Microsoft Dynamics 365

Agent Evaluation Series: Governance, Lifecycle Gates, and Operating Agents in Production

Presenter(s):
Priyanka Sinha – Senior Solution Architect

Co-Presenter(s):
Ahmet Yildirim – Senior Solution Architect
Akshat Singh – Senior Solution Architect

20th May 2026

TechTalk Series

- Session 1 – Agent evaluation framework foundation
- Session 2 – Designing Evaluation Sets, Metrics, and the Evaluation Blueprint
- **Session 3 – Governance, Lifecycle Gates, and Operating Agents in Production**

Agenda

- Governance
- Evaluation Lifecycle
- Evaluation Gates
- Production Evaluation Strategy
- Best practices and key take aways

Microsoft Dynamics 365

Agent Governance

Why Governance Matters for AI Agents

Why Governance Is Different for AI Agents

Agent quality is not static. Traditional D365 deployments reach a stable state after Go-Live.

Agent quality degrades silently as models update, grounding data shifts, or usage patterns evolve.

Someone must own agent quality, monitor it continuously, and have authority to act when it degrades.

In traditional D365 implementations, it's clear who signs off on each gate:

- The Business Owner approves the Solution Blueprint
- The Solution Architect validates technical readiness
- The Project Manager confirms Go-Live preparedness

With AI Agents, the Stakes Are Different

Agents don't just execute pre-programmed logic — they reason, make decisions, and take actions. This means:

Risk is continuous, not one-time: An agent can drift or degrade after deployment
Evidence replaces certainty: You can't "prove" an agent works; you accumulate evidence it behaves correctly
Accountability must be explicit: Someone needs to accept residual risk when an agent goes live

Key Insight

Without clear governance, agent projects stall in endless testing or ship with unclear accountability. A structured framework ensures evidence-based decision-making throughout the lifecycle.

Governance: Why Evaluation Doesn't Stop at Go-Live

AI agents are fundamentally different from traditional D365 software — they demand continuous evaluation

