

raia · THE AGENT FACTORY

Deploying your first Support AI Agent

Five phases, fourteen steps. What your team prepares, who is involved, how much time it takes, and what has to be true before each phase closes.

Scope · Training · Build · QA · Launch · Continuous improvement
Working session for the deployment team

The Agent Factory lifecycle

Each phase produces the inputs the next phase needs. Launch is not the end — it feeds Build and QA continuously.



Continuous improvement feedback loop: monitoring results, human interventions, conversation audits, user feedback, and incidents flow back into Build and QA. Instructions, workflows, skills, sources, and controls get revised, retested, documented, and released through a governed change process.

Every phase closes on evidence, not a date

A phase is finished when its exit criteria are met and a named person accepts them.

PHASE	PRIMARY OUTPUT	EXIT CRITERIA	WHO ACCEPTS
1. Scope	Approved agent definition and blueprint	Outcome approved, blueprint sufficiently defined, stakeholders assigned, boundaries and success measures agreed	Sponsor + agent owner
2. Training	Validated sources and application connections	Sources approved, connections working, permissions validated, training context traceable to authoritative sources	Functional Expert + Integration Specialist
3. Build	Functional agent configuration and workflow	Instructions implemented, primary workflow runs end to end, required skills available and tested	Technical Agent Builder + agent owner
4. QA	Validated agent and audit report	Critical issues resolved, remaining limitations documented and accepted, reviewers approve, audit report supports a launch decision	Functional Expert + sponsor
5. Launch	Production agent with ongoing governance	Safeguards verified, monitoring active, human oversight staffed, conversation audits scheduled, owners assigned for post-launch issues	Sponsor + security owner

Any phase can close with a different answer: proceed, narrow the scope, do more discovery, or hold. A hold is a valid and useful outcome.

Where each phase actually happens

Scope involves no platform work at all — and it is the phase that most often decides whether the agent succeeds.

1. SCOPE

Workshops and your own documentation

No platform work. Outcome, blueprint, and team are settled in the room and in your existing tools.

2. TRAINING

raia Command

You collect and curate sources. Command runs the connector library and processes everything into a trained vector store.

3. BUILD

raia cX — the Launch Pad

Where the agent is built, configured, and managed — instructions, workflow, skills, security, launch state.

4. QA

raia Copilot

Simulated tests, source inspection, ratings and comments, and the evidence behind the audit report.

5. LAUNCH

Your channel, watched from Copilot

Security sign-off, live monitoring, human takeover, and recurring conversation audits.

Instructions are not knowledge. Instructions define behavior, boundaries, and judgment. The knowledge layer supplies the facts and procedures the agent retrieves. Most defects trace to one or the other — the diagnosis always starts by deciding which.

The four roles — name a person for each

These are raia's four defined roles. Each has its own training path, so the person you name can be brought up to speed before the work starts.

Functional Expert

Your support domain specialist

- What a correct answer looks like
- Which source wins a conflict
- Whether the agent is right

Scope → QA · ~20 hrs

Functional Expert path · 6 modules

Agent User

The people who work the queue today

- How the work really happens
- Realistic testing and ratings
- Takeover in Copilot

Scope, QA, Launch · ~12 hrs

Agent User path · 3 modules

Technical Agent Builder

Your developer or prompt engineer

- System prompts and instructions
- Functions, webhooks, and the API
- n8n workflow integration

Build → QA · ~10 hrs

Technical Builder path · 6 modules

Agent Integration Specialist

Your IT, DevOps, or platform engineer

- Connectors and data ingestion
- MCP setup, identities, permissions
- Live Chat SDK and security

