



# W.

## Coday: Architecting the Enterprise Operating System at Whoz

---

A Case Study in Agentic AI,  
Memory-Driven Knowledge, and  
Organizational Transformation



# Table of Contents

<u>Table of Contents</u>	1
<u>Executive Summary</u>	3
<u>Introduction</u>	4
<u>1. Coday Defined: Architecture and Capabilities</u>	5
<u>1.1 Genesis of Coday</u>	5
<u>1.2 Orchestration Layer and Multi-Agent System</u>	5
<u>1.3 Tool-Grounded Workflow Engine</u>	7
<u>1.4 Governed Memory System</u>	8
<u>1.5 Standardization: Templates and Checklists</u>	8
<u>2. Why Coday Matters: Strategic Motivations</u>	10
<u>2.1 Overcoming the Tacit–Explicit Knowledge Divide</u>	10
<u>2.2 Accelerating Speed, Raising Quality, and Institutionalizing Consistency</u>	10
<u>2.3 From Assistant to Operational Teammate: Agents as Process Owners</u>	11
<u>2.4 Enhancing Decision-Making with Embedded Collective Intelligence</u>	12
<u>2.5 Future-Proofing Organizational Memory and Strategic Alignment</u>	13
<u>3. Organizational Integration: Use Across Functions</u>	15
<u>3.1 Enhancing Client Relations and Customer Success</u>	15
<u>3.2 Driving Operational Excellence</u>	15
<u>3.3 Building Robust HR and Compliance Foundations</u>	16
<u>3.4 Transforming R&amp;D, Engineering, and Product Workflows</u>	16
<u>3.5 Closing the Loop with Support and Field Operations</u>	17
<u>3.7 Toward Organizational Synergy</u>	17
<u>4. Strategic Impact and Measurable Outcomes</u>	18
<u>4.1 Standardization at Scale: Raising the Floor Across Functions</u>	18
<u>4.2 Quantified Benefits: Demonstrable Efficiency and Quality Gains</u>	19
<u>4.3 Knowledge Scalability and Organizational Memory</u>	23
<u>4.4 Strengthening Governance, Traceability, and Compliance</u>	24
<u>4.5 Alignment of Execution with Strategic Imperatives</u>	25
<u>5. Overcoming Polanyi's Paradox in Software Engineering</u>	26
<u>5.1 Background: The Nature and Relevance of Polanyi's Paradox</u>	26
<u>5.2 Coday's Approach: Systematic Capture and Institutionalization of Tacit Knowledge</u>	26
<u>5.2.1 Digital Capture of Tacit Interactions</u>	27
<u>5.2.2 Standardization and Synthesis</u>	27
<u>5.2.3 Human-AI Co-Learning and “Ask for Help” Loops</u>	28
<u>5.2.4 Continuous Codification and Evolution</u>	28
<u>5.3 Practical Impact and Organizational Benefits</u>	29
<u>5.3.1 Accelerated Learning and Onboarding</u>	29

<u>5.3.2 Standardization and Reduction of Cognitive Load</u>	29
<u>5.3.3 Sustainable Institutional Memory</u>	30
<u>5.3.4 Next-Level Human-AI Collaboration</u>	30
<u>5.4 Case Study: Coday in Action</u>	30
<u>5.4.1 Debugging Workflow</u>	30
<u>5.5 Limitations and Critical Reflection</u>	32
<u>5.6 Strategic Implication: The Enterprise Beyond Polanyi</u>	33
<u>6. Limitations and Mitigation Strategies</u>	35
<u>6.1 Hallucination and Context Drift</u>	35
<u>6.2 Integration and Permissions Complexity</u>	36
<u>6.3 Adoption and Change Management</u>	37
<u>6.4 ROI and Impact Evaluation</u>	38
<u>6.5 Cost and Performance Optimization</u>	39
<u>7. The Road Ahead: Vision and Roadmap</u>	41
<u>7.1 Memory Segmentation</u>	41
<u>7.2 Expanded Retrieval-Augmented Generation (RAG)</u>	42
<u>7.3 Security and Link Governance</u>	42
<u>7.4 Test and Documentation Automation</u>	43
<u>7.5 Agent Analytics and Expansion</u>	43
<u>7.6 Sales and Success Enablement</u>	44
<u>8. Measuring Success: Recommendations and Prospective Metrics</u>	45
<u>8.1 Evolving Measurement Practices Across Core Functions</u>	45
<u>8.2 Continuous Feedback and Adaptive Measurement</u>	46
<u>8.3 The Strategic Role of Prospective Metrics in Coday's Evolution</u>	46
<u>Conclusion</u>	48
<u>Appendix - Examples of Coday Agents</u>	49

# **Executive Summary**

The exponential growth of artificial intelligence (AI) and multi-agent systems has ushered in a new era of enterprise productivity. Whoz, a technology-driven organization, stands at the forefront of this revolution through the development and deployment of Coday—an internal, enterprise-grade orchestration layer and AI assistant. By embedding Coday within organizational workflows and connecting 40+ specialized agents across technical and business domains, Whoz has accelerated delivery, standardized knowledge, and reduced operational risk.

This paper explores the strategic significance of Coday as Whoz's "Enterprise Operating System" (EOS). It analyzes design principles, impact across functions, measurable outcomes, knowledge management innovations, compliance approaches, and the roadmap for further transformation. Special attention is paid to how Coday addresses Polanyi's Paradox, converting tacit employee know-how into explicit, reusable, and dynamically evolving digital assets.

# **Introduction**

Organizations globally are grappling with the challenge of scaling expertise and standardizing execution without stifling creativity or agility. Traditional knowledge management systems frequently fragment institutional memory, while isolated AI assistants fail to support end-to-end business outcomes. Whoz's response to these challenges is Coday: more than an assistant, it is a digital backbone—an orchestrator, workflow engine, governed operator, and institutional learning platform.

Originating in software development and product management, Coday's initial focus was to solve the acute needs of technical teams. This paper therefore devotes significant attention to engineering and product examples, reflecting both Coday's roots and the early manifestation of its value. Over time, however, Coday has evolved to become pervasive throughout the Whoz organization. It now supports workflows, standardization, and knowledge retention across departments—including sales, operations, client relations, HR, and compliance. Where relevant, examples from outside the technical domain are shared, and it is expected that further cross-functional exploration will take place.

Coday operates not just as a technical tool but as an organizational catalyst. It reduces silos, codifies best practices into accepted standards, and bridges intuition-driven, tacit knowledge with auditable, repeatable processes.

# 1. Coday Defined: Architecture and Capabilities

## 1.1 Genesis of Coday

Coday originated as the individual initiative of a Whoz developer, Vincent Audibert, who was motivated to harness the potential of AI and large language models (LLMs) to streamline and enhance his daily work. At the time, most available frameworks for agentic development and LLM integration were fundamentally Python-driven, offering little direct support for Java—the principal technology stack used at Whoz. Rather than undertake the considerable effort of retraining technical teams in Python, Audibert embarked on designing Coday as a purpose-built solution, optimized for Whoz's Java-centric environment.

Additionally, early external solutions lacked the flexibility to connect smoothly with third-party systems integral to Whoz's workflows, as mature multi-connector platforms (such as MCP) either did not exist or were only in their nascent stages. Recognizing both the technical limitations and the strategic opportunity, Whoz's leadership soon identified Coday as a foundation for the company's broader agentic ambition. It became clear that thriving in the evolving landscape would require adopting an agent-first mindset, where coordinated digital agents could augment and eventually orchestrate key aspects of the organization's operations.

As a result, Coday's adoption accelerated organically within Whoz, expanding from a developer-centric tool to one that also empowered product managers and, gradually, stakeholders across every department. Through continuous, hands-on refinement and real-world use, Coday became increasingly relevant and productive—its utility and efficiency growing in direct response to its roots in everyday organizational practice.

## 1.2 Orchestration Layer and Multi-Agent System

At its core, Coday operates as an adaptive orchestration engine designed to intelligently route requests to a catalog of specialized agents, each encapsulating distinct domains of operational or technical expertise. Unlike conventional workflow or rule engines—which follow static, predetermined process flows—Coday's architecture is dynamically adaptive: it leverages contextual

awareness, historical memory, and deep organizational understanding to deliver tailored outcomes.

Coday maintains a persistent memory system, recording past interactions and user preferences, and possesses in-depth, evolving knowledge of Whoz's organizational structure, business priorities, and technical platform. This allows Coday not only to optimize routing and agent selection based on real-time context, but also to surface relevant institutional knowledge, relevant reminders, and provide continuity across user experiences.

Its agents, therefore, extend far beyond simple automation, actively enhancing processes from release management, client support, and accessibility validation, to sales enablement, knowledge management, and compliance—each action grounded in both organizational history and current context.