Traditional D365 Testing

Deterministic — same input → same output

Predictable forms, workflows & record-change paths

Binary pass/fail test criteria

Tests what was built — configuration verified

AI Agent Behavior

Non-deterministic — same input → different valid outputs

Autonomous — acts on live data, often without human loop

Drifts over time — model updates & data changes erode quality

No binary result — correctness, safety & alignment need rubrics

Key Insight

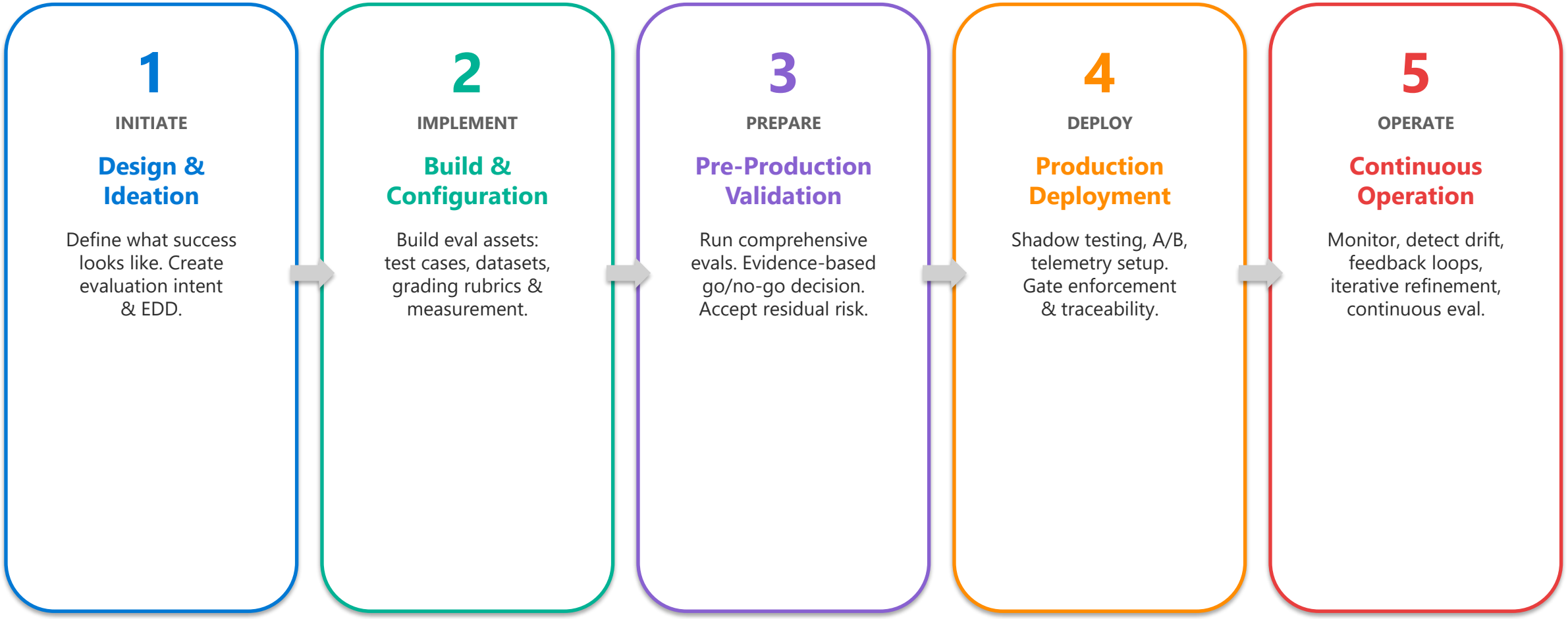
Risk is continuous, not one-time. Evidence replaces certainty — you can't "prove" an agent works; you accumulate evidence it behaves correctly. Accountability must be explicit — someone needs to accept residual risk when an agent goes live.

Microsoft Dynamics 365

Evaluation Lifecycle

The Evaluation Lifecycle: Five Stages of Evidence-Based Decisions

Aligned with Success by Design phases — evaluation is a first-class lifecycle artifact



Governance: Who Decides, When, and Based on What Evidence

Key roles that govern every evaluation decision across the delivery lifecycle


Business Owner


AI Quality Specialist


Responsible AI Reviewer


Release Authority


Service Owner

Primary Profile per Stage

1. Design & Ideation

- Define **success criteria** & **identify risk areas**
- **Align on intended** use & evidence needs

2. Build & Configuration

- Develop **test case**s & **datasets**
- **Validate** evaluation **approach**

3. Pre-Production Validation

- **Run evaluations** & **assess risks**
- Decide **go/ no-go** based on evidence

4. Production Deployment

- **Enforce** launch decision & document outcomes
- **Ensure traceable evaluation records**

5. Continuous Operation

- **Monitor performance** & detect drift
- **Adapt** & **re-evaluate** as needed

Evaluation Intent & Success Definition | **Evaluation Design & Setup** | **Readiness & Go/No-Go** | **Gate Enforcement & Traceability** | **Monitoring**

Stage 1: Design & Ideation

Evaluation Intent and Success Definition — "What does success look like for this agent?"

Evaluation Focus

- Define evaluation intent aligned to user & business outcomes (e.g., "95% inquiries resolved without escalation")
- Identify critical behaviors the agent must exhibit (e.g., "Always verify inventory before promising delivery")
- Declare expected interaction shape (single-turn vs. multi-turn, state/tool use)
- Map risk areas where evidence will be required (sensitive data, financial decisions)

Governance Focus

- Should this agent be built? What problem does it solve?
- How do we define “good”?
- What types of evidence will inform our go/no-go decision later?
- Map risk areas requiring evidence

Key Output

An Evaluation Design Document (EDD) specifying what “good” means and what types of evals will be needed.

