iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

A TAXONOMY OF FAILURE MODES FOR PRODUCTION LLM AGENTS

Akhil Reddy Mandadi
Independent Researcher, India

ABSTRACT
Large language model (LLM) agents have been deployed at scale in the enterprise, where they can reason, retrieve, plan, and communicate with external tools to complete tasks autonomously. All these are very useful features which enhance automation and decision making, but they also have caused a new class of failures in their operations, which is very different from conventional software systems. In contrast to deterministic applications, the production LLM agents behave probabilistically, dynamically understanding user intent, generating plans, calling external tools and communicating with changing knowledge sources. As a result, the failures tend to be a product of reason, grounding in the environment, execution, and externalities of the system more often than they are a result of isolated software issues. Although there are numerous production incidents reported publicly, organizations are still reporting failures in a variety of different terms, which affects effective communication, systematic debugging, and the creation of uniform mitigation strategies. In this work, we introduce a taxonomy of LLM agent failures, with five complementary dimensions of grounding fidelity failure, tool invocation correctness failure, plan coherence failure, intent alignment failure, and side effect scope failure. The taxonomy is developed by summarizing and analyzing published incident reports, engineering post mortems and practitioner experience to identify common operational patterns in a variety of deployments. To each of these categories, the taxonomy defines relationships between observable symptoms, causes, operational detection signals and mitigation approaches. The proposed framework is expected to help establish a consistent and shared vocabulary for incident classification, promote organizational learning, enhance observability in production, and pave the way for the evolution of standardized reliability engineering practices for autonomous LLM agent systems.

Keywords:
Large Language Models (LLMs), AI Agents, Autonomous Agents, Failure Taxonomy, AI Reliability, Production AI, Tool Invocation, AI Operations.

I. INTRODUCTION

1.1 Background
Large language models (LLMs) have revolutionized AI by making it more independent and proficient in handling complex tasks with minimal human oversight. Current production LLM agents combine reasoning with access to external tools, retrieval systems, databases, application programming interfaces (APIs), software services, and so on to achieve multi-step tasks that go beyond generating text. This has led organizations in various sectors such as healthcare, finance, manufacturing, software development, education, and customer service to start implementing autonomous agents to streamline processes, enhance productivity, and assist in making informed decisions (Jin et al., 2024).

Whereas conventional software runs executable code, production LLM agents perform uncertain actions, yielding a probability distribution of outputs in the context of their reasoning. They repeatedly understand user requests, fetch external data, devise execution plans, call up specialized tools, check intermediate data, and adjust subsequent actions based on environmental changes. These properties not only give them increased flexibility of operation but also create new types of reliability problems which were not considered during classical software design (Weber, 2024).

As autonomous agents become more part and parcel of production infrastructures, reliability, safety and governance are growing more and more in importance for their operations. Organizations are increasingly relying on LLM agents to interact with customers, provide software development support, manage infrastructure, retrieve enterprise knowledge, assist in financial analysis, and make decisions in operations. Therefore, even small errors in the reasoning can be cascaded throughout complex systems, resulting in unforeseen actions with business impacts. Additionally, the operation of such systems is further complicated by

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

recent advances in agent architectures such as Collaborative Multi-agent Systems where failure of an agent can propagate across interacting multiple agents before being detected (Tran et al., 2025; Wu et al., 2023).

Reliable frameworks that can handle the distinctive behaviors of these LLM agents are thus needed to move toward autonomous production systems. Organizations are increasingly aware that the failures that occur during operation are not primarily due to implementation of individual tools, but are the result of interactions between reasoning processes, grounding in context, planning behavior, tool execution and uncertainty in the environment. With the ongoing trend of intelligent agents taking on more operational responsibility, grasping these interconnected failure mechanisms becomes critical to ensure the trustworthy deployment of intelligent agents, enhance their resilience, and foster sustainable adoption within enterprise environments (Wang et al., 2025).

1.2 Emerging Challenges in Production

As production LLM agents have taken off, they have revealed a new set of operational issues which are significantly different from those found in traditional software systems. Numerous autonomy failures of customer-service agents, software engineering assistants, and enterprise automation platforms have been reported in the public eye and have been shown to come from inaccurate reasoning rather than deterministic failures. These accidents frequently include the use of hallucinatory tool parameters, wrong retrieval results, misunderstanding about the user's intentions, logically consistent plans that are contextually inappropriate, and unintended changes in external systems.