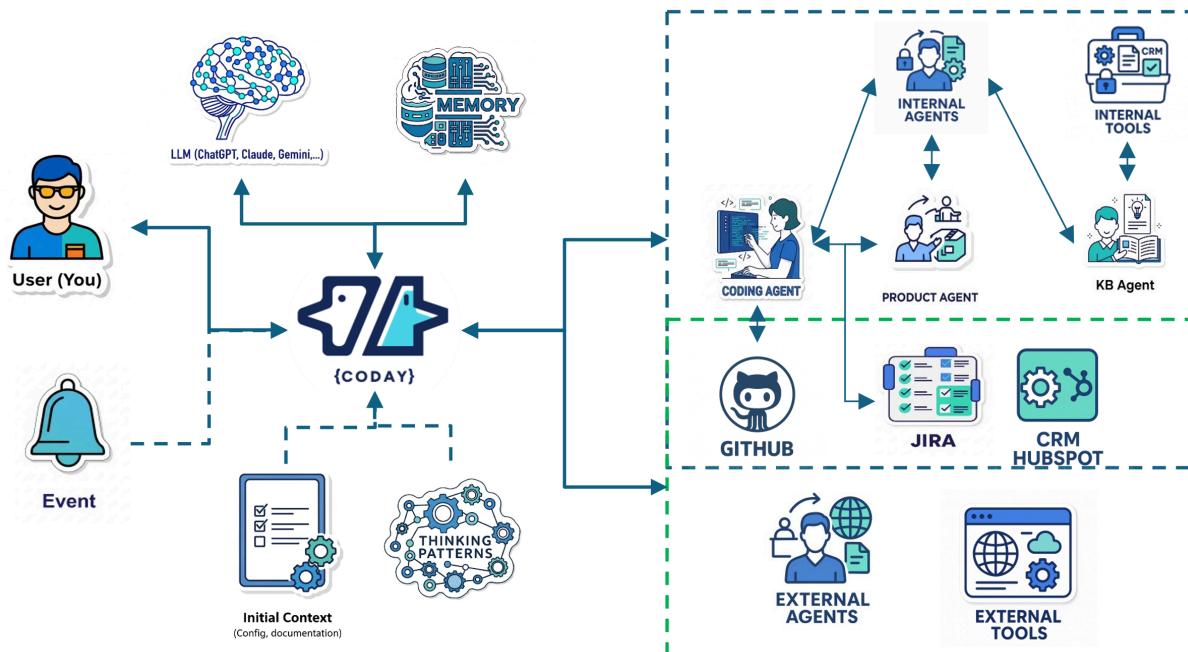
Unlike conventional enterprise tools—which typically address isolated functions or domains—Coday serves as an *Enterprise Operating System* (EOS): a foundational digital layer that unifies, coordinates, and supports all aspects of the organization's work. Just as a computer operating system manages resources, enforces standards, and provides essential services to applications, Coday orchestrates the flow of information, the execution of processes, and the memory of organizational interactions, ensuring coherence and resilience at scale. In this capacity, Coday delivers five core capabilities:

- **Breaking down departmental silos:** By intelligently routing information and requests between teams, Coday fosters seamless collaboration and alignment across functions that would otherwise be fragmented.
- **Providing ubiquitous access to knowledge:** The system provides all users—regardless of role, location, or team—with consistent, governed access to organizational intelligence, removing barriers to informed decision-making.
- **Automating coordination and routine tasks:** Coday takes ownership of repetitive, cross-team coordination—such as approvals, notifications, knowledge handoffs, and compliance checks—allowing human talent to focus on judgment, creativity, and high-value initiative.
- **Enabling holistic, data-driven management:** By capturing, contextualizing, and disseminating real-time data from across the enterprise, Coday

empowers teams and leaders to base their decisions on the organization's collective operational picture.

- **Preserving and evolving institutional memory:** Through persistent memory and documentation of interactions, workflows, and rationale, Coday safeguards critical knowledge—mitigating the risks posed by personnel transitions and ensuring continual organizational learning.

It represents Whoz's commitment to building not just innovative products, but also an innovative organization—one where information flows freely, decisions are informed by data, and human potential is maximized through intelligent support systems.



### 1.3 Tool-Grounded Workflow Engine

Coday is tightly integrated within Whoz's day-to-day systems—Jira, Confluence, Git, Angular/Nx toolchain, Gradle, Cypress, Playwright—allowing agents to read, write, and refactor code; run builds and tests; manipulate project files; and access documentation, all within controlled permissions and audit trails. Importantly, Coday's open, extensible architecture is designed to accommodate future integrations as organizational needs evolve.

Plans are already underway to incorporate additional platforms such as Notion for collaborative documentation, Basecamp for project management, Hubspot for CRM, and other tools that may arise as Whoz's technology landscape grows. This adaptability ensures that Coday remains a single, reliable orchestration hub—synchronizing operations across both current and emerging technologies as the enterprise ecosystem advances.

## 1.4 Governed Memory System

Unlike conventional AI that simply generates responses, Coday is distinguished by its selective, long-term memory, retained at three levels: user, project, and organization. This memory architecture is fundamental for enforcing organizational standards, preventing knowledge sprawl, and ensuring that the most current and relevant best practices are always accessible. Governance is embedded at every stage, with stringent permission controls and clear contextual boundaries for what is remembered and by whom it can be accessed.

Coday's long-term memory is a cornerstone of its effectiveness, offering two principal advantages. First, it guarantees that critical information—from daily user interactions or direct ingestion of reference documentation—is systematically captured and utilized to inform present and future interactions. Second, its memory can be actively curated: as users interact with Coday, significant insights and enduring solutions are identified for retention, while less valuable or obsolete information is pruned away. This ensures that Coday's knowledge base remains both high-signal and adaptive, evolving alongside the organization's needs and the latest standards.

## 1.5 Standardization: Templates and Checklists

Agentic outputs within Coday are anchored in deterministic templates and controlled glossaries—spanning translation workflows, release notes, accessibility themes, and troubleshooting guides. These standardized structures are fundamental for ensuring output consistency, expediting localization, and elevating organizational quality.

It is worth noting that if Coday's environment were comprised solely of machine-to-machine interactions, templates might eventually become obsolete. In such a world, agents could dynamically negotiate optimal communication modalities, evolving formats continuously to optimize data exchange. However, because the ultimate recipients of these outputs are humans, maintaining template-based, glossary-driven formats remains essential. Templates provide a reliably structured framework that aligns with human cognitive patterns, ensuring information is accessible, interpretable, and actionable for users across the organization.

The process is thus a blend of determinism and adaptivity: while the format is strictly defined, the content generated within that structure is non-deterministic. Coday synthesizes output by mobilizing the most relevant agents, integrating diverse sources, and adapting to the specific operational context—all while adhering to standardized presentation. Occasionally, due to the inherent limitations of large language models, Coday may deviate from the expected format. Such deviations can be viewed either as temporary errors to correct or, at times, as prompts for users to reconsider and evolve established conventions, sparking organizational reflection on effective communication.

In this way, Coday leverages deterministic structure as a bridge between the creative, adaptive power of generative AI and the enduring need for human-centric clarity and reliability.

## **2. Why Coday Matters: Strategic Motivations**

In the modern knowledge economy, the fundamental challenge of transforming individual expertise into scalable, repeatable organizational capability lies at the heart of sustained enterprise advantage. Michael Polanyi's observation—that "we know more than we can tell"—describes the persistent organizational dilemma of tacit knowledge: expertise that is internalized by practitioners but escapes explicit codification and systematization. Coday was architected explicitly to address this paradox—translating deeply held organizational intuition into accessible, actionable, and durable knowledge assets.

### **2.1 Overcoming the Tacit–Explicit Knowledge Divide**

Critical workflows across software engineering, product management, sales, marketing, legal, and support are suffused with practices and insights that are rarely formalized:

- Experienced technologists and operators solve problems and drive innovations based on pattern recognition and intuition developed over time.
- These solutions—often only partially articulated in meetings, chats, or one-off documentation—are lost in translation across teams and with team turnover.

Coday's multi-agent, memory-driven, and embedded design targets this friction by systematically capturing, distilling, and redeploying tacit knowledge. High-impact approaches and solutions become accessible at the right time, for the right stakeholder, closing the knowledge transfer gap at organizational scale.

### **2.2 Accelerating Speed, Raising Quality, and Institutionalizing Consistency**

Through agentic automation, Coday enables the organization to achieve new levels of throughput, quality, and reliability.

Agent workflows, powered by memorialized knowledge, compress cycle times in a wide range of activities—including documentation, code review, quality assurance, client support, project management, sales communications, onboarding, and more. This organizational automation empowers humans throughout all departments to focus on intent, judgment, and creative problem-solving rather than repetitive tasks. Standardized templates and controlled glossaries, natively embedded in these agent workflows, ensure that every artifact—regardless of its origin or business function—reflects the latest, validated best practices.

## **2.3 From Assistant to Operational Teammate: Agents as Process Owners**

Crucially, Coday's deployment represents a profound paradigm shift—from the classical AI-as-assistant model, where systems serve only as advisors or passive contributors, to AI-as-operational-teammate, where agents assume direct accountability for discrete business processes. In this new model, Coday agents do not simply suggest next steps: they initiate, execute, validate, and fully document organizational workflows. Each agent-driven process is anchored to explicit definitions of done—criteria that clarify not only the intended outcome but also the agent's zone of decision-making autonomy. Rather than passing through a rigid set of instructions, these agents devise and complete actions within established guidelines, exercising judgment on behalf of their human collaborators. Mechanical execution and process documentation are thus handled end-to-end by Coday, allowing human expertise to focus on creative problem-solving, intent-setting, and exception handling.

This shift brings critical implications for organizational reliability, measurement, and scalability. Workflows managed by digital teammates become less dependent on any individual, more consistent across iterations, and inherently auditable. Over time, the locus of complex knowledge work moves from the memory and availability of subject matter experts to the orchestration and transparent execution of digital agents—liberating employees to invest their energies in tasks that require uniquely human discernment or empathy.