Training, Build, Launch · ~8 hrs

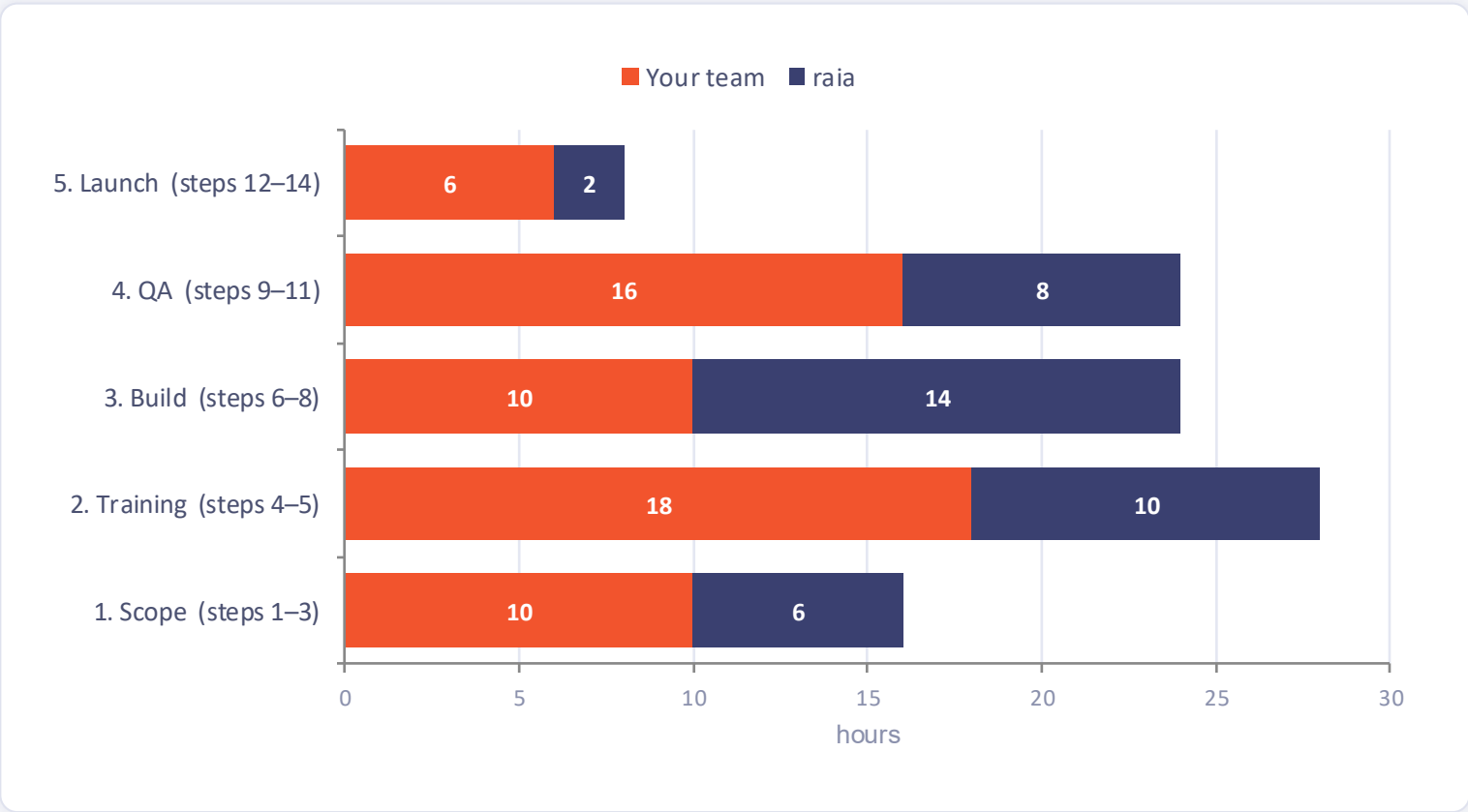
Integration Specialist path · 3 modules

Two more names, no training path needed: a **sponsor** who owns the value decision and accepts each phase (~3 hrs), and an **agent owner** who coordinates the four roles, keeps the decisions and evidence, and runs the beta (~10 hrs).

Start the training during Scope. There is no platform work in that phase, so it is free time to get each named person through their path at docs.raiaai.com.

Where the effort goes

Roughly 100 combined hours for a first narrow agent — about 60 on your side, 40 on ours.



