OBSERVE·AI

BUYER'S GUIDE · VOICE AI

Proof-based vendor selection.

A practical guide to evaluating enterprise Voice AI agents — by what they do in production, not what they show in a demo.

WHAT'S INSIDE

Evaluate the system, not the demo.

This guide turns vendor selection into a repeatable, evidence-based process — one that measures the agent and everything required to make it succeed: integrations, telephony, security, delivery, deployment controls, and observability.

1

— THE PREMISE

Why proof beats demos.

A Voice AI evaluation should answer one question: which platform can reliably operate in your environment — with your customers, systems, policies, and operational constraints?

A polished demonstration cannot answer that. Demos are built to show an agent under favorable conditions. The workflow is known. The caller is cooperative. The audio is clear. The integrations work. The conversation follows a path the vendor has already rehearsed.

THE SHIFT

Move the decision away from presentation quality and toward observable performance. Every vendor faces the same operating conditions, the same scenarios, the same success criteria, and the same scrutiny.

PRODUCTION IS DIFFERENT

Customers interrupt, change subjects, use incomplete language, call from noisy environments, and respond in ways no script anticipated. Systems time out. Knowledge changes. Transfers fail.

Compliance requirements vary by state, product, or segment. A minor configuration update can improve one workflow while quietly breaking another. The result shouldn't tell you which agent sounds best — it should show which vendor can help you build, deploy, govern, and continuously improve a production program.

2

THE FRAMEWORK

The four-layer evaluation model.

A credible evaluation has four layers. A vendor shouldn't win by performing well in one while creating unacceptable risk in the others. Weighting varies by organization — but all four get evaluated.

1

Agent performance

THE CORE

What the agent says, understands, and does — and whether the customer reaches the intended outcome. This is the core of the evaluation.

2

Enterprise execution

Whether the agent operates across real systems, workflows, telephony environments, security requirements, and handoff processes.

3

Delivery and governance

How the vendor moves your team from configuration to production — testing, version control, approvals, deployment, and operational ownership.

4

Continuous improvement

Whether the platform can identify failures, diagnose root causes, safely test changes, and monitor performance once the agent is live.

3

BEFORE TESTING

Define the decision.

Before vendors configure an agent, document the type of decision being made. Each situation carries a different threshold.

A **New platform selection**

May be decided on the strongest weighted score across finalists.

B **Build versus buy**

Must include internal engineering, maintenance, and operational costs in the comparison.

C **Incumbent replacement**

Must account for migration risk and require a meaningful improvement over current production performance.

D **Formal procurement**

Must keep product performance from being overshadowed by questionnaire responses and presentation quality.

DEFINE OUTCOMES FIRST

A valid result may be a clear winner, a limited production pilot between two finalists, a conditional selection with required remediation, or a finding that no vendor is ready. No vendor should be selected simply because the process is expected to produce one.

Define the caller variability.

Testing the same request with the same cooperative caller creates a false sense of reliability. Each workflow should be exercised through different caller behaviors and audio conditions.

- Speaks quickly
- Pauses frequently
- Interrupts the agent
- Out-of-order information
- Corrects a previous answer
- Sounds frustrated
- Informal language
- Loud setting
- Indirect answers
- Multiple facts at once

Voice AI must do more than map a clean sentence to an intent. It has to manage the structure and unpredictability of spoken conversation. Define a representative distribution of conditions — not only extreme edge cases.

WHY IT MATTERS

Run the same underlying scenario across multiple caller conditions. That's what separates a workflow problem from a speech, comprehension, or conversation-management problem.

4

FROM REAL OPERATIONS

Build the test foundation.

Don't ask a conference room to invent twenty ideal test calls. Start from the work the agent will actually perform — existing interactions, standard operating procedures, knowledge sources, business rules, and known failures. Weight coverage by both frequency and consequence.

FIVE KINDS OF SCENARIOS

Standard workflows

Common calls the agent should handle consistently.

Complex workflows

Multiple requests, exceptions, and conditional rules.

Environmental challenges

Noise, weak connections, accents, interruptions.

Failure conditions

Dead APIs, missing records, auth failures, timeouts.

Adversarial & high-risk

Attempts to bypass policy, expose data, or overcommit.

