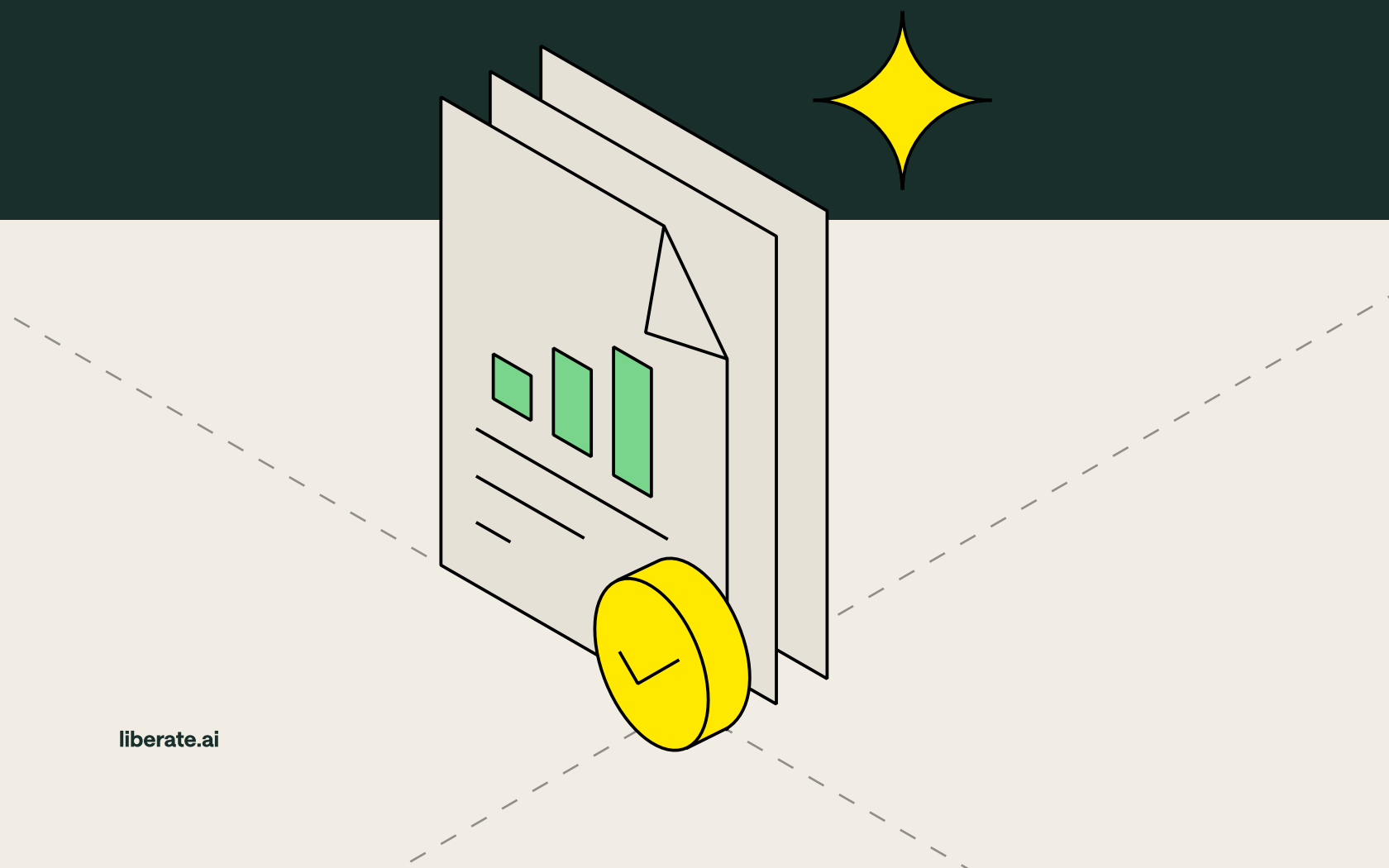
Liberate

From Demo to Production in Weeks

Why Results Are Won or Lost in the Last Mile



liberate.ai

There is a moment in almost every AI demo when the room gets excited.

