.2 Create a test set of 50 real-world inputs. Include: 20 standard cases, 20 edge cases, 10 adversarial inputs (confusing, incomplete, or off-topic).

4.3 Run all 50 tests. Measure: accuracy rate, escalation rate, response time, and output format compliance.

4.4 >90% accuracy on standard cases, measured against the predefined success criteria for each test case (defined in Step 2.1 Success Metric field).

4.5 For any failures, revise the system prompt. Re-test the full set after each revision.

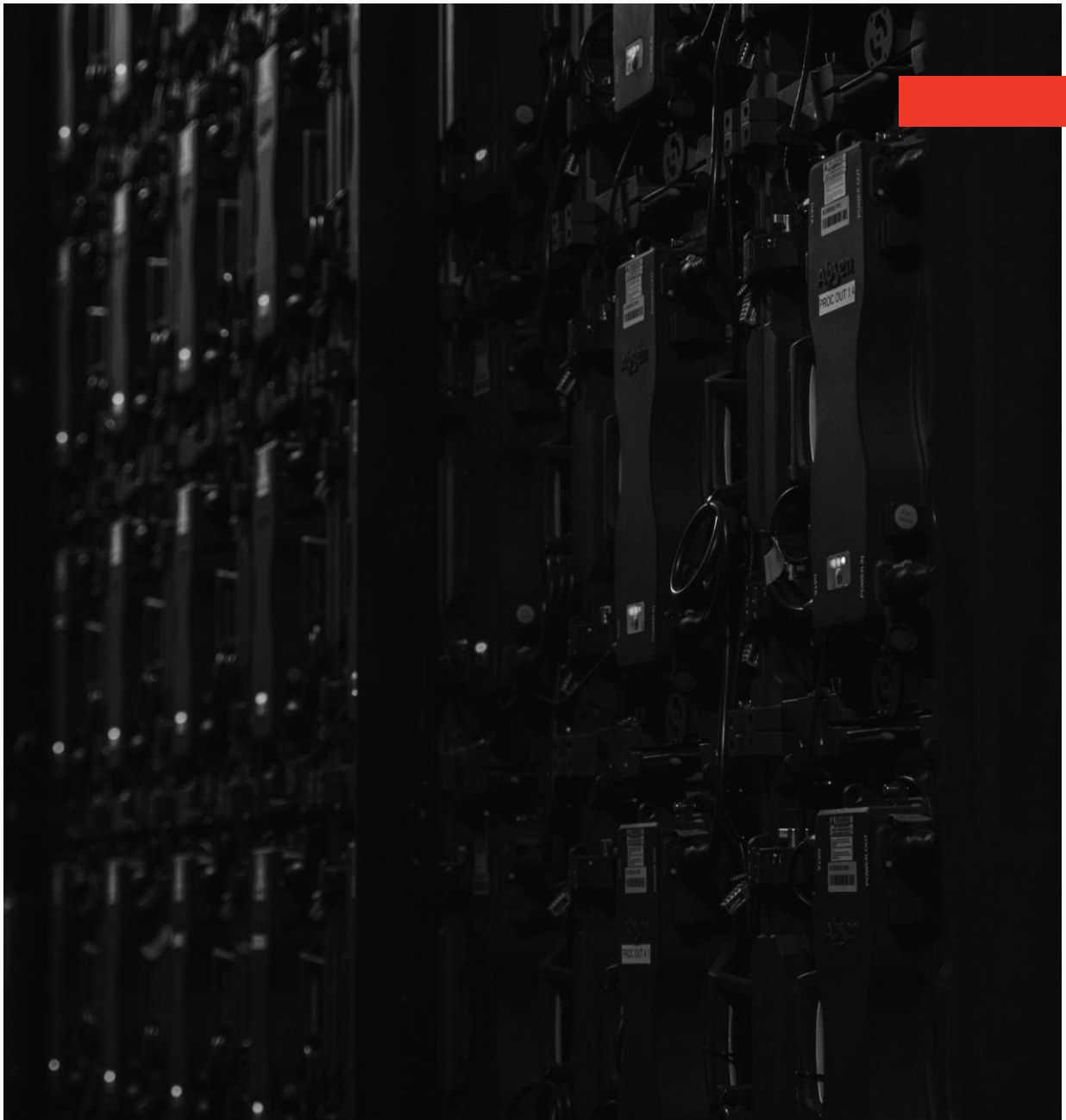
STEP 5 INTEGRATE AND DEPLOY

5.1 Integrate the agent into the MVP prototype from Guide B2.

5.2 Include the agent in user validation sessions. Observe how users interact with AI-generated outputs.

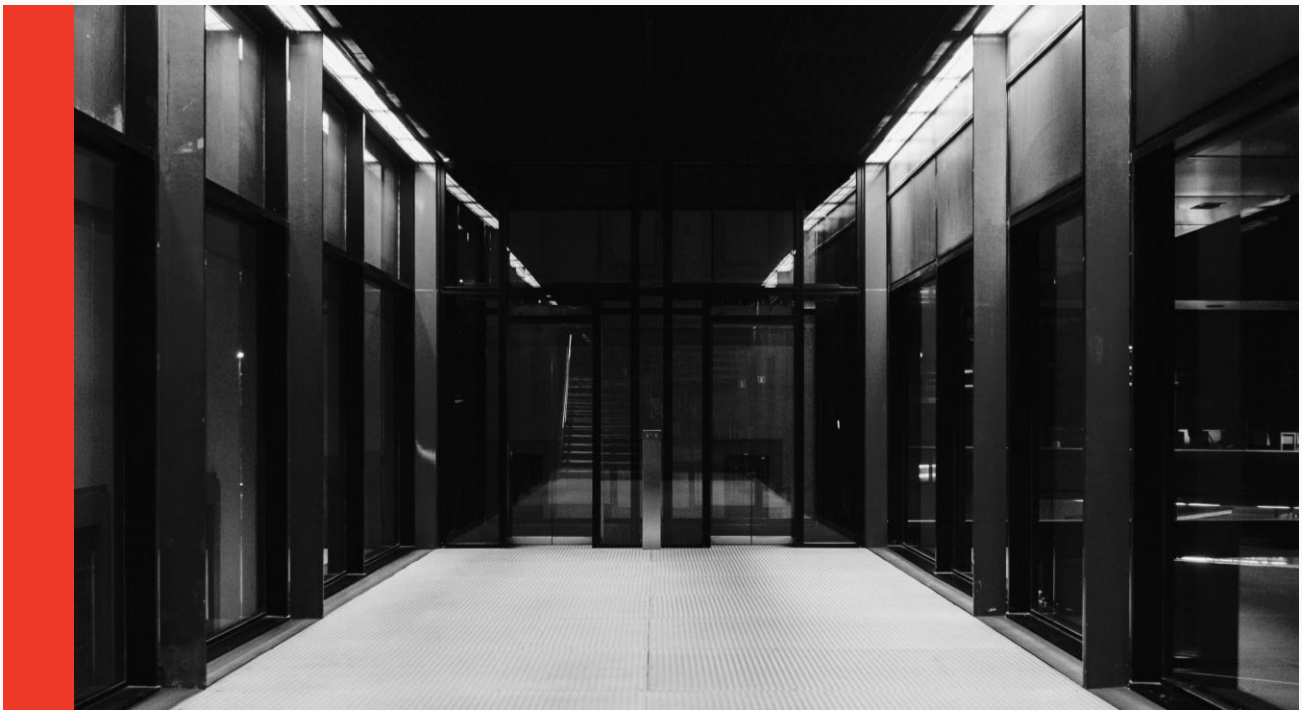
5.3 Document the escalation protocol: who receives the escalation, through what channel, and what is the expected response time?

5.4 Create an agent performance dashboard: track accuracy, escalation rate, and user satisfaction weekly.



TROUBLESHOOTING

ISSUE	LIKELY CAUSE	FIX
Agent produces hallucinated outputs on standard cases	System prompt constraints are too loose or output format is undefined	Tighten system prompt constraints and add explicit output format rules
Escalation rate is too high (>40%)	Edge case definitions are too narrow or prompts reject valid inputs	Review edge case definitions and broaden acceptable input conditions
API integration fails during testing	Invalid credentials, rate limits, or unstable connections	Verify credentials, check API rate limits, and test each connection individually before full workflow testing
Agent performs well in testing but fails in production	Production data differs from the test dataset	Expand the test set using real user inputs from the Guide A1 interview pool



7

NEXT STEPS

↘

Once your AI agents are configured, tested, and integrated into the MVP prototype, proceed to *Guide C1: How to Stress-Test Your Business Model and Unit Economics.*

