iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.80037

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Open AGI - Multi-Agent Automation Framework for AI-Orchestrated Software Development

Ms. Dhanashree Patil[1], Gauri Dongre[2], Laukik Patil[3], Shubham Maske[4]
Department of Artificial Intelligence and Data Science AISSMS Institute of Information Technology, Pune, India

Abstract: ***The quick growth of Artificial Intelligence (AI) and Large Language Models (LLMs) has sped up the creation of autonomous systems that can handle complex tasks similar to those performed by humans. Even with these improvements, software development still requires a lot of human effort, teamwork, and industry knowledge. This paper introduces Open AGI, a Multi-Agent Automation Framework designed for AI- managed software development. In this system, several intelligent agents work together to turn natural language requirements into complete software solutions that can be deployed. Each agent in Open AGI has a specific role, such as Product Manager, Architect, Developer, and Quality Analyst, and follows set Standard Operating Procedures (SOPs) to ensure consistency, accuracy, and reliability throughout the development process.***

The survey reviews current multi-agent and LLM-based frameworks like AutoGPT, LangChain, and MetaGPT, pointing out their strengths and weaknesses in automating software engineering tasks. By comparing these systems, the paper highlights the research gap that Open AGI fills: the lack of a combined, role-based automation framework capable of managing the entire software development process using AI management. This new approach not only simplifies project execution but also encourages scalability, flexibility, and less reliance on human input in software creation.

Keywords: Open AGI, Multi-Agent Systems, Artificial Intelligence, Software Automation, Natural Language Programming, Large Language Models (LLMs), SOP- Driven Development.

I. INTRODUCTION

Artificial Intelligence has progressed from basic rule-based systems to large neural networks that can reason and understand natural language. Large Language Models, like GPT-4, Gemini, Claude 3, and LLaMA-3, show a level of contextual understanding that enables them to draft documents, write code, and perform reasoning tasks with high accuracy. However, traditional software engineering still involves many steps and requires teams of experts and significant coordination. Human-centered workflows can limit scalability and lead to inefficiencies due to delays in communication, inconsistent documentation, and reliance on specialized skills. Automation in software engineering has mainly targeted specific tasks, such as code completion or test generation. Recent frameworks, including AutoGPT, LangChain, and Microsoft AutoGen, expand these abilities by coordinating several LLM-based agents to achieve specific goals. However, these frameworks often lack formal collaboration methods, clear task assignments based on domain needs, and a well- defined lifecycle structure. Their outputs are often isolated or not reproducible, which hampers full automation. To address these challenges, Open AGI presents a unified framework driven by standard operating procedures that organizes intelligent agents to independently handle each phase of software development. With clear role definitions, hierarchical communication, and persistent memory, Open AGI can create requirement documents, architecture diagrams, source code, test cases, and deployment scripts directly from natural language prompts. This paper reviews related research, outlines the architectural design of Open AGI, and discusses its benefits compared to current frameworks. The aim is to show that a system driven by multiple agents and LLMs can function like a self-sufficient software company, providing consistent and scalable project delivery.

The past decade has marked a decisive turning point in Artificial Intelligence (AI). Systems that once depended on explicitly programmed rules have been superseded by machine- learning architectures capable of reasoning, adaptation, and creativity. Among these advances, Large Language Models (LLMs)—massive neural networks trained on trillions of tokens—have proven to be the most transformative. Models such as GPT-4, Claude 3, Gemini 1.5, and LLaMA 3 can generate human-like text, synthesize code, summarize knowledge, and perform multi-step reasoning. Their success has led to the emergence of AI agents—software entities that use language as both an interface and a reasoning medium—to plan, act, and collaborate autonomously.

Despite this explosion of capability, the software- development process remains one of the few domains still dominated by human coordination. Developing reliable software requires orchestrating specialists across multiple disciplines: product management, architecture design, development, quality assurance, and operations.

©IJRASET: All Rights are Reserved

Each role contributes distinct expertise, and their collaboration follows structured Software Development Lifecycle (SDLC) phases such as requirement analysis, design, implementation, testing, and deployment. Even when assisted by AI-based tools—for example, code completion in IDEs or automated testing frameworks—humans remain the principal decision- makers. This human dependency creates bottlenecks of communication, delays in iteration, and inconsistencies in documentation.

Traditional automation efforts in programming have mostly targeted narrow tasks. Early systems such as rule- based expert frameworks could automatically generate small templates of code, but they lacked contextual understanding. Statistical language models and later neural architectures extended this to autocomplete code snippets, yet the scope remained at the function or module level. The arrival of transformer-based LLMs expanded the horizon by introducing reasoning chains and contextual retention, allowing models to produce entire classes, APIs, or configuration files. Nevertheless, these LLMs are still usually deployed as single monolithic entities responding to isolated prompts, not as members of a coordinated workflow.

To achieve truly autonomous software creation, researchers began exploring multi-agent collaboration, where several LLM instances play specialized roles and exchange structured messages to complete complex goals. Early frameworks like AutoGPT (2023) demonstrated how an agent could spawn sub-agents to plan, execute, and self- evaluate tasks. LangChain introduced the concept of agent "chains," providing modular connectivity to databases, APIs, and external tools. Microsoft AutoGen formalized multi- agent dialogues, defining explicit communication channels, memory sharing, and human-in-the-loop supervision. MetaGPT extended these foundations into a virtual software company, encoding Standard Operating Procedures (SOPs) for roles such as Product Manager, Architect, and QA Engineer.

