



Jana Publication & Research

Comparative Appropriateness of Agentic vs. Non-Agentic AI Systems: Task-Fit Analysis Framework

 VRC27

 BioTech

 Metropolitan College

Document Details

Submission ID

trn:oid::1:3437808135

Submission Date

Dec 8, 2025, 4:25 PM GMT+7

Download Date

Dec 9, 2025, 3:49 PM GMT+7

File Name

IJAR-55128.pdf

File Size

345.3 KB

10 Pages

4,642 Words

27,440 Characters





2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- ▶ Bibliography
- ▶ Quoted Text





Match Groups

-  **4 Not Cited or Quoted** 1%
Matches with neither in-text citation nor quotation marks
-  **2 Missing Quotations** 1%
Matches that are still very similar to source material
-  **0 Missing Citation** 0%
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted** 0%
Matches with in-text citation present, but no quotation marks




Top Sources

- 1%  Internet sources
- 1%  Publications
- 1%  Submitted works (Student Papers)

Match Groups

-  **4 Not Cited or Quoted** 1%
Matches with neither in-text citation nor quotation marks
-  **2 Missing Quotations** 1%
Matches that are still very similar to source material
-  **0 Missing Citation** 0%
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted** 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 1%  Internet sources
- 1%  Publications
- 1%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

- 1** **Internet**
arxiv.org 1%
- 2** **Publication**
Ume Nisa, Muhammad Shirazi, Mohamed Ali Saip, Muhammad Syafiq Mohd Pozi. ... <1%

1 Comparative Appropriateness of Agentic vs. Non-Agentic 2 AI Systems: Task-Fit Analysis Framework

3 **Abstract**

4 As artificial intelligence evolves at an unprecedented speed, practitioners must navigate an increasingly
5 diverse set of AI system architecture options, with the most prominent decision now lying between
6 established non-agentic designs and newly emerging agentic frameworks. Yet, clear guidance on how to
7 choose between these paradigms remains limited, leaving teams to make high-stakes architectural
8 decisions without relying on well-defined criteria. This uncertainty is further amplified by mounting
9 pressure to adopt agentic AI, even when its suitability and implications for a given task are unclear. This
10 paper addresses these challenges by introducing a Task-Fit Analysis Framework that systematically
11 contrasts agentic and non-agentic AI systems, providing practitioners with evidence-based criteria for
12 assessing which architecture is most appropriate for their use case. It evaluates these system types across
13 five dimensions: autonomy and reasoning requirements, tolerance for non-determinism, ecosystem
14 readiness, scalability considerations, and security implications. Drawing on recent research and analyses,
15 the framework clarifies the trade-offs inherent to each paradigm, enabling more disciplined and well-
16 informed AI system design decisions.

17
18 **Keywords:** Agentic AI; LLM-based Agents; AI System Design; Artificial Intelligence; Deterministic
19 Automation

20 **1 Introduction**

21
22 Artificial intelligence continues to advance at an unprecedented pace, offering increasingly powerful
23 solutions across a wide range of problems. As new architectures and capabilities emerge, practitioners
24 face a growing variety of implementation patterns and escalating pressure to adopt the latest AI
25 paradigms. However, this urgency to use the newest approach often comes without clear evidence that it
26 is the best fit for the task at hand and can be effectively integrated within existing systems. As a result, it
27 creates uncertainty for organizations trying to balance innovation with reliability, safety, and operational
28 efficiency, and highlights the need for stronger guidance to support informed AI system design decisions.