Stage 2: Build & Configuration

Evaluation Design and Setup — Translating intent into concrete evaluation assets

Evaluation Focus

- Create test datasets reflecting real-world usage (core scenarios, edge cases, long-tail)
- Design grading approaches (automated metrics, LLM-as-judge, human review)
- Define evaluation taxonomy so results are explainable (what was evaluated, and why)

Governance Focus

- Are we using appropriate metrics and methods?
- Do our eval datasets represent realistic usage?
- Is the evaluation design repeatable and auditable?

Key Output

A functional eval suite that can be run repeatedly, with clear documentation of what each eval measures and why it matters.

Stage 3: Pre-Production Validation

Readiness and Go/No-Go Signals — "Does the evidence support going live?"

Evaluation Focus

- Run comprehensive eval suites against the final agent version
- Assess dataset representativeness: what user intents are covered vs. missing
- Document known limitations and edge cases where the agent may fail
- Conduct adversarial testing (jailbreak attempts, prompt injections)
- Perform human evaluations (UAT equivalent)

Governance Focus

- Is the agent ready to go live based on available evidence?
- What risks remain, and are they acceptable?
- Go/No-Go (or conditional go) based on evidence and judgment
- Accept or address residual risk identified by gaps in evidence
- Determine if launch constraints are needed

Key Output

A Go/No-Go Decision with documented rationale: eval results summary, known gaps, conditions for launch, and re-evaluation triggers.

Stage 4: Production Deployment

Gate Enforcement and Traceability — Formalizing the decision and establishing operational contracts

Evaluation Focus

- Document who is accepting residual risk and under what conditions
- Package evaluation evidence and assumptions as part of the deployment record
- Ensure evaluations are traceable and interpretable (what was tested, what wasn't, what results mean)

Governance Focus

- Implement production gate; document decision context and responsible individuals
- Establish criteria for re-evaluation (model changes, new tools, new user groups)
- Establish monitoring and alerting thresholds

Key Output

A Production Deployment Record: eval evidence reviewed, risk acceptance docs, release approver(s), conditions for operation, and re-evaluation criteria.

Stage 5: Continuous Operation

Monitoring, Drift Detection, and Learning — Ensuring ongoing performance after go-live

Evaluation Focus

- Continuously monitor agent behavior in production (online evals, observability)
- Detect drift: Is the agent's behavior changing over time?
- Refresh eval datasets as usage patterns evolve
- Use evaluation insights to inform iteration: measure → learn → tune → re-evaluate

Governance Focus

- Is the agent performing as expected?
- Should we proceed, adjust, limit scope, or halt?
- Use evaluation data + human judgment to choose: continue, limit, refine, or end
- Reassess at milestones (significant changes, new scenarios, scope increases)

Key Output

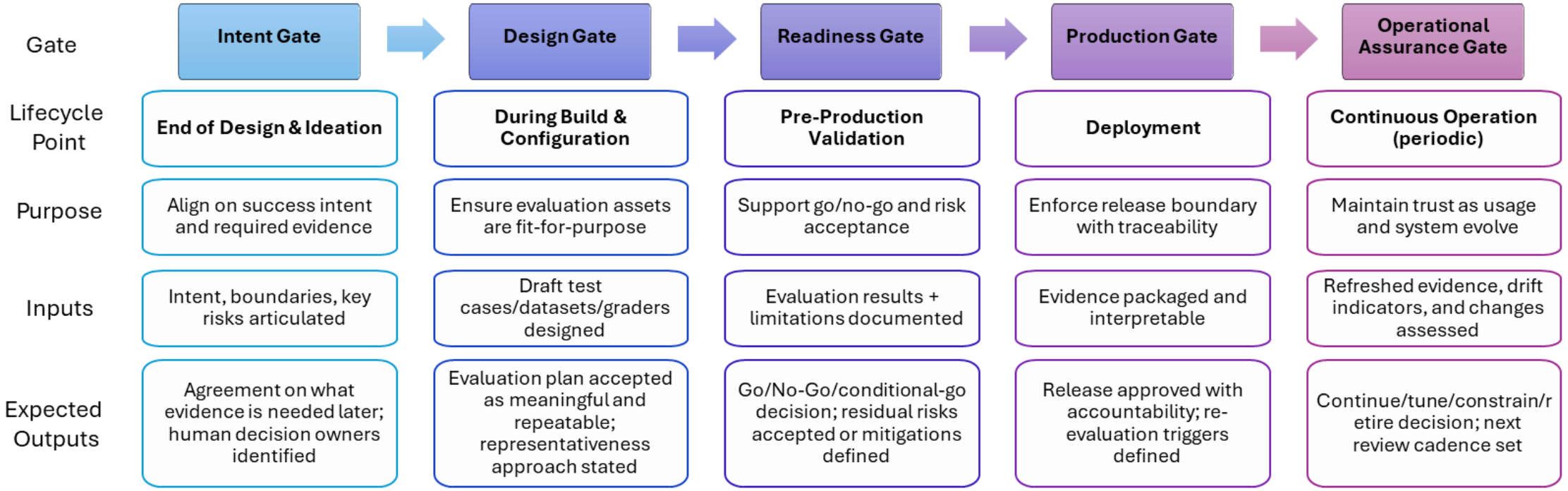
Ongoing: production error rate monitoring, user satisfaction tracking (CSAT), A/B testing, periodic regression evals, incident response & root cause analysis.