Although these frameworks have moved the field forward, each exhibits critical limitations when examined through the lens of complete software-project automation. AutoGPT and LangChain operate primarily as experimental sandboxes rather than standardized workflows. They lack persistent state management and inter-agent accountability; once an execution loop finishes, all context is lost. AutoGen improves dialogue structure but remains confined to short-term sessions. MetaGPT, while pioneering SOP encoding, provides only a partial pipeline and does not address continuous integration, deployment, or feedback-driven refinement. Furthermore, few of these frameworks include DevOps compatibility—version control, containerization, or cloud deployment—meaning the generated output rarely matures into production-ready systems.

II. BACKGROUND AND RELATED WORK

A. *Evolution of Intelligent Systems and Automation*

Artificial Intelligence has come a long way from the days when computers could only follow rigid, pre-programmed rules. In its earliest stages, AI was built on symbolic reasoning, where logic and inference were manually coded into the system. Programs like *ELIZA* and *SHRDLU* could mimic small parts of human conversation, but their understanding was shallow — more like following a script than real thinking.

The next generation of AI, in the 1980s, introduced expert systems — machines that could make decisions within a narrow domain using thousands of "if-then" rules. They were impressive for their time but quickly hit a wall. Each new rule had to be hand-crafted, and the systems could not adapt or learn.

The 1990s and early 2000s brought the machine-learning revolution. Instead of hardcoding rules, models learned patterns directly from data. Algorithms like decision trees, support vector machines, and random forests changed how we approached automation — but they still needed structured, labeled datasets and could not understand natural language.

The real transformation came with deep learning. Once neural networks became capable of scaling to millions (and later billions) of parameters, AI could begin to approximate the human brain's pattern-recognition ability. With better GPUs, larger datasets, and improved training techniques, deep learning became the foundation for the most powerful AI models in existence today.

Then came the transformer architecture, introduced in 2017, which completely changed how machines process information. Transformers made it possible for AI to understand context over long sequences of data — the key breakthrough behind today's Large Language Models (LLMs) such as GPT, Claude, Gemini, and LLaMA. These models can understand human intent, maintain conversation context, generate code, and even explain their reasoning.

With this shift, AI stopped being just a tool — it began to resemble a collaborator. Instead of following fixed instructions, it could understand goals and propose solutions. This ability to reason in natural language laid the groundwork for what we now call multi-agent systems, where many intelligent models can think, plan, and work together like a team of digital professionals.

III. LITERATURE SURVEY

Artificial Intelligence (AI) research has entered a stage where machines can perform increasingly complex reasoning and collaboration tasks that once required human intellect. The progression from rule-based systems to autonomous, conversational multi-agent systems has inspired researchers to explore the boundaries between human cognition, machine intelligence, and collaborative automation. This literature survey reviews key developments across these themes and highlights how they converge in the design philosophy behind **Open AGI**, a framework for AI-orchestrated software development.

A. Human–Machine Collaboration and Cognitive Learning

The idea of machines augmenting human decision-making has been studied for decades. Early work by Azad Abad et al. (2017) introduced the concept of *autonomous crowdsourcing*, where human and machine participants collaborate to solve tasks through shared learning. This research emphasized that effective collaboration depends not just on automation, but on how information is communicated and validated between participants — a concept central to Open AGI's *orchestration layer*, which structures communication among intelligent agents.

As the field evolved, researchers began examining how language models could simulate human reasoning and cooperation. Aher et al. (2023) demonstrated that large language models (LLMs) can simulate multiple human participants in controlled environments, effectively replicating social behavior and decision-making patterns. Such findings provided the groundwork for viewing AI not merely as a computational system, but as a socially interactive entity.

Similarly, Abdurahman et al. (2024) explored both the opportunities and risks of integrating LLMs into sensitive areas like psychology, showing that while they can model human- like responses, their reliability depends on context and oversight. These insights reinforce Open AGI's emphasis on SOP-driven governance and human-in-the-loop supervision — ensuring that agents' outputs remain interpretable and verifiable.

B. Emergence of Multi-Agent Collaboration

The transition from individual models to multi-agent frameworks marked a turning point in AI research. With LLMs capable of understanding and generating natural language, agents could now communicate, reason collectively, and divide labor — much like human teams. Ahn and Sentis (2021) proposed a Nested Mixture of Experts architecture to enable cooperative and competitive learning among agents, setting early theoretical foundations for distributed intelligence systems.

Recent studies have expanded this paradigm to social and competitive dynamics. For example, Abdelnabi et al. (2024) analyzed negotiation among LLM-based agents under conditions of cooperation, competition, and even malicious intent. Their work revealed that agentic systems exhibit emergent behaviors, including self-interest and conflict resolution, underscoring the need for robust orchestration and ethical oversight — both of which are explicitly addressed in Open AGI's hierarchical communication model.

Furthermore, Akbulut et al. (2024) discussed the risks of anthropomorphizing AI agents, warning that human-like behavior can mislead users into over-trusting machine reasoning. Open AGI mitigates this risk by using structured communication templates and validation checkpoints, ensuring that each agent's contribution is transparent, auditable, and aligned with project goals adapting general-purpose LLMs to agricultural domains and proposes future research directions. Limitations noted

C. Advances in Large Language Models and Their Technical Capabilities

The capabilities of Open AGI's agents depend heavily on advances in Large Language Models (LLMs). The GPT-4 technical report (Achiam et al., 2023) outlines how transformer-based architectures with instruction tuning and reinforcement learning from human feedback (RLHF) enable models to understand nuanced prompts and follow domain- specific instructions. This technology underpins Open AGI's agent specialization — allowing different agents (Product Manager, Architect, Developer, etc.) to interpret, reason, and act within their roles.