A major aspect of production agent failure is that it has multiple dimensions. One failure is not caused by one defect alone, but by a number of interacting mechanisms to bring about undesirable results. For instance, if the retrieval is mistaken it may fail to provide adequate contextual grounding, then the reasoning model will generate an incorrect execution plan and call on the wrong external tools. This cascading effect makes it difficult to debug, since the apparent failure mode is usually much different from what caused it. Recent work on operational AI systems (Wang et al., 2024) and LLM-based failure management (Zhang et al., 2024) highlights the critical need for understanding the interplay between reasoning, execution, and infrastructure behavior for production reliability

As autonomous agents are being deployed in safety-critical and enterprise applications, their risks of operations are further exacerbated. Multi-agent collaboration adds more dependencies, where the reasoning of the one agent can affect the decisions of other collaborating agents, thus causing localized failure to propagate through distributed workflows. Failure attribution has been investigated, and it is still a great challenge to detect the failure source of the agent that caused the failure in the downstream agents in collaborative autonomous systems (Zhang, S., et al., 2025). Likewise, role-aware coordination mechanisms have demonstrated that managing failure is not only increasingly reliant on the understanding of interactions between specialized agents, but also on understanding individual agents separately (Zhang, L., Zhai, et al., 2025).

Another emerging challenge is regarding uncertainty in autonomous planning and execution. Production agents often have to make sequential decisions that affect the future trajectory of production, and are making these decisions under incomplete information conditions. However, uncertainty detection is still a challenge prior to taking the wrong actions, since the traditional monitoring methods focus on monitoring execution results, not reasoning uncertainty. In recent years, research on uncertainty aware failure detection indicates that confidence scores can be tracked and updated during planning, thus enabling better detection of uncertain decisions at earlier stages of the planning process, and avoiding high operational consequences of such decisions (Zheng et al., 2024).

All the above challenges show that production LLM agent failures can't be explained by each and every software failure individually. Rather, organizations need systematic methods that might describe failures in reasoning, planning, grounding, execution, and coordination that are connected in a unified framework of operations.

1.3 Research Gap

Current software reliability models have been developed for deterministic computing systems, where the behavior of a system is known and is executed according to given rules, and failures are usually identified as particular implementation bugs. Existing methods like Failure Mode and Effects Analysis (FMEA), software fault classification, and other reliability engineering techniques have played valuable roles in enhancing system robustness for industrial systems. However, these approaches require relatively stable execution boundaries which are different from the probabilistic reasoning processes of the production LLM agents (Xia et al., 2024).

While recent research has advanced the work on LLM safety, alignment, risk assessment, and agent evaluation, most work has focused on specific failure categories individually and without a unified framework that can be

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

used to express operational failures in production settings. Studies are often conducted to analyze one of these aspects, such as hallucination, vulnerabilities of prompt engineering, safety assessment, uncertainty estimation or benchmark performance in isolation, without providing a consistent vocabulary to describe complex production incidents (Cui et al., 2024; Zhang, Z., et al., 2024).

Standardized terminology is hard to come by, which poses practical difficulties for engineering teams charged with deployment and maintenance of autonomous agents. Organizations may have different internal definitions for failures, depending on their operational experiences, which leads to misunderstandings between teams and reduces the scope for knowledge-sharing between organizations. This means that if a similar incident occurs, it can be classified differently even though it has the same root cause, making comparative analysis, incident reporting and systematic learning very challenging. Work in recent years looking at benchmarking practices has also pointed out that the current evaluation methods are unable to fully reflect the operational risks of actual production agents, especially for situations in which reliability and safety are essential requirements (Chen, Z., et al., 2025)

1.4 Research Objective

This research work focuses on creating a systematic taxonomy of production LLM agent failure modes to establish a common framework for categorizing, analyzing, and reporting operational failures in the deployment of autonomous agents in the real world. The taxonomy we propose aims to classify failures along five complementary dimensions grounding fidelity failures, correctness failures in the tool invocation, plan coherence failures, failures of intent alignment and failures due to scope of the side effects.

In addition to classification, the proposed framework seeks to create links and connections between the various operational failure symptoms observed, the root causes thereunder, the measurability of the detection signals, and the actions that can be taken to mitigate. The study aims to facilitate the organization's learning, enhance incident investigation, production monitoring and reliability practices for autonomous AI systems, and link these elements in a single taxonomy.

The taxonomy is also designed to serve as a common conceptual language for researchers and practitioners to consistently compare production incidents across various application areas. This standardization can help streamline evaluation approaches, facilitate reproducible research, and facilitate more systematic creation of operational safeguards for increasingly independent LLM agent ecosystems.