ROUTINE — HIGHER CONSEQUENCE

THE OUTPUT

A shared scenario library that represents the environment the chosen vendor will face in production — not the one the demo was built for.

Averages hide the failures that matter.

An overall completion rate can mask serious failure in the most important scenarios. A platform can look responsive on average while producing unacceptable delays in a meaningful share of calls. Score by workflow, by complexity, and across repeated runs — one successful call doesn't prove consistent performance.

TREAT AS MANDATORY GATES

Guardrail adherence

A vendor may score well overall and still be disqualified if it repeatedly exposes protected information or skips a required disclosure.

Conversational discipline

An agent that sounds warm but talks over the customer, gives long answers, or avoids necessary clarification is not delivering a strong experience.

EVIDENCE, NOT VERDICTS

The platform must let you move from a score to the conversation that produced it — and show the tool calls, workflow steps, guardrail triggers, and latency behind every result. That's the difference between a metric and something you can govern.

5

THE CORE SCORE

Score agent performance.

The bulk of the score belongs to the agent itself. Four dimensions give it structure — and two of them should act as mandatory gates, not averages.

1 Goal attainment

Did the agent complete the work? Go beyond containment — a contained call isn't a success if the answer was wrong, the system wasn't updated, or a step was skipped. Break results down by workflow and complexity.

2 Operational reliability

Latency, time to first audio, recognition, tool-call success, fallback, transfer success, context preservation. Examine the slow tail across repeated runs — not just the mean. Voice AI is probabilistic.

3 Conversation excellence

Pacing, turn-taking, interruption handling, clarification, tone. Naturalness and brand fit are scored criteria — not one executive's reaction to a voice that overrides the scorecard.

4 Guardrail adherence

GATE

Exact disclosures, authentication, privacy, payment controls, escalation, prohibited statements. The platform must show where and why a guardrail failed — pass/fail without evidence isn't enough.

6

AROUND THE AGENT

Evaluate the enterprise system.

The agent score is central, but it doesn't fully answer whether the vendor can succeed in your environment. Score these five areas separately.

1

Integration and workflow interoperability

Retrieve and update across CRM, case management, billing, scheduling, and core systems. Test auth errors, missing data, timeouts, and partial completion — not just the happy-path API call.

2

Telephony and transfer architecture

Routing, recording, DTMF, warm transfers, queue handling, and context preserved for the human teammate. A transfer that makes the customer repeat everything is not a successful handoff.

3

Knowledge and policy management

How knowledge enters the system, how updates are governed, and how outdated or unapproved information is kept out. Test how fast a policy can be updated, validated, approved, and deployed.

4

Security, privacy, and access controls

Encryption, retention, redaction, regional requirements, role-based access, and audit history. Test behavior in transcripts, recordings, logs, prompts, and tool calls — not just documentation.

5

Scale and performance management

How the system behaves as concurrent volume rises — rate limits, failure isolation, monitoring, and incident response. A few sequential test calls don't prove enterprise readiness.

7

PEOPLE AND PROCESS

Delivery and change control.

Voice AI programs don't succeed through software alone. The vendor's delivery model shapes how fast the agent reaches production and how well it performs once there. A capable team translates requirements into agent behavior, finds gaps in procedures, configures integrations, and guides launch.

TEST RESPONSIVENESS — DON'T ASSUME IT

After round one, give each finalist a defined set of findings. Measure how quickly they identify root causes, propose changes, implement them, and improve performance without creating regressions elsewhere. That reveals how they'll behave under real production issues.

REQUIRE A SAFE CHANGE AND DEPLOYMENT MODEL

An agent isn't finished when the evaluation ends. Prompts, policies, models, knowledge, and customer behavior all change. Ask each vendor to make a realistic configuration change during the evaluation — then test the controls around it.

SEPARATE	COMPARE	APPROVE	REVERSE
Distinct dev and production environments	Version history and config diffs	Pre-deployment testing and sign-off	Rollback and clear ownership

A vendor that can build the initial agent but can't manage ongoing change creates long-term operational risk.

8

SIMULATION & COMPARABILITY

Test at scale.

Manual test calls belong in the process — but they can't be the whole process. A credible evaluation runs the shared scenario library repeatedly, at enough volume to expose inconsistent behavior.