Parallel advancements have improved the interpretability and adaptability of language models. Anonymous (2024) introduced gradient-based domain generalization methods that enhance the ability of models to generalize across datasets without retraining. Techniques like Pareto-optimal gradient matching and federated domain adaptation reduce overfitting, improving consistency across diverse project types — a key strength for a general-purpose automation framework like Open AGI.

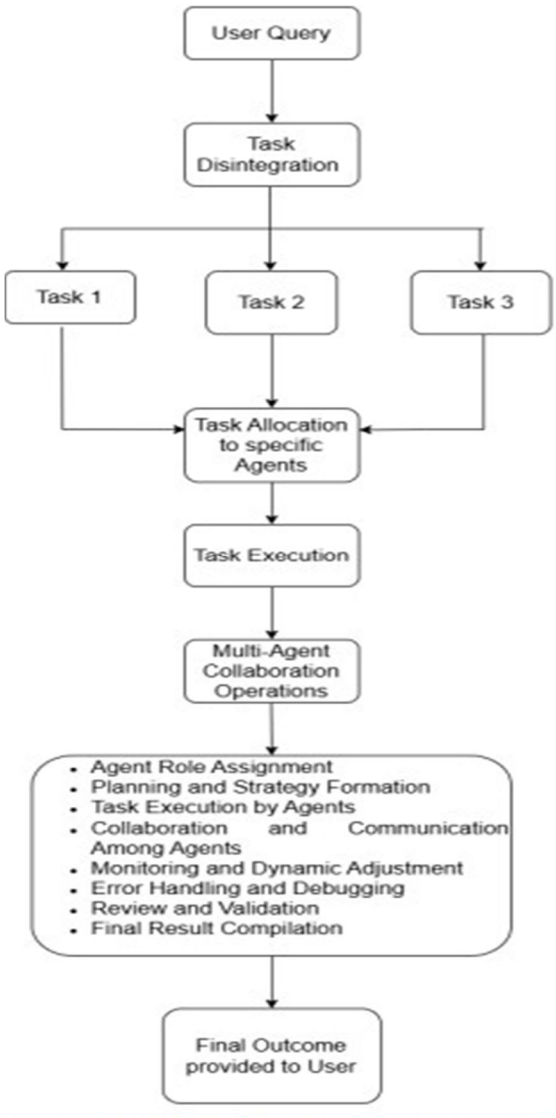
At the same time, researchers such as Gabriele Ansaldo (2023) have explored AgentSpeak architectures that embed LLMs within agent-based modeling environments. Ansaldo's work, which applied such systems to simulate human decision- making in electric vehicle adoption, shows how LLMs can serve as dynamic cognitive modules within multi-agent environments — directly influencing Open AGI's modular, role-based structure.

D. Frameworks for Multi-Agent Orchestration

The introduction of open-source frameworks like AutoGPT, LangChain, AutoGen, and MetaGPT accelerated the adoption of agentic AI. These systems demonstrated that LLMs could be configured as goal-driven entities capable of autonomous reasoning.

- AutoGPT (2023) pioneered recursive task decomposition, allowing an agent to plan, execute, and self-evaluate. Its weakness — limited memory and lack of coordination — inspired future models to adopt shared vector databases and orchestrators.
- LangChain provided modular pipelines for chaining tasks and integrating APIs, making it ideal for experimentation but lacking governance or lifecycle structure.
- Microsoft AutoGen introduced conversational collaboration between multiple agents, proving that AI entities could coordinate through dialogue, though it still relied heavily on human intervention.
- MetaGPT (2024) went further by defining SOPs and assigning LLMs to professional roles such as Product Manager or Developer. However, it lacked persistent memory and DevOps integration, leaving a gap for frameworks capable of *continuous, full-lifecycle automation.*

Open AGI builds upon these foundations, combining the SOP discipline of MetaGPT with the scalability and persistence lacking in earlier models. Its orchestration layer ensures that multiple agents can operate concurrently while maintaining a coherent shared context.



E. Ethical, Social, and Practical Considerations

As LLM-based multi-agent systems grow in capability, researchers have increasingly focused on ethical and societal implications. Akbulut et al. (2024) cautioned that anthropomorphic design choices can distort user perceptions of AI competence, leading to misplaced trust. Abdurahman et al. (2024) echoed these concerns, emphasizing transparency and accountability when using LLMs in human-facing contexts.

These warnings are highly relevant for Open AGI, which must ensure that AI-generated outputs — whether system designs or deployment scripts — are interpretable and traceable. Therefore, Open AGI incorporates explainable-agent pipelines where every decision and response is logged and validated by another agent, enforcing a form of peer review within the system.

Furthermore, Open AGI's modular design allows human oversight at every stage. Users can inspect decisions, revise prompts, and control final outputs, ensuring the framework supports human creativity rather than replacing it. This aligns with the vision described by Abad et al. (2017) — that true autonomy arises not from removing humans, but from designing systems where human and machine reasoning complement each other harmoniously.

F. Identified Research Gaps

From the reviewed literature, several consistent challenges and opportunities emerge:

1) Fragmented Collaboration: Most current frameworks handle only specific parts of the development lifecycle, lacking an integrated orchestration system.
2) Lack of Persistent Context: Even advanced frameworks lose memory once a session ends, reducing consistency across large projects.
3) Weak Evaluation Protocols: There is no universal benchmark for measuring agent collaboration quality or output coherence.
4) Limited Real-World Deployment: Most systems generate text or code but cannot manage deployment pipelines autonomously.
5) Ethical and Trust Issues: Anthropomorphism and hallucination risks require strict validation and transparency mechanisms.