1.5 Contributions

This research contributes to the growing body of work in the field of production LLM agent operations in five key ways. First, it proposes a taxonomy of the failure modes of operations, which is systematic and based on five interconnected dimensions that relate to the key phases of an autonomous agent's operation. Second, the taxonomy defines clear linkages between failure classes, observable symptoms, internal failure causes, detection signals that can be recognized during operation, and representative mitigation actions, going beyond failure classification to extend failure analysis.

Third, the proposed framework offers a common language to facilitate more consistent communication between researchers and engineers and organizations deploying production LLM agents. The use of common or standardized terms helps with documentation of incidents, helps to compare incidents across different deployment environments, and builds up transferable operational knowledge. Fourth, the taxonomy provides practical directions for reliability engineering, as it can reveal common operational patterns which can be used for monitoring, observability, governance, and risk management. These learnings align with the latest research on automated root cause analysis, AI diagnostics and intelligent failure management in the growing trend of automated software ecosystems (Maheshkar, 2025; Chandrasekaran, 2025; Chen, X., et al., 2025).

Last but not least, the study provides the conceptual basis for further studies on automatic failure attribution, self-correction mechanisms, the integration of reinforcement learning, grounded execution and adaptive production monitoring for autonomous LLM agents (Pan et al., 2023; Pternea et al., 2024; Rivkin et al., 2023). These contributions aim to progress academic knowledge and practical reliable engineering of production LLM agent systems.

2. RELATED WORK

2.1 Traditional Software Reliability Models

Structured methods like Failure Mode and Effects Analysis (FMEA), Fault tree analysis and software fault classification have been used for a long time in traditional software reliability engineering to identify, prioritize and mitigate operational risks prior to the deployment of systems. These approaches are based on the deterministic nature of execution, meaning that the actions of the system are constrained by rules and, in most

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

cases, known implementation defects can be blamed for the failure. In recent times, there have been attempts to modernize these approaches by incorporating LLMs with traditional risk management practices. For instance, Xia et al. (2024) showed that LLM-aided FMEA can enhance the effectiveness and structured engineering reasoning for failure mode identification in industrial systems. Similarly, Weber (2024) introduced a taxonomy for LLM-integrated software applications, noting that traditional software architectures need to be adapted when probabilistic language models are made executable components of a system. **Table 1** shows the key characteristics of typical software failures and production LLM agent failures to highlight the conceptual differences.

Table 1 Comparison between Traditional Software Failures and Production LLM Agent Failures

Characteristic	Traditional Software Systems	Production LLM Agents
Execution model	Deterministic	Probabilistic reasoning
Failure origin	Programming defects	Reasoning, grounding, planning, and tool interaction
Root cause identification	Generally traceable	Often distributed across multiple execution stages
Recovery strategy	Bug fixing and testing	Monitoring, validation, guardrails, and adaptive mitigation
Operational behavior	Fixed execution logic	Dynamic context dependent decision making

Source: Adapted from Weber (2024) and Xia et al. (2024)

It is clear from **Table 1** that the production LLM agents are fundamentally different from traditional software. For this reason, current models of reliability offer important engineering principles, but fail to account for the dynamic and connected nature of system failures for systems of autonomous reasoning.

2.2 Reliability Research for Large Language Models

Recent studies focus on the dependability, security, and hazards of LLM-based systems. Cui et al. (2024) created an extensive taxonomy of risks to safety, mitigation strategies, and evaluation benchmarks of LLM systems, and Kirk et al. (2023) explored personalization risks and alignment policies for human centric LLM systems. To complement these studies, Wang et al. (2025) proposed a comprehensive safety framework that covers aspects from data to model training, deployment, and operational governance, demonstrating that production reliability goes beyond the development of a model.

The possibility of enhancing model robustness after deployment has been investigated. While Zheng et al. (2024) explored uncertainty based approaches for detecting failures in closed loop planning, Pan et al. (2023) examined various self-correction methods that allow LLMs to detect and correct their own reasoning errors. All these studies suggest that more than simply model testing, continuous monitoring and adaptive intervention are necessary to increase operational reliability. Conceptually, the relations between major themes in reliability research are summarized in **Figure 1.**