Microsoft Dynamics 365

Evaluation gates

Evaluation Gates: Evidence-Based Go/No-Go Decisions

Structured decision points — evaluation evidence is required for advancement at each gate



Best Practice: Document every gate decision with evidence reviewed, risks identified, decision rationale, approver(s), and next review criteria.

Responsible AI Safety Gates Across the Lifecycle

RAI considerations validated at specific lifecycle gates with defined entry criteria and evidence requirements

Design (Initiate)

RAI requirements review

Assess risk profile
Define acceptable behavior boundaries
Document fairness criteria

Build (Implement)

Red-team testing

PII protection validation
Fairness assessment across segments
Adversarial prompt testing

Pre-Deploy (Prepare)

Production readiness gate

Documented evidence of RAI compliance
Known limitations documented

Production (Deploy/Operate)

Continuous monitoring

Safety drift detection
Compliance degradation alerts
Bias monitoring

Incident Response

Post-incident review gate

Root cause analysis
Feedback into eval sets
Guardrail updates

Core RAI Criteria (must be addressed in every EDD)

Fairness: Consistent performance across segments | Privacy & Security: No PII in logs, respect D365 security roles | Transparency & Accountability: Reasoning logged, traceable, users aware of AI

Sales Order Process Agent Example - RAI Safety Gates

RAI considerations validated at specific lifecycle gates with defined entry criteria and evidence requirements

Design (Initiate)

RAI requirements review

Define fairness: Define fairness: Consistent extraction accuracy across customer segments (variance <5%)

Identify privacy risks: Customer email addresses, contact names in attachments

Set safety requirements: Zero unauthorized order creation

Document human oversight: Manual review for orders >$50K or ambiguous customer matches

Build (Implement)

Pre-Production Safety Evaluation

Red-team testing: 500+ test cases:

- Ambiguous customer names (multiple matches in F&O)
- Missing product codes
- Unusual quantities or pricing
- International characters and formats

Fairness analysis: Test across 10 customer segments (enterprise, SMB, international, etc.)

Privacy testing: Verify PII is redacted in logs; test session boundaries
Result: 2 critical issues found (PII in error logs, hallucinated product codes) → Block until fixed

Pre-Deploy (Prepare)

Production readiness gate

Monitoring dashboard configured: Tracks extraction accuracy, exception rate, processing time, fairness metrics
Incident runbook: Defines response procedures for unauthorized orders, PII leakage, accuracy degradation
Audit logging validated: 100% of order creation attempts logged with full context
DPO sign-off: Privacy controls approved
Result: Approved for production with 10% shadow mode testing

Production (Deploy/Operate)

