Intelligent Advocacy: AI Agents and the Legal Profession

Authors:

Niamh Hughes

Samuel Adebayo, PhD

ISx4 • AI research - 2025

ISx4 • AI research - 2025

ISx4
AI & ANALYTICS
Insight, Strategy, Solutions, Services & Support

info@ISx4.com - http://www.ISx4.com

Executive Summary

Agentic workflows are advanced AI-driven processes that can plan, decide, and act through multiple steps autonomously. In essence, an agentic workflow combines large language models (LLMs) with tools and feedback loops to manage tasks independently, freeing humans to focus on higher-level decisions. Unlike a single-turn AI response, an agentic system involves an AI agent that reasons through problems, selects and uses external tools, adapts its plan based on outcomes, and completes a sequence of sub-tasks with minimal human guidance[4]. This white paper introduces agentic workflows and how to deploy them without the drama - i.e. with reliability, transparency, and control. We discuss the benefits and challenges of running AI agents in production, outline the anatomy of an agentic system, provide a legal use case, and compare internal (employee-facing) versus customer-facing agents. Finally, we conclude with practical advice and a call to action for partnering with ISx4 to implement these solutions with proper governance.

Table of Contents

Executive Summary.....2

Table of Contents.....3

Benefits of Agentic Workflows..... 4

Challenges of Deploying AI Agents (and How to Mitigate Them) 5

Anatomy of an Agentic System 7

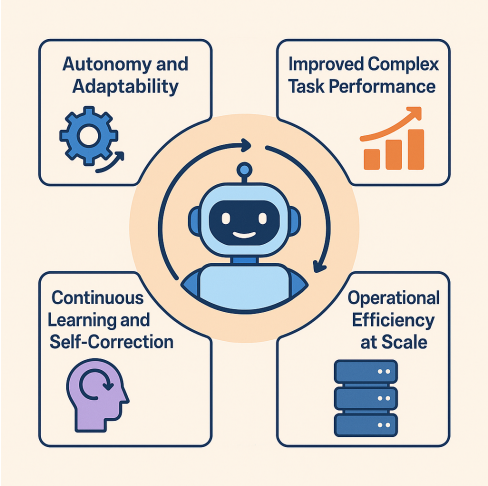
Use Case: Agentic Workflow in Legal Services 10

Internal vs. Customer-Facing Agents.....12

Conclusion and Call to Action.....13

Benefits of Agentic Workflows

Modern AI agents can coordinate complex tasks in ways traditional automations cannot. Key benefits include:



- **Autonomy and Adaptability:** Agentic workflows operate with goal-driven autonomy. They dynamically adjust to new inputs or changing conditions, rather than following a rigid script.[1][2] This flexibility means an agent can handle evolving situations or unexpected challenges by recalculating its actions in real time, staying aligned to its goal.[1] Where a static workflow would break, an AI agent can learn and adapt, leading to more robust performance.

- **Improved Complex Task Performance:** By breaking down complex jobs into manageable steps (a process known as task decomposition), agentic systems often outperform one-shot approaches on difficult tasks.[2] The agent "thinks" through a multi-step plan before acting, which can reduce errors and yield higher-quality results. For example, an agent might first outline a strategy, then execute sub-tasks in order, in a few cases trading some latency for better accuracy.[3] This multistep planning enables higher quality outputs in areas like high-skilled areas that would stump simpler bots such as coding, writing, or data analysis.

- **Continuous Learning and Self-Correction:** Agentic workflows can incorporate feedback loops to improve over time. Many agents use a reflection or review step to critique their own output and refine it [1]. They also maintain memory of prior interactions and results, enabling learning across sessions. This means the agent can avoid repeating mistakes and get better with each iteration.[2] [4] In enterprise settings, this self-optimisation translates to growing efficiency and accuracy as the system handles more cases.

- **Operational Efficiency at Scale:** When implemented correctly, AI agents can handle routine or high-volume tasks faster and with less manual effort. They excel at automating repetitive workflows (e.g. updating records, routing queries) with minimal oversight.[1] By offloading drudge work to agents, organisations boost overall productivity. Studies have found that autonomous AI systems with feedback loops increased process efficiency by 35% and sped up decision-making

[1] What are agentic workflows? Patterns, use cases, and what to watch in 2026
https://www.wrike.com/blog/what-are-agentic-workflows/

[2] What Are Agentic Workflows? Patterns, Use Cases, Examples, and More | Weaviate
https://weaviate.io/blog/what-are-agentic-workflows

[3] Building Effective AI Agents \ Anthropic
https://www.anthropic.com/engineering/building-effective-agents

[4] What Is an Agentic Workflow? A Guide to Autonomous AI Task Execution - SmartOSC
https://www.smartosc.com/what-is-an-agentic-workflow/

by 20% within months. [4] Moreover, agentic workflows can scale elastically - multiple instances or specialised sub-agents can run in parallel - allowing enterprises to tackle larger workloads without linear increases in staff.

