

Deploying Your First Support AI Agent

A business-led guide to the five-phase, fourteen-step lifecycle

Scope · Training · Build · QA · Launch · Continuous improvement

Audience

Business sponsors, support leaders, Functional Experts, Agent Users, Technical Agent Builders, Agent Integration Specialists, and agent owners.

Companion documents

Preparation and decision workbook (Appendix A) · Historical ticket export specification (Appendix B) · Setup documentation checklist (Appendix C) · Optional modules for enterprise governance and higher-risk agents.

Version 3.0

Contents

	SECTION	WHAT IT COVERS
1	About this guide	Who it is for, what it assumes, and how to use it
2	The Agent Factory lifecycle	The five phases, their outputs, and how they connect
3	The fourteen steps	Every step, its deliverable, and its owner
4	Where the work happens on raia	raia cX, raia Command, and what sits inside cX
5	The delivery team	The four roles, their training paths, and decision rights
6	Effort and schedule	Where the hours go, and what actually sets the timeline
7	Before Scope begins	The preparation with the longest lead times
8	Phase 1 — Scope	Steps 1–3: outcome, blueprint, team
9	Phase 2 — Training	Steps 4–5: sources and application connections
10	Phase 3 — Build	Steps 6–8: instructions, workflow, skills
11	Phase 4 — QA	Steps 9–11: simulated tests, human feedback, audit report
12	Phase 5 — Launch	Steps 12–14: security, monitoring, conversation audits
13	Continuous improvement	How signals return to the right phase
14	Change management and continuity	Versions, source pipelines, and documentation
15	A successful first deployment	What good looks like when the lifecycle completes
16	Optional modules	When to add governance or higher-risk controls
A	Preparation and decision workbook	Fillable tables for every decision in the lifecycle
B	Historical ticket export specification	What to request, from whom, and in what form
C	Setup documentation checklist	What to record so the agent survives a handover
	References	Where each part of this guide comes from

1. About this guide

This guide describes how a company deploys its first enterprise-grade Support AI Agent on raia. It follows the Agent Factory lifecycle: five phases and fourteen steps, from defining the intended outcome through to auditing live conversations and feeding what you learn back into the agent.

The guide is deliberately weighted toward process rather than platform configuration. The technical work on raia is fast once the decisions, sources, access, and owners exist. What determines whether a first agent succeeds is almost always the work that happens before and around the platform: whether the outcome was defined precisely, whether the right people were named and given time, whether the knowledge had owners, and whether the tests reflected real work.

1.1 Why support is a good first use case

Support combines the conditions an enterprise-grade agent needs. There is documented business knowledge. There is repeatable work with recognizable patterns. There is a large body of historical evidence in the ticket system. There are measurable outcomes. There are existing systems worth connecting to. And there is a natural place for human judgment to remain, which makes the human-in-the-loop design obvious rather than theoretical.

1.2 What this guide assumes

- One narrowly defined agent, not a platform rollout. Breadth comes later, on evidence.
- An internal-first path in most cases: the agent helps the support team before it speaks to customers.
- A co-build. raia supplies platform expertise and the method; your company supplies the business task, the experts, the knowledge, the access, and the real-world testing.
- Phases close on evidence, not on dates. Any phase can end with proceed, narrow, discover more, or hold.

1.3 How to use it

IF YOU ARE	READ
A sponsor or business leader	Sections 2, 5, 6, and 15. These cover what you are approving, what it costs, and what success looks like.
The agent owner	The whole guide, plus Appendix A, which you will maintain as the working record.
A Functional Expert	Sections 8, 9, 11, and 13. Your work concentrates in Scope, Training, and QA.
An Agent User	Sections 8.3, 10.2, 11, and 12. You describe how the work happens, test the agent, and operate alongside it.
A Technical Agent Builder	Sections 10 and 11.
An Agent Integration Specialist	Sections 9.2, 10, and 12.1.
A security or compliance reviewer	Sections 9.2, 10.5, 12.1, and Appendix C.

A note on scope. The most common cause of a stalled first agent is not technical. It is a use case that was never narrowed. If, at the end of Scope, the team cannot name the issue class and the exception conditions in a single sitting, narrow the scope before proceeding rather than discovering the problem during QA.

2. The Agent Factory lifecycle

The Agent Factory is a five-phase lifecycle for defining, training, building, validating, and operating an AI agent. Each phase produces the inputs the next phase requires. Launch is not the end: it creates an ongoing loop that returns monitoring results, human interventions, and conversation audits back into Build and QA.

PHASE	PURPOSE	PRIMARY OUTPUT
1. Scope	Define the agent's intended outcome, operating blueprint, and delivery team.	Approved agent definition and implementation blueprint
2. Training	Gather the information the agent needs and connect the applications required for training.	Validated training sources and application connections
3. Build	Create the agent's instructions, workflows, and reusable skills.	Functional agent configuration and workflow
4. QA	Test the agent, collect human feedback, and document quality findings.	Validated agent and audit report
5. Launch	Verify operational safeguards, introduce human oversight, and audit live conversations.	Production-ready agent with ongoing governance

2.1 The continuous improvement loop

Monitoring results, human interventions, conversation audits, user feedback, and operational incidents feed back into Build and QA. Instructions, workflows, skills, training sources, and controls are revised as needed, retested, documented, and released through a governed change process. Section 13 sets out which signal returns to which phase.

2.2 Phases close on evidence

A phase is not finished when the work is done. It is finished when its exit criteria are met and a named person accepts them. An unnamed acceptor is how a phase drifts for weeks with everyone assuming somebody else has signed off.

PHASE	EXIT CRITERIA	WHO ACCEPTS
1. Scope	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.	Sponsor and agent owner
2. Training	Sources are approved, application connections work, permissions are validated, and the agent's training context traces back to authoritative sources.	Functional Expert and Agent Integration Specialist
3. Build	Core instructions are implemented, the primary workflow operates end to end, and the required skills are available and tested.	Technical Agent Builder and agent owner
4. QA	Critical issues are resolved, remaining limitations are documented and accepted, human reviewers approve the experience, and the audit report supports a clear launch decision.	Functional Expert and sponsor
5. Launch	Safeguards are verified, monitoring is active, human oversight is staffed, conversation audits are scheduled, and owners are assigned for issues discovered after launch.	Sponsor and security reviewer

Four valid answers at every gate: proceed, narrow the scope, do more discovery, or hold. A hold is a useful outcome — it is far cheaper than a failed QA phase.

3. The fourteen steps

Each step produces a named deliverable. Together these deliverables form the record of the build, and they are what the setup documentation in Appendix C is assembled from.