Effort is not schedule

The platform work is fast once decisions, sources, access, and owners exist. Elapsed time is set by how quickly your side can decide, extract, and approve.

- What stretches a timeline**
- Waiting on a ticket export
 - Unowned or unpublished knowledge
 - Security review started late
 - Expert time that keeps slipping
 - Approvals waiting on one person

Illustrative average across the four roles and the raia team — not a fixed estimate or a calendar commitment.

What to have ready before Scope begins

Two of these have long lead times. Start them today, not at the phase where they are needed.

People

One name for each of the four raia roles, plus a sponsor, an agent owner, and a deputy for the Functional Expert. Time confirmed, and training paths started.

Lead time Now

Knowledge assets

A list of the sources that matter, who owns each one, and whether they are current. Access for the people who will curate them in Training.

Lead time 1–2 weeks

Ticket history

An export of historical support tickets. Needs a data owner and usually a privacy decision — the spec is later in this deck.

Lead time 2–4 weeks

System access

The applications the agent may need to reach, the owner of each, and whether a connector, API, or MCP path already exists.

Lead time 2–6 weeks

Also decide upfront

■ Which support domain is the candidate — one, not several

■ Where the historical tickets live and who can extract them

■ Who the first users are, and whether they are internal

■ Whether any of this needs a privacy or legal review

1 Scope

Define why the agent exists, how it will operate, and who is responsible — before anything is built.

1

Define Agent — Outcome

Business outcome, intended users, primary use cases, boundaries, success criteria, and the conditions that require a human.

DELIVERABLE

Agent outcome statement and measurable success criteria

2

Build Blueprint

The proposed experience, required capabilities, knowledge needs, application dependencies, workflow stages, governance requirements, and expected handoffs.

DELIVERABLE

Approved agent blueprint

3

Assign Team

Business owner, subject-matter experts, builders, reviewers, security and compliance stakeholders, operational owners — and their decision rights.

DELIVERABLE

Named delivery team with assigned ownership

Exit criteria: the intended outcome is approved, the blueprint is sufficiently defined, the required stakeholders are assigned, and the team agrees on the boundaries and success measures for the agent.

Most failed agents fail here. A vague outcome and an unassigned team produce work that nobody can accept at the end of QA.

1

Define the agent and its outcome

Step 1. A use case describes a result, not a feature. Fill the brackets in the room.

Help **[specific support user]** resolve or progress **[defined issue class]** by **[approved behavior and workflow]**, producing **[defined output]**, while transferring **[exception conditions]** to **[human role]**.

Success criteria you can measure

- Current baseline: ticket volume, handling time, resolution and escalation rates
- What should improve, and by roughly how much
- Where the measurement comes from, and who collects it
- What result would justify expanding — and what would justify stopping

The first boundary

- One support domain
- One primary user group, normally internal
- A limited, written intent set
- Capability: answer, guide, retrieve, summarize, or draft — writes come later
- Explicit exclusions: what this agent will not handle at all

Conditions requiring a human

- Commitments, exceptions, and approvals
- Sensitive or high-risk situations
- Anything outside the written intent set
- What the agent says when knowledge or a system is unavailable
- Stop conditions that require pause, restriction, or disable

Baseline first: if nobody can state the current handling time or resolution rate, there is nothing to measure the agent against at QA.

1 Build the blueprint, assign the team

Steps 2 and 3. The blueprint is what Training, Build, and QA are all working from.