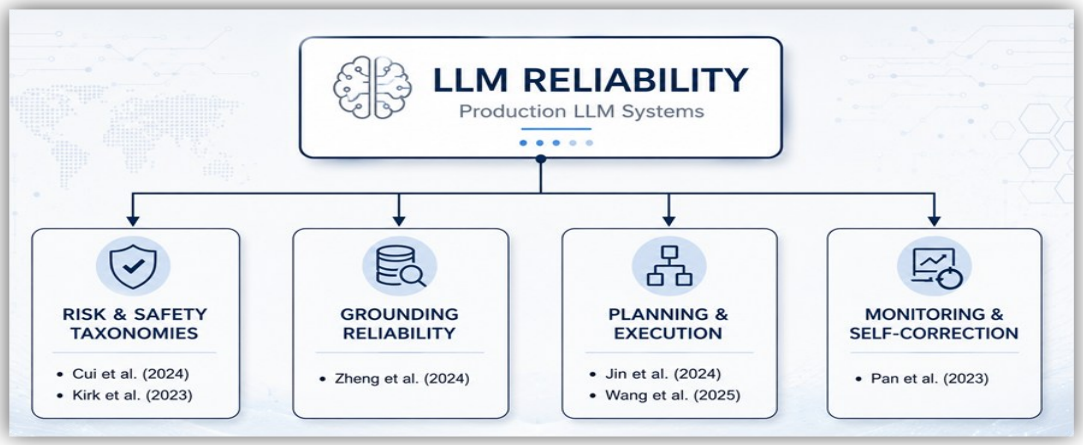

Figure 1. Conceptual landscape of reliability research for production LLM systems.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

Source: Developed by the authors based on Cui et al. (2024), Kirk et al. (2023), Pan et al. (2023), Zheng et al. (2024), Jin et al. (2024), and Wang et al. (2025).

As shown in **figure 1**, this research in reliability has taken multiple complementary paths in the present era, from safety assessment, to grounding, to planning, to monitoring, to self-correction. However, the above research lines tend to focus on isolated aspects of reliability and neglect to integrate the full cycle of autonomous agent operations.

2.3 Existing Agent Evaluation Frameworks

The shift from single LLMs to independent multi-agent systems has sparked significant research into assessing, collaborating, and evaluating the performance of agents. Jin et al. (2024) surveyed LLM-based software engineering agents, and found that planning, using tools, and reliability during execution are significant research challenges. AutoGen was introduced by Wu et al. in 2023, which facilitates multi-agent conversations, and Tran et al. in 2025, who surveyed collaboration mechanisms that allow autonomous agents to reason in a coordinated way across distributed agents. Conventional assessment methods also have been highlighted in recent benchmarking studies as not being very effective in capturing operational safety and production risk. For instance, Z. Chen et al. (2025) proposed that operational risk is the most important metric for assessing financial-domain agents instead of benchmark accuracy, while Z. Zhang et al. (2024) proposed Agent-Safety Bench to measure safety behaviors of autonomous financial LLMs.

While these frameworks offer valuable insights into the autonomous agent's performance capabilities, they tend to focus on individual aspects like planning error, group efficiency, or safety adherence. Relatively little effort has been invested in the creation of a single operational taxonomy to systematically link grounding failures, tool invocation errors, planning inconsistencies, intent misalignment and unintended side effects in a common framework for analysis. This unresolved gap serves as the main motivation for the taxonomy proposed here that aims to develop a common language to describe and analyze production LLM agent failures in various deployment settings.

3. METHODOLOGY

3.1 Taxonomy Development Process

Through a qualitative taxonomy development approach, a structured taxonomy of production LLM agent failure modes was built. The taxonomy was developed by synthesizing all the peer-reviewed studies, industrial technical reports, engineering post mortems, best practices frameworks, and publicly documented production incidents involving autonomous LLM agents. The review did not concentrate on specific failures of technical aspects of music making, instead it concentrated on identifying repetitive operation patterns that relate to reasoning, planning, grounding, music making with tools, and autonomous execution. This evidence-driven method is in line with the current research that supports a systematic approach to classify risk and assess operational reliability of production AI systems (Cui et al., 2024; Zhang, L., Jia et al., 2024; Wang et al., 2025).

The literature sources were chosen based on the inclusion criteria already identified to make it relevant to the issue of production agent reliability.

Table 2 Literature Selection Criteria for Taxonomy Development

Criterion	Description
Publication relevance	Studies addressing LLM agents, reliability, safety, or operational failures
Technical scope	Research involving production deployment, benchmarking, monitoring, or multi-agent systems
Evidence source	Peer-reviewed articles, surveys, technical reports, and engineering case studies
Analytical focus	Failure causes, detection mechanisms, mitigation strategies, or operational evaluation

Source: Developed by the authors based on Cui et al. (2024), Wang et al. (2025), and Zhang, L., Jia, et al. (2024)

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

The selection of criteria guaranteed that the taxonomy was taken from representative literature that captured the cutting edge of research and the deployment experience, which increased the applicability of the taxonomy in a wide range of production situations.