#	STEP	DELIVERABLE	PRIMARY OWNER
1	Define Agent — Outcome	Agent outcome statement and measurable success criteria	Functional Expert
2	Build Blueprint	Approved agent blueprint	Agent owner
3	Assign Team	Named delivery team with assigned ownership	Sponsor
4	Collect Data Sources for Training AI	Curated and approved training-source library	Functional Expert
5	Connect to Applications for Training	Validated application connections and access map	Agent Integration Specialist
6	Build Instructions	Version-controlled agent instructions	Technical Agent Builder
7	Develop Workflow	Implemented agent workflow	Technical Agent Builder
8	Setup Skills	Tested skill and tool configuration	Technical Agent Builder
9	Run Simulated Tests	Test results with identified defects and risks	Functional Expert
10	Human Feedback	Consolidated feedback and prioritized revisions	Agent Users
11	Build Audit Report	Agent QA and audit report	Agent owner
12	Verify Security / Compliance	Security and compliance launch approval	Security reviewer
13	Monitor / Human in the Loop	Operational monitoring and human-oversight process	Operational owner
14	Audit Conversations	Recurring conversation-audit findings and action plan	Functional Expert

Primary owner is not sole participant. Every step involves more than one role. The owner is the person accountable for the deliverable existing and being fit for the next phase.

4. Where the work happens on raia

Two products carry the build. Everything else your team will use sits inside raia cX.

PRODUCT	WHAT IT IS	USED IN
raia cX	The agent platform. Inside it, the Launch Pad is where each agent is built, configured, and managed — identity, instructions, skills, security, launch state, and reporting. Its tabs configure the agent; they are not a project tracker.	Build, QA, Launch
raia Command	The knowledge layer. Bring source material together, run the library of connectors to common business systems, and process it into a trained vector store the agent retrieves from.	Training

4.1 Inside raia cX

AREA	WHAT IT DOES
Launch Pad	Where the agent is built, configured, and managed.
Copilot	Test the agent, inspect retrieved sources, rate and comment on responses, monitor live conversations, and take over when needed.
Chat	The customer-facing surface, once the evidence supports opening one.
Skills	How you expand what the agent can do over time — communication channels, integrations and actions, knowledge and retrieval, admin and security.
API	Drive agents from your own applications and processes.

4.2 Which phase happens where

PHASE	WHERE IT HAPPENS
1. Scope	Workshops and your own documentation. No platform work at all — and this is the phase that most often decides whether the agent succeeds.
2. Training	raia Command. You collect and curate; Command runs the connectors and processes everything into a trained vector store.
3. Build	raia cX, in the Launch Pad.
4. QA	raia Copilot — simulated tests, source inspection, ratings and comments, and the evidence behind the audit report.
5. Launch	Your chosen channel, watched from Copilot. Security sign-off, live monitoring, human takeover, and recurring conversation audits.

4.3 Instructions are not knowledge

raia distinguishes the agent's instructions from its knowledge. Instructions define behavior, boundaries, response expectations, and judgment. The knowledge layer supplies the facts, procedures, and reference material the agent retrieves. A successful deployment needs both, and almost every defect traces to one or the other. Diagnosis always begins by deciding which.

4.4 Connecting to your systems

raia connects to other applications in five ways. Choose the pattern with your Agent Integration Specialist; the decisions that matter to the business are in section 9.2.

PATTERN	WHAT IT DOES
MCP — applications to raia	Other applications connect to your raia agent and call it as a tool.
MCP — raia to applications	Your raia agent connects out to third-party tools at runtime.
Functions	The agent calls one of your APIs mid-conversation and uses the result.
Webhooks	raia pushes an event out to your systems after a message or conversation.
raia API	Create, manage, test, and deploy agents from your own applications.

Scheduled or multi-step orchestration is normally handled by a workflow engine such as n8n or your own standard tool.

5. The delivery team

raia defines four roles, each with its own training path and its own ceiling of technical engagement. Name a person for each before Scope begins. Two further names — a sponsor and an agent owner — carry accountability rather than a training path.

ROLE	WHO THIS IS	WHAT THEY OWN	PHASES	EFFORT
Functional Expert	Your support domain specialist	What a correct answer looks like, which source wins a conflict, and whether the agent is right	Scope → QA	~20 hrs
Agent User	The people who work the queue today	How the work really happens, realistic testing and ratings, and takeover in Copilot	Scope, QA, Launch	~12 hrs
Technical Agent Builder	Your developer or prompt engineer	System prompts and instructions, functions, webhooks, the API, and n8n workflow integration	Build → QA	~10 hrs
Agent Integration Specialist	Your IT, DevOps, or platform engineer	Connectors and data ingestion, MCP setup, identities and permissions, Live Chat SDK and security	Training, Build, Launch	~8 hrs
Sponsor	The business leader who owns the outcome	Whether the value justifies the investment; accepts each phase and removes blockers	All gates	~3 hrs
Agent owner	The project coordinator	Coordinates the four roles, keeps the decisions and the evidence, and runs the beta	All phases	~10 hrs

5.1 Training paths

Each of the four roles has a role-based training path at docs.raiaai.com. Start them during Scope: there is no platform work in that phase, so it is free time to bring each named person up to speed before the work that needs them begins.

PATH	FOR	COVERS	LENGTH
Functional Expert	Business users — support, sales, finance, legal	High-level agent building, prompt writing and refinement, knowledge base training, automation goals and escalation	6 modules
Agent User	Operators of Copilot	The Copilot interface and takeover, feedback and rating, monitoring	3 modules
Technical Agent Builder	Developers and prompt engineers	System prompts and instructions, APIs, functions and webhooks, n8n workflow integration	6 modules
Agent Integration Specialist	IT, DevOps, platform engineering	Connectors and data ingestion, MCP server setup, Live Chat SDK and security	3 modules

5.2 Decision rights

Naming people is not the same as assigning authority. For every recurring question, one person decides. Write those down before Training begins.

QUESTION	WHO DECIDES
What is a correct answer in this domain?	Functional Expert
Which source wins when two sources disagree?	Functional Expert, with the source owner
Is the agent useful in the real workflow?	Agent Users
Is this integration safe to build?	Agent Integration Specialist and the security reviewer
Is the agent ready to leave a phase?	The named acceptor for that phase (section 2.2)
Should scope expand, hold, or stop?	Sponsor

The role that decides the project: the Functional Expert. If that person cannot protect a recurring block of time from Scope through QA, narrow the scope until it fits the time they actually have — or wait until it can be protected. Name a deputy at the same time; an absence during QA stalls everything downstream.