This transformation is best framed through the lens of operational autonomy as conceptualized by Azeem Azhar and his "99% step-length" theory. Traditional AI progress is often measured by metrics such as model size, processing speed (FLOPS), token counts, or benchmark scores. Yet, as Azhar argues, these technical indicators fail to capture the fundamental shift now underway: the real inflection point is the duration for which an AI system can reliably operate at or above 99% autonomy without human intervention. Below this threshold, autonomy proves illusory—continuous oversight and frequent correction by humans are necessary. At 99%, however, AI systems approach true operational independence, reliably managing workflows and executing tasks with minimal human oversight.

Remarkably, Azhar observes that this "99% step-length"—the number of sequential steps an AI system can take before requiring intervention—has been doubling approximately every seven months. If this trend persists, AI systems like Coday will, by the early 2030s, autonomously manage not just standalone workflows but entire, multi-month strategic programs: the digital equivalent of a human career, executed continuously and safely by intelligent agents.

For Whoz, the implications are transformative. As Coday's agents steadily increase their "step-length," the company moves beyond basic automation, toward an era where collective organizational intelligence, memory, and process integrity can be sustained and compounded for months at a time. This evolution does not replace human ingenuity; rather, it augments and liberates it—enabling personnel to focus on intent, design, and vision, while entrusting compliance, execution, and continuous learning to their digital teammates.

## 2.4 Enhancing Decision-Making with Embedded Collective Intelligence

By surfacing the highest-signal, contextually relevant knowledge at the moment of need, Coday embeds collective intelligence directly into the organization's workflows. Agentic action is always grounded in artifact-based traceability, ensuring quality and explainability. This mechanism not only accelerates decision-making but allows each case and outcome to feed a virtuous learning

cycle, continuously shrinking the gap between individual intuition and institutional best practice.

## 2.5 Future-Proofing Organizational Memory and Strategic Alignment

Finally, Coday's architecture of selective, permissioned memory plays a critical role in ensuring that organizational knowledge persists despite personnel transitions. This has led to tangible benefits such as shorter onboarding cycles, reduced knowledge loss, and greater resilience—even as priorities and teams evolve—while also enabling closer alignment between day-to-day activities and strategic direction.

Yet, as the scope and sophistication of agentic workflows grow, memory management emerges as perhaps the greatest challenge ahead. The value of Coday's memory lies in its potential to transform isolated expertise into collective organizational intelligence. However, the design of memory—whether shared or isolated—raises profound issues, especially in sensitive domains like Human Resources. For example, creating an agent capable of assessing individual performance could deliver significant value, but it also risks exposing personal data if memory is shared across contexts. Conversely, if memory is too narrowly isolated, the benefits of cross-team learning and shared best practices are diminished.

Striking the right balance—maximizing the leverage of collective experience while protecting confidentiality and individual privacy—will be a central challenge as Coday and agentic paradigms advance. Ongoing work at Whoz is thus focused on building robust memory governance: frameworks and permission models that allow for dynamic, context-aware memory sharing, ensuring both the power of aggregated learning and the assurance of compliant, ethical information handling.

---

In summary:

Coday is not simply an automation platform or a digital assistant. By transforming intelligent agents into operational teammates, Coday enables Whoz to overcome Polanyi's Paradox; process ownership and embedded memory raise the floor of institutional performance and accelerate adaptation. This agentic infrastructure forms the foundation not only for rapid delivery and consistent quality, but also for the emergence of an intelligent, self-improving enterprise.

This evolution is directly aligned with Whoz's Agentic AI vision: in the near future, every client instance of Coday will be capable of orchestrating autonomous workforce operations across thousands of actions, achieving human-level reliability for extended periods without intervention. Already, our internal Agentic Operating System structures agents in step-based orchestration loops, positioning Whoz at the forefront of mastering the "99% frontier"—where operational autonomy and persistent organizational intelligence become the rule, not the exception.

## **3. Organizational Integration: Use Across Functions**

The value of Coday as an enterprise operating system resides not only in its technical architecture, but in its capacity to permeate and elevate the day-to-day work of every major business function at Whoz. Its multi-agent, memory-driven design has enabled deep integration with workflows spanning client engagement, operations, engineering, compliance, and more. Rather than functioning as an isolated support layer, Coday has become an active agent of organizational unification—translating best practices and expertise into tailored, context-sensitive automation across departmental boundaries.

### **3.1 Enhancing Client Relations and Customer Success**

Coday has been instrumental in transforming client-facing workflows, particularly in the articulation and delivery of product changes and support solutions. With specialized agents orchestrating the mapping of product release content to client-specific tickets, communications have become both more consistent and more responsive to real-world needs. The generation of tailored release notes—including automated analysis and synthesis across Jira, and Confluence—has helped minimize review cycles and improve the clarity of executive communication. Meanwhile, the standardization of troubleshooting documentation and structured knowledge base articles empowers client support teams to provide rapid, effective, and traceable resolutions, leading to measurable reductions in escalation rates and increased client satisfaction.

Coday further bridges the gap between UX research and product adoption by automating the synthesis of usability insights. This enables client-facing teams to move more quickly from research to actionable recommendations, ensuring that feedback not only reaches the right stakeholders but informs subsequent releases in a tangible, data-driven manner.

### **3.2 Driving Operational Excellence**

This real-time linkage between operations and product initiatives fosters a shared language between technical and non-technical teams, accelerating consensus

and elevating the precision of both forecasting and execution. By making institutional expertise accessible through operational workflows, Coday operationalizes resilience and agility within the core of the business.

### **3.3 Building Robust HR and Compliance Foundations**

In the domains of human resources and compliance, Coday systematizes previously ambiguous processes. By maintaining onboarding guides, clarifying permission and role mapping, and providing templated GDPR/sub-processor narratives, Coday minimizes confusion and accelerates both internal ramp-up and external due diligence. The system actively reduces risks associated with information silos and compliance misstatements—ensuring that evolving legal, operational, and security requirements are embedded within daily organizational routines.

### **3.4 Transforming R&D, Engineering, and Product Workflows**

Within software engineering and product management, Coday agents have become trusted collaborators throughout the development lifecycle. Automated code and quality reviews, end-to-end test generation, and Storybook documentation are all synchronized with Jira tickets and linked to standard acceptance criteria. API conformance checks and accessibility validations are executed proactively by agents, reducing manual errors and improving release quality across the board.

Moreover, Coday streamlines interactions between product, R&D, and UX teams, codifying user stories, test case methodologies, and release communications into orchestrated workflows. By transforming lessons learned and proven expertise into accessible, template-driven standards, Coday eliminates unproductive variance and empowers teams to move faster, with greater confidence.

## **3.5 Closing the Loop with Support and Field Operations**

In support and field operations, Coday helps automate the classification, and initial response for client tickets. Standardized troubleshooting documentation reduces ambiguity and improves searchability, while permission-aware knowledge sharing guarantees that only the right individuals access sensitive information. This has led to faster meaningful responses, and increased self-service success, ultimately elevating the client experience.

## **3.6 Toward Organizational Synergy**

The integration of Coday across these diverse functions is not simply additive—it is synergistic. By creating interoperable workflows, enforcing shared standards, and embedding process-specific memories, Coday enables continuous learning and adaptation as the needs of the business evolve. The sum total is a more cohesive, adaptive, and high-performing organization, with knowledge infrastructure that scales in lockstep with Whoz's strategic ambitions.

## **4. Strategic Impact and Measurable Outcomes**

The ultimate measure of an enterprise system's value is not only its theoretical capabilities, but its demonstrable effect on organizational performance. Having examined Coday's foundational design, its systematic approach to overcoming Polanyi's Paradox, and the agentic re-engineering of core workflows, we turn now to evidence of transformation. This section synthesizes quantitative and qualitative impacts observed at Whoz since Coday's deployment, spanning error reduction, time savings, client engagement, knowledge retention, and alignment with key business imperatives. Through concrete examples and targeted metrics, it becomes possible to distinguish between automation as incremental productivity and Coday as a catalyst for deep operational leverage.

### **4.1 Standardization at Scale: Raising the Floor Across Functions**

One of the most immediate and pervasive effects of Coday at Whoz has been the systematization of core deliverables and communications. Prior to Coday, essential artifacts such as release notes, troubleshooting documentation, and test case definitions often varied not only between teams, but even between individuals within the same department. This inconsistency created friction in review cycles, led to communication breakdowns with clients, and sometimes introduced compliance and quality risks.

With the adoption of deterministic templates, enforced glossaries, and agent-driven quality checks, Coday has materially elevated the organization's baseline output. Notably, release notes now pass through a multi-stage "Game" workflow, beginning with a creative rough draft and progressing through detailed, structurally validated iterations. Interestingly enough, while anchored in standardized formats, this process is fundamentally interactive and non-deterministic. In this process, structured templates provide a foundation, but the iterative exchange between agents and human contributors enables a dynamic, adaptive evolution of content. This collaborative interplay—where AI surfaces suggestions, humans inject context and refinement, and both sides learn from each cycle—ensures that the final output synthesizes the completeness and consistency of automation with the nuance and creativity of expert judgment, resulting in both higher quality and greater adaptability.

