FRACTO

2026 EDITION · ENTERPRISE AI SERIES

Agentic AI Workflow Design Playbook

A Practical Guide for Designing, Orchestrating & Governing AI Agents in the Enterprise

Agent Architecture | Orchestration Patterns | Integration | Security & Governance | 90-Day Roadmap

What's Inside

- Reference architectures for single-agent and multi-agent systems
- Nine agentic workflow patterns with decision criteria and when to use each
- Tool integration, permissioning, and observability checklists
- Orchestration models: centralized, decentralized, and hybrid
- Security, guardrails, and human-in-the-loop design
- 90-day pilot-to-scale roadmap with phases and success metrics
- 30-point Agentic AI Readiness Checklist
- Example prompts and orchestration flows for enterprise workflows

fracto.ie · contact@fracto.ie

Executive Summary

Most enterprises are notlimitedby AItools—they are limited by how they design and deploy AI workflows. You can have the best modelsand infrastructure in the world, but without the right agent architecture, orchestration patterns, and governance, agentic AI initiatives stall in pilot purgatory or fail when a workflow hits an edge case no one anticipated.

The data is unambiguous. A Perplexity–Harvard field study of hundreds of millions of agent interactions found that over half of all agent activity is real cognitive work: 36% of queries involve productivity and workflows, 21% involve learning and research. Enterprise platforms like Salesforce Agentforce and Microsoft Copilot Agents are reporting 30–60% reductions in manual workload when agents handle common workflows end-to-end. Futurum Group reports that 89% of CIOs now consider agent-based AI a strategic priority.

This playbook is a complete operating reference: how to choose your agent architecture, which orchestration patterns apply when,howto integrate agentssafely into your stack, and a 90-day roadmap to movefrom MVP to scale.

36%
of agentqueriesareproductivity & workflow tasks

89%
of CIOs rate agent-based AI as a strategic priority

60%
reduction in manual workload reported by early adopters

Three commitments for 2026

1. **A named workflow owner for each agentic deployment** — with clear scope, tool permissions, and KPI accountability tied to business outcomes, not AI outputs.

2. **An orchestration pattern chosen before engineering begins** — single-agent, orchestrator–workers, or hybrid — matched to the complexity and risk of the workflow.

3. **Guardrails and human-in-the-loop checkpoints defined up front** — least-privilege access, action whitelists, approval thresholds, and escalation paths before any agent touches production.

How to use this playbook

Read Parts 1–3 before your next architecture review to align on agent patterns and orchestration models. Use Part 4 (Integration & Governance) as the security checklist for any new agentic deployment. Run the Part 5 roadmap as the backbone of your 90-day pilot plan. Use the Part 6 checklist quarterly with your AI steering committee.

Contents

PART 1 What Is Agentic AI

1.1 From Assistants to Agents

Traditional LLMassistants generate content or answer questions in response to prompts. Agentic AI goes further — it reasons, plans, acts, and adapts across multiple steps without step-by-step human instruction.

Capability	Traditional Assistant	Agentic AI
Goal pursuit	Single-turn response	Multi-step task execution toward a defined goal
Autonomy	Waits for each prompt	Pursues sub-goals independently between checkpoints
Tool use	Text output only	Reads and writes to APIs, databases, SaaS tools, RPA
Adaptability	Stateless per session	Adjusts plans based on new data, errors, or feedback
Memor y	Context window only	Persistent state across steps and sessions

1.2 The Four Pillars of Agentic AI

Autonomy
Agents pursue goals over multiple steps without human instruction at each step. They break high-level objectives into sub-tasks and execute them in sequence or in parallel.

Intentionality
Agents maintain an internal representation of objectives and progress. They track what has been done, what remains, and what to do when a step fails or produces unexpected output.

Adaptability

Agentsadjust their plans based on new data, errors, or feedback. If a tool call fails or returns unexpected data, the agent reroutes — it does not simply stop.

Tool Use
Agents interact with external systems — APIs, SaaS apps, databases, file systems, RPA — to read, write, and update information as part of their workflow execution.

1.3 Where Agents Are Actually Used Today

Researchacrosshundredsofmillionsofagent interactions shows that agentic AI is already deeply embedded in knowledge work, with usage concentrated in two categories:

Category	Share of Agent Queries	Representative Tasks
Productivity & Workflow	36%	Planning projects, generating and editing documents, organizing information across Google Docs, Notion, Asana
Learning & Research	21%	Synthesizing reports, explaining complex concepts, exploring markets and technologies
Digital & Marketing	High stickiness	Campaign coordination, content generation, analytics workflows — once adopted, daily usage grows
Sales & Management	Power users 9× avg	Account research, pipeline management, meeting prep, stakeholder communications

The pattern: start low-stakes, scale fast

Users and teams typically start with low-stakes tasks (travel, trivia, search) and quickly graduate to complex, high-value work once trust is established. Design your first agent deployment to create that trust — then expand.