H

Historical scenarios

Reproduce known requests and known failures — proof the agent can handle the reality the business already understands.

S

Synthetic scenarios

Vary caller behavior, wording, noise, emotion, and system conditions to expose failures not yet present in the data.

Score automatically while routing ambiguous or high-risk cases to human review — use judgment where it adds value, not on every generated call.

RUN EVERY VENDOR UNDER COMPARABLE CONDITIONS

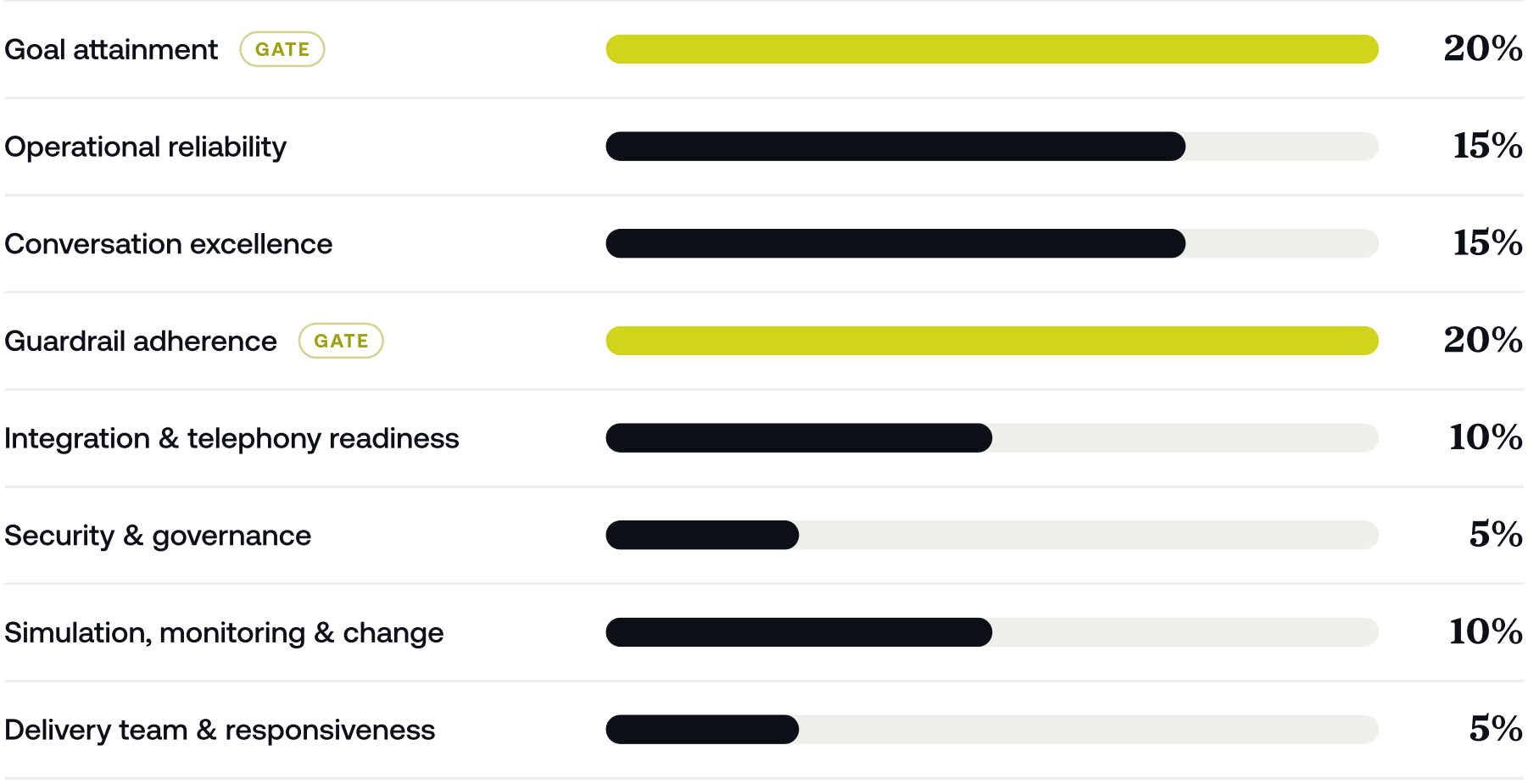
Same workflow definitions, system access, knowledge, caller variability, scoring rules, and repetitions. Run in parallel where possible. Produce aggregate results and drill-down evidence — by workflow, by caller condition, down to the individual interaction. Don't hide tradeoffs inside one composite number.