6. Effort and schedule

A first enterprise-grade agent typically represents around 100 combined hours of effort across your team and raia — roughly 60 hours on your side and 40 on ours. This is an illustrative planning average for a narrowly scoped first agent, not a fixed estimate and not a calendar commitment.

PHASE	YOUR TEAM	raia	WHERE THE TIME GOES
1. Scope (steps 1–3)	~10 hrs	~6 hrs	Workshops, the outcome statement, the blueprint, and naming the team
2. Training (steps 4–5)	~18 hrs	~10 hrs	Locating, curating, and validating sources; agreeing and testing connections
3. Build (steps 6–8)	~10 hrs	~14 hrs	Instructions, taxonomy, workflow, and skills — the one phase raia carries more of
4. QA (steps 9–11)	~16 hrs	~8 hrs	Extracting tickets, running tests, giving feedback, and writing the audit report
5. Launch (steps 12–14)	~6 hrs	~2 hrs	Security verification, standing up monitoring, and the first conversation audits
Total	~60 hrs	~40 hrs	

6.1 Effort is not schedule

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

WHAT STRETCHES A TIMELINE	WHY
Waiting on a ticket export	Usually needs a data owner and a privacy decision. Start it in week one.
Unowned or unpublished knowledge	Nobody can approve a source that has no owner, and nothing downstream can proceed without approved sources.
Security review started late	Access approvals cannot be compressed. Involve the reviewer during Scope, not at step 12.
Expert time that keeps slipping	The Functional Expert is on the critical path from Scope through QA.
Approvals waiting on one person	Name a deputy for every acceptor identified in section 2.2.

Plan on readiness, not duration. Set the next workshop against the completion of a deliverable, not against a date. A phase that closes two weeks late but closes properly is far cheaper than one that closes on time and unravels during QA.

7. Before Scope begins

Four things should be in motion before the first workshop. Two of them have long lead times and should start immediately, regardless of when the phase that needs them arrives.

WHAT	WHAT IS NEEDED	LEAD TIME
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 against real calendars, and training paths started.	Now
Knowledge assets	A list of the sources that matter, who owns each one, and whether it is current. Access for the people who will curate them during Training.	1–2 weeks
Ticket history	An export of historical support tickets. Needs a data owner and usually a privacy decision. The full specification is in Appendix B.	2–4 weeks
System access	The applications the agent may need to read from or act on, the owner of each, and whether a connector, API, or MCP path already exists.	2–6 weeks

7.1 Decisions to make up front

- Which support domain is the candidate — one, not several.
- Who the first users are, and whether they are internal.
- Where the historical tickets live, and who is able to extract them.
- Whether any of this requires a privacy or legal review, and who initiates it.

Two items set the schedule. The ticket export and system access. Assign an owner and a date to both before the first workshop ends, whatever else is still undecided.

8. Phase 1 — Scope

Scope defines why the agent is being created, what outcome it must achieve, how it will operate, and who is responsible for delivering it. It establishes a shared foundation before any training or development begins. It is the cheapest phase to get right and the most expensive to skip.

STEP	BREAKDOWN	DELIVERABLE
1. Define Agent — Outcome	Define the business outcome, intended users, primary use cases, boundaries, success criteria, and the conditions requiring escalation to a human.	Agent outcome statement and measurable success criteria
2. Build Blueprint	Document the agent's proposed experience, required capabilities, knowledge needs, application dependencies, workflow stages, governance requirements, and expected handoffs.	Approved agent blueprint
3. Assign Team	Identify the business owner, subject-matter experts, builders, reviewers, security or compliance stakeholders, and operational owners. Clarify decision rights and responsibilities.	Named delivery team with assigned ownership

8.1 Step 1 — Define Agent — Outcome

A use case describes a result, not a feature. Objectives such as “build a support chatbot” or “use AI to answer questions” are not use cases. Define who needs help, which work is in scope, what outcome should occur, and what the agent should produce.

Outcome statement 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

- The current baseline: ticket volume, handling time, resolution rate, escalation rate, backlog, or service quality.
- What should improve, and roughly by how much.
- Where the measurement comes from, and who collects it.
- What result would justify expanding the agent — and what result would justify rescoping or stopping.

The first boundary

- One support domain.
- One primary user group, normally internal.
- A limited intent set, written down.
- Capability: answer, guide, retrieve, summarize, or draft. Write actions and autonomy come later, on evidence.
- Explicit exclusions: what this agent will not handle at all.

Conditions that require a human

- Commitments, exceptions, and approvals.

- Sensitive, high-risk, or emotionally charged situations.
- Anything outside the written intent set.
- What the agent says and does when knowledge or a system is unavailable.
- Stop conditions that require the agent to be paused, restricted, or disabled.

Baseline first. If nobody can state the current handling time or resolution rate, there is nothing to measure the agent against during QA, and no defensible basis for the expansion decision at the end.

8.2 Step 2 — Build Blueprint

The blueprint is not a design document that goes unread. Training uses its knowledge needs and application dependencies; Build uses its workflow stages and handoffs; QA tests against its proposed experience and governance requirements.

BLUEPRINT SECTION	WHAT TO DOCUMENT
Proposed experience	How a request arrives, what the agent does, and what the user sees at each point.
Required capabilities	What the agent must be able to do in order to reach the outcome defined in step 1.
Knowledge needs	Which subject areas the agent has to be grounded in, and to what depth.
Application dependencies	Which systems it needs to read from, and which it may need to act on.
Workflow stages	The end-to-end path, including decision points, approvals, and exception routes.
Governance requirements	Approvals, data rules, retention expectations, and audit obligations.
Expected handoffs	Where a person takes over, what context travels with the handoff, and what happens next.

8.3 Step 3 — Assign Team

Section 5 defines the four raia roles and their training paths. Step 3 is where those roles are attached to real people, with real availability and explicit decision rights.

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

8.4 Preparing the wider team

Adoption is a readiness activity, not a communications exercise at the end. Before the agent reaches real users, the affected team should understand:

- Why the agent is being introduced, and which outcome it should improve.
- Which work the agent performs and which decisions remain human.
- What the agent does not know, cannot access, or is not permitted to do.
- When a person must verify an answer before relying on it.
- How to rate, explain, and categorize agent responses.
- When and how a person takes over a conversation.
- Which information must never be entered.
- How a suggested improvement becomes an approved change.

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

9. Phase 2 — Training

Training prepares the knowledge and application context the agent needs to perform its role. It focuses on collecting appropriate source material and securely connecting the systems that provide relevant information or examples. This is the heaviest phase for your team, and almost none of the work is technical.