PART 2 Reference Architectures

2.1 Architecture Selection Criteria

Chooseyourarchitecturebeforeengineeringbegins. The three primary patterns differ on complexity, scalability, and governance fit.

Criterion	Single Agent + Tools	Orchestrator + Workers	Hybrid / Federated
Workflow complexity	Low–medium	Medium–high	High
Number of systems	1–3 tools	3–8 tools / roles	8+ tools / domains
Implementation speed	Fastest	Medium	Slowest
Obser vabilit y	Simple	Good with tracing	Requires governance layer
Failure blast radius	Low	Medium (isolated workers)	Contained by domain
Best for	Pilot MVP, focused tasks	Most enterprise use c ases	Multi-domain programs

2.2 Pattern A: Single Agent with Tools

One LLM-based agent withdirect accessto a defined set of tools and APIs. The agent reasons over each step and calls tools as needed to complete the goal.

> **When to use**
>
> Ideal for focused, linear workflows with 1–3 systems: e.g. "research an account → draft outreach → log to CRM" or "triage support ticket → draft response → update status." Fast to implement, easy to observe, low operational overhead. Start here for every new use case.

- **Tools to include:** Internal RAG / knowledge search, CRM read/write, ticketing, docs/email, scheduling
- **Pros:** Low complexity, fast to build, easy to debug and trace
- **Cons:** Limited scalability for multi-stakeholder or multi-domain workflows
- **Key requirement:** Well-scoped goal definition — broad or ambiguous goals cause single-agent drift

2.3 Pattern B: Orchestrator + Specialized Workers

Anorchestratoragent receives user goals, decomposesthem intotasks, selects appropriate specialized agents (workers), and manages the overall workflow. Workers focus on narrow capabilities.

Component	Responsibility	Example Specializations
Orchestrator	Goal intake, task decomposition, agent selection, result aggregation	Planning agent, routing agent
Worker agents	Execute a single narrow capability using specific tools	Contract reader, CRM updater, pricing modeller, email writer
Communication layer	Pass state, context, and results between agents with structured messages	LangGraph, n8n, AutoGen state
Memory store	Persist context, intermediate results, and workflow state across steps	Vector DB, key-value store

> **The single biggest orchestration mistake**
>
> Building worker agents that are too broad. A worker agent that can "handle all CRM tasks" becomes as complex as a single agent and defeats the purpose. Keep each worker to one well-defined capability with a small tool set. **Narrow workers = modular, testable, replaceable.**

2.4 Pattern C: Hybrid / Federated Orchestration

Domain-level orchestrators (e.g. Finance, Support, Sales) each manage their own worker agents, operating under a global governance and policy layer. Best suited for large programs spanning multiple business units.

Model	Structure	Best Fit	Risk
Centralized	One orchestrator, all workers report up	Early-stage, high governance need	Bottleneck at scale
Decentralized	Agents coordinate peer-to-peer	Flexible, resilient systems	Harder to audit and govern
Hybrid	Domain orchestrators under global policy	Multi-BU enterprise programs	Policy drift between domains

Most enterprises start centralized for control and auditability, then evolve toward hybrid as capabilities and trust mature. **Centralize early. Federate as capabilities mature.**

PART 3 Nine Workflow Patterns

These nine patterns cover the full range of agentic workflow designs. Most enterprise deployments combine 2–3 patterns. Match patterns to workflow structure, not to the technology stack.

PATTERN 01
Prompt Chaining

Sequential stepswhere each output feeds the next. The simplest agentic pattern.

- Analyze → Plan → Draft → Review
- Low complexity, high predictability
- Best for: linear, well-defined workflows

PATTERN 02
Plan-and-Execute

Agent builds a complete plan first, then executes steps, reviews outcomes, and iterates (PDCA loop).

- Separates planning from execution
- Good for complex multi-step tasks
- Best for: research, analysis, proposals

PATTERN 03
Evaluator–Optimizer

Oneagentgenerates outputs; a second critiques and feeds improvements back in a loop.