Open AGI directly addresses these gaps by combining persistent memory (vector databases), standardized workflows (SOPs), and DevOps integration (Git, Docker, CI/CD) into a unified ecosystem. It aims not only to automate programming tasks but to institutionalize machine collaboration as a disciplined, transparent, and scalable process.

The literature shows a clear evolution: from human–machine collaboration (Abad et al., 2017) to emergent multi-agent reasoning (Abdelnabi et al., 2024), and from rule-based automation to adaptive LLM ecosystems (Achiam et al., 2023). While existing systems like MetaGPT made valuable progress toward role-based AI, none achieve the full orchestration, lifecycle integration, and persistent intelligence that Open AGI delivers.

Thus, Open AGI positions itself as a bridge between research and real-world automation — a practical step toward AI- orchestrated software development, where intelligent agents work like a true team, following procedures, learning continuously, and creating deployable systems with minimal human intervention.

IV. ROLE OF LARGE LANGUAGE MODELS

Large Language Models (LLMs) are the central intelligence driving the Open AGI framework, acting as the reasoning core that allows its multiple autonomous agents to think, plan, and collaborate like members of a professional software team. These models form the foundation upon which the system's intelligence is built, enabling each agent—whether functioning as a Product Manager, Architect, Developer, Quality Analyst, or DevOps Engineer—to understand natural-language instructions, generate meaningful reasoning, and produce structured outputs. Unlike traditional automation tools that execute predefined rules, LLMs understand intent, maintain context across long interactions, and adapt their responses based on the project's evolving state. The transformer architecture that underlies these models allows them to relate complex dependencies and preserve coherence, making them ideal for orchestrated multi- agent collaboration. Within Open AGI, each agent is guided by structured prompts and predefined Standard Operating Procedures (SOPs), which shape the model's behavior and define its specific role. This approach allows a single LLM architecture, such as GPT-4, Claude, or LLaMA, to emulate distinct professional specializations. Through these carefully designed prompts, LLMs transition from general-purpose text generators into role-specific experts capable of producing requirement documents, design blueprints, code implementations, and quality reports. The models also enable fluent natural-language communication between agents, allowing them to exchange feedback, validate outputs, and reason collectively. To preserve consistency and context across interactions, Open AGI incorporates vector-based memory systems that store semantic representations of previous exchanges, enabling long-term recall and continuity in complex projects. Beyond reasoning, LLMs in Open AGI also act as bridges between human-like understanding and machine execution.

They integrate seamlessly with external tools such as compilers, APIs, and CI/CD pipelines, empowering the system to transform generated content into deployable, verifiable results. By combining reasoning with action, the framework ensures that its outcomes are grounded in functional execution rather than theoretical predictions. Another vital contribution of LLMs in this project is their ability to learn and adapt through continuous feedback loops. Each agent's performance is periodically reviewed by the Orchestrator and peer agents, allowing Open AGI to refine prompts and evolve its internal coordination strategies without retraining the models themselves. This adaptive learning process makes the system progressively more efficient over time. Moreover, Open AGI's architecture is model-agnostic—it can dynamically assign different models to tasks depending on cost, performance, or privacy requirements, allowing flexibility across projects. The use of LLMs also brings creativity into the engineering workflow. They generate multiple design possibilities, propose innovative solutions, and significantly reduce cognitive load for users by allowing them to communicate objectives in simple language while the system handles the complex reasoning and execution behind the scenes. However, despite their transformative potential, LLMs are not without challenges. They can occasionally produce incorrect or biased information, which Open AGI mitigates through rigorous cross-validation and human-in-the-loop review for critical tasks. Each agent's output is peer-checked by another, ensuring that only validated and coherent results progress through the pipeline. Ethical safeguards are embedded throughout the system to maintain transparency and accountability, preventing over-reliance or anthropomorphism of AI behavior. In essence, LLMs act as the cognitive engine of Open AGI—infusing intelligence, reasoning, and creativity into every layer of the framework. Their capacity for natural- language understanding, multi-agent communication, and adaptive learning transforms Open AGI from a set of static automation modules into a living, cooperative ecosystem of intelligent agents capable of end-to-end, AI-orchestrated software development.

V. MULTIMODAL AI: VISION, SPEECH, AND LANGUAGE INTEGRATION

The integration of multimodal artificial intelligence within Open AGI marks a major step toward creating a system that perceives, understands, and interacts with the world in a more human-like way. Unlike traditional AI systems that rely solely on textual input, Open AGI combines vision, speech, and language processing to form a unified cognitive environment where agents can comprehend and reason across multiple types of data. The visual intelligence layer enables the framework to interpret images, diagrams, or screenshots—an ability particularly valuable in analyzing system architectures, UI mockups, and even error logs shared as images. Through computer vision modules powered by deep convolutional and transformer-based models, the system can detect objects, interpret context, and transform visual content into structured data for reasoning by other agents. The speech processing module adds another dimension to the interaction, allowing users to communicate naturally through voice commands and receive spoken feedback. Using automatic speech recognition (ASR) and text-to-speech (TTS) models, Open AGI converts speech into textual instructions that are then interpreted by language agents, making it accessible to non-technical users and removing barriers for those who may prefer verbal over written communication. The language integration layer, built on large language models, serves as the cognitive bridge connecting these modalities. It interprets text, links it with corresponding visual or auditory information, and ensures coherent understanding across formats. For example, when a user describes a system verbally while uploading a design image, the system's multimodal layer synchronizes these inputs— allowing the Architect agent to reason over both spoken requirements and visual references simultaneously. This integration of multiple sensory streams mirrors human perception, where understanding arises from blending sight, sound, and language together. Beyond improving interaction, multimodal AI enhances contextual comprehension and accuracy within Open AGI.