## 4.2 Quantified Benefits: Demonstrable Efficiency and Quality Gains

Within software engineering, the influence of tacit knowledge is perhaps most profound—and the returns from its operationalization most measurable. Prior to the deployment of Coday, Whoz's R&D teams, like many in the industry, navigated a "valley of dilution": as team sizes increased or as projects became more sophisticated, individual productivity tended to decrease. Informal, undocumented expertise—crucial for debugging, solution design, and complex system delivery—remained siloed or was lost with attrition, leading to rework and inconsistent standards.

Coday's integration initiated a step change in this paradigm. Through agents that both *preserve* and *deploy* best practices, standardize architecture, facilitate code reviews, suggest optimizations, and actively capture human-machine collaboration, the system has reversed the classic productivity decline associated with organizational growth and technical complexity.

This impact can be quantified using the Linear Dilution Model for productivity decay as teams scale:

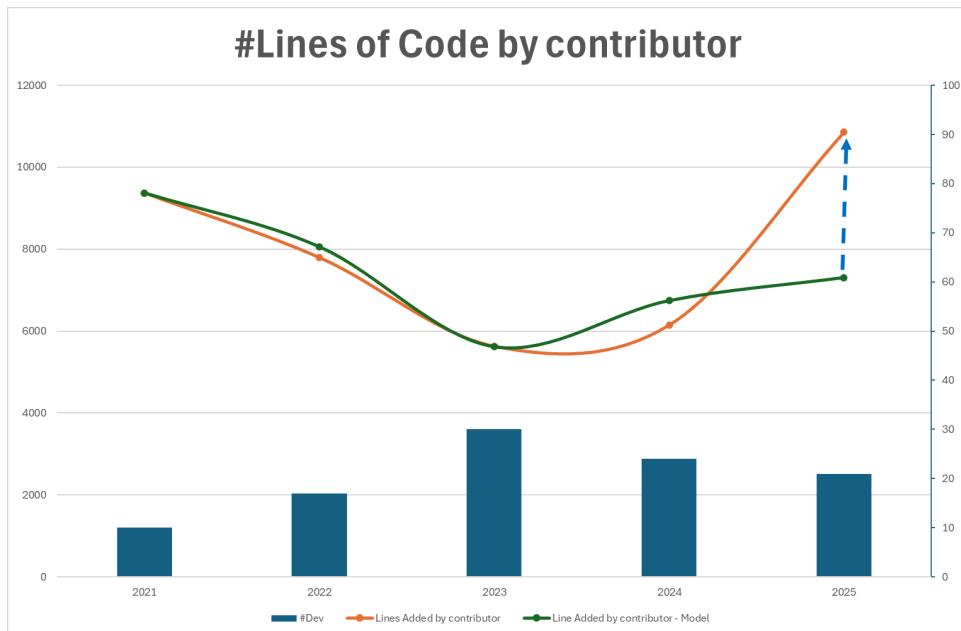
$$P = P_0 \times (1 - k \times (N - N_0) / N_0)$$

where:

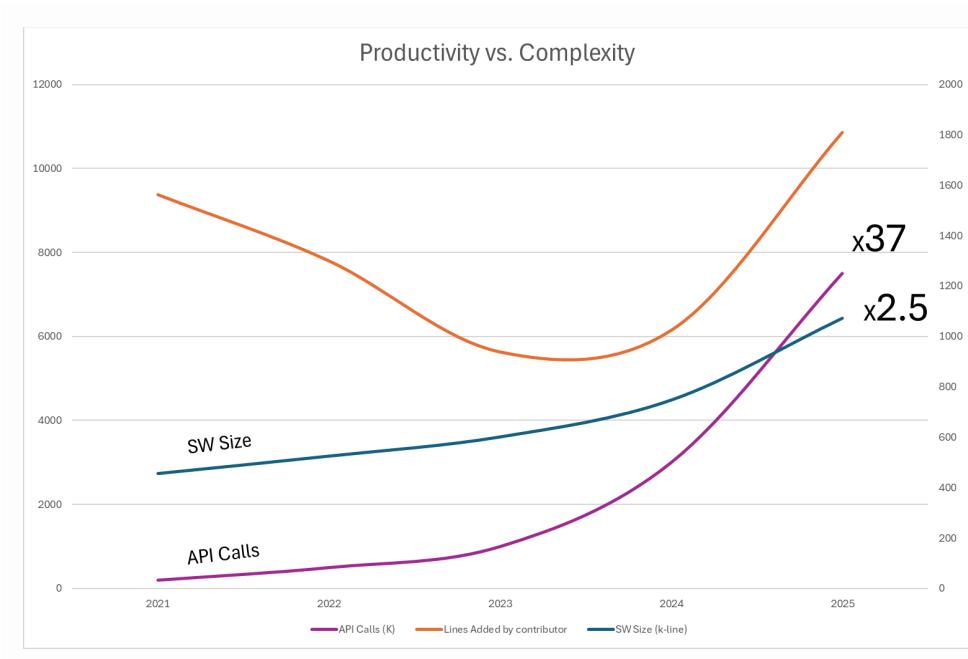
- $P$  is the new productivity per person;
- $P_0$  is the baseline productivity per person;
- $N$  is the new team size;
- $N_0$  is the baseline team size;
- $k$  is a dilution factor, empirically observed in the range 0.15 – 0.25 (with Whoz standardizing at  $k=0.2$ ).

Traditionally, this model predicts a sharp decline in per capita output as engineering teams scale. However, post-Coday, a different trend emerges at Whoz. Integration of agent workflows and codification of tacit knowledge led to a

measured improvement of 49% in global code delivery productivity—even as both team size and the complexity of delivered solutions increased.

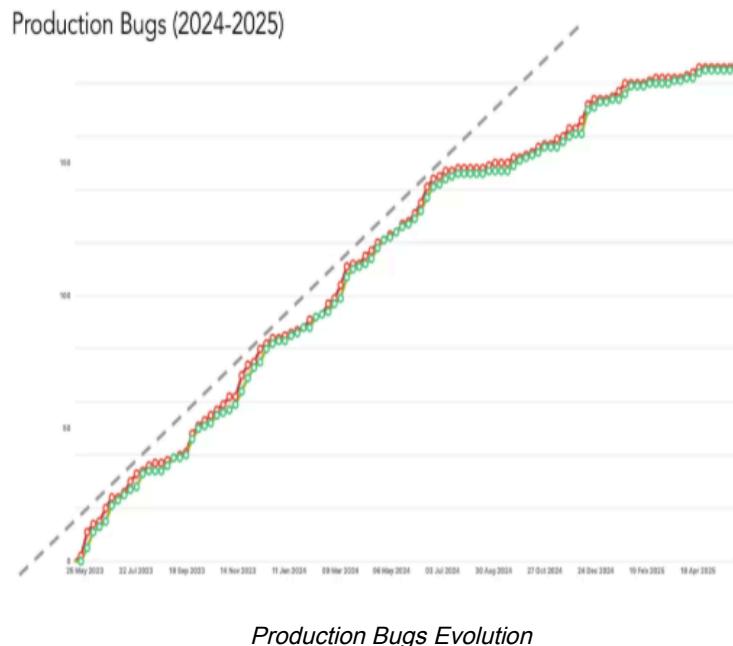


*Output increase by 49% vs. model prediction*



*Solution Complexity Increase*

The empirical experience at Whoz has shown that not only is overall output higher, but that more complex problems are successfully tackled without trade-offs in quality or speed. This defies the typical industry trend, wherein technical complexity and larger group size challenge coordination, dilute responsibility, and raise cognitive load.



Moreover, by closing the gap between what engineers “know but cannot tell” and what the system can suggest, standardize, and automate, Coday has shifted team culture from firefighting and rework to structured innovation and cumulative learning. Architectural decisions, non-obvious debugging techniques, and nuanced approaches to refactoring are retained and propagated, substantially flattening the learning curve for junior staff and reducing bottlenecks that traditionally required attention from senior technical contributors.

In summary, Coday’s deployment in software engineering has converted tacit expertise into a renewable resource and reversed the natural law of productivity dilution: team expansion and problem complexity now correlate with rising—not falling—project velocity and code quality.

And these improvements are not isolated. They stem from a single, systemic shift: the ability to encode organizational wisdom into automation, thereby converting experience and experimentation into ongoing, replicable advantage.

## 4.3 Knowledge Scalability and Organizational Memory

A particularly transformative effect of Coday lies in the area of knowledge retention and dissemination. Where once the departure of a technical lead or the reassignment of a client manager could render weeks or months of accrued expertise inaccessible, Coday agents now ensure that the "institutional brain" of Whoz is not only preserved but actively grows with each new case handled. New hires benefit from immediate access to senior-level solutions, best practices, and historical decisions, all curated and contextualized within the same tools in which they work. This has resulted in a marked reduction in the learning curve for complex product modules and has insulated Whoz from the productivity drags typically associated with internal mobility or turnover.

This organizational benefit is not merely an emergent property of technology, but the result of deliberate cultural and structural initiatives. From its inception, the adoption of Coday has been championed at the highest level, with the CEO and top management providing visible endorsement and impetus. Regular presentations, internal workshops, and dedicated sessions at company events have introduced employees across all departments to Coday's potential, fostering a climate of curiosity and experimentation.