STEP	BREAKDOWN	DELIVERABLE
4. Collect Data Sources for Training AI	Identify, collect, organize, and validate the documents, examples, policies, procedures, FAQs, transcripts, and other approved sources needed to train or ground the agent. Remove obsolete, duplicated, restricted, or low-quality material.	Curated and approved training-source library
5. Connect to Applications for Training	Connect the approved applications, repositories, and data systems that provide training context. Confirm access permissions, data boundaries, synchronization requirements, and source ownership.	Validated application connections and access map

9.1 Step 4 — Collect data sources

Your sources do not need to arrive AI-ready. Your responsibility is to collect, curate, upload, or connect the relevant material. raia Command performs the cleaning, labeling, processing, chunking, and embedding that turns it into a usable vector store, and already includes a library of connectors to common business systems.

What to collect

SOURCE CATEGORY	WHAT WE NEED FROM IT	WHO NORMALLY OWNS IT
Product and service knowledge	Manuals, feature documentation, supported versions, service descriptions, technical references	Product or documentation team
Support procedures	Triage, troubleshooting, diagnostic sequences, issue classification, resolution paths	Support operations
Policies and commitments	Eligibility, warranty, service terms, privacy, and approved commitment language	Policy, legal, or service owner
Known issues and changes	Incident guidance, workarounds, releases, deprecated behavior, maintenance notices	Engineering or release management
Existing knowledge base	FAQs, approved articles, help-center material, internal support 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, handoff rules	Support operations
Terminology	Product names, acronyms, synonyms, error wording, user-language variants	Functional Expert

Broad, relevant coverage is generally beneficial, because the agent can reason across more of your organizational knowledge than it can from a single source. Testing then tells you which sources improve the agent and which should be removed, corrected, or supplemented.

Curation rules

- Remove what is obsolete, duplicated, restricted, or low quality before it is trained.
- Name a business owner for every source. A source without an owner cannot be approved and cannot be corrected later.
- Decide which source wins when two sources disagree, and record the decision.
- Limit each connection to the correct products, versions, teams, languages, and content states.
- Agree how obsolete content is retired, and how emergency guidance is published and expired quickly.

The source of truth never moves. The originating application remains authoritative. The vector store is a processed derivative. Correcting an answer inside a conversation fixes nothing — the change belongs in the originating system, and flows back through Command.

Closing gaps

The first training cycle is not expected to be complete. Uploading or connecting sources and then testing the agent is what reveals the gaps.

WHAT TESTING REVEALS	WHAT YOU DO ABOUT IT	WHO OWNS IT
The agent has no answer	Add another source, or write approved content that does not yet exist	Functional Expert
Two sources disagree	Decide authority at the source, then correct or remove the loser	Source owner
A source adds noise	Remove or filter it — more is not automatically better	Functional Expert
A procedure is unclear to the agent	Write a clearer derivative guide or an explicit decision path	Functional Expert
Wrong product or version answered	Improve source scope, filters, and metadata	Source owner and raia
The right content exists but is not retrieved	Adjust source organization, separation, or training structure	raia
The request cannot be supported at all	Escalate to a person, or exclude it from scope deliberately	Support lead

9.2 Step 5 — Connect to applications

System connections serve two distinct purposes, and confusing them is the most common architecture mistake in a first deployment.

PURPOSE	WHAT HAPPENS	SUPPORT EXAMPLE
Knowledge training	Documents and data from files, databases, SaaS applications, and repositories are processed into the vector store.	The agent learns the how, why, and what behind approved support procedures.
Real-time access	During a conversation, the agent uses a scoped call to obtain current, case-specific data.	The agent retrieves a particular ticket, entitlement, order status, or incident.

The strongest support workflows combine both: retrieve the live ticket, then reason about it using trained procedural knowledge.

Hybrid data placement

DATA TYPE	RECOMMENDED TREATMENT
Stable, approved procedures, policies, product knowledge, and reference content	Retain in the vector store for reliable retrieval, deeper reasoning, and scalable reuse.
Personal data, sensitive records, entitlements, live status, user-specific operational data	Keep in the source systems and retrieve through scoped access only when needed.
Data needed only during the current interaction	Apply minimum-necessary retention.
Credentials and secrets	Never in training content, and never in conversation content.

What to confirm for every connection

Record all of the following before a connection is built. Your Agent Integration Specialist and security reviewer approve identity, permissions, credentials, network path, and audit approach.

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	Environment (dev / test / production)	Trigger and expected response
Timeout behavior	Missing-record behavior	Access-denied behavior
Invalid-response behavior	Duplicate prevention	Human approval point
Logging and audit	Credential owner	How to disable it

Least privilege. Give the agent only the access it needs to perform the approved job, using a distinct service identity per connection wherever the target system supports one.

Exit criteria. The required sources are approved, the relevant application connections are working, permissions are validated, and the team can trace the agent's training context back to authoritative sources.

10. Phase 3 — Build

Build converts the approved scope and training context into an operational agent. Nothing here should be discovered during configuration: every input was decided during Scope or Training.

STEP	BREAKDOWN	DELIVERABLE
6. Build Instructions	Create the agent's role, goals, operating rules, response standards, constraints, escalation behavior, tool-use guidance, and handling requirements for ambiguous or unsupported requests.	Version-controlled agent instructions
7. Develop Workflow	Design the end-to-end workflow, including triggers, decision points, information retrieval, application actions, approvals, exception paths, human handoffs, and completion criteria.	Implemented agent workflow
8. Setup Skills	Configure the specialized skills, tools, reusable procedures, integrations, and knowledge-retrieval capabilities required for the agent's assigned use cases.	Tested skill and tool configuration

10.1 Step 6 — Build instructions

Instructions are the agent's operating rules. They are versioned, reviewed, and changed deliberately — not edited mid-conversation.

INSTRUCTION AREA	WHAT IT DEFINES
Role and goals	What the agent is for, who it serves, and what it is trying to achieve.
Operating rules and constraints	How it behaves, what it must never do, and where its authority ends.
Response standards	Required format, level of detail, tone, language, and grounding behavior.
Escalation behavior	When it hands off, what it says, and what context it collects first.
Tool-use guidance	When to call a live system, with what parameters, and how to handle the result.
Ambiguous or unsupported requests	When to ask a clarifying question, when to decline, and when to route to a person.

10.2 Step 7 — Develop workflow

Do not automate the legacy process. A workflow designed for people contains navigation, copying, searching, and waiting that an agent does not need. At the same time, a person may apply informal judgment or hold system access that must be made explicit before an agent can perform the work.