2 Build Blueprint

Proposed experience	How a request arrives, what the agent does, what the user sees
Required capabilities	What the agent must be able to do to hit the outcome
Knowledge needs	Which subject areas it has to be grounded in
Application dependencies	Which systems it needs to read from or act on
Workflow stages	The end-to-end path, including exception routes
Governance requirements	Approvals, data rules, retention, audit expectations
Expected handoffs	Where a person takes over, and with what context

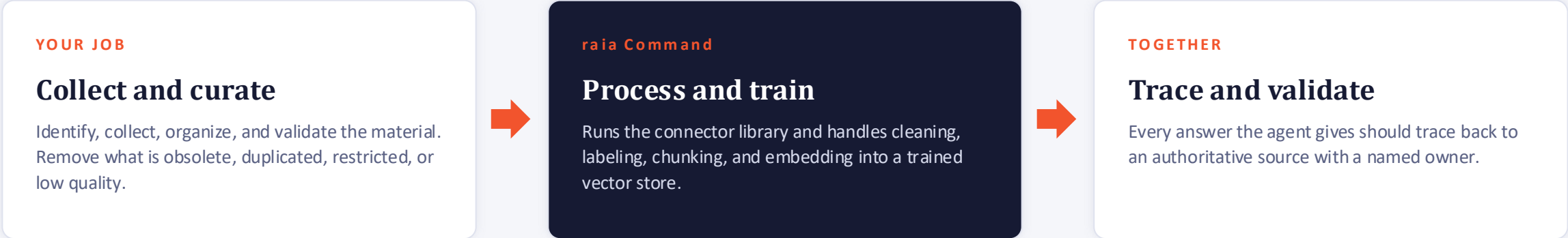
3 Assign Team

Business owner	Owns the outcome and accepts each phase
Functional Expert	Defines correct behavior and resolves source conflicts
Agent Users	Represent the work as it happens today; test and give feedback
Technical Agent Builder	Instructions, workflow, functions, and skills
Integration Specialist	Connectors, MCP, identities, permissions, security
Reviewers	Security, compliance, privacy, and legal where applicable
Operational owner	Runs the agent after launch and owns post-launch issues

Clarify decision rights, not just names. For every open question, one person decides — write down who, before Training starts.

2 Collect the training sources

Step 4. Sources do not need to arrive AI-ready. They do need an owner and a decision about which one wins.



SOURCE CATEGORY	WHAT WE NEED FROM IT	WHO NORMALLY OWNS IT
Product and service knowledge	Manuals, feature docs, supported versions, technical references	Product or documentation team
Support procedures	Triage, troubleshooting, diagnostic sequences, resolution paths	Support operations
Policies and commitments	Eligibility, warranty, service terms, approved commitment language	Policy, legal, or service owner
Known issues and changes	Incident guidance, workarounds, releases, deprecated behavior	Engineering or release management
Existing knowledge base	FAQs, approved articles, help-center content, internal notes	Knowledge manager
Transcripts and examples	Resolved conversations that show how the work is actually done	Support operations
Escalation information	Severity, queue ownership, required context, service hours	Support operations
Terminology	Product names, acronyms, synonyms, error wording, user language	Functional Expert

2

Connect the applications

Step 5. Two different things: what gets trained into the agent, and what it reaches for live during a conversation.

TRAINED INTO THE VECTOR STORE

Stable, approved knowledge

- Procedures, policies, product and reference content
- Owned at the source and refreshed on a monitored pipeline

RETRIEVED LIVE, NEVER TRAINED

Sensitive and current data

- Personal data, entitlements, live status, user-specific records
- Reached through scoped calls at runtime, with minimum retention

Confirm all of this for every connection, before it is built

Business purpose	Data fields used	Fields explicitly prohibited	Service identity
Authentication method	Read / draft / write authority	Access permissions and scope	Data boundaries
Synchronization requirements	Source ownership	Timeout and missing-record behavior	Access-denied behavior
Human approval point	Logging and audit	Credential owner	How to disable it

Least privilege: a distinct service identity per connection wherever the target system supports one. Your Agent Integration Specialist and security reviewer approve identity, permissions, credentials, network path, and audit approach.

3

Build

Steps 6, 7, and 8. Nothing here should be discovered during configuration — every input was decided in Scope or Training.