Departmental leaders have played a vital role in identifying practical opportunities to create and use agents tailored to their specific workflows, further embedding Coday into daily routines. Simultaneously, the 3D Factory has been proactive in dedicating resources—not only to support the creation and refinement of agents, but also to improve the overall Coday experience in terms of user interaction, memory fidelity, and interface design.

Organic adoption has also been a key driver: as individuals discovered the practical value of Coday—particularly its contextual memory and deep, embedded understanding of Whoz's organizational structure, technical architecture, and operating procedures—usage spread quickly. Unlike generic copilots or isolated

automation tools, Coday requires users to specify far less; much of the necessary context is already present in the system, allowing tasks to be accomplished without repeated explanations or manual setup. Just as importantly, Coday's natural language interface lowers the barrier to entry: users can engage, request, and achieve results using everyday language, bypassing the more complex requirements of traditional enterprise tools such as Jira.

This ease of use and contextual relevance have made Coday the obvious platform of choice for efficiently getting work done, and they strongly signal an evolutionary shift underway in digital work habits. In the longer run, Whoz anticipates that agent-based interaction will become the primary and most intuitive avenue for productivity, with traditional software platforms receding to background middleware roles—serving the seamless, human-centered experiences users will increasingly expect.

Looking ahead, Whoz is also considering a strategic transition to reinforce this agent-first mindset: progressively restricting direct user access to certain enterprise tools and requiring interactions to take place exclusively through Coday's agents. By combining the powerful "carrot" of superior usability and embedded knowledge with the "stick" of mandated adoption, the organization aims to ensure both the comprehensiveness of knowledge capture and the full realization of a truly agentic, context-aware digital workplace.

Nevertheless, leadership recognizes that true systematic knowledge capture will require ongoing effort. Leveraging local champions, further integrating Coday with emerging workplace tools, and developing more automated connections to existing information repositories remain priorities to ensure that relevant expertise is continuously and comprehensively captured as part of everyday work.

## **4.4 Strengthening Governance, Traceability, and Compliance**

Coday's role as an orchestration layer extends firmly into the domains of governance and risk management. Every automated action, whether modifying code, generating client communications, or producing legal narratives, is permissioned and logged. The use of systems such as the Chaos microservice ensures these operations cannot cross inappropriate boundaries, with role-based

access rigorously enforced. The upshot is accelerated compliance cycles—particularly in sensitive processes such as GDPR and sub-processor documentation—alongside higher confidence for clients and auditors alike that Whoz's communications and practices are both accurate and consistent.

Moreover, Coday's compliance capabilities reach into product quality and user experience through specialized agents such as the a11y agent, which helps ensure that accessibility standards are taken into account—including the latest WCAG 2.2 requirements. By integrating accessibility checks directly into development, design, and release workflows, Coday not only reduces the risk of non-compliance but instrumentally supports Whoz's commitment to building inclusive, regulation-compliant products from the ground up. This thorough, automated scrutiny further reassures clients and regulators, making compliance demonstrable, explainable, and scalable within the organization.

## **4.5 Alignment of Execution with Strategic Imperatives**

Perhaps most notably, Coday's impact is felt not just in operational efficiencies or isolated business victories, but in the enterprise's ability to maintain alignment between its day-to-day execution and its broader strategic vision. The system's memory, , and validation rules ensure that ongoing activity remains tethered to evolving objectives, whether optimizing for international expansion, onboarding enterprise clients, or achieving industry-leading accessibility and performance standards. Coday does not merely support these initiatives; it actively shapes them by surfacing relevant intelligence, codifying best practices, and ensuring that progress is both measurable and visible across all levels of the organization.

## 5. Overcoming Polanyi's Paradox in Software Engineering

### 5.1 Background: The Nature and Relevance of Polanyi's Paradox

Polanyi's Paradox, first articulated by Michael Polanyi in his seminal 1966 work, *The Tacit Dimension*, asserts: "We know more than we can tell." In the context of knowledge work and, specifically, software engineering, this paradox manifests as a persistent gap between what experts do and what they are able to fully describe or document. High performers operate based on intuition built over years of problem-solving; best practices are internalized and rarely codified in accessible formats.

Implications for Software Development:

- **Abundance of Tacit Knowledge:** Expert developers leverage intuition to debug, make architectural decisions, and implement optimizations, often without explicit documentation.
- **Variation by Expertise:** Senior engineers may use tacit shortcuts or pattern recognition that juniors cannot easily access.
- **Informality of Knowledge Transfer:** Most organizational learning happens "in the moment" via oral conversations, code reviews, or chat exchanges, leading to knowledge silos.
- **Productivity and Standardization Risk:** Companies risk dependence on individual contributors and suffer from reduced productivity, slower onboarding, and inconsistent output as a result.

### 5.2 Coday's Approach: Systematic Capture and Institutionalization of Tacit Knowledge

Coday aims to systematically dissolve Polanyi's Paradox within Whoz through four interlocking mechanisms. It strives to make organizational intuition explicit, accessible, and actionable—not just for documentation, but for the automation and augmentation of daily work.

### 5.2.1 Digital Capture of Tacit Interactions

Coday is built to listen, observe, and participate in all relevant developer interactions:

- Every solution to a bug, nuanced workaround, architectural decision, and bit of feedback is logged (with privacy/compliance guardrails).
- Critical contextual metadata such as the nature of the problem, the code involved, human rationale, and outcome are extracted.

This process is mutually reinforcing: developers are not only contributing knowledge for the benefit of the group but also directly reducing their own cognitive burden. Users understand that by entrusting Coday with the nuances of each situation, they are effectively “saving” insights for future retrieval—both for their colleagues and for themselves.

For example, when a senior developer debugs a tricky timezone issue in the casting engine and documents their approach, Coday extracts that pattern and tags it to both the codebase and support knowledge base, making it retrievable for similar future issues. The next time the same developer, or anyone else, encounters a related challenge, Coday can proactively recall the prior solution, offer reminders, or even warn against previously observed pitfalls—transforming real-world experience into continuously accessible organizational memory.

This feedback loop motivates genuine participation: by sharing knowledge, users foster collective intelligence and simultaneously gain an always-available prompt and safety net for their own work in the future.

### 5.2.2 Standardization and Synthesis

Captured knowledge is transformed from idiosyncratic notes or chats into structured, reusable formats:

- **Templates and Frameworks:** Whether for troubleshooting docs, test cases, or release notes, captured best practices are templated, with contextual placeholders for variables.
- **Controlled Glossaries:** Company-specific lexicons enforce consistent use of terminology across code reviews, support documents, and user-facing materials.
- **Memory Indexing:** Insights are metadata-tagged for fast retrieval by project, module, type of issue, or outcome.

As an example, a recurring solution to front-end accessibility color-contrast issues is converted from an ephemeral chat to a validated template, automatically checked in every pull request and available as a decision recommendation in project planning.

### 5.2.3 Human-AI Co-Learning and “Ask for Help” Loops

Coday's architecture recognizes the persistent limits of AI-based reasoning: When faced with ambiguous, novel, or unsolved problems, it actively enlists human expertise:

- **Prompting for Human Input:** Coday escalates questions or requests guidance, learning from developer responses.
- **Virtuous Feedback Loop:** Each human intervention is logged, contextualized, and memorialized, raising Coday's proficiency over time for similar cases.
- **“Reverse Tacitization”:** By structuring queries to humans and synthesizing their responses, Coday forces otherwise implicit knowledge to be articulated, then curated.

During E2E test generation, if Coday cannot interpret a Jira ticket's acceptance criteria, it requests clarification from the author; the subsequent clarification updates both the test case and Coday's template logic for future tickets.

#### 5.2.4 Continuous Codification and Evolution

Coday's memory system is dynamic:

- New "Tacit Insights" are Regularly Validated: User feedback, success/failure outcomes, and evolving standards feed back into Coday's knowledge base.
- Systematic Update of Templates: As organizational practices improve or adapt (e.g., a new translation flow or testing methodology), Coday propagates updated standards across all relevant workflows and agents.
- Agent Self-Evaluation: Performance analytics (cycle times, correction frequencies, adoption metrics) prompt ongoing refinement of memorialized knowledge.

After a series of recurring data synchronization issues are flagged in Performance Center, Coday, in partnership with human leads, updates the canonical troubleshooting template, ensuring future rounds leverage the most effective diagnostic steps.

### 5.3 Practical Impact and Organizational Benefits

#### 5.3.1 Accelerated Learning and Onboarding

Employees gain access to a continuously evolving, up-to-date repository of senior "moves"—making expert thinking immediately available and reducing time to proficiency.

Onboarding for complex modules drops from weeks to days, as new hires find concrete examples and templated solutions directly in their daily tools.

#### 5.3.2 Standardization and Reduction of Cognitive Load

Engineers, support, and compliance all benefit from heightened predictability:

- Fewer duplicated workarounds or undocumented fixes
- Consistent application of company lexicon and communication standards
- Templated workflows that surface only the exceptions, reducing noise

### 5.3.3 Sustainable Institutional Memory

With turnover or role changes, knowledge loss is reduced. Client-specific practices, technical workarounds, and domain-specific reasoning persist in agent-accessible formats.