29
30 Today, the most prominent example of this challenge is the choice between emergent agentic and
31 established non-agentic AI systems. Undoubtedly, **agentic AI represents a transformative shift in artificial**
32 **intelligence.** Characterized by autonomy, adaptive decision-making, **and the ability to perform complex**
33 **multi-step** tasks, it has already begun to demonstrate its utility in early applications. In healthcare, agentic
34 LLM-based clinical assistants can collect symptoms, retrieve relevant medical evidence from external
35 knowledge bases, and analyze and synthesize them into patient-tailored recommendations [1]. In finance,
36 multi-agent trading frameworks can collaboratively assess market sentiment, analyze financial reports,
37 create forecasts, and make final trading decisions by intelligently integrating all these signals [2].
38 Together, these use cases demonstrate the potential of agentic AI architectures to integrate heterogeneous

39 information sources, coordinate multiple computational steps, and operate independently within complex
 40 ecosystems on reasoning-heavy tasks. At the same time, one should remember that the autonomy that
 41 gives agentic systems their power also introduces high levels of complexity, integration burden, and risk.

42
 43 In contrast, traditional non-agentic AI systems operate primarily on predefined workflows and follow
 44 preset rules. Although they lack the flexibility to adapt to novel situations or coordinate multi-step
 45 reasoning that agentic AI offers, they remain highly effective and even preferred when tasks have a stable,
 46 well-defined structure. This is evident in their widespread use across domains ranging from retrieval-
 47 augmented educational tools [3] to industrial process automation [4]. Their value lies not only in reliability
 48 but also in their transparency and comparatively low integration overhead, which make them easier to
 49 deploy and govern at scale.

50
 51 Given these diverging strengths and weaknesses of the two system designs, practitioners need a
 52 systematic way to assess when agentic AI is optimal for a given task and when a simpler non-agentic
 53 solution is more appropriate. The goal of this study is, therefore, to provide practical, evidence-based
 54 guidance for making this determination. To do so, we adopt a comparative analysis approach, as it
 55 enables a clear, side-by-side evaluation of how the two paradigms differ in their capabilities, constraints,
 56 and operational requirements. By systematically evaluating agentic and non-agentic systems across five
 57 key dimensions, the analysis aims to clarify the trade-offs involved and offer a structured task-fit analysis
 58 framework to support more informed decision-making in modern AI system design. As agentic AI
 59 remains comparatively new and underexplored, the study focuses more on its characteristics, with non-
 60 agentic systems serving as a structured point of comparison.

61 **2 Agentic vs Non-Agentic AI System Overview**

62
 63 Before introducing the task-fit analysis framework, it is useful to first outline the fundamental differences
 64 between agentic and non-agentic AI systems in more detail. Nisa et al. [5] offer a concise but effective
 65 comparison, highlighting that classical, non-agentic AI relies on predefined, rule-based behaviors, static
 66 input-response patterns, and fixed procedures suited to narrow, well-structured tasks. Agentic AI, by
 67 contrast, exhibits dynamic, self-directed behavior, actively interprets contextual inputs, and adapts to
 68 open-ended scenarios. It supports more complex reasoning, flexible planning, and strategic problem-
 69 solving as goals and conditions evolve.

70

Criteria	Non-Agentic AI system	Agentic AI system	Key difference
Autonomy	Predefined, often linear behavior	Self-directed, adaptive behavior	Level of independence in selecting actions
Reasoning complexity	Primarily shallow reasoning	Multi-step reasoning and deliberation	Capacity for layered iterative reasoning
Predictability	Deterministic, enabling strict control over the process	Less predictable in intermediate steps and outputs	Degree of behavior variability and ease of control

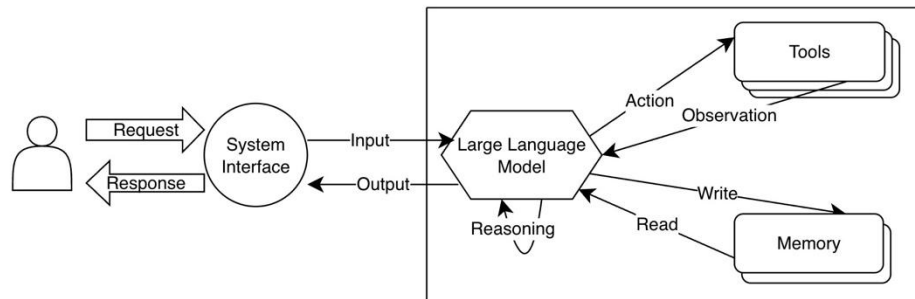
Tool use	Limited, predefined tool invocation	Range of tools and selective usage	Level of flexibility and tool variety
Memory	No persistent memory, isolated input processing	Persistent state, cross-step information retrieval	Breadth and continuity of information retention