6

Build Instructions

- Role and goals
- Operating rules and constraints
- Response standards
- Escalation behavior
- Tool-use guidance
- How to handle ambiguous or unsupported requests

DELIVERABLE

Version-controlled agent instructions

7

Develop Workflow

- Triggers and decision points
- Information retrieval
- Application actions
- Approvals
- Exception paths
- Human handoffs and completion criteria

DELIVERABLE

Implemented agent workflow

8

Setup Skills

- Specialized skills for the assigned use cases
- Tools and reusable procedures
- Integrations
- Knowledge-retrieval capabilities
- The minimum set for this stage — expand later on evidence

DELIVERABLE

Tested skill and tool configuration

Exit criteria: the core instructions are implemented, the primary workflow operates end to end, the required skills are available, and the agent is ready for structured quality testing.

3

Designing the workflow

Step 7 in detail. Do not automate the legacy process — redesign it for a team that now includes an agent.

The method: an Agent User narrates one real ticket end to end. Every step gets classified into one of four types. Steps that turn out to be navigation, copying, searching, or waiting are removed rather than automated.

AGENT

Knowledge and reasoning

Retrieves trained knowledge, interprets context, recommends an answer or a next step.

LIVE CALL

Current-data retrieval

Pulls the live ticket, account, order, entitlement, or incident mid-conversation.

SYSTEM

Deterministic system action

Performs a defined update — behind human approval wherever risk requires it.

PERSON

Human judgment

Exceptions, empathy, sensitive decisions, commitments, approvals, anything unsupported.

Then define all eleven elements for every workflow you keep:

Trigger

Required context

Decision point

Retrieval

Application action

Approval

Human handoff

Completion criteria

Exception path

Fallback

Evidence

Escalation and human coverage

Designed in Build, operated at Launch. Two mechanisms, nine situations, six decisions each.

Escalation — the agent hands off

Configured as a skill in cX. The agent decides it should not continue and notifies a person by email, SMS, or webhook — into whatever system your team already works in.

Takeover — a person steps in

Happens in Copilot. Someone watching the conversation replies directly, the agent yields, the transcript stays unified, and control returns when the issue is resolved.

NINE SITUATIONS THAT NEED AN AGREED ANSWER

Someone explicitly asks for a person

There is no approved answer

Sources conflict or are incomplete

Troubleshooting has failed repeatedly

Sentiment turns negative or the situation is sensitive

A restricted topic or prohibited action comes up

An account or system change is required

An integration is unavailable

A security or privacy concern appears

Six decisions per situation

- What the agent says
- What context it collects and passes on
- Where it goes, and who is notified
- Who is covering, and during which hours
- How quickly a response is expected
- What happens when no one is available

4

Run simulated tests

Step 9. Roughly 300 conversations drawn from real tickets, run as structured tests against defined acceptance criteria.

The ticket export you need — start it early

Systems	Every helpdesk or ticketing system in scope
Date range	Usually 6–12 months, enough to cover seasonality
Statuses	Resolved and closed; escalated included, spam excluded
Fields	Subject, full thread, category, resolution, escalation flag, handle time, product and version, language
Exclusions	Out-of-scope products, teams, or languages
Personal data	Agreed treatment — redact, pseudonymize, or restrict access
Owner and format	A named person, and a format your analysts can read

What the test set must cover

Representative use cases	Wording variants
Edge cases	Failure scenarios
Restricted requests	Incomplete inputs
Workflow exceptions	Escalation paths
Live-data calls	Combined reasoning
Tone and experience	Regression

EVERY TEST CASE DEFINES

The user situation and context	Preconditions: version, data, state, identity	The expected business outcome	The ideal output and its sources	The expected live call and its result	Whether a human reviews, approves, or takes over
--------------------------------	---	-------------------------------	----------------------------------	---------------------------------------	--

Human feedback and the audit report

Steps 10 and 11. Agree how the numbers are counted before anyone reports one.

AI resolution rate = **in-scope conversations that reached the intended outcome without human takeover** ÷ all in-scope conversations tested. “In-scope” means inside the boundary written in Step 1 — out-of-scope requests are excluded, not counted as failures.