### 5.3.4 Next-Level Human-AI Collaboration

The relationship between human experts and digital agents is reframed as a symbiotic, iterative partnership:

- AI scales known solutions and asks for help when stumped
- Humans provide judgment only for edge cases or new domains
- Organization's "collective brain" expands as each human-agent interaction is memorialized

## 5.4 Case Study: Coday in Action

### 5.4.1 Debugging Workflow

A recurring bug occurs in the scheduling engine under certain international date formats.

- The first time, a senior developer identifies the root cause using a series of code traversal steps—none of which are documented anywhere.
- Coday observes, logs the sequence (including tool commands, transient insights, and decision rationale), and tags the resolution under "scheduling/date-format-bug".
- Three months later, the issue recurs in a slightly different module. A junior developer encounters it but, prompted by Coday, is served both the sequence (step-by-step investigation) and the solution. The problem is solved in hours, not days.

- The agent logs the successful reuse, tying it to improved mean time to resolution and fewer code review loops, all recorded in Coday's analytics dashboard.

#### 5.4.2 Legal Knowledge and Contractual Risk Mitigation

Coday's agentic memory and feedback mechanisms extend well beyond technical workflows, providing significant organizational leverage in legal and compliance management. Consider, for example, the evolution of contractual know-how. Suppose that during the review of a client contract, the legal team discovers that a critical clause regarding data protection is ambiguously worded or entirely absent—creating unanticipated risk exposure for Whoz. This oversight may become apparent only after a challenging negotiation or, worse, as a result of a near-miss during a regulatory audit.

With Coday in place, the risk does not become a forgotten anecdote. The agent observes the issue, logs the problematic language or missing section along with contextual metadata: contract type, jurisdiction, applicable law, and the sequence of legal reasoning that led to its detection. The system tags the insight appropriately—perhaps, “GDPR/data-protection-clause/missing” or “liability-wording/UK-contracts”—and stores any mitigation steps that were taken, such as recommended language or consultation notes with outside counsel.

Several months later, when a new contract is drafted with a similar client profile or under a comparable legal framework, Coday's legal agent flags the precedent at the outset: surfacing past mistakes, required inclusions, and jurisdiction-specific attention points, all contextualized for current law and regulation. Informed by the most recent legal evolutions and regulatory interpretations, the agent may suggest updated language, further reducing the likelihood of repeated error and ensuring that learnings are systematically incorporated.

This virtuous cycle strengthens over time. Every contract negotiation, resolved dispute, or audit triggers further refinement: Coday curates and surfaces both the “what” and the “why” of contractual best practices. As a result, knowledge once scattered in emails, isolated legal opinions, or individual memory is made

actionable enterprise-wide—directly enhancing risk mitigation, accelerating contract review, and supporting continuous compliance.

At the same time, these legal use cases illustrate again that memory governance remains a central challenge for the agentic organization. For Coday to provide the most value, its memory must be actively curated; ways must be found to flag which clauses, interpretations, or case histories are enduringly relevant and which become obsolete as laws evolve. Just as critical, contractual information often contains sensitive or privileged details. Determining what legal insight or precedent may be safely generalized and shared organization-wide, versus what must remain compartmentalized by client, matter, or jurisdiction, is essential to avoid inadvertent confidentiality breaches.

Thus, the promise of agentic knowledge in legal workflows will depend not only on technical sophistication but also on the careful development of memory taxonomies, visibility controls, and curation protocols. These governance safeguards will be key to harnessing the full benefits of collective learning while upholding the rigorous standards of discretion and ethics demanded by the legal domain.

## 5.5 Limitations and Critical Reflection

While Coday represents a breakthrough in the codification and operationalization of practical knowledge, certain limitations and enduring challenges remain:

- Some dimensions of human intuition, creativity, or highly contextual judgment are likely to remain irreducible to templates, frameworks, or agent-driven synthesis.
- True organizational benefit demands not just technological adoption but a genuine cultural transformation; shifting to an agent-first mindset challenges longstanding habits and assumptions, requiring not only user adoption but also critical literacy—ensuring that teams remain thoughtful and discerning about agent outputs, rather than passive consumers.
- The management of memory itself—balancing collective intelligence with confidentiality, curating relevant information while discarding the obsolete, and carefully defining access boundaries—is perhaps the most complex

challenge of all, with wide-reaching implications for trust, compliance, and organizational adaptability.

- Finally, systematic feedback mechanisms and processes for selective memory curation are essential to prevent knowledge sprawl or drift into irrelevance, ensuring that what is retained truly elevates organizational capability rather than diluting it.

Ultimately, Coday's promise will be realized only to the extent that the organization develops not just the technical infrastructure, but also the cultural and governance foundations required for agentic work at scale.

## 5.6 Strategic Implication: The Enterprise Beyond Polanyi

Coday's continuous dissolution of Polanyi's Paradox is not only a technical achievement—it signals the emergence of an organization in which:

- The marginal cost of disseminating best practices approaches zero
- Organizational learning compounds at the speed of feedback cycles, not personnel turnover
- The business achieves unprecedented alignment, adaptability, and resilience

By operationalizing tacit knowledge and institutionalizing it via agent-driven workflows, Coday transforms Polanyi's insight from a barrier into a springboard for the next era of collective intelligence. This positions Whoz not just as a product innovator, but as a pioneer in the architecture of the intelligent organization.

---

In summary:

The transformative promise of Coday rests not only in its capacity to operationalize tacit organizational knowledge and continuously dissolve Polanyi's Paradox, but also in its ability to stand at the intersection of two rapidly compounding technological forces. On the one hand, advancements in large language models—driven by external pioneers such as OpenAI, Mistral, and

others—are poised to accelerate exponentially, bringing ever greater capabilities to bear without requiring Whoz to build foundational AI infrastructure in-house. On the other hand, Whoz's concentrated efforts on the long-term memory challenge—making organizational learning persistent, curated, and context-aware—ensure that each new leap in model power is captured, institutionalized, and applied for genuine field improvement.

By systematically combining access to frontier AI models with a robust, evolving knowledge memory, Whoz is positioned not only to ride but to magnify each wave of technological advancement. The result is a new form of exponential improvement: every gain in the broader AI ecosystem is immediately leveraged and made durable within Whoz's agentic operating fabric, accelerating the pace of organizational intelligence, adaptability, and collective performance over time.

## 6. Limitations and Mitigation Strategies

As with any transformative enterprise AI system, Coday's rollout revealed risks and operational challenges. It is important to recognize that Whoz remains at the early stages of this journey; the organization is learning in real time, continually uncovering both opportunities and limitations by putting Coday into practice. Each step brings new revelations—whether around usage patterns, gaps in impact measurement, or the evolving needs of a growing business. Many issues only become fully visible through hands-on adoption, and the process of surfacing, understanding, and addressing these challenges is ongoing.

Nevertheless, Whoz is committed to remaining at the forefront of the agentic enterprise market. Our approach balances technical solutions, organizational adaptation, disciplined measurement, and a willingness to iterate rapidly. The considerations that follow represent not only current observations but also a prospective agenda—food for thought and a living roadmap that will guide the continued evolution of Coday. By tackling these challenges head-on, we aim to ensure that Coday becomes ever more robust, impactful, and perfectly attuned to the needs and ambitions of Whoz.

### 6.1 Hallucination and Context Drift

Challenge:

Large Language Model (LLM)-based agents can produce answers that are plausible but not grounded in fact (hallucinations), especially when working with ambiguous inputs. As agent context windows grow (i.e., memory of previous interactions and documents), there is risk of context drift: moving away from single-source-of-truth answers or introducing subtle inconsistencies in outputs.

Symptoms:

- Inconsistent release notes or support responses due to outdated memory
- Tool misuse due to shifting context (e.g., making incorrect API calls)

Mitigation Strategies:

- **Tool-First Design:** Coday agents leverage structured data (e.g., Jira tickets, Confluence docs, source code) whenever possible, using LLMs primarily for orchestration, reasoning, and templated language generation.
- **System-of-Record Grounding:** Decisions reference and cite artifacts from the system of record, creating observable traceability (e.g., linking a bug fix to the relevant Git commit or Jira ticket).
- **Deterministic Templates and Glossaries:** Outputs are shaped by immutable formats (e.g., release note templates, troubleshooting HTML), limiting free-form text generation and enforcing terminological consistency through company glossaries.
- **Human-in-the-Loop Gatekeeping:** Critical outputs (e.g., major release notes, client-facing materials) require human validation before execution/deployment.

Example:

When producing a release note, Coday's agent retrieves the actual ticket data and fills out a locked template. Only specific fields (summaries, KPIs) use generative language, with links back to Jira tickets to verify content accuracy.

## 6.2 Integration and Permissions Complexity

Challenge:

As Coday scales across teams and tools, role-based access control (RBAC) and secure integration become crucial. Agents must not “overreach” their access nor expose sensitive information across boundaries (e.g., between clients, departments, or sensitive modules).

Symptoms:

- Erroneous publication of support response
- Unintended disclosure of restricted data
- Workflow interruptions due to permission errors

#### Mitigation Strategies:

- **Workspace-Aware Search and Deep-Link Guards:** Search results and links rendered by agents are strictly filtered based on current user/agent workspace and RBAC settings.
- **Centralized Permission Checks:** The Chaos microservice enforces permissions at navigation and action points, preventing unauthorized access or operations.
- **Explicit Documentation:** Intended cross-workspace behaviors are clearly described and documented, with corrective flows for errors.
- **Audit Logging:** Every sensitive agent action (e.g., cross-boundary data generation) is logged and reviewable for compliance and forensic analysis.