Visual cues help verify code logic or UI design, speech adds fluid command input, and textual reasoning binds everything into a coherent thought process. Together, these components make Open AGI more intuitive, responsive, and versatile in real-world development environments. Moreover, this fusion of modalities reduces cognitive friction between humans and machines—users no longer need to conform to rigid interfaces but can interact as they naturally would with another person. Ethical safeguards are also embedded into this multimodal framework to prevent misuse of visual or voice data, ensuring privacy, consent, and transparency in every interaction. In essence, the integration of vision, speech, and language transforms Open AGI from a text-based automation tool into a truly perceptive, multimodal ecosystem capable of understanding, reasoning, and collaborating across multiple forms of human communication—bringing it one step closer to the reality of general, adaptive intelligence.

VI. RAG-ENABLED KNOWLEDGE FRAMEWORK FOR MULTI-AGENT AUTOMATION

Retrieval-Augmented Generation (RAG) plays a vital role in enhancing the reasoning and factual accuracy of Open AGI's multi-agent ecosystem by bridging the gap between static language models and dynamic, real-world information. Traditional large language models operate primarily on knowledge encoded during training, which can become outdated or incomplete over time.

RAG overcomes this limitation by enabling the system to retrieve relevant, up-to- date information from curated databases, knowledge repositories, or live APIs before generating a response. Within Open AGI, each intelligent agent—whether functioning as a Product Manager, Developer, or Quality Analyst—uses the RAG mechanism to access contextual information specific to its domain, ensuring that all reasoning and output are grounded in verified data rather than relying solely on model memory. For example, when the Architect agent designs a system based on recent framework updates or emerging technologies, it first retrieves current documentation or technical guidelines from trusted online sources, which are then synthesized by the LLM to produce an accurate, context-aware response. Similarly, the QA agent can use retrieval to reference testing standards or existing defect databases before recommending validation strategies. This combination of retrieval and generation ensures that every agent's decision is both informed and adaptive, preventing hallucination and misinformation—two common limitations of standalone language models. RAG also serves as a shared memory layer for the entire Open AGI organization, allowing agents to store and retrieve project-specific information from previous sessions. This means that when a new project begins, agents can learn from earlier development patterns, recall architectural principles, or reuse tested modules, significantly improving system efficiency and reducing redundancy. The integration of vector databases and semantic search within this mechanism allows for fast, precise information retrieval based on meaning rather than keywords, making the system's memory both intelligent and scalable.

Beyond accuracy, RAG fosters collaboration among agents, as retrieved knowledge is accessible across the multi-agent network, ensuring that every agent works with the same up-to- date information and maintains consistency throughout the software development lifecycle. This shared context strengthens collective reasoning and project coherence, allowing Open AGI to behave more like a well-informed human organization. The incorporation of RAG also reinforces transparency and traceability, since every piece of external information used to generate output is logged and can be verified—addressing ethical and accountability concerns in AI- driven decision-making. In essence, RAG transforms Open AGI's knowledge system into a living, evolving repository that continuously learns, retrieves, and refines information to support autonomous software creation. It allows the framework to move beyond pre-trained intelligence toward dynamic, retrieval-driven cognition, where every generated output reflects not only the model's learned reasoning but also the most relevant and reliable knowledge available at that moment.

VII. CHALLENGES AND LIMITATIONS

Despite its transformative potential, the Open AGI framework faces several challenges and limitations that stem from the inherent complexities of large-scale artificial intelligence, multi-agent coordination, and real-world integration. The first and most fundamental challenge lies in the reliability of large language models (LLMs) themselves. While LLMs enable reasoning, planning, and collaboration, they are still prone to generating hallucinated or factually incorrect information, particularly when working on open- ended or ambiguous tasks. This issue can cascade within a multi-agent environment, where one agent's incorrect output may influence others, amplifying small inconsistencies into larger system-level errors. To address this, Open AGI employs multi-layered validation and peer review among agents, but achieving absolute reliability remains a technical hurdle. Another major challenge involves context management and memory persistence. Even with vector-based databases and retrieval-augmented generation, maintaining coherence across long and complex projects requires efficient memory governance. Over time, the growing amount of stored information can lead to redundancy, retrieval delays, or irrelevant context recall, impacting both speed and accuracy. Optimizing this memory architecture without sacrificing scalability is an ongoing research problem.

Communication overhead represents a third limitation. As the number of agents increases, message passing and decision synchronization can become computationally expensive, particularly when multiple agents work in parallel. Designing lightweight yet effective coordination protocols is crucial for sustaining scalability in larger deployments. Similarly, task orchestration and conflict resolution within a diverse team of AI agents present significant challenges. When multiple agents independently generate creative or alternative solutions, integrating these into a unified and consistent output requires a sophisticated arbitration mechanism. The Orchestrator plays a vital role here, but as the system grows in complexity, balancing autonomy with control becomes a delicate task.