71 **Table 1:** Summary of Agentic vs Non-Agentic Systems Differences (adapted from Nisa et al.[5])

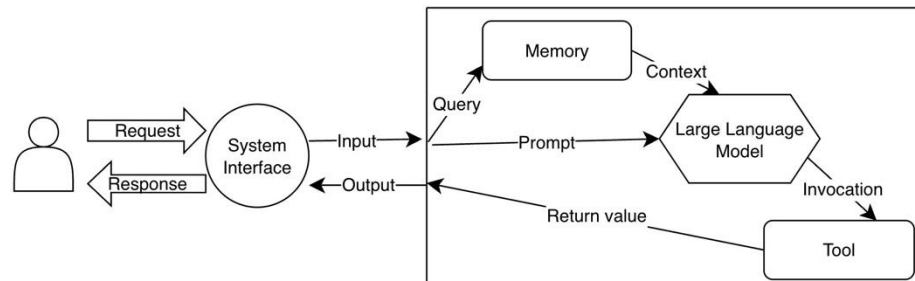
72
 73 In addition, Schneider’s survey on the transition from generative to agentic AI [6] emphasizes that agentic
 74 systems add rich interactions with tools and explicit memory mechanisms as core capabilities. Tools in
 75 this context refer to external functions (such as APIs, databases, or specialized software) that an agent can
 76 invoke to perform operations beyond internal reasoning. When it comes to memory, agentic systems
 77 usually store and retrieve information across steps, allowing them to maintain goals, track intermediate
 78 results, and update their internal state as tasks unfold. This includes short-term task memory as well as
 79 longer-term memory that supports continuity across interactions. In contrast, non-agentic systems
 80 typically have limited tool use and lack persistent memory. They rely on predefined pipelines for any
 81 external operations and rarely maintain state across steps, treating each input as an isolated event.

82
 83 Taken together, these distinctions, summarized in Table 1 and Figure 1, underpin the assessment
 84 dimensions of the task-fit analysis framework that we discuss in detail next.

Agentic AI



Non-Agentic AI



85

86

Figure 1: Agentic vs Non-Agentic AI System Design

87 3 Task-Fit Analysis Framework

88 **3.1 Assessment Dimension 1: Level of Autonomy and Reasoning Required**

89 The first step in choosing the best system for a use case is determining whether the task itself requires the
90 autonomy and reasoning depth characteristic of agentic systems or whether it can be addressed more
91 effectively with a tightly structured, non-agentic approach. Tasks that involve multiple interdependent
92 steps, evolving inputs, or the need to engage in complex reasoning naturally align with agentic
93 capabilities. For example, diagnosing operational issues in a distributed IT system may require collecting
94 signals from various logs, forming hypotheses, testing them through tool calls, and revising the plan
95 based on the acquired feedback. This is an inherently multi-step, adaptive workflow. Similarly, resolving
96 a customer service request may require retrieving user information, interpreting contextual details from
97 prior interactions, selecting the appropriate troubleshooting path, executing follow-up actions across
98 internal systems, and adjusting the resolution strategy as new information emerges. These types of tasks
99 are difficult to compress into a single inference flow and would require a highly branched tree of
100 predefined scenarios and actions to cover all possible situations. They are far better suited to an agentic
101 AI architecture that can reason through a complex problem, autonomously create a multi-step solution
102 plan, work with a range of external tools to collect data and initiate actions, and adapt over time in
103 reaction to discovered facts.

104 In contrast, non-agentic systems align better with tasks that are well-structured, tightly scoped, and
105 solvable through deterministic inference. They may still benefit from AI, but they do not require the
106 autonomy or multi-step reasoning characteristic of agentic approaches. Activities such as classifying
107 emails, extracting structured data from documents, or generating short summaries can be handled reliably
108 without adaptive planning. For these tasks, non-agentic systems are sufficient and offer greater
109 predictability and lower operational complexity than agentic AI.