9

AN ILLUSTRATIVE MODEL

The recommended scorecard.

Assign most of the weight to agent performance, with meaningful weight reserved for enterprise readiness and delivery. Tune to your workflow and risk profile.



Yellow bars are mandatory gates — non-negotiable regardless of total score. **100%**

COST AGAINST THE FULL OPERATING MODEL

Weigh platform fees, usage, implementation, integrations, services, internal staffing, testing, and monitoring. A low per-minute rate isn't a low total cost if the platform needs extensive engineering or manual review.

10

WHAT TO AVOID

Avoid the credibility traps.

✕ The test is built around the demo

Fix Build and weight the test before the vendor presentation influences it.

✕ Vendors receive different tests

Fix Equivalent conditions are required even when architectures differ.

✕ The team reviews too few calls

Fix Automated evaluation at scale, supported by targeted human review.

✕ The score changes after results arrive

Fix Document weights up front; require a written reason for any change.

✕ Voice preference becomes a veto

Fix Treat naturalness and brand fit as scored criteria from the start.

✕ The evaluation ignores delivery

Fix Evaluate the future operating model, ownership, and required skills.

✕ The test stops at selection

Fix Carry scenarios and thresholds into acceptance and production monitoring.

✕ The process takes too long

Fix Limit the field, run in parallel, automate scoring, go deep only on finalists.

11

ONE CONTINUOUS SYSTEM

From selection to production quality.

The most valuable output isn't the final scorecard — it's the evaluation system you've built. Workflow definitions become operational requirements. The scenario library becomes the regression suite. Scoring criteria become production evaluations. Thresholds become alerts and governance controls.

Once the agent is live, **Interaction Intelligence** — or an equivalent observability layer — evaluates real conversations continuously, then feeds every finding into the next improvement cycle.

1. **Identify** The issue, from a live metric.
2. **Diagnose** The root cause behind it.
3. **Propose** A change to address it.
4. **Test** Through simulation first.
5. **Approve** Review and sign off.
6. **Deploy** Via controlled release.
7. **Monitor** The result in production.

↻ From metric to conversation, to the failed workflow step, to the prompt, policy, tool call, or integration behind it. **Selection and production quality become one system — not two projects.**

12

SHOW, DON'T DESCRIBE

Questions every finalist must answer.

A finalist should be able to **show** — not just describe — how its platform answers each of these.

- Complete our workflows across common, complex, and failure conditions?
- Execute disclosures, authentication, privacy, and escalation consistently?
- Inspect tool calls, workflow steps, guardrail triggers, and latency?
- Preserve context during transfers and downstream case creation?
- Compare a config change against production before deployment?
- Evaluate all production conversations, not a small manual sample?
- Define the delivery team's role before and after launch?
- Hold conversational quality across realistic caller and audio variability?
- Show why a conversation succeeded or failed?
- Connect to our systems without a fragile custom architecture?
- Test historical, synthetic, and adversarial scenarios at scale?
- Approve, release, monitor, and roll back changes safely?
- Let our team operate the system without vendor services for every change?
- Diagnose and correct failures found during the evaluation — quickly?

THE TELL

A vendor that can't answer these with evidence is asking you to accept risk that only becomes visible after deployment.

IN CLOSING

Confidence in a production program.

The best Voice AI vendor isn't the one with the most impressive prepared conversation. It's the one that proves it can perform across your workflows, caller conditions, systems, policies, and operating requirements.

✓

A decision you can defend

Shared tests, predefined weights, repeated execution, automated evaluation, and transparent evidence — grounded in the conditions the agent will actually face.

✓

A reusable quality standard

The same assets guide implementation, acceptance testing, production monitoring, and every agent change that follows.

Evaluate the work, not the demo.

Observe.AI brings VoiceAI Agents, the Companion Agent, and Interaction Intelligence together on one CX platform — built-in evaluation, simulation, and governance, end to end.

Book a demo