Of course, these benefits only materialise if agents are *built right*. A poorly designed agent could flounder or even introduce new problems. Next, we address the challenges and how to mitigate them.

Challenges of Deploying AI Agents (and How to Mitigate Them)

Increased Complexity & Unpredictability	Guardrails, narrow scope, approval for high-stakes steps
Error Handling & Safety	Fallbacks and retries, safe rollback, human-in-the-loop checkpoints
Latency & Efficiency	Parallelism/batching, caching, lighter models, prompt design
Observability & Auditability	Full traces (prompts, tool calls), span logs, replay

While promising, agentic workflows come with practical challenges that must be managed to avoid "drama" in production:

- **Increased Complexity and Unpredictability:** Giving an AI more autonomy also adds uncertainty. Unlike a hard-coded workflow, an agent may take an unexpected path or output, since its decisions are often probabilistic[5]. This can reduce reliability if not properly constrained. The solution is to add guardrails and limits on the agent's actions. For example, one can define allowed tools, set role boundaries, or require the agent to get approval for high-stakes steps. As a best practice, start simple: do not use an agent where a straightforward script suffices. [5] [6] Reserve agentic approaches for problems that truly need adaptive decision-making and keep the agent's scope as narrow as possible.

- **Error Handling and Safety:** In a multi-step agent run, many things can go wrong – an API call might fail, the model might hallucinate a step, or the agent might get stuck in a loop. Robust error handling is critical. Agents should be designed to fail gracefully: for example, by catching exceptions and trying an alternative strategy or reverting to a safe state. One effective pattern is

[5] What Are Agentic Workflows? Patterns, Use Cases, Examples, and More | Weaviate
https://weaviate.io/blog/what-are-agentic-workflows

[6] Building Effective AI Agents \ Anthropic
https://www.anthropic.com/engineering/building-effective-agents

to implement fallbacks or retries for tool calls and model queries.[7] If the primary language model is unavailable or produces an error, the agent can automatically switch to a backup model or a predefined contingency plan. [7] Additionally, human-in-the-loop checkpoints can be inserted for oversight on critical decisions. These measures ensure the agent doesn't go off the rails without intervention.

- **Latency and Efficiency:** An agentic workflow often involves multiple LLM calls and tool invocations, which can introduce latency. Users might face a noticeable delay if an agent is naively orchestrated. In fact, many AI agents trade off speed for better results[8], but in customer-facing scenarios, long waits are unacceptable. To mitigate this, teams focus on optimisation: measure and trace the agent's execution to find bottlenecks, then streamline. For instance, if an agent currently makes five sequential model calls taking 20 seconds, perhaps those calls could be batched or run in parallel to cut response time. [7] Using a more efficient model for certain steps or caching frequent results are other tactics to reduce latency. Careful prompt design can also avoid unnecessary loops. By instrumenting the agent with timing metrics, developers can profile and speed up slow steps to meet production SLAs.

- **Observability and Auditability:** Building trust in an autonomous agent requires making its behaviour transparent. Without good observability, an AI agent is a black box – if it takes a wrong turn, it's hard to diagnose why.[9] For enterprise-use and especially in regulated industries, it's crucial to log what the agent did and why, so that you can audit its decisions. Teams should implement tracing for each agent execution: recording prompts, model outputs, tool calls, and intermediate reasoning steps.[10] Modern LLM observability tools represent an agent's run as a trace composed of spans, where each span is a step like an API call or LLM response.[10 11] This rich telemetry turns the "black box" into a "glass box". With full trace logs, one can debug errors, pinpoint latency issues, and even replay or analyse the agent's chain of thought. Observability is not a nice-to-have – it's essential for debugging, cost monitoring, and safety in production.[11] For example, trace logs enable root-cause analysis when an agent outputs something unexpected, and they provide an audit trail to understand decisions (useful for compliance or reviewing a customer complaint). [11] By continuously monitoring these logs and user feedback, organisations can improve the agent iteratively and ensure it stays within bounds.