The method

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

WORK TYPE	PERFORMED BY	WHAT IT COVERS
Knowledge and reasoning	The agent	Retrieves trained knowledge, interprets context, and recommends an answer or a next step.
Current-data retrieval	A live call	Pulls the current ticket, account, order, entitlement, incident, or repository item mid-conversation.
Deterministic system action	A system call or workflow	Performs a defined update, initially behind human approval wherever risk requires it.
Human judgment	A person	Exceptions, empathy, sensitive decisions, commitments, approvals, and anything unsupported.

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	

Watch for hidden work. Informal expert judgment and undocumented system access must be made explicit before an agent can perform a step. Both only surface when someone narrates a real ticket, rather than describing the process in the abstract.

10.3 Step 8 — Setup skills

Skills are how you expand what the agent can do. Enable the minimum set the approved workflow requires at this stage, and add capability later on evidence rather than by default.

SKILL CATEGORY	WHAT IT ADDS
Communication	The channels through which the agent talks to users.
Integrations and actions	Pushing and pulling data from external systems, and calling tools mid-conversation.
Knowledge and retrieval	Extending what the agent can reach beyond its trained knowledge.
Admin and security	Monitoring, auditing, scoring, feedback, escalation, retention, and usage controls.

10.4 Escalation and human coverage

Human involvement is a design, not a failure mode. raia provides two mechanisms, configured in different places:

MECHANISM	HOW IT WORKS
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.

For each of the following situations, six decisions must be agreed before the agent meets real users: 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, and what happens when no one is available.

SITUATION	WHY IT MATTERS
Someone explicitly asks for a person	The most common trigger, and the one users judge you on.
There is no approved answer	Determines whether a gap becomes a safe handoff or a fabricated answer.
Sources conflict or are incomplete	Requires human review and a source correction, not a guess.
Troubleshooting has failed repeatedly	Needs a stop condition and a structured handoff.
Sentiment turns negative or the situation is sensitive	Tone, priority, and human coverage all change.
A restricted topic or prohibited action comes up	Refusal, alert, and in some cases an incident path.
An account or system change is required	Human approval or an authorized workflow.
An integration is unavailable	A fallback process and an accurate status message — never a false claim of success.
A security or privacy concern appears	Immediate restriction, alert, evidence, and a named owner.

The decision teams skip: what happens when no human is available. Answer it for all nine situations before the agent meets real users.

10.5 Setting the control profile

Your existing controls already cover most of what an agent needs. Review the agent through three familiar lenses, then scale the depth to its actual exposure.

LENS	CONTROLS TO INHERIT OR ADAPT
The agent as an application	Ownership, authentication, access, environments, testing, release, monitoring, incident response, rollback.
The agent as a data system	Classification, source authority, least privilege, privacy, residency, retention, deletion, lineage, logging.
The agent as a workforce participant	Role definition, instructions, conduct, communication standards, supervision, performance measures, escalation.

CONTROL AREA	BASELINE INTERNAL AGENT	HIGHER-OVERSIGHT AGENT
Access	Named users, minimum roles	Role restriction, periodic review, separation of duties
Knowledge	Approved sources with named owners	Classification, filtering, audit, source-change testing
Runtime data	Minimum necessary live access	Fine-grained scopes, stronger monitoring, stricter retention
Oversight	Feedback and available escalation	Active monitoring, defined coverage, scoring, mandatory takeover
Testing	Representative functional and failure tests	Larger corpus, simulations, adversarial and integration tests
Change	Authorized administrator or owner	A customer approval process before release

An internal agent used by a handful of people does not need the controls of an agent discussing sensitive data with hundreds of customers. Decide the profile deliberately rather than by default.

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.

11. Phase 4 — QA

QA evaluates the agent against expected, unexpected, and high-risk scenarios. Simulated tests reveal repeatable issues; human reviewers evaluate quality, judgment, usability, and alignment with the intended outcome. The phase ends with a documented recommendation, not an impression.

STEP	BREAKDOWN	DELIVERABLE
9. Run Simulated Tests	Test representative use cases, edge cases, failure scenarios, restricted requests, incomplete inputs, workflow exceptions, and escalation paths. Compare results with defined acceptance criteria.	Test results with identified defects and risks
10. Human Feedback	Have business owners, subject-matter experts, reviewers, and representative users assess accuracy, usefulness, tone, consistency, workflow behavior, and the quality of human handoffs.	Consolidated human feedback and prioritized revisions
11. Build Audit Report	Document test coverage, findings, unresolved risks, remediation actions, approvals, limitations, and the launch recommendation.	Agent QA and audit report

11.1 Step 9 — Run simulated tests

For a Support AI Agent, testing draws on your own history. A larger corpus of historical tickets is analyzed to understand the domain; a separate set of roughly 300 representative conversations is what the agent is actually tested against. The number expands when the domain is wider, the language more variable, the integrations deeper, or the risk higher.

What the historical analysis produces

- The intent and issue taxonomy for your support domain.
- Volume and frequency by category, so scope goes where the value is.
- The real language your users type, including error wording and variants.
- Resolution paths that already work, and the points where escalations happen.
- Where live system data is genuinely required.
- Knowledge gaps and conflicts to fix before testing.
- The baseline measures the value case rests on.
- The seed material for the test conversations.

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

What every test case defines

FIELD	DESCRIPTION
User situation	The request and the relevant context.
Preconditions	Agent version, data, system state, identity, and environment.
Expected business outcome	Resolved, progressed, escalated, declined, or another approved result.
Ideal output	Required facts, steps, structure, tone, and source behavior.
Expected live call	The tool called, its parameters, the current result, and how status is handled.
Human behavior	No takeover, optional review, mandatory escalation, or required approval.
Criticality	Ordinary, important, or a hard gate.
Evidence	The conversation, the retrieved sources, the tool result, logs, rating, and reviewer comment.

The test program should not require one exact sentence unless policy demands exact language. It should require the correct outcome, grounded in the correct sources, with the correct human behavior.

Appendix B has the export specification. The ticket export is the longest-lead item in the whole lifecycle. Agree the specification during Scope, not at the start of QA.

11.2 Step 10 — Human feedback

A negative rating is diagnostic evidence, not a verdict. It should always be traced to a cause, and the cause fixed at its origin rather than in the conversation.

WHO ASSESSES	WHAT THEY ASSESS
Functional Expert	Accuracy, domain judgment, and whether the outcome was correct.
Agent Users	Usefulness in the real workflow, tone, consistency, and clarity.
Technical Agent Builder	Instructions, functions, error handling, and environment behavior.
Agent Integration Specialist	Connections, identity, access boundaries, and retention.
Sponsor	Alignment with the intended outcome.
Reviewers	The quality of the human handoffs, and any compliance concern.