Ethical and transparency concerns also remain a pressing limitation. Since Open AGI automates parts of software engineering that traditionally require human judgment, ensuring accountability and explainability is essential. Each agent must provide interpretable reasoning for its actions to prevent "black- box" decision-making, especially in safety-critical applications. Moreover, protecting user data and proprietary information during retrieval and generation processes demands rigorous privacy safeguards. The integration of external APIs, version- control systems, or cloud services introduces potential vulnerabilities that must be continually monitored and mitigated.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Another limitation lies in model dependency and adaptability. Open AGI's performance heavily depends on the underlying LLMs it uses. Variations in model accuracy, token limits, or API latency can directly affect task outcomes. Furthermore, as LLMs evolve rapidly, maintaining compatibility and ensuring consistent agent performance across different versions can be challenging. There is also the issue of cost efficiency, as running multiple large-scale models simultaneously demands substantial computational power and infrastructure, which may not be practical for small organizations or local deployments.

Finally, while Open AGI represents a major step toward autonomous software development, achieving true general intelligence—where agents exhibit reasoning and creativity comparable to humans—remains an ambitious goal. Current systems can simulate teamwork and structured workflows but lack emotional intelligence, intuition, and long-term strategic foresight. The framework's reliance on textual interaction limits its ability to interpret non-verbal cues, emotional tone, or human intent beyond linguistic meaning. Overcoming this will require future integration of affective computing, real-time learning, and more sophisticated multimodal understanding.

In summary, while Open AGI introduces a powerful paradigm for AI-orchestrated software development, it also highlights critical areas for improvement. Ensuring factual accuracy, reducing communication overhead, optimizing memory, enhancing explainability, maintaining ethical transparency, and managing computational costs remain open challenges. These limitations do not undermine the system's innovation but instead define a roadmap for continuous refinement—guiding the evolution of Open AGI from an intelligent automation framework toward a truly adaptive, human-aligned artificial intelligence ecosystem.

VIII. RESEARCH GAPS

Even though significant progress has been made in the field of multi-agent artificial intelligence and autonomous software systems, there remain several research gaps that limit the full realization of AI-orchestrated software development. The first major gap lies in the lack of unified orchestration frameworks. Most existing systems—such as AutoGPT, LangChain, or MetaGPT—focus on specific aspects of autonomy like task decomposition or role-based simulation, but they do not offer an integrated environment that governs the complete software lifecycle. These frameworks operate in silos, often lacking standardization in communication, task validation, and memory management. Open AGI begins to close this gap by introducing a structured orchestration layer and standardized Standard Operating Procedures (SOPs), but research is still needed to formalize these orchestration protocols, establish inter-agent communication standards, and ensure interoperability across heterogeneous AI systems. A second research gap is related to long-term memory and contextual continuity. While Retrieval-Augmented Generation (RAG) and vector databases provide partial solutions, most current systems struggle with maintaining persistent, meaningful context across large-scale, multi-phase projects. Once project size or duration increases, agents often lose consistency, misinterpret earlier decisions, or repeat redundant tasks. Addressing this limitation requires deeper research into *context compression*, *hierarchical memory management*, and *semantic state representation*—areas where the field has yet to achieve reliable, scalable results. Open AGI's vector- memory integration offers a starting point, but developing efficient lifelong memory architectures remains an open challenge.

Another critical gap exists in the evaluation and benchmarking of multi-agent intelligence. Unlike traditional AI systems that can be measured by accuracy, F1 score, or BLEU metrics, multi-agent ecosystems lack clear, standardized performance indicators. Evaluating reasoning quality, collaboration efficiency, and communication relevance is still a subjective process. There is an urgent need for robust benchmarking frameworks that can assess agent cooperation, task completion, reliability, and adaptability under different conditions. Establishing such standardized metrics would help validate claims of intelligence and autonomy in systems like Open AGI, providing the research community with consistent tools for comparison and improvement.

A further gap lies in the ethical and accountability mechanisms of autonomous AI teams. As AI systems become more capable of self-directed decision-making, ensuring transparency, explainability, and fairness becomes increasingly important. Current research provides only partial solutions through post-hoc interpretability methods or human-in-the- loop validation. However, there is still no universally accepted framework for ethical orchestration—one that embeds moral reasoning and accountability directly into the agents' communication and decision-making protocols. Open AGI mitigates this partially by implementing validation hierarchies and explainable task logs, but future work must focus on defining AI governance standards that ensure every autonomous decision can be traced and justified.

Scalability and resource optimization also represent persistent research challenges. As the number of collaborating agents grows, computational costs, latency, and communication complexity increase dramatically. The scalability bottleneck prevents most multi-agent systems from being deployed in real-world, enterprise-scale environments. Research into distributed orchestration, decentralized communication models, and lightweight agent architectures could help overcome these constraints, enabling systems like Open AGI to function efficiently even with hundreds of interacting agents.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

There is also a significant gap in human-AI collaboration models. While many frameworks allow AI agents to assist humans, few focus on balanced co-creation, where AI and human participants share mutual understanding and decision authority. Current designs either over-rely on automation or demand excessive human supervision, which limits productivity and adaptability. More research is needed to establish seamless hybrid workflows where humans and agents can dynamically adjust roles, delegate responsibilities, and co- evolve solutions. Open AGI lays the foundation for this through its interactive orchestration interface but leaves room for further innovation in emotional intelligence, adaptive trust calibration, and cognitive alignment between humans and machines.

Finally, an emerging gap involves cross-domain generalization and multimodal integration. Most multi- agent frameworks operate primarily in text-based environments, limiting their ability to process or correlate information from other modalities such as vision, audio, or real- time sensor data. While Open AGI's architecture supports multimodal extensions, there is still a lack of extensive research on how agents can simultaneously reason over multimodal inputs, adapt to domain shifts, and transfer learned strategies across contexts without retraining. Future exploration into unified multimodal reasoning could make AI systems more robust, adaptable, and capable of operating in diverse real- world settings.