The agent answers naturally. It understands the question. It moves through the workflow. Someone says, “Wow.”

That moment matters. But it proves surprisingly little.

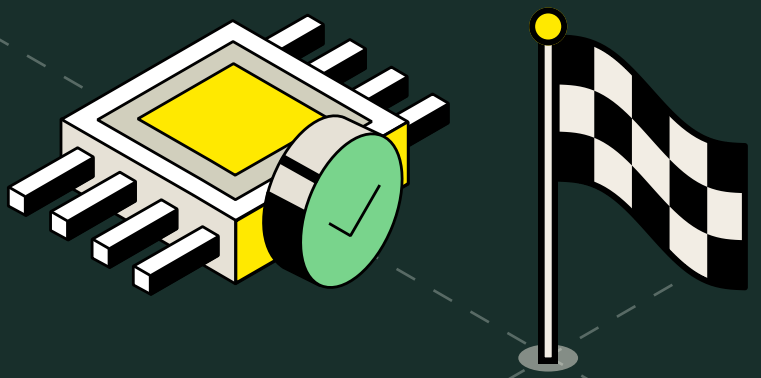
The real test begins when that same agent has to work inside an insurer’s environment, authenticate a caller against the right system, find the right policy, write information back to the right fields, transfer a call to the right queue, satisfy compliance requirements and keep performing when real customers stop following the happy path.

That is the last mile. And in enterprise AI, the last mile is the product.

Getting an AI agent into production means much more than building a good conversational experience. It means making the agent function as part of an insurer’s operating environment, with the systems, controls, testing and human handoffs required to do real work.

The demo shows possibility. Production is the receipt.

In enterprise AI, the last mile is the product.



Liberate

The Smoothest Implementations Start Before Week One

When companies talk about speeding up AI deployment, the conversation often jumps to engineering velocity. How quickly can the agent be configured? How much code can be reused? How fast can integrations be built?

What many insurers don't realize is that speed directly correlates with the quality of preparedness before engineering begins.

Before the build begins, the engineering team works with the insurer to surface the dependencies that could otherwise create starts and stops later: sandbox API access, production credentials, middleware requirements, telephony arrangements, compliance approvals, operational ownership and system documentation.

That preparation can get surprisingly specific.

- ☐ Who owns the telephony infrastructure?
- ☐ How long does provisioning take?
- ☐ Which queues receive transferred calls?
- ☐ What does production API credential provisioning require?
- ☐ Do credentials expire?
- ☐ Is sandbox data representative of production?
- ☐ Who approves the AI agent's voice, tone or greetings?
- ☐ Who receives an alert if something breaks?

Every one of these questions matters when launch day arrives, but they must be answered before implementation is underway. Receipts-driven AI launches identify and remove potential barriers before implementation starts.

WEEK ZERO: GET READY

Before engineering work begins, align on the details that matter.

- ✓ Confirm scope and use cases
- ✓ Identify key stakeholders and owners
- ✓ Verify sandbox API access and progress toward production access
- ✓ Confirm telephony / call forwarding
- ✓ Review required legal and compliance approvals
- ✓ Validate operational dependencies and data access

FIND THE BLOCKERS BEFORE THE CLOCK STARTS.

Production AI Has to Fit Each Insurance Operation

Insurance operations contain thousands of workflow variations that rarely appear in an AI demo.

Consider something as ordinary as a policy number.

One carrier might use a specific prefix and a fixed number of digits. Another might use an entirely different format. That distinction can affect whether an AI agent successfully looks up and authenticates the person on the phone.

The same problem appears throughout the workflow.

An insurer may store a standard data element in a custom field. A claims system may support phone-number lookup while another does not. Middleware may expose only certain endpoints. A transferred caller may need to enter a particular queue, with a screen pop that tells the human agent why the transfer occurred and what has already happened.

CASE IN POINT: POLICY NUMBERS

Small formatting differences can become big workflow problems.

Insurer A: Uses a two-letter prefix followed by eight digits

Example:

AB12345678

Insurer B: Uses 10 digits with no prefix

Example:

1234567890

Same workflow. Different data structure.
If the AI expects the wrong format, lookup can fail before the conversation gets started.