Continuous monitoring gate

Weekly reports: Automated tracking of RAI metrics (fairness variance, PII compliance, exception rate)
Monthly review: Human evaluation of 100 sampled orders for quality and appropriateness
Real-time alerts: Trigger on any unauthorized order attempt, PII detection, or accuracy drop >5%
Result: One alert in Week 2 (fairness variance 7% for international customers) → Investigation and prompt adjustment

Incident Response

Post-incident review gate

Incident: Agent created order for wrong customer due to similar company names
Root cause: Insufficient customer validation logic when multiple fuzzy matches found
Mitigation: Immediate rollback to manual processing for ambiguous matches
Preventive action: Enhanced matching algorithm; additional test cases; lower confidence threshold for escalation
Result: Updated EDD and evaluation suite to prevent recurrence

Microsoft Dynamics 365

Production Evaluation Strategy

Production Evaluation Strategies

Three experimentation modes for validating agents in production — each serves a different risk profile

Shadow Mode

Risk: Very Low | User Impact: None

- Run agent in parallel with existing system
- Log predictions without affecting users
- Validate performance risk-free
- Real-world data collection
- Ideal for high-stakes domains

A/B Testing

Risk: Low–Moderate | User Impact: Limited

- Control vs Treatment groups
- Statistical significance testing
- Sample size calculations
- Compare two well-defined alternatives
- Collect user feedback data

A/B/n Testing

Risk: Moderate–High | User Impact: Moderate

- Multiple variants simultaneously
- Traffic split across n versions
- Test different prompts/configs
- Requires sufficient traffic
- Exploration of multiple approaches

Transition from Offline to Online Evaluation

Several critical shifts occur when moving from controlled testing to production evaluation

1. Baseline → Dynamic Context

Offline: 200 curated order emails
Online: Actual customer emails with varied formats, typos, incomplete info, unexpected attachments

2. Periodic → Continuous Evidence

Offline: Weekly regression test runs
Online: Real-time monitoring of extraction accuracy, exception rate, processing time on every order

3. Completeness → Proxy Measurement

Offline: Human review of 100% of test cases
Online: Confidence scores as proxy; human review of sampled cases (10%) and all exceptions

Offline evaluation establishes the authoritative baseline. Online evaluation ensures continued performance in real-world conditions.

Production Evaluation Strategy: Shadow Mode Deployment

What is Shadow Mode?

Agent runs in parallel with existing systems

Processes real production inputs WITHOUT taking real actions

Zero user exposure — risk-free evaluation

Collects production-grade signals on real data

Reveals patterns not in offline test sets

When to Choose Shadow Mode

High-stakes domains (financial, medical, legal)

Introducing never-before-validated capabilities

Rolling back from incidents or safety violations

Compliance mandates zero user exposure during testing

Sales Order Agent — Shadow Mode Example

Shadow: Extract orders from emails but DON'T create them in F&O. Compare agent extractions against manual processing. Validate extraction accuracy, exception routing, and processing time on real customer emails — without any business impact.

Never skip shadow testing. Production data often contains patterns not represented in offline test sets.

Production Evaluation Strategy: A/B Testing

What is A/B Testing?

A/B testing is a production experimentation method where you route a percentage of live traffic to a new agent variant (the **Challenger**) while the rest continues using the current approved agent (the **Champion**). You then compare performance across both groups using predefined metrics to determine which performs better.

When to Choose A/B Testing

Comparing two well-defined alternatives with similar risk profiles
Making **incremental improvements** to established functionality
Collecting user feedback data to inform future development
A **production baseline exists** for statistical comparison

Sales Order Agent — A/B Testing Example

Sales Order Agent currently extracts order details from customer emails and creates orders in Dynamics 365 F&O. You've improved the extraction prompt to better handle product descriptions (instead of product codes) — a pattern that caused 15% of production exceptions.

The A/B test validated that the prompt improvement generalizes to real production data — not just the offline test set — before promoting it to 100% of traffic.

Production Evaluation Strategy: A/B/n Testing

What is A/B/n Testing?

A/B/n testing extends A/B testing by comparing multiple agent variants simultaneously against the current production agent (Champion). Instead of testing just one Challenger, you test 2 or more Challengers at the same time — each receiving a portion of live traffic.