While Open AGI advances the current frontier by combining role-based orchestration, persistent memory, and DevOps integration, it also exposes the broader research landscape's unmet needs. There remains much to be explored in long- term contextual reasoning, ethical orchestration, benchmarking, and scalable communication design. Bridging these gaps will not only strengthen the Open AGI framework but also push the field closer to realizing true cooperative intelligence—an era where AI agents can autonomously create, evaluate, and evolve software systems with reliability, transparency, and human alignment at their core.

IX. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Ms. Dhanashree Patil, Department of Artificial Intelligence and Data Science, AISSMS Institute of Information Technology, Pune, for her invaluable guidance, encouragement, and continuous support throughout the development of this research work. Her insightful feedback, patient supervision, and dedicated involvement greatly contributed to the clarity, direction, and refinement of this project. The authors also extend their heartfelt appreciation to Dr. Riyaz A. Jamadar, Head of the Department of Artificial Intelligence and Data Science, for providing an inspiring academic environment, constant motivation, and access to essential resources that enabled the successful completion of this study. The authors further acknowledge the faculty members, technical staff, and fellow students of the department for their cooperation, constructive discussions, and unwavering moral support during the entire course of this work. Their collective contributions have played a significant role in shaping the outcomes of this research.

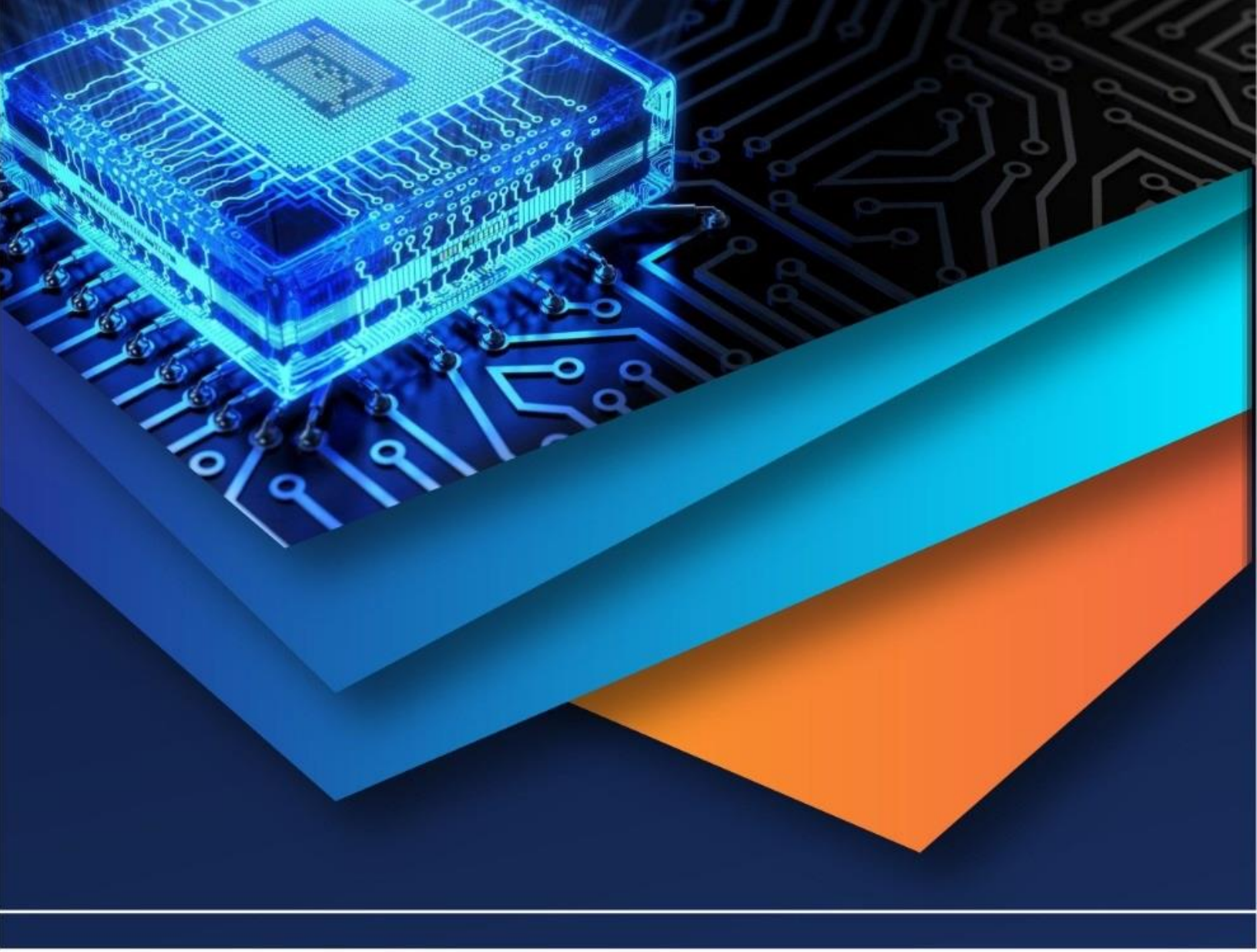
This survey has thoroughly explored the rapidly evolving landscape of LLM-based multi-agent automation systems with a focused examination of Open AGI, a collaborative AI framework designed to transform software development through intelligent orchestration. We have traced the progression from early rule-based automation and isolated LLM tools to today's coordinated multi-agent ecosystems capable of reasoning, planning, and executing complex development workflows. Our analysis highlights that while advances in large language models, retrieval-augmented generation, multimodal understanding, and role-based prompting have opened new possibilities for autonomous software engineering, the path to practical deployment still requires addressing critical challenges such as long-term memory consistency, scalable agent coordination, ethical transparency, and robust integration with real-world development tools. Nevertheless, the emergence of frameworks like Open AGI marks a significant step toward trustworthy, adaptive, and collaborative AI systems that can operate as intelligent partners throughout the software lifecycle, bridging the gap between human creativity and machine-driven automation.