These are not edge details. They determine whether the AI implementation works.

That is why a receipts-driven process starts from production-proven use cases and then adapts them to the insurer's environment.

Standard mappings and workflows provide the foundation, but customer-specific field mappings, authentication rules and conversational requirements still have to be aligned.

There is leverage in having deployed the use case before. But additional engineering is still required to make it work for each customer. Both are essential.

Liberate

Reuse What's Known; Engineer What's Different

A blank-sheet implementation sounds flexible. In practice, it can mean reinventing the wheel over and over.

A receipts-driven implementation process takes a different approach.

That matters because production AI has two kinds of complexity.

During the first two weeks, the deployment team aligns the technical build plan with the insurer, using use cases already delivered in production as the starting point. The plan incorporates the conversational experience, system mappings, customer-specific fields and any language that needs explicit approval.

The first is repeatable.

Claims intake has common patterns. Policy servicing has common patterns. Carrier systems have familiar structures. Transfer logic, authentication, greetings, closings and workflow orchestration often have reusable components.

The second is local.

This insurer uses a custom field. That carrier handles policy numbers differently. This call center uses a specific transfer queue. That compliance team requires particular language.

Smooth implementation depends on knowing the difference.

The goal is not to reinvent what has already been proven. It is to spend engineering time on the pieces that genuinely vary. That is one reason an insurance-specific AI implementation can move in weeks rather than months.

WEEKS 1-2: APPROVE DESIGN (AKA: THE BLUE GATE)

Turn your use cases into a detailed, agreed-upon build plan.

- Leverage production-proven templates and use cases
- Align conversational experience (greetings, flows, closings)
- Map systems (e.g., claims and policy systems)
- Define customer-specific field mappings
- Confirm language and scripts

ACHIEVE BUILD-PLAN ALIGNMENT FOR FEWER PRODUCTION SURPRISES.

Build Quality Is Measured by Gates, Not Hope

AI systems create a peculiar kind of temptation. Because the experience feels conversational, it can be easy to judge readiness conversationally.

> **Does it sound good?**
>
> **Did that call work?**
>
> **Would I be comfortable showing this to the customer?**

Those questions are useful. They are nowhere near enough.

That's why a receipts-driven launch uses formal quality gates during implementation.

After the build plan is aligned, a technical design review, known as the blue gate, confirms that the proposed architecture, mappings and customer requirements are ready before the main build proceeds.

QUALITY GATES

Blue gate

Blue: confirms that architecture, mappings and customer requirements are ready.

Green gate

Green: confirms that the initial build meets the desired production-quality bar.

Later, the green gate asks a harder question: Is what we built ready to meet the production-quality bar?

That review goes beyond whether the voice experience sounds right. Product leaders and senior engineers look at functionality, analytics, eventing, alerting and observability before the customer begins user acceptance testing.

The distinction matters.

A customer should not discover that monitoring is incomplete after launch. They should not be the first people to find basic integration failures. They should not have to become the vendor's QA department.

This leads to one of the most important principles in the process:

UAT Should Confirm the Build, Not Discover It.

Liberate

WEEKS 3-6: BUILD + PROVE (AKA: THE GREEN GATE)

Forward-deployed engineers configure your agent, integrate with your systems and rigorously test.

- Configure voice and backend workflows
- Integrate with your systems and endpoints
- Run hundreds of simulated conversations (Agent Arena)
- Conduct human QA and iteratively refine
- Address customer-specific details (e.g., policy number formats, authentication)

CONFIGURE. TEST. REFINE.

What UAT Is, and What It Shouldn't Be

User acceptance testing has a bad habit of becoming something else. A customer gets access to a system, starts trying things and suddenly everyone is debugging. That may be called UAT. It is really outsourced QA.

The customer should not be expected to test the engineers' work.

A receipts-driven launch arrives at UAT differently.

THE HUMAN EAR MATTERS

Automation makes the testing process faster and broader, but it does not eliminate judgment.

The Agent Arena runs tests at scale, allowing teams to pressure-test many scenarios quickly, but it also has limits.