---

[7] AI Agents in Production: Observability & Evaluation | ai-agents-for-beginners
https://microsoft.github.io/ai-agents-for-beginners/10-ai-agents-production/

[8] Building Effective AI Agents \ Anthropic
https://www.anthropic.com/engineering/building-effective-agents

[9] Observability in Multi-Agent LLM Systems: Telemetry Strategies for Clarity and Reliability | by Kirill Petropavlov | Medium
https://medium.com/@kpetropavlov/observability-in-multi-agent-llm-systems-telemetry-strategies-for-clarity-and-reliability-fafe9ca3780c

[10] LLM Observability for AI Agents and Applications - Arize AI
https://arize.com/blog/llm-observability-for-ai-agents-and-applications/

[11] AI Agents in Production: Observability & Evaluation | ai-agents-for-beginners
https://microsoft.github.io/ai-agents-for-beginners/10-ai-agents-production/

- **Governance and Control:** With great power comes great responsibility. An autonomous agent might have access to sensitive data or be allowed to execute transactions, which raises governance questions: How do we prevent misuse? How do we ensure compliance with policies and laws? It's important to limit the agent's privileges to the minimum needed for its tasks and to enforce policies at multiple levels. For instance, tool usage can be sandboxed – an agent that writes code could be restricted to a test environment. All actions should be permissioned and logged (who/what triggered the agent to do X). Human override mechanisms are also key: if the agent is about to do something potentially harmful or incorrect, there should be an alert or approval step. Indeed, mature agent frameworks include features for output validation, decision overrides, and human-in-the-loop checkpoints out of the box.[12] By instituting such controls, enterprises ensure that the AI agent remains a helpful assistant, not a rogue actor.

By acknowledging these challenges and designing accordingly, one can deploy agentic workflows that are **reliable, observable, and safe**. The goal is to reap the benefits of autonomous AI without the drama of ungoverned behaviour. Next, we looked into how these agents work under the hood.

Anatomy of an Agentic System

A Planner module breaks down goals into tasks. The Executor carries out each task, often by calling external Tools (APIs, databases, etc.). The agent uses Memory to store context and results, enabling it to maintain state across steps and learn from feedback. A Feedback loop connects back to the planner, allowing the agent to adjust its plan based on outcomes (success or failure of tasks) before the process repeats or terminates.

Underneath the buzzword "agent" lies a modular system of coordinated components. An agentic workflow can be viewed as an AI **orchestrator** that integrates several building blocks to achieve autonomy.[13] The core components include:

- **Planner:** The planner is the strategic brain of the agent. It takes a high-level goal or request and decomposes it into a sequence of actionable steps.[13] For example, if the goal is "analyse this contract and draft a summary", the planner might produce a plan: (1) extract key clauses, (2) check each clause against compliance rules, (3) summarise findings, (4) compile a report. The planner uses the AI's reasoning capabilities (often an LLM prompt or specialised planning module) to decide what needs to be done and in what order. A good planner is dynamic – it can adjust the task list on the fly if new information arises or if certain steps fail.[13] In effect, this component injects goal-oriented decision-making into the workflow.

[12] Agentic AI Explained: Workflows vs Agents | Orkes Platform
https://orkes.io/blog/agentic-ai-explained-agents-vs-workflows/

[13] What Is an Agentic Workflow? A Guide to Autonomous AI Task Execution - SmartOSC
https://www.smartosc.com/what-is-an-agentic-workflow/

- **Executor:** The executor is the hands of the agent – it carries out the tasks the planner specifies.[13] Often the executor is an LLM itself (instructed to perform a specific subtask) or a controller that calls various functions. It interfaces with external systems to do the work: e.g. querying a database, calling an API, drafting text via the LLM, etc. Each step in the plan is dispatched through the executor. The executor can handle synchronous actions (waiting for immediate response) or orchestrate asynchronous operations if a task will take time. In advanced setups, the executor can spawn sub-agents or parallel processes for efficiency. This component is all about action – whatever the planner decides, the executor makes it happen, step by step.[13]

- **Tools and Integrations:** No agent is an island; useful agents leverage external tools to extend their capabilities.[14] Tools could be anything from a web search API, a calculator, a database, an internal CRM system, to even another ML model. By calling tools, the agent can retrieve up-to-date information and take actions in the real world (for instance, sending an email or updating a record). A robust agentic framework provides a library of tool integrations and a standardised way for the agent to invoke them (for example, via the emerging Model Context Protocol standard[14]). Tool-use is often orchestrated by the executor: the plan might say "use the PDF reader tool on document X" and the executor handles that. Tools dramatically increase an agent's usefulness – they allow it to act on data and services beyond what it "knows" in its base model. However, each tool call is also a potential point of failure (hence the need for error handling and permission controls mentioned earlier). When designing agentic workflows, deciding which tools to incorporate (and their APIs/permissions) is a crucial architectural step.