REFERENCES

[1] Cassio Pennachin and Ben Goertzel, 2007. "Contemporary Approaches to Artificial General Intelligence", Artificial general intelligence - Springer Berlin Heidelberg, pp. 1-30.
[2] Ben Goertzel, 2023. "Generative AI vs. AGI: The cognitive strengths and weaknesses of modern LLMs", ArXiv preprint arXiv:2309.10371
[3] Nanyi Fei, Zhiwu Lu, et al, 2022. "Towards artificial general intelligence via a multimodal foundation model", Nature Communications, 13/1, pp. 3094.
[4] Fernando Diaz, and Michael Madaio, 2024. "Scaling laws do not scale", Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. 7, pp. 341-357.
[5] Qingyun Wu, Gagan Bansal, et al, 2023. "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation", arXiv preprint arXiv:2308.08155, ¾
[6] Rahul M. Samant, et al, 2013. "A Study on Feature Selection Methods in Medical Decision Support Systems", International Journal of Engineering Research & Technology (IJERT), 2/11, pp. 615-619.
[7] Ben Goertzel, " Artificial General Intelligence : Concept, State of the art, and Future prospects", Journal of Artificial General Intelligence, 5/1, 2014.
[8] J. Grudin and Richard Jacques, 2019. "Chatbots, Humbots, and The Quest for Artificial General Intelligence", Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1-11.

[9] Peter Voss, 2007. “Essentials of General Intelligence: The Direct Path to Artificial General Intelligence”, Springer Berlin Heidelberg, pp. 131– 157.
[10] Sébastien Bubeck, Varun Chadrasekaran, et al, 2023. “Sparks of Artificial
[11] General Intelligence: Early Experiments with GPT-4 ”, ArXiv, 2303.
[12] Rahul Samant and Srikantha Rao, 2013. “A study on Comparative Performance of SVM Classifier Models with Kernel Functions in Prediction of Hypertension”, International Journal of Computer Science and Information Technologies, 4/6, pp. 818-821.
[13] Minqi Jiang, Tim Rocktäschel, Edward Grefenstette, 2023. “General intelligence requires rethinking exploration”, Royal Society Open Science, 10/6, pp. 230539.
[14] Lin Zhao, Lu Zhang, Zihao Wu, Yuzhong Chen, Haixing Dai, Xiaowei
[15] Yu, et al, 2023. “ When brain inspired AI meets AGI ”, Meta-Radiology
[16] - Elsevier, 1/1, pp. 100005.
[17] James Babcock, János Kramár and Roman Yampolskiy, 2016. “The AGI containment problem”, Artificial General Intelligence: 9th International Conference, AGI 2016, New York, NY, USA, July 16-19, 2016, Proceedings 9, Springer International Publishing, pp. 53-63.
[18] Andrzej Cichocki and Alexander P Kuleshov, 2021. “Future trends for Human-AI collaboration: A Comprehensive Taxonomy of AI/AGI using Multiple Intelligences and Learning Styles”, Computational Intelligence and Neuroscience - Hindawi, 2021/1, pp. 8893795
[19] Rahul Samant and Srikantha Rao, 2013. “Evaluation of Artificial Neural Networks in Prediction of Essential Hypertension”, International Journal of Computer Applications, 81/12.
[20] T. Everitt, G. Lea and M. Hutter, 2018. “ AGI Safety Literature Review ”, International Joint Conference on AI.
[21] D. B. Acharya, K. Kuppan and B Divya, 2025. “Agentic AI: Autonomous intelligence for complex goals – A comprehensive survey”, IEEE Access
[22] Erik Miehling, Karthikeyan Natesan Ramamurthy, Kush R Varshney, Matthew Riemer, Djallel Bouneffouf, John T Richards, Amit Dhurandhar, et al, 2025. “Agentic AI Needs a Systems Theory”, arXiv preprint arXiv:2503.00237, IBM Research.
[23] Shanmugasundaram Sivakumar, 2024. “Agentic AI in Predictive AIOps: Enhancing IT Autonomy and Performance”, International Journal of Scientific Research and Management (IJSRM), 12/11, pp. 1631- 1638.
[24] Antonio Lieto, Mehul Bhatt, Alessandro Oltramari, David Vernon, 2018. “The role of cognitive architectures in General Artificial Intelligence”, Cognitive Systems Research, Volume - 48, pp. 1-3.
[25] Steve J Bickley and Benno Torgler, 2023. “Cognitive architectures for artificial intelligence ethics”, AI & Society, 38/2, pp. 501-519.
[26] Qian Niu, Junyu Liu, et al, 2024. “Large Language Models and Cognitive Science: A Comprehensive Review of Similarities, Differences, and Challenges”, arXiv preprint arXiv:2409.02387.
[27] Henry Hexmoor, Johan Lammens, et al, 2025. “Behaviour Based AI, Cognitive Processes, And Emergent Behaviors in Autonomous Agents”, WIT Transactions on Information and Communication Technologies, Volume-1.
[28] Teppo Felin, Matthias Holweg, 2024. “Theory Is All You Need: AI, Human Cognition, and Decision Making”, Strategy Science, 9/4, pp. 346- 371.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)