WHAT THE FEEDBACK REVEALS	THE UNDERLYING FIX
Wrong or stale fact	Correct or replace the source.
No answer available	Add source content, create approved derivative material, or escalate.
The right content exists but was not retrieved	Improve source scope, organization, or training structure.
Poor tone or format	Refine the instructions and the response examples.
Wrong workflow	Correct the trigger, sequence, parameters, or authority.
Integration failure	Correct authentication, mapping, availability, or error handling.
Incorrect escalation	Adjust the trigger, routing, context, or human coverage.
Unsupported capability	Exclude it, redesign it, connect another system, or keep it human.
Security or privacy concern	Restrict the behavior, remediate access or retention, and retest.

11.3 Measuring the result

Agree how the numbers are counted before anyone reports one. Two teams counting resolution differently will produce two different figures from the same test run.

AI resolution rate = in-scope conversations that reached the intended outcome without human takeover, divided by 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.

The initial maturity benchmark is that roughly 80% of representative, in-scope support tickets are resolved using AI without human takeover. The remaining share is a controlled exception pool: it tells you exactly what is missing — knowledge, an unsupported requirement, an integration limit, or work that genuinely needs human judgment. With sustained use, human feedback, better sources, refined instructions, and improved workflows, the unresolved share can move toward roughly 5%.

These block launch regardless of the percentage: critical data exposure; an unauthorized action taken on a system; fabricated consequential guidance; a mandatory escalation that did not fire; a false claim that a workflow succeeded. High aggregate performance never excuses one of these.

11.4 Step 11 — Build the audit report

The audit report is the artifact the launch decision is made from. It should be readable by someone who was not in the room.

SECTION	WHAT IT RECORDS
Test coverage	What was tested, how many conversations, and across which categories.
Findings	Defects identified, their severity, and their status.
Unresolved risks	What remains open, and what the exposure is.
Remediation actions	What was changed, by whom, and what was retested afterwards.
Approvals	Who reviewed and accepted, and on what date.
Accepted limitations	What the agent will not do well, stated plainly.
Launch recommendation	Proceed, proceed with conditions, narrow, or hold — with the reasoning.

Exit criteria. Critical issues are resolved, remaining limitations are documented and accepted, human reviewers approve the experience, and the audit report supports a clear launch decision.

12. Phase 5 — Launch

Launch moves the agent into controlled production use. It confirms that security and compliance requirements are satisfied, establishes human oversight, and introduces ongoing review of live interactions. Launch is a state, not an event.

STEP	BREAKDOWN	DELIVERABLE
12. Verify Security / Compliance	Confirm access controls, data handling, privacy requirements, retention rules, approved integrations, auditability, content safeguards, and applicable organizational policies before production release.	Security and compliance launch approval
13. Monitor / Human in the Loop	Monitor agent performance, failures, escalations, user feedback, workflow completion, and operational exceptions. Maintain clear human review, intervention, approval, and override procedures.	Operational monitoring and human-oversight process
14. Audit Conversations	Review a representative sample of agent conversations to identify accuracy issues, unsafe behavior, process gaps, emerging user needs, training-data deficiencies, and opportunities for improvement.	Recurring conversation-audit findings and action plan

12.1 Step 12 — Verify security and compliance

AREA	WHAT TO CONFIRM
Access controls	Who can use the agent, who can change it, and whether those rights match the risk.
Data handling and privacy	What the agent may access, display, retain, or train on, and how that is enforced.
Retention rules	Conversation retention, vector-store retention, and deletion procedures.
Approved integrations	Only the connections authorized during Training are live, with the agreed scopes.
Auditability	Activity and conversation logging is in place and reviewable.
Content safeguards	Monitoring for sensitive data exposure, prompt manipulation, and policy violations.
Organizational policy	Any additional requirement your industry, regulator, or internal policy imposes.

12.2 Step 13 — Monitor and keep a human in the loop

Human oversight must be staffed, not merely available. Agree who is covering, during which hours, and what happens outside them.

MONITOR	WHAT TO WATCH FOR
Performance	Whether conversations reach the intended outcome, and how quickly.
Failures and escalations	What triggered them, whether the route was correct, and how they resolved.
User feedback	Ratings, comments, and the reasons given.
Workflow completion	Whether actions actually succeeded in the target system.
Operational exceptions	Timeouts, unavailable integrations, denied access, and duplicates.

Maintain clear procedures for human review, intervention, approval, and override — and make sure the people named in them know they are named.

12.3 Step 14 — Audit conversations

Reviewing a representative sample of live conversations on a schedule is what turns operation into improvement. Audits look for:

- Accuracy issues that testing did not surface.
- Unsafe behavior, including anything approaching the launch blockers in section 11.3.
- Process gaps where the workflow does not match how the work actually arrives.
- Emerging user needs — requests that are becoming common but sit outside the current scope.
- Training-data deficiencies, including stale, missing, or conflicting sources.
- Opportunities to standardize a pattern that is working well.

12.4 Widening exposure

Exposure should increase one step at a time, with human feedback and takeover remaining active throughout.

STAGE	EXPOSURE AND AUTHORITY
Development	Expert testing on a development agent or a copy.
Internal beta	Named Agent Users, with feedback and takeover available.
Supervised pilot	One channel, a limited intent set, active monitoring, and human fallback.
Broader deployment	Expanded audience, volume, channels, products, languages, or authority — each step on evidence.

Before a customer-facing channel opens, all of the following should be true.

The resolution benchmark is met	Escalation, takeover, and staffing are operational
Quality is at or above the existing human baseline	Channel identity, notices, consent, and hours are approved
Customer experience meets the agreed standard	Integrations are reliable, with no false success claims
Knowledge use is grounded, with gaps exposed safely	Safety and privacy tests pass, with no open critical defect
Disable and rollback restore the manual process	The agent owner, support operations, the Technical Agent Builder, and the security reviewer have signed off

Exit criteria. Safeguards are verified, monitoring is active, human oversight is staffed, conversation audits are scheduled, and owners are assigned to resolve issues discovered after launch.

13. Continuous improvement

Launch is not the end of the Agent Factory lifecycle. Monitoring results, human interventions, conversation audits, user feedback, and operational incidents feed back into Build and QA. The value of the loop comes from routing each signal to the phase that can actually fix it — teams that treat every signal as a Build problem end up rewriting instructions to fix what is really a source or scope issue.