- Improves output quality iteratively
- Adds latency — use for high-value outputs
- Best for: content, legal, code review

PATTERN 04
Orchestrator–Workers

Orchestrator assignssubtasks to parallel specialized workers, then aggregates results.

- Concurrent execution — reduces latency
- Workers are modular and replaceable
- Best for: multi-system enterprise workflows

PATTERN 05
Routing

Aclassifier agent routes tasks to the most appropriate specialized agent based on content or context.

- Decouples intake from execution
- Enables multi-domain coverage
- Best for: support triage, intake workflows

PATTERN 06
Human-in-the-Loop

Workflow pauses atdefined checkpoints for human review or approval before continuing.

- Required for high-risk actions
- Builds trust in early deployments
- Best for: finance approvals, contracts, HR

PATTERN 07
Memory-Augmented

Agent persists structured knowledge across sessions and retrieves relevant context for each new task.

- Enables personalisation at scale
- Requires memory design and hygiene
- Best for: account management, onboarding

PATTERN 08
Agentic RAG

Agentdynamically retrieves from multiple knowledge sources during execution rather than using a single static context.

- Handles large, changing knowledge bases
- More expensive than static RAG
- Best for: legal, compliance, research

PATTERN 09
Reflection Loop

Agent critiques itsown output against defined criteria before returning a final result.

- Reduces hallucination and errors
- Adds compute and latency cost
- Best for: data extraction, factual tasks

Pattern Selection Guide

Workflow Type	Recommended Pattern(s)	Avoid
Linear, predictable task	Prompt Chaining (01)	Evaluator–Optimizer (adds unnecessary cost)
Complex research or analysis	Plan-and-Execute (02) + Agentic RAG (08)	Single-step chaining
Multi-system enterprise workflow	Orchestrator–Workers (04) + Routing (05)	Single agent (limits modularity)
High-risk or regulated actions	Human-in-the-Loop (06) + Reflection (09)	Fully autonomous execution
Repeating customer-facing tasks	Memory-Augmented (07) + Routing (05)	Stateless chaining (loses context)
Content or document generation	Evaluator–Optimizer (03) + Reflection (09)	Single-pass chaining for final outputs

PART 4 Integration & Governance

4.1 Connect to Systems of Record — Don't Replace Them

Alagentsbecomeexecution layers ontopof existingsystems, which remainsources of truth and governance. Agents read and write through well-defined APIs; they do not bypass security or data governance layers.

System Type	Agent Interaction Pattern	Governance Requirement
CRM / ERP	Read pipeline data, write activity logs and status updates via scoped API	Field-level write permissions; audit log on all writes
HRIS	Read employee data for routing and context; never write without approval	PII handling policy; human approval for any writes
Ticketing / ITSM	Read and triage tickets; write draft responses; escalate via defined rules	Escalation policy; auto-close rules reviewed quarterly
Finance / Procurement	Read spend and contract data; flag anomalies; trigger — not complete — approvals	Hard dollar threshold for human approval; full audit trail
Document stores	Search and retrieve; write drafts to staging areas only	DLP on all retrieved content; no direct publish to production

4.2 State, Memory & Observability

Robustorchestration frameworks(LangGraph, n8n, CrewAI, Ray, AutoGen) share four core infrastructure requirements. Treat these as non-negotiables before moving any agent to production.

Requirement	What It Covers	Key Tools
State management	Persistent context across workflow steps and sessions; ability to resume after failure	Redis, PostgreSQL, LangGraph state, AutoGen memory
Communication protocol	Structured messages and events between agents; typed schemas to prevent misrouting	JSON schemas, message queues, gRPC
Tracing & debugging	Replay agent runs step by step; inspect LLM decisions; tune prompts from production data	LangSmith, Langfuse, W&B Traces, Helicone
Fallback & escalation		

Requirement	What It Covers	Key Tools
	What happens when a tool call fails, an agent times out, or confidence is below threshold	Retry logic, circuit breakers, human escalation queues

> **Treat traces as first-class assets**
>
> Agent traces and decision logs are your primary tool for evaluation, governance, and continuous improvement. Design logging into your orchestration framework from day one — retrofitting observability is expensive and incomplete.