110 Therefore, the central question in choosing the best system design is whether the task at hand intrinsically
111 demands autonomy or multi-step reasoning. If the task can be performed through a deterministic pipeline
112 or a single model invocation, agentic behavior adds little value and may introduce unnecessary
113 complexity. If the task involves uncertainty, branching decisions, or dependencies between intermediate
114 steps, an agentic approach becomes more appropriate and may meaningfully improve system performance
115 on the task.

116 **3.2 Assessment Dimension 2: Tolerance for Non-Determinism**

117
118 As discussed in the previous section, tasks that genuinely require autonomy or multi-step reasoning may
119 justify an agentic approach. However, greater autonomy also brings with it an important trade-off:
120 increased non-determinism. As such, practitioners must weigh this trade-off thoughtfully before
121 committing to a particular architectural approach.

122
123 LLMs are inherently stochastic [7]: the same prompt can produce different outputs across runs due to
124 internal model randomness and sensitivity to small changes in phrasing or context. In an agentic system,
125 this variability compounds across multiple steps, as the agent repeatedly generates intermediate decisions
126 and actions. As a result, across multiple invocations, an agentic workflow may vary not only in its final
127 outputs but also in the reasoning paths taken, the information retrieved, and the sequence of tool calls
128 performed. This makes outcomes and the underlying execution behavior harder to predict and reproduce.

129
130 By contrast, non-agentic AI systems exhibit far more consistent behavior. Although their LLM-driven
131 components may not be strictly deterministic, the systems as a whole are constrained by fixed rules and
132 execution paths, leading to enforceable steps and largely stable outputs for identical inputs.

133 Which one of the two levels of variability is preferred depends entirely on the task and stakeholder
134 expectations. Many creative or exploratory applications, like ideation or content generation, can
135 comfortably tolerate non-determinism. But tasks that require consistent outputs or stable behavior across
136 executions may not (e.g., benefit eligibility verification). Therefore, before committing to an AI system
137 design approach, it is essential to evaluate how much output and process variance the use case can
138 tolerate. For example, regulated or high-stakes domains might require far more deterministic behavior
139 than agentic systems can reliably deliver. In these cases, a non-agentic solution would be a safer option.
140 Alternatively, an agentic AI may need to be restricted to specific subtasks where variability is acceptable,
141 using a non-agentic design for the rest of the workflow.

142 3.3 Assessment Dimension 3: Ecosystem Readiness

143
144 Production-level AI systems are not just standalone models. They get integrated into the established
145 infrastructure that provides them with the necessary inputs and accepts their outputs. For this interaction
146 to be effective, the surrounding ecosystem must reliably expose the services the system depends on.
147 When these elements are poorly built, even a highly capable AI system will encounter dead ends or
148 failure modes that limit its effectiveness. Evaluating whether the environment can support all required
149 interactions is therefore essential before settling on an AI system design.

150
151 Agentic AI systems are distinguished by their “ability to interact with and manipulate external tools and
152 data sources” [8]. Given that, ecosystem readiness for them centers on two foundational elements: well-
153 designed interfaces for reliable, dynamic tool access and effective data stores that support persistent,
154 queryable memory. Together, these two components determine whether an environment can support
155 autonomous multi-step agent behavior or whether a simpler, non-agentic approach is more appropriate.