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 and 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

13.1 Turning resolved work into knowledge

Human resolutions are a knowledge asset. A completed ticket can trigger a drafted article, but your existing content governance decides review and publication.

OPTION	EXAMPLE PROCESS
Manual	A support representative or knowledge owner writes and publishes the article.
AI-assisted	An agent drafts a richer article from the conversation, and the existing review process approves it.
Automated	An approved workflow generates and publishes content according to conditions you define.

13.2 The operating rhythm

CADENCE	WHAT HAPPENS
Daily, at first	Review conversations, takeovers, and any critical event.
Weekly	Work the gap log, correct sources, and retest what changed.
Monthly	Resolution rate, adoption, escalation patterns, and integration health.
Quarterly	Access review, retention check, and the scope or expansion decision.
On change	Any material source, instruction, or system change triggers a retest.

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. This is why the last row exists.

14. Change management and continuity

14.1 Development and production copies

Deploy separate copies of the agent for development and production. raia versions the agent instructions natively, while Command records source versions and upload timing. The upstream system remains the primary control for continuously integrated content: a change in CRM, support, repository, or knowledge content can alter agent behavior even when the instructions are unchanged.

14.2 Source pipeline controls

CONTROL	RECOMMENDED PRACTICE
Source owner	Name the business or content owner for every connected source.
Publication state	Synchronize only approved or published content where the source system supports it.
Filters	Limit product, version, team, language, date, and content type.
Validation	Check connection, mapping, completeness, expected volume, and processing status.
Monitoring	Detect failed synchronization, stale content, and unexpected source changes.
History	Be able to identify what changed and remove or replace a harmful derivative version.
Retest	Rerun retrieval and regression tests after any material source or pipeline change.

Document the setup. There is no formal handoff requirement, but the setup should be recorded so that transition, maintenance, troubleshooting, and future change do not depend on one person's memory. Appendix C is the checklist.

14.3 A change flow for higher-risk agents

raia does not provide a native change-approval workflow; the authorized administrator or owner controls publishing. Where separation of duties is required, use your existing ITSM, governance, or CI/CD process, and automate through the raia API where that helps.

STAGE	ACTIVITY
Change record	Document the reason, the expected benefit, the owner, and the affected agent.
Impact assessment	Determine whether scope, data, audience, integrations, permissions, or autonomy change.
Development build	Apply the change to a development copy or a new version.
Testing	Run functional, regression, integration, security, and human review appropriate to the risk.
Approval	Follow your required business, technical, security, or compliance process.
Authorized release	The administrator or owner publishes, manually or through API automation.
Observation	Monitor efficacy, feedback, takeover, and incidents after release.
Rollback	Restore or redeploy the last acceptable configuration if necessary.

15. A successful first deployment

The first deployment has succeeded when the combined evidence shows all of the following.

DIMENSION	EVIDENCE
Business task	The agent performs a clearly defined task inside the real business environment.
Adoption	Leadership and the intended users accept it as a useful addition to how the work gets done.
Value	It creates measurable improvement against the success criteria written in step 1.
Efficacy	It resolves the agreed share of representative in-scope support tickets without human takeover.
Human safety net	The remaining work is reliably handled through escalation, takeover, or a deliberate manual process.
Knowledge	Source coverage, training, gaps, and source-of-truth ownership are understood and owned.
Integration	Live connections, workflows, access, failures, and fallbacks operate as designed.
Controls	Security, privacy, retention, logging, and oversight are appropriate to the use case.
Improvement	The team can identify gaps, diagnose causes, and route each one to the right phase.
Continuity	Setup, controls, dependencies, limitations, and operations are documented.
Decision	The company can now choose to expand, open a channel, build another agent, keep refining, or stop.

The goal. Not a perfect AI demonstration, but a working human-agent system that resolves most in-scope work, transfers exceptions to people safely, and creates measurable business value.

16. Optional modules

The lifecycle in this guide applies to a first Support AI Agent. Larger or more regulated organizations may add either of the following. Both are optional and independent of any individual agent project — do not impose them on a first, narrow build.

MODULE	USE WHEN
Enterprise Agent Governance	You are running multiple agents and want portfolio scoring, an agent registry, agent tiers, a center of excellence, reusable standards, or API-connected approvals.
Higher-Risk Agent Controls	The agent is customer-facing, high-volume, financially or operationally consequential, deeply integrated, sensitive, regulated, or materially autonomous.

Appendix A · Preparation and decision workbook

Use these tables as a shared working record throughout the lifecycle. Link to controlled source documents rather than copying sensitive information or credentials into the workbook.

A.1 Initiative summary

FIELD	RESPONSE
Initiative name	
Business unit / support team	
Sponsor	
Agent owner	
Support problem	
First user group	
First scope	
Explicit exclusions	
Intended business outcome	
Current phase	Scope / Training / Build / QA / Launch
Status	Preparing / In progress / Blocked / Paused / Complete

A.2 Delivery team

ROLE	NAMED PERSON	DEPUTY	DECISION RIGHTS	TIME CONFIRMED
Functional Expert				Yes / No
Agent User				Yes / No
Technical Agent Builder				Yes / No
Agent Integration Specialist				Yes / No
Sponsor				Yes / No
Agent owner				Yes / No
Security reviewer				Yes / No
Operational owner				Yes / No

A.3 Step 1 — Outcome and success criteria

QUESTION	APPROVED ANSWER
Who is the first user?	
What issue class is in scope?	
What happens today?	
What should be true when the interaction ends?	
What exact output should the agent produce?	
What remains human-only?	
What must the agent never do?	
What should happen when it cannot complete the task?	
Current baseline (volume, handling time, resolution rate)	
Target after stabilization	
Measurement source and owner	
Result that would justify expansion	
Result that would justify rescope or stop	

A.4 Step 1 — Guardrails

ID	THE AGENT MUST NEVER...	RISK CONTROLLED	SAFE BEHAVIOR	TEST ID
G-01				
G-02				
G-03				
G-04				
G-05				

A.5 Step 2 — Blueprint

SECTION	DOCUMENTED DECISION
Proposed experience	
Required capabilities	
Knowledge needs	
Application dependencies	
Workflow stages	
Governance requirements	
Expected handoffs	

A.6 Step 4 — Source inventory

ID	SOURCE / APPLICATION	BUSINESS OWNER	AUTHORITY	SENSITIVE?	STATUS
K-01			Primary / Supplemental	Yes / No	
K-02			Primary / Supplemental	Yes / No	
K-03			Primary / Supplemental	Yes / No	
K-04			Primary / Supplemental	Yes / No	
K-05			Primary / Supplemental	Yes / No	