4.3 Security, Guardrails & Human-in-the-Loop

Control	What It Does	Implementation
Least privilege	Each agent uses a scoped service account with only the permissions required for its tools	Dedicated service accounts per agent; no shared admin keys
Policy-aware orchestration	Tool calls and data access pass through policy checks based on identity, context, data sensitivity	OPA (Open Policy Agent), custom middleware, attribute-based access
Input guardrails	Filter and validate all user inputs before the agent processes them	Prompt injection detection, input schema validation, content filters
Output guardrails	Screen all agent outputs before they are delivered or acted upon	DLP scanning, PII redaction, hallucination detection, action whitelists
Human-in-the-loop	Require explicit human review before high-risk or irreversible actions	Approval workflows in Slack/ Teams, async review queues, hard dollar thresholds
Action whitelisting	Agents can only call tools and actions explicitly approved for the workflow	Tool manifests with allowed-action lists; deny-by-default posture

PART 5 90-Day Pilot & Scale Roadmap

Phase	Weeks	Key Activities	Success Criteria
Phase 1 Identify & Scope	1–3	Interview 10–20 knowledge workers across 1–2 functions. Surface high-friction, multi-step workflows. Select one workflow with clear value and reasonable data readiness. Define KPIs.	One workflow selected with stakeholder sign-off and baseline metr ics established
Phase 2 Build MVP	4–6	Choose single-agent + tools pattern. Define narrow goal. Integrate 1–3 tools via APIs with least-privilege service accounts. Build basic trace logging. Run internal testing.	Agent completes end-to-end workflow in test environment with >80% task success rate
Phase 3 Harden & Observe	7–9	Run in shadow mode (agent proposes, humans approve). Instrument full traces and logs. Measure task success, time saved, error types. Add guardrails. Refactor to orchestrator + workers if complexity warrants.	Shadow mode operating stably; guardrails tested; error taxonomy documented; team trained
Phase 4 Expand & Standardize	10–13	Move low-risk tasks to active execution. Document patterns in internal agent library. Define agent governance policies. Identify adjacent workflow and reuse architecture. Plan next deployment.	One workflow in active production; governance policy published; second use case scoped

Phase 1 — Workflow Identification

Focus onworkflowswhere employeesalreadyuse AI informally. The Perplexity–Harvard research shows agents naturally gravitate to productivity & workflow (36%) and learning & research (21%) tasks — start your search there.

- Look for copy-paste, manual lookups, and repetitive status updates — these are agent-ready friction points
- Prioritise workflows with clear start/end states and measurable outcomes (time saved, error rate, cycle time)
- Avoid workflows with ambiguous success criteria or heavy regulatory exposure in the first pilot
- Score candidates on: value (time × frequency), data readiness, integration complexity, and risk level

Phase 2 — Single-Agent MVP

Resist the urge to build a multi-agent system on day one. A well-scoped single agent demonstrates value faster, is easier to debug, and builds the internal trust you need to expand.

- Define the goal in one sentence: "Take a support ticket from new → triaged + draft response in under 2 minutes"
- Limit tool access to exactly what the task requires — no extra permissions
- Log every decision, tool call, and output from day one
- Set approval thresholds early — define which outputs humans must review before action

Phase 3 — Shadow Mode

Shadow mode is not optional forenterprise deployments. Run the agent alongside your existing process, compare outputs to human decisions, and build a golden test set before moving to active execution.

- Measure task success rate, latency, error type distribution, and human override frequency
- Identify failure patterns and update prompts, tools, and guardrails accordingly
- Use shadow period to train the team on how to work with agents, interpret outputs, and escalate appropriately

> **Go/No-Go criteria before active execution**
>
> Before moving any agent from shadow to active: task success rate >90% on golden test set; all guardrails tested and passing; human escalation paths defined and tested; team trained; governance policy signed off by security and legal.

PART 6 30-Point Agentic AI Readiness Checklist

Use this checklist before launching any new agentic workflow. Run it quarterly with your AI steering committee for active deployments. Items marked **P0** are blockers — do not proceed to production without them.

Strategy & Use Cases

- [ ] At least one high-value, multi-step workflow identified and scoped for agentic automation
- [ ] Clear business KPIs defined: time saved, error rate, throughput, CSAT, or revenue impact
- [ ] Stakeholders and domain experts engaged in workflow design — not just IT
- [ ] Baseline metrics captured for the target workflow before agent deployment
- [ ] Success criteria defined: what does "working" look like after 30, 60, and 90 days?