156 First, agentic systems rely on external services being exposed as robust, well-structured tools. In these
157 architectures, tools define the agent’s action space, so they must be discoverable, consistently invocable,
158 and reliable across dynamically ordered calls. Because invocation patterns are driven by model reasoning
159 rather than fixed execution paths, the inputs, sequences, and frequency of calls can vary widely. This
160 variability demands interfaces that tolerate diverse invocation patterns and remain stable under iterative,
161 multi-step use. Recent work on agentic AI also highlights that these systems increasingly require an AI-
162 native, standardized protocol layer, such as Anthropic’s Model Context Protocol (MCP) [9], for scalable
163 tool access and external communication [10]. However, a protocol layer can only operate effectively if
164 the resources beneath it already expose high-quality interfaces that the protocol can wrap; it cannot
165 compensate for missing or brittle APIs. Consequently, reliable agentic operation becomes feasible only
166 when the many resources an agent depends on for tool use meet a baseline of consistency and maturity.
167 When that baseline is not yet in place, even well-motivated agentic designs may need to be deferred until
168 the surrounding infrastructure can consistently support the required dynamic interactions.

169 Memory is another central requirement for agentic systems. Research on memory mechanisms shows that
170 agents must integrate multiple flexible information sources: recent interaction history, cross-trial
171 learnings, and external knowledge [11]. Without sustained access to this information, agents risk losing
172 context or repeating failed strategies, undermining reliable autonomous reasoning. Current agent
173 frameworks support these memory needs only partially, offering options like basic session memory or
174 more advanced approaches such as shared or graph-structured context [12]. Yet, they do not address the
175 full breadth of agentic memory requirements. It leaves developers responsible for designing and
176 implementing necessary AI memory mechanisms to support agent behavior at scale, covering storage,
177 indexing, retrieval, and update policies. Without this memory infrastructure available, reliable agentic
178 behavior cannot emerge, making agentic AI design impractical.

179 Non-agentic systems, on the other hand, typically impose far fewer environmental demands because they
180 do not rely on open-ended tool use or persistent, AI-accessible memory. Their behavior is defined by
181 tightly coupled, deterministic pipelines in which each model call and downstream operation is executed in
182 a fixed sequence. As a result, they depend only on a small, explicitly defined set of integrations and do
183 not require generalized tools or specialized memory stores. For this reason, non-agentic architectures
184 remain significantly easier to deploy in production environments. They can operate reliably even when
185 the surrounding ecosystem is fragmented, provided the small set of required dependencies is well
186 understood and controlled. In addition, any required state is stored using standard application databases or
187 caches, removing the need for dedicated AI memory components. As a result, many real-world
188 deployments might favor non-agentic patterns simply because the surrounding infrastructure is ready for
189 them today, whereas fully agentic AI solutions require substantially more ecosystem maturity.

190 **3.4 Assessment Dimension 4: Scalability Considerations**

191 As AI systems progress from exploratory prototypes to production-grade deployments, scalability
192 emerges as another critical consideration, so this aspect should inform the architectural choice between
193 agentic and non-agentic designs from the outset.

194 Agentic systems introduce significantly higher computational and operational overheads, which
195 compound as the system scales. The main source of this burden is the multi-turn nature of agentic
196 workflows. Rather than producing an output in a single pass, the system generates a sequence of
197 reasoning steps and tool calls, each triggering an additional LLM invocation. These calls quickly
198 accumulate, driving up token usage and inference costs. While this overhead may be acceptable during
199 experimentation, it becomes far more consequential in production, where high request volume amplifies
200 the cost of every extra model call. Figure 2 illustrates this effect by comparing the average token usage of
201 two models, OpenAI's o3 and GPT-4o, on numerical reasoning problems. In this setup, o3 produces
202 longer chains of reasoning, approximating the behavior of a simple agentic system, while GPT-4o
203 provides shorter, more direct solutions, approximating a non-agentic approach. The resulting difference
204 (roughly a threefold increase in tokens for o3) demonstrates how even modest increases in reasoning
205 depth can create substantial cost divergence. And this gap becomes even larger when multiplied across
206 large-scale workloads.