Voice AI is experienced by humans, and humans notice things that might look minor in a technical log. A strange pause. A clumsy authentication flow. An awkward transfer. A response that is technically acceptable but feels wrong.

Small imperfections can erode trust quickly, so automated testing and human testing work together. One expands coverage. The other evaluates the experience.

Production readiness requires both.

Before handoff, the agent should have already gone through internal testing that combines automation with significant human testing.

The Agent Arena can run simulated conversations across defined scenarios at scale, while people on the team also make calls to evaluate behavior that automated simulations may not fully capture.

The customer then receives more than a testing link or a phone number to call. They receive evidence of how their AI already performs.

That can include an Agent Arena testing report, a defined UAT test plan and links to successful recordings or transcripts in the supervisor platform. The customer can reproduce the scenarios, listen to recordings, review transcripts or give feedback on a specific test case.

The objective is to make UAT a confirmation step.

- These are the requirements we agreed on.
- These are the scenarios we built for.
- Here is the evidence that they have already been tested.
- Now confirm that the experience meets your expectations.

WEEKS 7-8: VALIDATE + LAUNCH

You receive the evidence and confirm it meets your needs.

- Provide UAT line
- Share Agent Arena testing report
- Deliver customer test plan
- Provide successful recordings / transcripts in the supervisor platform
- Track and resolve open items in a shared issue tracker
- Customer sign-off on production readiness

CONFIRM THE AGENT MEETS YOUR REQUIREMENTS IN REAL-WORLD SCENARIOS.

Going Live Is Just the Beginning

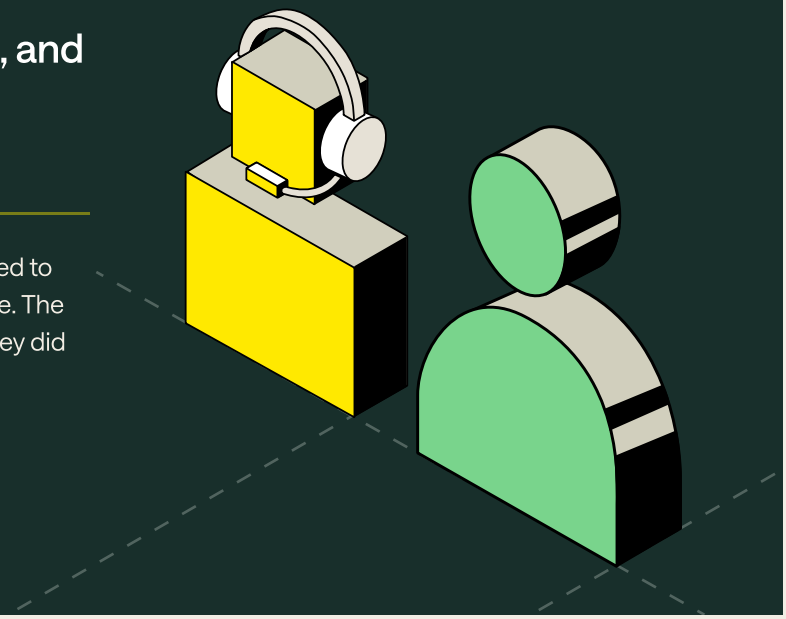
There is another mistake buried in the language of software implementation. The go-live moment is commonly called a “launch,” suggesting that the important work ends when the system goes live.

However, for AI, going live reveals something new: the agent is now operating inside a real organization for the first time. This is where the agent begins encountering the real-world variation that no test plan can reproduce perfectly.

AI IS NOW YOUR COWORKER

The AI is now part of your team, and that framing changes how you think about the last mile.

A coworker has to hand work off cleanly. They need to operate inside the same systems as everyone else. The people around them need to understand what they did and what happens next.



The response to that variation should not be frantic tweaking.

To prepare for that variation, insurers can use a narrow production launch where the experience is turned on for a limited period and reviewed before broader rollout, depending on the insurer’s telephony setup and operating preferences.