- **Memory (Context Storage):** Memory is the agent's contextual glue. Unlike a stateless API call, an agent carries state across multiple steps – it "remembers" what has happened so far.[15] This could include the conversation history with a user, the intermediate results it has generated, or key facts it has extracted. Memory can be short-term (within a single session) and long-term (persisted across sessions or available for future runs). For example, an agent doing legal research might store that "Cases A, B, C were already fetched" or "User's preference is to prioritise recent rulings" so it doesn't repeat work or contradict itself. Technically, memory can be stored in vectors, databases, or passed along in prompts (though prompt length limits constrain how much can be "remembered" at once). Effective use of memory enables context continuity – the agent's actions in step 5 can be informed by what happened in step 1.[15] It also supports learning: by logging outcomes to memory, an agent can avoid past errors and refine its approach over time.[15] In sum, memory turns a series of disjointed actions into a coherent workflow.

[14] What are agentic workflows? Patterns, use cases, and what to watch in 2026
https://www.wrike.com/blog/what-are-agentic-workflows/

[15] What Is an Agentic Workflow? A Guide to Autonomous AI Task Execution - SmartOSC
https://www.smartosc.com/what-is-an-agentic-workflow/

- **Feedback Loop:** The hallmark of agentic systems is the closed feedback loop for continuous improvement.[15] After the executor performs a task and gets a result, the agent doesn't just move on blindly – it examines the outcome. This could involve a Verifier module or simply the planner reconsidering its next move. If the result of a step is unsatisfactory or indicates a new problem, the planner can revise the plan or append follow-up steps.[16] For instance, if a tool returns an error (say an API is down), the planner might decide to try an alternative tool or ask for human help. Or if the agent's draft summary is too long, a reflection step might trim it. This feedback mechanism is often implemented via additional LLM queries that evaluate prior outputs (as in the reflection pattern[16]). The feedback loop is what makes the workflow agentic rather than linear – the system can iterate until it converges on a solution or reaches a stopping condition. It's important to set boundaries on the looping (to avoid infinite cycles) and define clear criteria for success vs. failure. When done right, the feedback loop dramatically improves quality and robustness, as the agent can catch its mistakes or adapt to new info mid-flight.

In practice, various architectures exist for these components. For example, recent research from Stanford proposed an AgentFlow framework with a **Planner – Executor – Verifier – Generator** structure coordinated by an explicit memory.[17] Many popular agent frameworks (LangChain, AutoGen, etc.) implement similar ideas under the hood. The take-home point is that an agentic workflow is modular: it's an AI system composed of distinct parts for planning, memory, tool use, execution, and self-reflection.[15] This modularity makes agents powerful – and also gives us multiple levers to manage and govern their behaviour (e.g. instrument the planner's prompts, restrict certain tools, log the memory state, etc.). Next, let's ground this in a concrete example from the legal domain.

[16] What are agentic workflows? Patterns, use cases, and what to watch in 2026
https://www.wrike.com/blog/what-are-agentic-workflows/

[17] Stanford Researchers Released AgentFlow: In-the-Flow Reinforcement Learning RL for Modular, Tool-Using AI Agents - MarkTechPost
https://www.marktechpost.com/2025/10/08/stanford-researchers-released-agentflow-in-the-flow-reinforcement-learning-rl-for-modular-tool-using-ai-agents/

Use Case: Agentic Workflow in Legal Services

To illustrate how agentic workflows can transform professional services, consider a **Legal Document Assistant** agent for a law firm. Legal work often involves labour-intensive processes like reviewing documents, researching case law, and ensuring compliance with regulations. An AI agent can automate large parts of this workflow:

- **Document Ingestion and Analysis:** The agent takes a new contract (or a stack of documents) as input. Using its planner, it outlines a plan: e.g.,
  "`First, read the contract and identify key clauses (payment terms, liabilities, termination clause, etc.). Next, compare each clause against our standard library or known regulatory requirements. Then, flag any deviations or risky terms. Finally, summarise the contract and any compliance issues in a report.`"
  The executor then carries out these steps. It may use a *PDF reading tool* to ingest the contract text, and an *LLM prompt* to extract clause summaries.