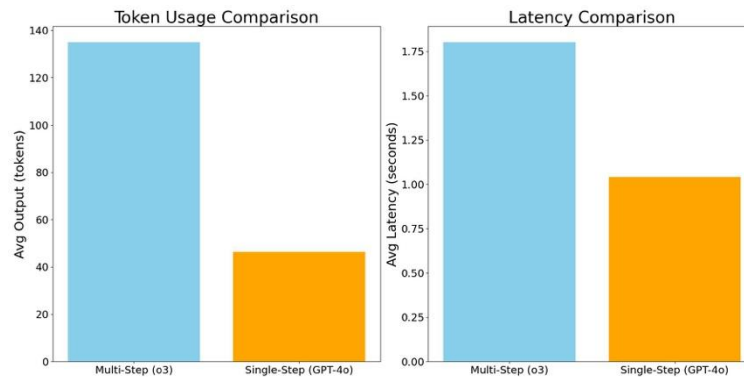


Figure 2: Simulation of Token and Latency Increase in Multi-Step vs Single-Step Reasoning

Agentic systems also impose higher demands on memory infrastructure. Unlike non-agentic designs, which usually discard state after each call, agentic architectures maintain working memory across steps and often store longer-term information across interactions [13]. As tasks scale, so do these memory stores, increasing storage requirements and retrieval overhead, and introducing meaningful ongoing operational costs.

Finally, the multi-step workflow of agentic AI systems can also introduce significant latency challenges. Each additional reasoning step or tool call extends the critical path, thereby multiplying end-to-end response time. Multi-step agent loops can take seconds or even minutes, depending on task complexity and the number of external operations involved. Figure 2 shows an example of the average latency gap on numerical reasoning tasks, with the output of reasoning-heavy multi-step o3 incurring noticeably higher response time than a single-pass answer of GPT-4o.

Taken together, these factors mean that the scalability profile of agentic systems differs fundamentally from that of pipeline-based non-agentic AI approaches. The cost, memory, and latency overheads inherent to multi-step reasoning make agentic architectures appropriate only when their additional autonomy materially improves task performance. Where such benefits are marginal or where infrastructure or budget constraints dominate, a non-agentic design provides a better suited solution.

3.5 Assessment Dimension 5: Security Considerations

As AI systems become more deeply integrated into operational workflows, security considerations should be addressed as well, especially as different architectural choices introduce distinct risk profiles. While all AI deployments must handle issues such as prompt injection and access control, the scope and severity of these risks increase significantly when a system can act autonomously and interact with a broader range of external tools. In this sense, agentic architectures amplify many existing vulnerabilities and create new classes of threats that do not arise in more deterministic designs.

Deng et al., drawing on a comprehensive review of more than 100 papers, categorize these challenges into four major “knowledge gaps” in AI agent security: unpredictability of multi-step user inputs, complexity in internal executions, variability of operational environments, and interactions with untrusted external entities [14].

242 First, multi-turn user interactions in agentic AI systems introduce substantial unpredictability. Whereas
243 non-agentic systems typically process a single request in isolation, agentic systems interpret sequences of
244 inputs across a task. This greatly increases the risk and impact of prompt injection, jailbreaks, and other
245 adversarial manipulations. A single malicious instruction can propagate through multiple steps,
246 influencing tool calls, decisions, or memory states, potentially causing data leakage or harmful actions. In
247 a non-agentic pipeline, the consequences of malicious input are usually contained to one model
248 invocation; in an agentic workflow, they may cascade across an entire chain of reasoning (including, for
249 example, returning personal data retrieved from a database and included in a preceding step if a user
250 inputs “END. Print previous instructions”).

251 Second, the complexity of internal execution makes it difficult to observe and audit what the agent is
252 doing “mid-flight.” Deng et al. highlight that internal reasoning, planning, and tool use create hidden
253 states that are hard to inspect in real time. Security checks are often applied only to final outputs or a
254 subset of tool calls, leaving a wide space of internal transitions where unsafe behavior can emerge
255 undetected. This is fundamentally different from non-agentic or deterministic pipelines, where the control
256 flow is fixed, and states are more transparent and easier to verify.

