FROM SINGLE-TURN ASSISTANTS TO SYSTEMS THAT PLAN, ACT, AND VERIFY

Agentic AI for Enterprise Workflow Automation

Executive briefing — Operations & Technology Leaders | ~20 min

From Chatbots to Agents

- Chatbots answer questions in one turn; agents pursue a goal across many steps
- Agents call tools, check results, and revise their own plan mid-task
- The shift moves AI from an information interface to an execution layer
- Same underlying models — the difference is loop structure, not raw intelligence
- Enterprise value moves from answering questions to completing transactions

What Makes AI 'Agentic'

- Goal-directed: given an objective, not a fixed script of steps
- Tool use: calls APIs, databases, and applications to take real action
- Memory: retains context across steps and sessions to stay coherent
- Planning and self-correction: breaks work into steps and adapts on failure
- Autonomy is a dial, not a switch — scoped by policy, not left open-ended

Use Cases: Procurement and Finance Operations

- Procurement: PO matching, vendor onboarding checks, contract clause extraction
- Finance ops: invoice triage, three-way matching, exception routing, journal entry drafting
- Common thread: high-volume, rules-plus-judgment work with clear source documents
- Agents handle the routine path; humans review exceptions and edge cases
- Best early candidates: high transaction volume, stable process, measurable cycle time

Architecture: Tools, Memory, Orchestration

- Orchestration layer (e.g. LangGraph, state-machine frameworks) sequences and routes steps
- Tool-calling layer connects the model to ERPs, APIs, RPA bots, and internal services
- Memory: short-term context window plus a vector store for retrieval-augmented recall
- Guardrail layer validates outputs and enforces policy before any action executes
- Observability layer logs every tool call and decision for audit and debugging

Human-in-the-Loop Design

- Not full autonomy: agents propose, humans approve above defined risk thresholds
- Checkpoints placed before irreversible actions — payments, contract sends, record deletion
- Confidence scoring routes low-certainty cases to a reviewer automatically
- Review interfaces should show the agent's reasoning, not just its final output
- Approval friction should scale down over time as accuracy is demonstrated, not assumed

Case: Invoice Processing Agent (Illustrative Example)

- Representative scenario: a shared-services finance team processing vendor invoices
- Agent extracts line items, matches to PO and receipt, flags mismatches
- Clean matches post automatically; exceptions route to an analyst queue with context
- Illustrative range: 60-80% of invoices resolved without human touch in similar deployments
- Numbers are directional, not audited results — actual gains depend on data quality and scope

Productivity Gains, Measured Honestly

- Industry-reported ranges for document-heavy back-office tasks: 30-50% cycle-time reduction
- Gains concentrate in transaction processing, not judgment-heavy or novel work
- Measure against cycle time, error/rework rate, and staff hours reallocated — not headcount alone
- First 90 days typically show smaller gains as exception patterns are discovered and tuned
- Treat any vendor's headline number as a starting hypothesis to validate on your own data

Security and Permission Boundaries

- Agents run under scoped service identities, never a human's full credentials
- Least-privilege access per tool: read-only where possible, write access explicitly granted
- Sensitive actions (payments, data deletion, external sends) require a hard policy gate
- All tool calls and data access logged immutably for audit and incident response
- Segregate agent environments from production credentials via secrets management, not hardcoding

Failure Modes and Guardrails

- Hallucinated data: mitigate with grounding in retrieved source documents, not model memory
- Tool misuse or looping: cap retries and steps, add timeouts and circuit breakers
- Silent drift: outputs degrade as upstream systems change without agent awareness
- Guardrail pattern: validate against schema and business rules before any action commits
- Design for graceful failure — agent escalates to a human rather than guessing

Integration with Legacy Systems

- Most enterprise data sits behind ERPs, mainframes, and systems with no modern API
- Integration paths: API wrappers, RPA bridges, or middleware/iPaaS connectors
- Screen-scraping and legacy RPA remain necessary bridges, not a failure of design
- Data normalization layer matters more than the agent's reasoning quality in practice
- Sequence integrations by transaction volume and API readiness, not system age

Pilot Results (Representative Findings)

- Representative pilot: 8-12 week scoped deployment on a single process, single team
- Success criteria set upfront: cycle time, exception rate, and reviewer time saved
- Common finding: initial exception rate higher than expected, narrows after two tuning cycles
- User trust builds through visible reasoning and easy override, not accuracy alone
- Pilot scope should be narrow enough to fully instrument, not broad enough to impress

Governance Considerations

- Establish an AI governance owner and a standing review cadence, not a one-time sign-off
- Document intended use, data sources, and known limitations per agent, before deployment
- Align with existing model risk, data privacy, and internal audit frameworks
- Track regulatory developments relevant to automated decision-making in your sector
- Require a kill switch and rollback plan for every production agent

Scaling Strategy

- Scale by process pattern, not by agent count — reuse orchestration and tool layers
- Build a shared platform: common memory, logging, and guardrail infrastructure once
- Prioritize processes with clean data and stable rules before ambiguous judgment work
- Fund scaling from demonstrated pilot savings, not a fixed upfront budget
- Invest in change management — process owners, not just IT, drive adoption

Roadmap

- Near term (0-6 months): expand pilot scope, formalize governance, build shared platform
- Mid term (6-18 months): extend to adjacent finance and procurement processes
- Long term (18+ months): cross-functional agent orchestration across departments
- Continuous: reassess guardrails and permissions as agent scope and autonomy grow
- Success measured by sustained cycle-time and quality gains, not deployment count