To further illustrate this point, if a support agent tries to reference a Jira issue in another client's workspace, deep-link guards block the action. Chaos microservice logs the attempt, and the agent is notified of access limits.

## 6.3 Adoption and Change Management

#### Challenge:

AI and automation can face cultural resistance—employees may distrust “black box” systems or fear loss of control/ownership over processes.

#### Symptoms:

- Inconsistent usage/adoption between teams
- Manual workarounds or bypassing agents
- Feedback of “opaque” or “inflexible” agent outputs

#### Mitigation Strategies:

- **Transparent Outputs:** Today agents clearly explain “why” and “how” they generate outputs, citing tools, source documents, and reasoning steps; sensitive decisions are linked back to primary data.

- **Embedded Standards and Quick Wins:** Well-crafted templates, clear structure, and department champions help drive early adoption and demonstrate quick benefits (e.g., faster documentation cycles).
- **Department Champions and Training:** Cross-team “superusers” champion best practices; lightweight onboarding/trainings encourage trust.
- **Feedback Loops:** Collect and act upon user feedback, surfacing discomfort or requests for change.

Example:

A Client Success Manager can preview step-by-step how a CxO level communication is built—seeing agent prompts, selected templates, and linked Jira records—building confidence in automation reliability.

## 6.4 ROI and Impact Evaluation

Challenge:

Proving the business value and return on investment (ROI) of automation goes beyond anecdotes; consistent measurement is needed to justify investment and guide priorities. This challenge is compounded by the fact that Coday's time-saving impact often enables work that previously could not have been attempted at all. As automation frees up capacity, teams are able to pursue new projects, explore innovations, or address long-neglected initiatives—creating qualitative gains and incremental value that are difficult to quantify and even harder to attribute explicitly to Coday. Thus, measuring ROI must account not only for time and cost savings, but also for the new opportunities and growth unlocked, raising the bar for what constitutes rigorous, meaningful impact assessment.

Symptoms:

- Lack of visibility into which agents produce the most value
- Difficulty aligning automation metrics to business KPIs (e.g., client satisfaction, revenue, quality)
- The intangible and distributed nature of new opportunities makes it hard to determine what additional value or innovation is directly attributable to

Coday, versus what might have emerged organically or through other concurrent initiatives

#### Mitigation Strategies:

- **Baseline Measurement:** All core workflows are baselined for current cycle time, error rates, and user satisfaction before agentification, providing an anchor for measuring both direct efficiency and downstream impact.
- **Process-Specific Success Metrics:** Each agent workflow is tied not only to traditional quantifiable KPIs (e.g., document review cycles, support time-to-first-response, test coverage delta), but is also monitored for the emergence of ancillary activities such as new projects launched, products enhanced, or cross-departmental collaborations enabled as a result of freed capacity.
- **Agent Analytics Dashboards:** Usage, quality, and adoption data is graphed in real time; process-specific before-and-after metrics are tracked alongside survey instruments or qualitative reporting designed to capture new initiatives or value streams originating from Coday-enabled bandwidth.
- **Continuous Improvement and Attribution Analysis:** Insights from analytics drive prioritization, agent refinement, and roadmap adjustments, but are supplemented by regular strategic reviews—where teams consciously reflect on what new capacities and successes may be linked to automation, even absent perfect attribution. Over time, the organization aspires to develop more sophisticated models and narratives for recognizing and scaling opportunity-driven impact.

#### Example:

After automating the Release Note Game, draft-to-approval cycle time is reduced by 30%, and the average number of review loops drops from four to two, as measured over three consecutive releases.

## 6.5 Cost and Performance Optimization

#### Challenge:

Inference costs for LLM-based agents and latency from tool integrations can throttle scalability, notably as agent catalog and workflow complexity expand.

Symptoms:

- Delayed output in high-complexity workflows
- Cloud compute or API costs exceeding forecasts

Mitigation Strategies:

- **Artifact Caching:** Frequently used data (e.g., translation glossaries, release note templates) are cached across sessions and only recalculated on update.
- **Batch Operations:** Heavy tasks (e.g., mass documentation generation or test runs) are grouped into batch jobs, reducing calls and waiting time.
- **Tiered Agent Execution:** "Light" agents serve high-frequency, low-computation tasks, while "heavy" agents (requiring external system calls or intensive computation) are invoked only as needed.
- **Monitoring Dashboards:** Real-time and retrospective tracking of agent cost and latency guides optimization efforts.

Example:

Large-scale E2E test generation is run overnight in batch mode, with results cached and distributed to all relevant teams the next business day, balancing compute cost and time-to-value.

## 7. The Road Ahead: Vision and Roadmap

Coday's roadmap is designed for continuous expansion, targeting ever-stronger integration, governance, productivity, and measurement. The next 12 months focus on deepening impact in four pillars: analytics, security, automation, and business enablement.

### 7.1 Memory Segmentation

Initiatives:

- **Granular Memory Layers:** Develop explicit segmentation of Coday's memory architecture at the user, team, department, and enterprise levels, allowing tailored access, contribution, and retrieval of knowledge according to context and sensitivity.
- **Contextual Sharing Protocols:** Design dynamic protocols for when and how memory fragments can be promoted from an individual's or team's context into the collective organization—balancing the power of shared learning with the imperative of confidentiality, especially for HR, legal, and client-sensitive domains.
- **Governance and Auditing:** Implement fine-grained permissioning, audit trails, and memory curation workflows, giving data owners and compliance leaders confidence in both the accessibility and the protection of information across the organization.

Example:

A legal advisor identifies a new risk in a client contract. The insight is initially stored in a private, client-specific memory segment. As similar risks appear over time, a compliance officer—using Coday's governance controls—flags this pattern for broader dissemination, promoting the anonymized learning to a company-wide legal guidelines library, while still preventing exposure of any client-identifying details.

## 7.2 Expanded Retrieval-Augmented Generation (RAG)

Initiatives:

- **Deeper Internal Integrations:** Expand RAG to ingest and index a wider set of organizational documents—Confluence wikis, Slack transcripts, product meeting notes, structured Airtable datasets.
- **Dynamic Document Context:** Agents will dynamically retrieve and reason over the most recent internal intelligence when generating answers or recommendations.
- **Workspace/Role Filtering:** RAG systems obey strict workspace and role boundaries, ensuring knowledge is surfaced only where authorized.

Example:

A PM requests a market impact analysis for a new feature: Coday's agents pull from recent UX research, meeting transcripts, and sales commentary, surfacing a cross-functional summary in minutes, filtered based on the PM's permissions.

## 7.3 Security and Link Governance

Initiatives:

- **Stronger Link Controls:** All inter-system links are protected via tokenization, expiration times, and Multi-Factor Authentication (MFA) for especially sensitive contexts.
- **Advanced Audit Trails:** Every access is logged with user/session/intent metadata, supporting forensics and compliance audits.
- **Role and Data Boundary Enforcement:** Tighten RBAC model to every agent, ensuring no cross-leakage as the agent catalog grows.

Example:

A troubleshooting doc with embedded links to client configuration is only accessible for three days and requires secondary verification for non-owners, with all accesses fully logged.

## 7.4 Test and Documentation Automation

Initiatives:

- Expanded E2E Test Automation: More code modules are subject to automated test generation, execution, and regression analysis.
- Storybook/Documentation Expansion: Automated agents expand beyond code, auto-generating Storybook UI documentation and developer reference summaries.
- Reinforced API/Accessibility Checks: More frequent, automated accessibility (WCAG) and API contract validation at scale; failures are flagged, and remedial tickets pre-drafted.

Example:

The entire Angular component library is automatically tested for accessibility compliance each sprint, with violating color pairings flagged and corrective PRs initiated by agents.

## 7.5 Agent Analytics and Expansion

Initiatives:

- **Agent Usage Dashboards:** Rolling out comprehensive dashboards that visualize agent usage rates, workflow speeds, error rates, and quality/adoption metrics. Enables leaders to quickly identify high-impact and underutilized agents.
- **Impact Metrics:** For each workflow, tie analytics to business outcomes (e.g., support resolution times, code review defect rates, sales plan generation speed).
- **Before/After Studies:** Launch A/B testing and controlled process baselining to rigorously validate agent impact, feeding back into continuous improvement.

Example:

Support leadership can compare time-to-first-meaningful-response pre- and post-automation, while QA can see the percentage increase in E2E coverage after introducing automated testing agents.

## 7.6 Sales and Success Enablement

Initiatives:

- **Automated Value Realization Tracking:** Sales and Customer Success agents will generate before/after snapshots, impact maps, and ROI summaries ahead of renewals and client reviews.
- **Auto-Preparation for QBRs:** Agents will pre-fill Quarterly Business Review decks with matched KPIs, client outcomes, and relevant documentation assets.
- **Client Impact Dashboards:** External-facing dashboards summarize adoption metrics and realized client value, powered by Coday-generated data.

Example:

Before a customer renewal, the account manager receives a Coday summary with improved staffing times, unstaffed rates, ticket deflection rates, and new feature adoption curve, ready to present to the client.