Architecture & Orchestration

- [ ] **[P0]** Architecture pattern chosen: single-agent, orchestrator + workers, or hybrid
- [ ] Tool and system integrations mapped: APIs, permissions, data flows, and write scopes defined
- [ ] Orchestration framework selected and evaluated for state management and tracing support
- [ ] State management and context persistence designed for multi-step workflows
- [ ] Agent goal definition written in one sentence — narrow, measurable, and unambiguous

Security & Governance

- [ ] **[P0]** Agents use dedicated, least-privilege service accounts — no shared or admin credentials
- [ ] **[P0]** Policies guard high-risk actions and sensitive data access via OPA or equivalent
- [ ] **[P0]** Guardrails implemented on inputs, outputs, and tool calls before production
- [ ] **[P0]** Human-in-the-loop checkpoints defined for all sensitive or irreversible operations
- [ ] AI security and governance teams have reviewed the design and given sign-off
- [ ] Action whitelist defined — agent can only call explicitly approved tools and operations

Evaluation & Monitoring

- [ ] **[P0]** Golden test set created for the target workflow before production deployment
- [ ] Metrics tracked: task success %, latency, per-run cost, error taxonomy, safety incidents
- [ ] Tracing and logging instrumented on all agent runs from day one
- [ ] Online monitoring and canary rollout process defined for major prompt or tool changes
- [ ] Red-teaming or adversarial testing completed against critical agents before launch
- [ ] Shadow mode run completed with documented outcomes before active execution

Adoption & Change Management

- [ ] Target user groups trained on how to work with agents and interpret outputs
- [ ] Clear expectations set: what the agent will and will not do, and when to escalate
- [ ] Feedback channels established for users to report issues and suggest improvements
- [ ] Escalation path defined and communicated for when the agent fails or produces unexpected output
- [ ] Success stories captured and shared to build organisational trust and momentum

PART 7 Common Enterprise Use Cases

7.1 Customer Support & Service Operations

The highest-adoption enterprise use case.AgenticAliswell-suited for multi-step service workflows where rules are complex but outcomes are measurable.

Workflow	Agent Actions	Reported Impact
Ticket triage	Classify, route, and prioritise incoming tickets across channels based on content and context	40–60% reduction in manual triage time
First-response drafting	Read ticket, search knowledge base, draft response, flag for human review above confidence threshold	30–50% reduction in response time
CRM and KB updates	Auto-update case status, log activities, keep knowledge base current after resolution	Eliminates 15–25 min of admin per ticket
Proactive follow-up	Monitor outcomes, trigger follow-up messages based on resolution status and SLA	15–20% improvement in CSAT scores

> **Design principle: handle → escalate, not handle → replace**
>
> The winning pattern is agents handling the 60–70% of cases that follow known patterns, while escalating the remainder to humans with full context already assembled. Agents that try to handle everything autonomously create more problems than they solve.

7.2 Sales, Marketing & RevOps

- Research accountsand contacts;summarise buying committees and recent news before calls
- Generate and personalise outreach sequences at scale based on account context
- Keep CRM fields clean and up-to-date; log activities automatically after meetings
- Coordinate campaigns across email, ads, and content with consistent messaging
- Monitor pipeline health; flag at-risk deals; draft re-engagement outreach

7.3 Finance, Procurement & Supply Chain

- Continuous monitoring ofspend, anomalies,andbudget variances with automated alerting
- Autonomous drafting of purchase requisitions and approval requests (human approval required)
- Contract review: read contracts, flag non-standard clauses, summarise risk for legal review

- Demand forecasting support: multi-agent analysis across inventory, logistics, and sales data

> **Finance boundary: trigger, don't execute**
>
> In finance and procurement workflows, agents should trigger approval workflows and surface recommendations — they should never execute financial transactions autonomously. Design hard thresholds: any action above a defined dollar value requires human sign-off, enforced at the guardrail layer, not the prompt layer.

7.4 Internal Knowledge & Productivity

- Unified search across internal wikis, tickets, code repositories, and documents
- Meeting summarisation, action item extraction, and automatic task creation in Asana/Jira/Notion
- "Ops copilot" agents: watch Slack/Teams, triage alerts, propose next actions to the right owner
- Onboarding agents: guide new hires through documentation, answer questions, track completion

7.5 ROI Benchmarks by Use Case

Use Case	Primary Metric	Typical Range	Time to Value
Support triage & drafting	Manual workload reduction	30–60%	4–8 weeks
Sales research & CRM hygiene	Time saved per rep/ week	3–6 hours	6–10 weeks
Procurement monitoring	Spend anomalies caught	2–4× baseline	8–12 weeks
Internal knowledge search	Search time per query	60–80% reduction	4–8 weeks
Meeting summarisation	Admin time per meeting	15–25 min saved	2–4 weeks

PART 8 Frequently Asked Questions

What's the difference between an AI assistant and an AI agent?