When to Choose A/B/n Testing

Optimizing multiple configuration parameters simultaneously (e.g., different prompts, temperature settings, grounding strategies)
System has sufficient traffic for statistical significance across all variants
In an exploration phase where multiple approaches have merit and you want to find the best one
Time-to-insight matters more than minimizing user exposure — testing in parallel is faster than sequential A/B tests

D365 CRM User Onboarding Agent — A/B/n Testing Example

our Onboarding Agent provisions D365 CRM access for new users. You want to improve the quality of the IT notification email it generates. Three different approaches look promising — rather than testing them sequentially over 3 months, you run A/B/n to find the winner in one cycle.

Testing 3 alternatives sequentially would take ~9 weeks (3 × 3 weeks each). A/B/n found the winner in ~5 weeks — nearly half the time. The trade-off is higher traffic requirements per variant to reach statistical significance.

Observability vs. Monitoring vs. Evaluation

A clear distinction is essential for operating agents in production

Monitoring

Tells you THAT something happened

"Order processing failed"

Real-time dashboards, alerts, SLO/SLA tracking, error rates

Observability

Tells you WHAT happened

"F&O API timeout during validation step"

Traces, logs, distributed context, deep debugging capability

Evaluation

Tells you WHETHER it met expectations

"Exception routing was appropriate; accuracy within threshold"

Quality scoring, regression testing, fairness analysis, drift detection

All three are required in production. Monitoring detects issues fast. Observability enables root cause analysis. Evaluation confirms the agent still meets business expectations.

Online Evaluation Signals: Continuous vs. Periodic

Near Real-Time (Continuous)

Safety guardrail activations (target: 0)

Policy-adherence violations

Quality degradation indicators (confidence < 90%)

Latency & reliability (processing time > SLA)

Anomaly detection (exception rate spikes)

Aggregated/Periodic

Weekly quality scorecards (human eval of 20-30 orders)

Longitudinal drift metrics & trends

Cohort analysis by order type, customer, geography

Monthly regression: full eval suite on prod logs

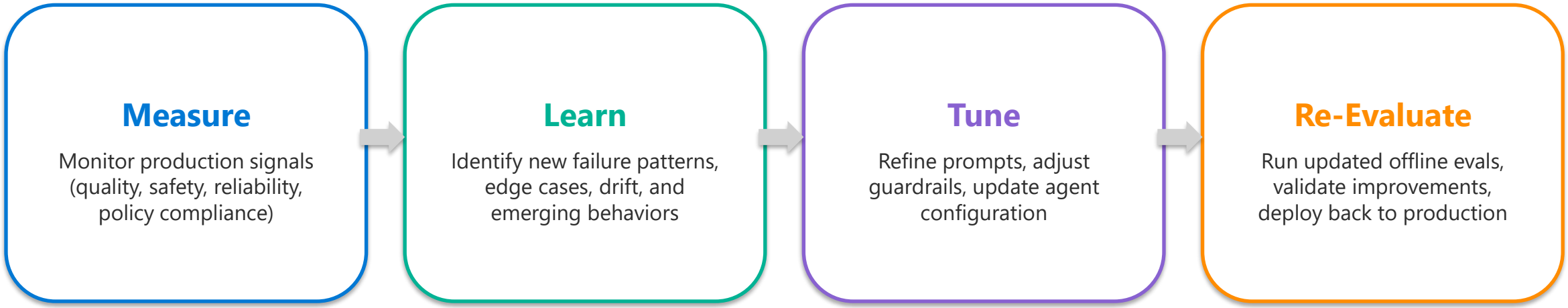
Quarterly fairness audits across segments

Validating Offline Assumptions in Production

Offline assumption: Agent achieves 95% extraction accuracy on test dataset
Online observation: Production accuracy = 92% (below threshold)
Investigation: New customer email format not represented in test data
Action: Add new format to test dataset → retrain/adjust agent → re-evaluate