- **Cross-Checking and Research:** For each key clause identified, the agent invokes relevant tools or databases. For instance, it might query an internal knowledge base of regulatory rules or past contracts for comparison. If a liability clause is outside the norm, the agent could use a legal database API to pull up any pertinent laws or prior cases about such clauses. This is where the agent's ability to integrate with external systems is crucial – it acts like a diligent junior attorney

doing targeted research. The agent's memory stores the facts it has gathered (e.g., "Clause X deviates from policy Y" or "Regulation Z requires ABC") and uses them to formulate its next actions.

- **Summarisation and Reporting:** Once analysis is done, the feedback loop comes into play. Suppose the agent found 3 clauses of concern and retrieved regulatory text for each. It now synthesises these findings. The planner might add a step: `"Summarise each issue with a recommendation."` The executor (via the LLM) drafts a summary, such as: `"Clause 5 (Non-compete) is broader than allowed under [Jurisdiction] law; recommend narrowing scope."` The agent then compiles an executive summary of the entire document, highlighting risks and suggesting changes. This summary goes to the human lawyer for review.

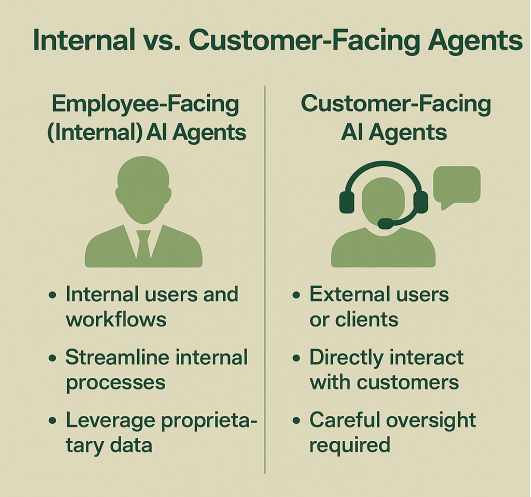
- **Validation and Learning:** After the human reviews the agent's report, any feedback (e.g., corrections or additional insights) can be fed back into the agent's memory. Over time, the agent learns the firm's preferences – e.g., what issues the lawyers care most about, or how to phrase recommendations. The next time a similar contract is reviewed, the agent is even more attuned. Importantly, throughout this process, every action the agent took (clauses found, databases queried, content generated) was logged. This audit trail gives the legal team confidence: they can inspect why the agent flagged something. It also helps with compliance – if asked to justify advice, they have the supporting references collected by the AI.

This illustrative use case shows an agent handling internal legal workflows: document review, case summarisation, compliance checking – exactly the kind of multi-step, knowledge-driven drudgery that AI is poised to streamline.[18] In fact, leading firms and tools are already moving in this direction. For example, Microsoft's Copilot Studio envisions an "automated contract review agent" that detects risky clauses and compliance issues, recommending edits to speed up legal reviews.[19] Likewise, AI copilots can summarise legal research and draft preliminary advice, cutting down the time attorneys spend on rote tasks.[19] By deploying an agentic workflow, a legal department can handle higher contract volumes, respond faster, and reduce human error in analysis – all while keeping human experts in control of final judgments.

[18] AI agents for legal document management: Applications and use cases, benefits and implementation
https://www.leewayhertz.com/ai-agents-for-legal-documents/

[19] Using Copilot in Legal (Copilot Scenario Library) – Microsoft Adoption
https://adoption.microsoft.com/en-us/scenario-library/legal/

Internal vs. Customer-Facing Agents



One of the core factors that could also heavily inform your decision on your agentic workflow design is who the end-user's are. As such, it is important to distinguish between **employee-facing** agentic workflows (used within an organisation by staff) and **customer-facing** agents (which interact directly with clients or end-users). Both share similar technology under the hood, but their design and governance differ:

Employee-Facing (Internal) AI Agents: These are agents that act as digital assistants or co-pilots for your teams. The purpose is to streamline internal workflows – from IT and HR tasks to project management and research.[20] Because they operate internally, they can be deeply integrated with company systems and data. For example, an employee-facing agent in Slack might retrieve data from an internal database, generate a report, or answer an employee's question using confidential company documents. These agents are context-aware about the business – they have access to files, knowledge bases, and tools like your project management app or CRM.[20] This rich context allows them to provide very personalised and accurate assistance to employees. A well-implemented internal agent acts like a "digital teammate" that can handle busywork, from drafting status updates to triaging support tickets.[20]Companies often start with internal agents because they can boost productivity and can be monitored/tweaked in a controlled environment before exposing AI to customers.[21]