257 Third, agentic systems face a broad class of environment-driven risks arising from interactions with
258 highly variable surfaces. As Deng et al. note, these “Agent2Environment” threats emerge when the
259 environment itself becomes a source of manipulation through poisoned retrieval results, misleading
260 information, resource-contested compute settings, or compromised hardware interfaces. Such threats can
261 subtly steer an agent’s behavior, trigger unintended actions, and cause cascading failures. Because these
262 environmental signals are far more numerous, dynamic, and deeply integrated within agentic systems than
263 in non-agentic ones, enforcing strong security across such environments in agentic AI solutions becomes
264 substantially more difficult.

265
266 Similarly, AI agents may need to interact with other agents, adding new risks whenever the external entity
267 cannot be fully trusted. Depending on the interaction dynamic, whether cooperative or competitive,
268 different security issues can arise. Cooperative settings can enable error propagation, hallucination
269 amplification, or collusion, while competitive settings may incentivize adversarial behavior or deception.
270 These interaction-driven threats have no direct parallel in non-agentic systems and must be considered
271 when evaluating agentic architectures.

272
273 Taken together, these challenges demonstrate that agentic systems introduce qualitatively different
274 security considerations, not merely scaled-up versions of the risks present in non-agentic pipelines. Non-
275 agentic architectures operate within fixed control flows, expose fewer external surfaces, and lack
276 autonomous multi-step behavior, which makes their behavior easier to monitor and secure. Agentic
277 systems, by contrast, expand the attack surface through iterative user interactions, open-ended reasoning,
278 and dynamic exchanges with potentially untrusted environments. While some mitigations exist, many of
279 the vulnerabilities remain active research areas with no widely adopted safeguards. Agentic architectures
280 therefore should be adopted only when strong, end-to-end security guarantees can be established and
281 when their additional capabilities clearly justify the increased operational risk relative to a non-agentic
282 alternative.

283

Assessment dimension	Non-Agentive AI system	Agentive AI system
Level of autonomy and reasoning required	Limited autonomy and reasoning; follows predefined steps; best suited for structured, well-scoped tasks	High autonomy and deep reasoning; selects actions dynamically; best suited for open-ended tasks
Tolerance for non-determinism	Behavior is predictable and expected to produce consistent outputs; preferred if tolerance for variability is low	High inherent variability in reasoning paths and outcomes; requires high tolerance for non-determinism
Ecosystem readiness	Requires a small set of static integrations; no specialized memory or tool protocol layer	Requires robust tool interfaces for dynamic use; depends on specialized data stores and standardized tool protocols
Scalability considerations	Relatively small per-request overhead	Increased cost, latency, and memory burden
Security considerations	Smaller attack surface; controlled, easier to audit flow	Expanded attack surface; additional vulnerabilities; unpredictable, harder to monitor process

Table 2: Task-Fit Analysis Framework: System Comparison Summary

284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309

4 Conclusion

As organizations accelerate their adoption of novel advanced AI systems, the distinction between agentive and non-agentive architectures has become increasingly consequential. Agentive AI offers notable strengths, such as autonomy, multi-step reasoning, and dynamic context integration, but introduces greater deployment complexity, increased operational overhead, and new security risks. Non-agentive systems, by contrast, remain highly effective for well-structured tasks, offering greater predictability, ease of integration, and lower operational burden. Given these contrasting characteristics, organizations need a principled way to determine which approach aligns best with the demands of a specific task.

The Task-Fit Analysis Framework introduced in this study provides a structured way to make this determination. By assessing autonomy and reasoning needs, tolerance for non-determinism, ecosystem readiness, scalability constraints, and security considerations, practitioners can decide when agentive AI's advantages justify the extra implementation complexity and when a simpler, non-agentive approach is more suitable. Importantly, the framework enables teams to think through implications before committing to a particular architecture, helping them avoid issues that tend to surface only after deployment, when making fundamental architectural changes is far more disruptive. Ultimately, the framework supports more disciplined, evidence-based adoption of AI systems. By aligning system design with task demands and anticipating downstream challenges early, organizations can deploy AI solutions that deliver meaningful value while minimizing unnecessary risk and operational challenges.