If someone hears one bad call and asks for a change, the first step is to determine whether that interaction reflects a broader pattern. Trends must be examined to determine whether an issue is widespread or isolated before changing the experience.

That discipline matters because fixing one unusual interaction can create unintended consequences somewhere else.

Hypercare Shepherds the Last Mile

Once the agent is live, the team enters HYPERCARE – a phase in which engineers closely monitor production behavior. During this phase, a forward-deployed engineer and forward-deployed PM may work directly with the insurer's operational teams, including on site.

They help verify that transferred calls arrive where they should, that screen pops contain the necessary information and that the human receiving the work understands why it was transferred.

This is where "integration" stops being an architecture diagram.

1. A claim reaches a person
2. The right information appears
3. The handoff makes sense
4. The customer keeps moving

That is what receipts-driven implementation looks like when it works.

POST-LAUNCH: HYPERCARE

Go live with confidence and a team by your side.

- Monitor performance and system health
- Validate real-world outcomes
- Resolve issues quickly
- Refine and optimize
- Support operational teams
- Sustain target performance

PROOF IN PRODUCTION.

Where Promises Become Receipts

AI is full of impressive promises.

- Better service.
- Lower cost to serve.
- More capacity.
- Faster response.
- More consistent execution.

Those outcomes matter, but an insurer cannot collect them from a demo.

They appear only when the agent makes it through the last mile and becomes part of the operation.

A receipts-driven insurance AI launch requires:

01 Securing readiness before engineering begins.

02 Starting from patterns that have already worked in production, then adapting them to the insurer's environment.

03 Applying technical quality gates, automated testing, human judgment and structured UAT.

04 Monitoring production behavior and staying close enough to the operation to see whether the handoffs work in the real world.

None of those things will produce the flashiest demo at a conference, but time and again, they produce something better: a working and highly effective AI agent.

In production. Doing the job.



From Demo to Production in Weeks

Why Results Are Won or Lost in the Last Mile

1 **WEEK 0**

Get Ready

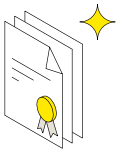


Before engineering work begins, align on the details that matter.

- Confirm scope and use cases
- Verify sandbox API access
- Review legal and compliance approvals

2 **WEEKS 1–2**

Approve Design

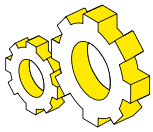


Turn your use cases into a detailed, agreed-upon build plan.

- Leverage production-proven templates and use cases
- Align conversational experience
- Map systems

3 **WEEKS 3–6**

Build + Prove

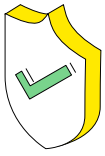


Forward-deployed engineers configure your agent, integrate with your systems, and rigorously test.

- Configure voice and backend workflows
- Run hundreds of simulated conversations
- Conduct human QA and iteratively refine

4 **WEEKS 7–8**

Validate + Launch

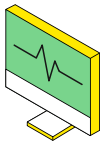


You receive the evidence and confirm it meets your needs.

- Provide User Acceptance Testing line
- Share Agent Arena testing report
- Customer sign-off on production readiness

5 **POST-LAUNCH**

Hypercare



Go live with confidence and our team by your side.

- Monitor performance and system health
- Resolve issues quickly
- Sustain target performance

Liberate

Liberate

About Liberate

Liberate builds AI agents that do the work of property and casualty insurance. When a policyholder calls, texts or emails, whether it's a billing question or a claim on the worst night of their life, Liberate's agents resolve the request end to end and write every action into the carrier's core systems, including Guidewire, Duck Creek and Applied Epic. A built-in supervisor layer governs every interaction, brings in a human the moment one is needed and leaves an audit trail a regulated industry can stand behind.

Insurance-native AI agents that resolve sales, service and claims end to end, on every call, email and text

10M+

automated transactions a month

Across voice, SMS and email with 70+ carriers, agencies & brokers

$100B+

in P&C premium now runs through Liberate's AI agents,

resolving sales, service and claims requests end-to-end.

ONE

platform, three functions: sales, service & claims

FNOL, claim status, policy servicing, billing and payments, quoting, lead qualification, certs and ID cards, Rate/Quote/Bind