Assistants primarily respond — they answer questions and generate content. Agentsboth reason and act:

they plan multi-step workflows, call tools, update systems, and adapt over time. In enterprise contexts, that means less copy-paste and more end-to-end workflow execution. The LLM is the reasoning core; the agent layer is the hands and feet.

Should we build our own agents or use vendor platforms?

Pre-builtenterprise agent platforms(SalesforceAgentforce, Microsoft Copilot Agents, ServiceNow Now Assist) deliver faster ROI and stronger built-in governance, especially for workflows centred on the vendor's own systems. DIY frameworks (LangChain, CrewAI, LangGraph, AutoGen) offer more customisation but require significantly more engineering and operational investment.

> **Recommended approach**
>
> Start with vendor platforms for core systems (CRM, ITSM, ERP). Use DIY frameworks for niche, cross-system, or highly custom workflows. The right answer is almost always a hybrid — don't force everything through one vendor's platform.

How do we prevent agents from going "rogue"?

Design for constrainedautonomyfromthe start: narrow,well-scoped goals; least-privilege tool access; explicit action whitelists; output guardrails; and human-in-the-loop checkpoints for high-risk actions. Treat agents like powerful new colleagues who need supervision — not infallible systems that can be trusted unconditionally.

The three most common failure modes are: (1) too-broad goal definitions that allow the agent to take unintended paths; (2) overly permissive tool access that enables unintended writes; and (3) missing escalation paths that let errors compound silently.

How is this different from traditional RPA or workflow automation?

Traditional RPAautomatesdeterministic, rule-based processes —it follows explicit instructions perfectly but breaks when conditions change. Agentic AI adds reasoning, adaptability, and language understanding, making it suitable for messy, semi-structured tasks where rules are incomplete or changing.

In practice, enterprises should combine both: RPA for stable, high-volume back-office processes; agents for dynamic knowledge work. They are complementary, not competing.

What should we measure to know if our agents are working?

Metric Category	Specific Metrics
Task performance	Task success rate, golden test set pass %, human override frequency
Efficiency	Time saved per workflow run, cycle time reduction, throughput increase
Cost	Cost per workflow run, cost vs. human baseline, cost trend over time
Quality	Error rate by type, escalation rate, user satisfaction scores
Safety	Guardrail trigger rate, policy violations, anomalous action rate

Where should we start?

Followthedata:agents naturallygravitate to productivity & workflow and learning & research tasks. Start with one painful, high-friction workflow for a high-value team — support triage, sales research, or procurement analysis are consistently high-ROI starting points. Design a single-agent MVP with clear success criteria, run it in shadow mode, and iterate from there.

The 90-day roadmap in Part 5 of this playbook is designed to get you from identification to a production-ready deployment with confidence.

WORKSHOP & NEXT STEPS

How Fracto Can Help

Fracto works with enterprise teams to move from slides about agents to measurable impact in operations. We bring architecture expertise, implementation experience, and the external perspective to challenge assumptions before they become expensive mistakes.

Agentic AI Workflow Assessment

A structured engagement designed for teams evaluating or actively planning their first (or next) agentic AI deployment. In a focused workshop, we:

- Identify high-ROI candidate workflows using the prioritisation framework in this playbook
- Map your current systems, data readiness, and integration complexity
- Choose the appropriate agent architecture — vendor vs. DIY, single vs. multi-agent
- Build a 90-day implementation roadmap with KPIs, guardrails, and governance checkpoints
- Identify risks and pre-empt the most common failure modes before engineering begins

What You Get

Workflow Prioritisation Matrix

Scoredlist of candidate workflowsranked by ROI potential, data readiness, and integration complexity — so you start with the highest-confidence deployment.

Architecture Blueprint

Recommended agent pattern, tool integrations, orchestration model, and governance design tailored to your specific workflow and stack.

90-Day Roadmap

Phase-by-phase implementation plan with milestones, KPIs, resource requirements, and go/no-go criteria at each phase gate.

Governance & Security Review

Checklist-based security andgovernance review mapped to your specific workflow risk profile — guardrails, access controls, and escalation paths defined before engineering.

Implementation Support

For teams that need hands-on support beyond the assessment, Fracto provides embedded implementation expertise across the full agentic AI stack — from orchestration framework selection and tool integration to prompt engineering, evaluation design, and production readiness review.

Book an Agentic AI Workflow Assessment

Movefrom slides aboutagentstomeasurableimpactin your operations. We work with a limited number of enterprise clients per quarter.

📧 contact@fracto.ie · 🌐
fracto.ie