Feedback Loops: Learning from Production

Online evaluation does not replace offline evaluation — it continuously informs it



Sales Order Agent — Feedback Loop Example

Week 1-4 Production Observations:
- 15% of exceptions: orders with product descriptions instead of product codes
- 8% of international orders: lower extraction accuracy due to date format variations
- 3% of orders: quantities in non-standard units (dozens, cases)

Feedback to Offline Evaluation:
→ Add 20 new test cases with product descriptions + 15 international orders + 10 non-standard unit orders
→ Update evaluation criteria: add "product description handling" dimension
→ Adjust agent prompts: enhance instructions for product descriptions & non-standard units
→ Re-run offline evaluation → validate improvements → deploy to production

Microsoft Dynamics 365

Best Practices and Key take aways

Common Pitfalls & Lessons Learned

Learning from common mistakes accelerates evaluation maturity

Evaluation Myopia

Focusing on only one dimension (usually correctness) while ignoring safety, alignment, efficiency

→ **Address all five dimensions; document priorities**

Evaluation-Production Mismatch

Offline evals that don't reflect production conditions give false confidence

→ **Use production telemetry to validate offline assumptions**

Neglecting Cost & Latency

A "better" agent that's 3x slower or 5x more expensive isn't better

→ **Include efficiency metrics in every evaluation**

Over-Optimization on Narrow Metrics

Agents game a single KPI while degrading overall experience

→ **Measure multiple complementary dimensions**

Treating Evals as Checklist Items

Running evaluations without using results to drive decisions

→ **Tie eval results to hard gates — if it doesn't block, it doesn't count**

Not Re-Evaluating After Changes

Prompt updates, model changes without re-running comprehensive evals

→ **Define re-evaluation triggers; automate evals in CI/CD**

Best Practices for Operating Agents in Production

✓ Start Small, Scale Gradually

Begin with 20-30 eval cases, not 1,000. Quality over quantity.

✓ Automate Early

Integrate evaluations into CI/CD from the start.

✓ Use LLM-as-Judge Wisely

Combine automated and human evals — validate LLM judges against human judgment.

✓ Version Everything

Track prompts, evaluation suites, and datasets with version control.

✓ Monitor Continuously

Production is where real issues emerge. Real-time + periodic signals.

✓ Iterate Based on Failures

Every evaluation failure is a learning opportunity — feed back into improvement.

✓ Align with Business Metrics

Evaluations should predict business outcomes, not just technical metrics.

✓ Make Accountability Explicit

Every production agent needs a documented trail: who decided, based on what evidence.

Key Takeaways

1. **Production evaluation is fundamentally different from offline testing.**

   Real users, real data, real consequences demand shadow mode, A/B testing, and canary deployments.

2. **Continuous improvement requires systematic feedback loops.**

   Instrument production signals, detect drift early, integrate human judgment, feed learnings back.

3. **Governance and lifecycle gates turn evaluation into accountability.**

   Evidence-based go/no-go decisions at each stage ensure RAI principles are enforced, not aspirational.

4. **Learning from common mistakes accelerates success.**

   Avoid narrow-metric optimization, eval-production mismatch, and poor baselines.

5. **Operating agents at scale demands a culture, not just tooling.**

   Transparent monitoring, stakeholder alignment, and iterative refinement must be organizational habits.


QUESTIONS

Dankie Faleminderit **Shukran** Chnorakaloutioun Hvala Blagodaria

Děkuji **Tak** Dank u Tänan Kiitos **Merci** Danke Ευχαριστώ A dank

Mahalo ՇՆՈՐՀ. **Dhanyavād** Köszönöm Takk Terima kasih **Grazie** Grazzi

Thank you!

감사합니다 Paldies Choukrane Ačiū **Благодарам** ありがとうございました

谢谢 Баярлалаа **Dziękuję** Obrigado Mulţumesc **Спасибо** Ngiyabonga

Ďakujem Tack Nandri Kop khun **Teşekkür ederim** Дякую Хвала Diolch


Microsoft Dynamics 365