Customer-Facing AI Agents: These agents engage with external users – prospects, customers, partners – often as front-line service or sales bots. As against the usual scripted chatbots these agents use reasoning and tools to handle more complex queries or transactions autonomously.[20] A customer-facing agent might help a user troubleshoot an issue, recommend products, or even negotiate a service plan, all in natural language. They aim to deliver a personalised experience at scale, maintaining your brand's voice and adhering to policies when interacting with people.[20] Unlike internal agents, customer-facing ones must be extra cautious – a mistake or offensive output directly impacts customers. Therefore, businesses impose strict content filters, fallback to human agents seamlessly when needed, and enforce that the AI doesn't overstep (for instance, it might have read-only access to account data and must escalate any sensitive changes to a human). The context they draw on includes customer profiles, interaction history, and public info – but typically not the full breadth of internal knowledge for confidentiality reasons.[22] When done right, these agents can provide 24×7 responsive service without burnout, and hand off to human support when queries get too tricky or sensitive.

[20] Employee-facing AI Agents: A Guide to Boosting Internal Productivity | Slack
https://slack.com/resources/why-use-slack/employee-facing-ai-agents

[21] Transforming Retail: What to Consider Before Deploying AI Agents
https://www.mytotalretail.com/article/transforming-retail-what-to-consider-before-deploying-ai-agents/

[22] Employee-facing AI Agents: A Guide to Boosting Internal Productivity | Slack

In practice, many enterprises will deploy both types. For example, in a legal context, an internal agent might assist lawyers as described, while a customer-facing legal AI might be a client-facing chatbot that answers common legal questions or helps clients fill out intake forms (with the agent knowing when to loop in a human attorney). The key is recognising the different requirements: internal agents can be more experimental and leverage proprietary data heavily, whereas external agents need rock-solid reliability, user-friendly behaviour, and careful oversight. Both should be built with governance in mind – but the stakes of an error differ (internal mistakes waste employee time; external mistakes could lose a customer or create liability). By addressing internal and external use cases, organisations can drive efficiency inside while also improving user experience outside.[22] [23]

Conclusion and Call to Action

Agentic workflows – multi-step, tool-using, goal-driven AI processes – represent a powerful evolution in how work gets done. They enable automation of complex tasks that previously required constant human intervention. As we've seen, deploying AI agents in production can yield major benefits in efficiency, decision quality, and scale. However, *agentic doesn't have to mean chaotic*. With careful design, robust guardrails, and the right partner, you can implement agentic systems **without the drama** of unreliability or loss of control.

At **ISx4**, we specialise in helping enterprise clients implement reliable AI solutions. We bring deep expertise in AI strategy and engineering to ensure your agentic workflows are *production-grade from day one*. That means we emphasise **auditability**, building in the observability tools and logs so you can always trace what the agent did.[24] It means focusing on **governance** – setting up the necessary permissions, human oversight points, and ethical guidelines so that your AI agents comply with your policies and industry regulations. And it means **infrastructure control** – deploying these agents on secure, scalable infrastructure (your cloud or on-premises) with integration into your existing tech stack and data stores. The result is an AI agent that works *for you* and with you, under the transparency and control that enterprise IT demands.

https://slack.com/resources/why-use-slack/employee-facing-ai-agents

[23] Internal Company AI: Use Cases and Best Practices - 1up.ai
https://1up.ai/blog/internal-company-ai/

[24] AI Agents in Production: Observability & Evaluation | ai-agents-for-beginners
https://microsoft.github.io/ai-agents-for-beginners/10-ai-agents-production/

If you're looking to leverage agentic AI workflows in your organisation – be it to supercharge internal operations or to deliver smarter customer-facing services – ISx4 is your trusted partner to make it happen. We invite you to reach out for a strategy discussion or pilot project. Our team will help you identify high-impact use cases, implement the agentic architecture tailored to your needs, and put the safeguards in place so you can deploy with confidence. With ISx4's guidance, you can harness autonomous AI agents to drive innovation and efficiency, *without the drama*. Let's build the future of reliable, goal-driven AI workflows together. **Contact ISx4** to get started.