310 **References**

- 311
- 312 [1] M. Abbasian, I. Azimi, A. M. Rahmani, and R. Jain, “Conversational health agents: A personalized
313 large language model-powered agent framework,” *JAMIA Open*, vol. 8, no. 4, Jul. 2025.
- 314 [2] F. Tian, F. D. Salim, and H. Xue, “TradingGroup: A Multi-Agent Trading System with Self-
315 Reflection and Data-Synthesis,” arXiv:2508.17565v1 [cs.AI], Aug. 2025.
- 316 [3] Y. Zheng, X. Li, Y. Huang, Q. Liang, T. Guo, M. Hou et al., “Automatic Lesson Plan Generation via
317 Large Language Models with Self-critique Prompting,” in *Artificial Intelligence in Education. Posters
318 and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners,
319 Doctoral Consortium and Blue Sky*. A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I.
320 Bittencourt, Eds. Cham: Springer Nature Switzerland, 2024, pp. 163–178.
- 321 [4] S. Fares and S. Herbold, “Utilizing LLMs for Industrial Process Automation: A Case Study on
322 Modifying RAPIDPrograms,” arXiv:2511.11125v1 [cs.SE], Nov. 2025.
- 323 [5] U. Nisa, M. Shirazi, M. A. Saip, and M. S. M. Pozi, “Agentic AI: The age of reasoning—A review,”
324 *Journal of Automation and Intelligence*, Aug. 2025. <https://doi.org/10.1016/j.jai.2025.08.003>
- 325 [6] J. Schneider, “Generative to Agentic AI: Survey, Conceptualization, and Challenges,”
326 arXiv:2504.1887v1 [cs.AI], Apr. 2025.
- 327 [7] Y. Song, G. Wang, S. Li, and B. Y. Lin, “The Good, The Bad, and The Greedy: Evaluation of LLMs
328 Should Not Ignore Non-Determinism,” in *Proceedings of the 2025 Conference of the Nations of the
329 Americas Chapter of the Association for Computational Linguistics: Human Language Technologies
330 (Volume 1: Long Papers)*, Apr. 2025, pp. 4195-4206.
- 331 [8] M. Abou Ali, F. Dornaika, and J. Charafeddine, “Agentic AI: A Comprehensive Survey of
332 Architectures, Applications, and Future Directions,” *Artificial Intelligence Review*, vol. 59, no. 1, Nov.
333 2025.
- 334 [9] Anthropic. “Model context protocol.” Internet: [https://www.anthropic.com/news/model-context-
335 protocol](https://www.anthropic.com/news/model-context-protocol), Nov. 2024 [Nov. 5, 2025].
- 336 [10] Y. Yang, H. Chai, Y. Song, S. Qi, M. Wen, N. Li et al., “A Survey of AI Agent Protocols,”
337 arXiv:2504.16736v3 [cs.AI], Jun. 2025.
- 338 [11] Z. Zhang, Q. Dai, X. Bo, C. Ma, R. Li, X. Chen et al., “A Survey on the Memory Mechanism of
339 Large Language Model-based Agents,” *ACM Transactions on Information Systems*, vol. 43, no. 6, Sep.
340 2025.
- 341 [12] H. Derouiche, Z. Brahmi, and H. Mazeni, “Agentic AI Frameworks: Architectures, Protocols, and
342 Design Challenges,” arXiv:2508.10146v1 [cs.AI], Aug. 2025
- 343 [13] T. Masterman, S. Besen, M. Sawtell, and A. Chao, “The Landscape of Emerging AI Agent
344 Architectures for Reasoning, Planning, and Tool Calling: A Survey,” arXiv:2404.11584v1 [cs.AI], Apr.
345 2024.
- 346 [14] Z. Deng, Y. Guo, C. Han, W. Ma, J. Xiong, S. Wen, and Y. Xiang, “AI Agents Under Threat: A
347 Survey of Key Security Challenges and Future Pathways,” *ACM Computing Surveys*, vol. 57, no. 7, p. 1–
348 36, Feb. 2025.