## 8. Measuring Success: Recommendations and Prospective Metrics

Translating the promise of an agentic operating system like Coday into lasting organizational advantage hinges upon the ability to track, analyze, and act on meaningful metrics. While Whoz has already established foundational practices for assessing time savings, standardization, and task-level cycle times, a more systematic and comprehensive measurement discipline remains aspirational and is a critical part of the ongoing development roadmap.

### 8.1 Evolving Measurement Practices Across Core Functions

To date, initial impacts of Coday have been observed in areas such as faster support response, reduced review cycles in documentation, and improvements in defect rates and code quality. For example, anecdotal evidence and episodic analysis show shortened time-to-first-meaningful-response in support, as well as increased reuse of troubleshooting content. Similarly, code review logs and QA outcomes point towards improved standardization and a reduction in production defects.

However, the ambition moving forward is to embed continuous measurement and analytics directly into every key function. Prospective metrics include:

- **In support:** systematic tracking of time-to-resolution, escalation rates, and self-service deflection, connected to agent intervention.
- **In engineering:** benchmarking of agent-driven test coverage and automated assessment of defect leakage compared with pre-agent baselines.
- **In sales and customer success:** measuring cycle times for account planning and renewal deliverables, and the ratio of agent-generated versus manually crafted content.
- **For documentation and communications:** calculating average draft-to-approval times, localization cycle times, and the number of required review loops before final sign-off.

- **At an enterprise-wide level:** monitoring overall agent adoption rates across departments and gathering structured user satisfaction scores to assess cultural integration.

Currently, some of these measurements are captured in ad hoc reports or emerging pilot dashboards, but their integration into everyday operational analytics represents an ongoing ambition for Whoz as Coday matures.

## 8.2 Continuous Feedback and Adaptive Measurement

Recognizing the fluidity of organizational priorities and digital transformation objectives, Whoz has aligned its measurement roadmap with an iterative, feedback-driven philosophy. Regularly scheduled user interviews, department champion sessions, and experimental analytics dashboards are being developed to close the loop between measurable outcomes and operational adjustments. The intention is for these insights not only to highlight progress, but to illuminate where agent workflows require refinement or expanded scope.

For instance, early feedback loops in engineering have led to refinements in how code review agents present suggestions and flagged the need for stronger alignment between agent outputs and evolving architectural guidelines. Over time, as data aggregates and analytics systems mature, management will be equipped to draw more robust, longitudinal conclusions regarding Coday's transformative impact.

## 8.3 The Strategic Role of Prospective Metrics in Coday's Evolution

In this context, the measurement framework for Coday is both a retrospective tool and a forward-looking enabler. It serves as a mechanism for transparent accountability—demonstrating realized benefits where possible—and as a guidepost for identifying new opportunities, targeting operational pain points, and adapting to changing organizational demands.

In summary, while the groundwork for robust performance measurement has been laid and early returns are promising, Whoz acknowledges that the full realization of analytics-driven, outcome-based management for agentic workflows is an ongoing, iterative journey. As further adoption and technical integration proceed, measurement will move from pilot and ad hoc initiatives to institutional practice—ensuring that future success is not only claimed, but quantifiably demonstrated.

## Conclusion

Coday represents a pivotal evolution from digital assistant to enterprise operating system—a connective, memory-driven fabric that orchestrates not just tasks, but the very way Whoz learns, adapts, and scales. By codifying best practices, surfacing otherwise tacit expertise, and institutionalizing agent-based ownership of core workflows, Coday addresses the complex realities of Polanyi's Paradox and positions Whoz as an early exemplar of the agentic organization.

Central to this transformation is the sophistication of Coday's long-term, selectively curated memory—one governed not only by technology, but by embedded protocols balancing the benefits of collective intelligence with the demands of ethical, auditable information handling. The ongoing work on memory segmentation and governance speaks both to the challenges and the promise of distributing institutional know-how safely and effectively across the enterprise.

Yet, Coday's potential is enabled—and bounded—by culture as much as code. True leverage will rely on sustained leadership stewardship, department-level champions, and a whole-organization willingness to embrace agent-forced workflows, transparency, and an ethic of constructive critical engagement with digital teammates. In Whoz's vision, the blend of "carrot and stick"—from natural language ease-of-use to exclusive, agent-mediated access—will further embed Coday's role as the backbone of day-to-day work.

Anchored in this culture, Whoz now stands at the threshold of an era where the marginal cost of sharing proven solutions nears zero. As Whoz leverages both the exponential advances in foundational AI models provided by industry leaders and its own institutionally retained knowledge, the organization is positioned to compound improvements in organizational intelligence, adaptability, and resilience. With each step, Coday evolves not simply as a technical platform, but as the orchestrator of Whoz's collective growth—a foundation for an enterprise where learning accelerates, silos disappear, and every user benefits from the ongoing synthesis of human and machine intelligence.

---

# Appendix - Examples of Coday Agents

Here's the full list of available agents as of October 6th 2025, organized by high-level categories (as generated by Coday):

## Core Orchestration

- Coday — Default orchestrator that routes requests to the right specialist agent and provides general assistance.

## Accessibility

- a11y — Web accessibility expert ensuring designs and implementations comply with WCAG and best practices.

## API and Data Contract Validation

- APIValidator — Validates API implementations against Stoplight/OpenAPI specs to detect schema, field, and endpoint inconsistencies.

## Documentation and Content

- DocWriter — Produces clear, structured documentation in the required format (HTML/Markdown) from provided inputs.
- ClientRedactor — Crafts personalized, executive-ready client communications based on release notes and RT analyses.
- Keynote — Builds compelling product release keynotes and narratives for internal/external presentations.
- SalesSummarizer — Transforms meeting notes into concise, CRM-ready summaries with structured insights.
- SocialReposter — Generates tailored LinkedIn repost content aligned with persona, focus, and content relevance.

## Release Notes and Client Communications

- releasenote — Runs the multi-stage "Release Note Game" to produce complete, engaging, and technically accurate release notes.
- RiRi — Analyzes release notes and RT tickets to produce client-specific impact analyses and summaries.

## Jira/Ticketing and Field Feedback

- Jiray — Internal JIRA ticketing and resolution workflow assistant for efficient issue handling.
- BugReporter — Drafts precise, reproducible bug tickets with clear impact, classification, and expected behavior.

- JiJi — Customer Success ticket analysis agent providing root cause, prioritization, and resolution guidance.
- Jirep — Produces professional, client-facing responses to JIRA tickets based on JiJi's analysis.
- WhozRtFinder — Mines RT (Remontées Terrain) tickets to detect trends, patterns, and actionable product insights.

### **Sales and Account Strategy**

- AccountPlanCoordinator — Orchestrates sub-agents to deliver cohesive, end-to-end account plans.
- ApproachStrategyPlanner — Designs approach strategies and plays for winning and expanding accounts.
- CompanyProfileAnalyst — Researches and analyzes company profiles to inform account strategy and planning.
- SalesAccountPlanMaker — Builds structured, data-driven account plans from prospect analysis.
- NeedsAnalysisSpecialist — Elicits and formalizes customer needs to align solutions and value.
- ObjectivesKPISpecialist — Defines measurable objectives and KPIs aligned to commercial strategy.
- StakeholderMapExpert — Maps decision makers and influencers to guide engagement strategies.
- ValuePropositionExpert — Crafts tailored value propositions that connect Whoz capabilities to client outcomes.
- SalesColdCaller — Assists with cold call scripts, targeting, and cadence for Tier-2 prospects.
- ArnaudAmirault — Sales persona agent for human-style sales interactions, messaging, and role-play scenarios.

### **Customer Success and Support**

- CustomerSupportExpert — Provides comprehensive platform support, troubleshooting, and user enablement.
- ICEMAP — Guides Sales and Success through the ICE + MAP process to prove and realize client business impact.

### **Product Strategy and Design**

- WhozDesigner — Explores product problems, generates solutions, and converts ideas into actionable user stories.
- PMChallenger — Helps PMs break down complex problems to maximize user value while minimizing cost.

- SolutionAgent — Pre-sales partner for solution shaping, scoping, and feasibility assessments.

## **Engineering, Development, and QA**

- Dev — Autonomous development agent able to implement changes with human oversight on critical decisions.
- Howzi — Reviews Angular code against Whoz frontend best practices to provide actionable improvements.
- Storybook — Creates and maintains Storybook stories with robust component analysis and JSDoc.
- E2E — Designs and implements end-to-end tests from JIRA tickets following codebase patterns.
- TestCaseRedactor — Produces comprehensive non-regression test cases with clear steps and expected results.
- UserStoryKiller — Reads JIRA stories, asks clarifying questions, and builds global implementation plans.
- us-redactor — Writes concise, standard-compliant JIRA user stories ready for backlog integration.
- KB43D — Knowledge base assistant that retrieves, correlates, and synthesizes technical/product information.
- Webot — Browser-enabled agent to execute complex web tasks and validations (while respecting auth constraints).

## **UX Research and Usability**

- Digeo — Usability assistant (French only) analyzing user tests and interviews to extract actionable insights.
- WhozUserResearcher — Performs UX data analysis and synthesis to inform product decisions and roadmaps.

## **Legal and Contracts**

- MSA — Reviews and negotiates MSAs and Order Forms, ensuring risk control and contractual clarity.

## **Finance**

- Financial — Handles financial questions, calculations, and scenario analyses supporting business decisions.

## **Recruiting**

- Recruiter — Verifies candidate fit against job requirements and provides structured assessment.