Who assesses what	
Functional Expert	Accuracy, domain judgment
Agent Users	Usefulness, tone, consistency
Technical Builder	Instructions, functions, errors
Integration Specialist	Connections, identity, access
Business owner	Alignment with the outcome
Reviewers	Quality of the human handoffs

Diagnose, do not just record	
Wrong or stale fact	Fix the source
No answer available	Add approved content
Right content, not found	Improve source structure
Poor tone or format	Refine instructions
Wrong workflow	Fix trigger or authority
Integration failed	Fix auth or errors

The audit report must contain
• Test coverage and findings • Unresolved risks • Remediation actions and owners • Approvals obtained • Accepted limitations • A clear launch recommendation

Separate from the percentage — these block launch regardless of how good the number looks: critical data exposure · an unauthorized action on a system · fabricated consequential guidance · a mandatory escalation that did not fire · a false claim that a workflow succeeded

5 Launch

Steps 12, 13, and 14. Controlled production use, with oversight staffed and audits scheduled before the agent goes live.

12 Verify Security / Compliance

- Access controls
- Data handling and privacy requirements
- Retention rules
- Approved integrations only
- Auditability
- Content safeguards
- Applicable organizational policies

DELIVERABLE

Security and compliance launch approval

13 Monitor / Human in the Loop

- Performance, failures, and escalations
- User feedback and workflow completion
- Operational exceptions
- Clear review, intervention, approval, and override procedures
- Named coverage and hours

DELIVERABLE

Operational monitoring and human-oversight process

14 Audit Conversations

- A representative sample, reviewed on a schedule
- Accuracy issues and unsafe behavior
- Process gaps
- Emerging user needs
- Training-data deficiencies
- Improvement opportunities

DELIVERABLE

Recurring audit findings and action plan

EXPOSURE WIDENS ONE STEP AT A TIME

Development — expert testing on a copy

Internal beta — named Agent Users, feedback and takeover

Supervised pilot — one channel, limited intents, active monitoring

Broader deployment — more audience, volume, or authority, on evidence

The continuous improvement loop

Monitoring results, interventions, audits, feedback, and incidents flow back into Build and QA — through a governed change process.

FEEDBACK SIGNAL	TYPICAL RESPONSE	RETURNS TO
Repeated inaccurate or incomplete responses	Improve training sources or agent instructions, then rerun simulated tests	Training → QA
Workflow failures or unnecessary handoffs	Revise workflow logic and validate the updated path	Build → QA
New business or user requirements	Revisit the scope and update the blueprint	Scope
Security, privacy, or compliance concern	Apply immediate controls, complete targeted QA before relaunch	Launch → QA
High human-intervention rate	Improve instructions, skills, or workflow decision points while preserving required oversight	Build
Positive performance trend	Document the successful pattern and standardize it across relevant use cases	Scope

The operating rhythm

Daily at first — conversations, takeovers, critical events. Weekly — work the gap log and re-test what changed. Monthly — resolution rate, adoption, escalation and integration health. Quarterly — access review, retention check, expansion decision.

The upstream system is the real control

A change in the helpdesk, CRM, repository, or knowledge base shifts agent behavior even when nobody touched the instructions. Any material source or system change triggers a re-test.

LEAVING THIS SESSION

Four things, with an owner and a date

1

Name the people

One person for each of the four raia roles, plus a sponsor, an agent owner, and a deputy for the Functional Expert. Confirm the hours and start their training paths.

2

Draft the outcome

One filled-in outcome statement and the measurable success criteria behind it. Rough is fine; specific is essential.

3

Start the ticket export

Agree the spec, name the data owner, and raise any privacy question now — this is the long pole in the whole lifecycle.

4

List the applications

Every system the agent may need to read from or act on, the owner of each, and whether a path already exists.

Scope answers one question: is this use case ready to proceed, does it need to be narrowed, does it need more discovery, or should it wait until the people, knowledge, access, or approvals are actually in place?

A narrower agent that reaches Launch beats a broader one that stalls in Training.