A.7 Step 4 — Gap log

ID	MISSING OR WEAK KNOWLEDGE	HOW IDENTIFIED	PROPOSED FIX	OWNER	RETEST
KG-01					
KG-02					
KG-03					
KG-04					

A.8 Step 5 — Connection record

FIELD	CONNECTION 1	CONNECTION 2
Business purpose		
Data fields used		
Fields prohibited		
Service identity		
Authentication method		
Read / draft / write authority		
Environment		
Timeout behavior		
Missing-record behavior		
Access-denied behavior		
Duplicate prevention		
Human approval point		
Logging and audit		
Credential owner		
Disable procedure		
Security approval		

A.9 Steps 6–8 — Build record

AREA	APPROVED CONTENT OR LINK
Role and goals	
Operating rules and constraints	
Response standards	
Escalation behavior	
Tool-use rules	
Ambiguous-request handling	
Current instruction version	
Taxonomy summary	
Skills enabled and why	
Workflow diagram or link	

A.10 Step 6 / 13 — Escalation and coverage

SITUATION	AGENT MESSAGE	CONTEXT PASSED	DESTINATION	COVERAGE	NO-HUMAN FALLBACK
Explicit request for a person					
No approved answer					
Conflicting knowledge					
Repeated failed resolution					
Negative sentiment					
Restricted topic					
System change required					
Integration unavailable					
Security or privacy concern					

A.11 Step 9 — Test case log

ID	CATEGORY	USER SITUATION	EXPECTED OUTCOME	CRITICAL?	RESULT
T-001	Representative			Yes / No	Pass / Fail
T-002	Wording variant			Yes / No	Pass / Fail
T-003	Edge case			Yes / No	Pass / Fail
T-004	Restricted request			Yes / No	Pass / Fail
T-005	Escalation path			Yes / No	Pass / Fail
T-006	Live-data call			Yes / No	Pass / Fail
T-007	Failure scenario			Yes / No	Pass / Fail
T-008	Regression			Yes / No	Pass / Fail

A.12 Step 11 — QA summary

MEASURE	RESULT	TARGET OR RULE
Conversations tested		~300 or the agreed scope-based total
AI resolution rate		Initial benchmark ~80%
Human takeover rate		Diagnostic — review the reasons
Knowledge-gap rate		Track and reduce
Integration success rate		Agreed measure
Mandatory escalations that fired correctly		All critical cases pass
Critical safety or privacy failures		Zero unresolved
Unauthorized actions		Zero
False workflow-success claims		Zero
Open critical defects		Zero before launch

A.13 Phase acceptance log

PHASE	DECISION	CONDITIONS OR ACTIONS	ACCEPTED BY	DATE
1. Scope	Proceed / Narrow / Discover / Hold			
2. Training	Proceed / Narrow / Discover / Hold			
3. Build	Proceed / Narrow / Discover / Hold			
4. QA	Proceed / Narrow / Discover / Hold			
5. Launch	Proceed / Narrow / Discover / Hold			

Appendix B · Historical ticket export specification

The ticket export is the longest-lead item in the lifecycle. It normally requires a data owner and a privacy decision, and both take longer than teams expect. Agree this specification during Scope and start the request immediately.

ITEM	WHAT TO SPECIFY	TYPICAL ANSWER
Systems	Every helpdesk or ticketing system in scope for this agent.	One primary system; note any secondary queue
Date range	Enough history to cover seasonality and product change.	6–12 months
Statuses	Which ticket states to include.	Resolved and closed; escalated included; spam and test excluded
Fields	The columns the analysis needs.	Ticket ID, created date, subject, full conversation thread, category or queue, resolution or closing note, escalation flag, handle time, product and version, language, channel
Exclusions	Content that must not be included.	Out-of-scope products, teams, or languages; restricted accounts
Personal data	How personal data is handled before analysis.	Redacted, pseudonymized, or access-restricted — decided with your privacy owner
Volume	Approximate row count, so the analysis can be sized.	Stated up front, not discovered mid-analysis
Format	A format your analysts and raia can both read.	CSV or JSON export, one row per ticket with the thread preserved
Owner	The named person who produces the export.	Named, with a backup
Delivery	How the export is transferred and where it is stored.	Agreed with your security reviewer

B.1 Two corpora, one export

The same export normally serves two purposes. The larger body of tickets is analyzed to understand the domain — intents, volumes, language, resolution paths, escalation patterns, and knowledge gaps. From it, roughly 300 representative conversations are selected as the test set the agent is actually measured against. Keep the two separate in your records; they answer different questions.

If the export cannot be produced, the analysis can run on a smaller sample. But the baseline measures and the test corpus both weaken, and the resolution figure becomes much harder to defend at the launch decision. Raise this early rather than absorbing it quietly.

Appendix C · Setup documentation checklist

Record the following so that transition, maintenance, troubleshooting, and future change do not depend on one person's memory. This can live inside your existing application, service, or change-management documentation.

ITEM	LINK OR RECORD	OWNER	UP TO DATE
Agent purpose, owner, scope, users, and intended outcome			Yes / No
Approved blueprint			Yes / No
Current system instructions and version			Yes / No
Taxonomy or knowledge-graph summary			Yes / No
Command projects, uploads, connectors, and source owners			Yes / No
Enabled skills and the purpose of each			Yes / No
Integrations, authentication method, and credential owner (no secrets)			Yes / No
Escalation configuration and human coverage			Yes / No
Logging, retention, and audit configuration			Yes / No
Platform roles, administrator, owner, and backups			Yes / No
Development and production copies			Yes / No
Historical analysis and test corpus			Yes / No
QA audit report and accepted limitations			Yes / No
Current resolution rate and gap analysis			Yes / No
Regression test set			Yes / No
Source update and synchronization process			Yes / No
Monitoring and conversation-audit schedule			Yes / No
Incident, immediate disable, manual fallback, and rollback procedures			Yes / No
Known limitations and dependencies			Yes / No
Key decisions, approvals, and contacts			Yes / No

References

TOPIC	SOURCE
Role-based training paths	docs.raiaai.com/paths
raia cX and the Launch Pad	docs.raiaai.com/products/cx
raia Command	docs.raiaai.com/products/command
raia Copilot	docs.raiaai.com/products/copilot
Skills directory	docs.raiaai.com/products/skills
Human in the loop	docs.raiaai.com/products/human-in-the-loop
MCP integrations	docs.raiaai.com/developers/mcp
Security and compliance	docs.raiaai.com/security
Agent Factory process lifecycle	Agent Factory Process Lifecycle (companion document)