The agent performance dashboard you built in Step 5.4 should be active before you begin the C-series guides.

CHECKLIST

PROCESS MAPPING

☐ All tasks performed by the team and customers in the core use case listed in full

☐ Every task rated on two dimensions: Frequency (how often) and Complexity (how much human judgment required)

☐ 2x2 Frequency/Complexity matrix completed — High Frequency/Low Complexity tasks marked as automation candidates

☐ High Complexity and High Stakes tasks identified as escalation points — NOT automation candidates

AGENT ARCHITECTURE

☐ Agent specification written for each automation candidate: Name, Trigger, Input, System Prompt, Output, Escalation Rule, Success Metric

☐ Minimum 2 agents specified: one customer-facing, one internal operations

☐ System prompts written using the template structure: role, primary task, target user, key behaviors, constraints, edge case responses, output format, escalation signal

☐ Escalation protocol defined: which condition triggers it, who receives it, through what channel, and expected response time

TESTING

☐ Test set of 50 real-world inputs created: 20 standard cases, 20 edge cases, 10 adversarial inputs

☐ All 50 tests run and results recorded: accuracy rate, escalation rate, response time, output format compliance

☐ Agent accuracy above 90% on standard test cases

☐ Edge case escalation working as designed — agent flags correctly and does not produce false outputs

☐ Zero failures on adversarial inputs — agent responds with escalation, not incorrect output

☐ System prompt revised after any failure; full test set re-run after each revision

DEPLOYMENT & MONITORING

☐ Agents integrated into the MVP prototype from Guide B2

☐ AI agent behavior included in Guide B2 user validation sessions — user reactions to AI outputs observed

☐ Agent performance dashboard configured: accuracy, escalation rate, and user satisfaction tracked weekly

☐ Escalation protocol tested end-to-end: escalation triggered, received, and responded to correctly