3.2 Classification Criteria

Failure modes were systematically grouped based on common operational characteristics, such as the way that failures occur, failure symptoms, failure root causes, failure detection signals, and failure mitigation patterns as found in the literature reviewed. It proved to be a multidimensional classification approach which allowed similar failures to be grouped into coherent classes, yet also maintained meaningful class distinctions between reasoning, grounding, planning, tool invocation, intent interpretation and side-effect propagation.

Table 3 Operational Criteria Used for Failure Classification

Classification Criterion	Purpose
Operational behavior	Distinguishes how failures emerge during execution
Observable symptoms	Identifies externally visible manifestations
Root cause	Determines underlying contributing factors
Detection signals	Supports monitoring and diagnosis
Mitigation pattern	Associates each failure with practical corrective strategies

Source: Developed by the authors from the synthesized literature

The resulting criteria give a consistent analytical framework for comparing heterogeneous production incidents and help to categorize failures in a consistent way across autonomous LLM agent deployments.

3.3 Validation Strategy

Preliminary taxonomy was further developed by comparing and contrasting various studies several times to ensure conceptual consistency and to remove duplicate taxonomies. Cross-study synthesis enabled the identification of recurring patterns of failure, enhanced the definition of categories, and also contributed to the clarity of the operations. This iterative validation approach aligns with recent research that has focused on systematically evaluating, executing with uncertainty, attributing failures automatically, building multi-agent reliability, and diagnosing operational failures using AI (Chen, X., et al., 2025; Chen, Z., et al., 2025; Zheng et al., 2024; Chandrasekaran, 2025).

4. PROPOSED TAXONOMY OF PRODUCTION LLM AGENT FAILURES

4.1 Grounding Fidelity Failures

The grounding fidelity failures happen when an LLM agent builds responses or execution plans based on incomplete, outdated and unreliable contextual information. These are often due to the failure to retrieve the information, to the presence of outdated knowledge bases, or to external information sources that are inaccurate, resulting in outputs that look coherent but are not related to the operational context. Many retrieval-augmented generation failures and downstream planning and execution errors often stem from inadequate grounding, which is heavily reliant on external knowledge repositories as autonomous agents grow in usage of such sources (Zhang, L., Jia, et al., 2024; Rivkin et al., 2023). Therefore, enhancing retrieval quality, source validation and contextual consistency is an important prerequisite for successful production deployment.

4.2 Tool Invocation Correctness Failures

Cases of failure when invoking tools, come when agents misinteract with external systems, even if they selected a suitable reasoning strategy. Common failures are selecting the wrong tools, schema violations, hallucinated parameters, malformed API requests, and execution sequences. Because production agents are constantly feeding and linking in databases, APIs, cloud services, and enterprise apps, the wrong tools can be used to introduce operational errors outside of the language model. Recent research highlights that validating tools,

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

enforcing schemas, and monitoring their execution significantly boosts the reliability of deployments and minimizes unintended behavior in systems (Jin et al., 2024; Weber, 2024).

4.3 Plan Coherence Failures

Plan coherence failures refer to cases where an agent fails to keep reasoning or performing a task logically throughout several steps. They can take the form of partial workflow, loops of reasoning, premature stoppage or contradictory decisions in the middle of the process, etc. Each of the reasons may seem correct separately, but the shortcomings become apparent when further actions aren't meant to achieve the initial goal. These behaviors grow increasingly evident in multi-agent environments that involve collaborative planning with several independent components that are dependent on one another (Tran et al., 2025; Wu et al., 2023). **Table 4** summarizes the distinguishing operational characteristics of these three dimensions, as summarized in the first three.

Table 4 Operational Characteristics of Core Taxonomy Dimensions

Taxonomy Dimension	Primary Failure Manifestation	Representative Mitigation
Grounding Fidelity	Incorrect or incomplete contextual understanding	Retrieval validation and source verification
Tool Invocation Correctness	Invalid tool usage or parameter generation	Schema validation and execution guardrails
Plan Coherence	Inconsistent multi-step reasoning	Workflow monitoring and planning constraints

Source: Developed by the authors based on Jin et al. (2024), Weber (2024), Tran et al. (2025), and Zhang, L., Jia, et al. (2024)

Table 4 shows that each taxonomy dimension focuses on a specific stage in the operation but are very closely related when executed independently.

4.4 Intent Alignment Failures

Intent alignment failures are when the intent that the user has is misread by the agent in some way. These failures are not only related to language but also to the differences between user expectations and autonomous decision making. These symptoms may be excessive autonomy, unauthorized assumptions, and over-scoped task execution. Accurate interpretation of human intent is a key requirement for autonomous trustworthy operation, which is consistently reported in the research literature on personalization, human LLM interaction, and operational safety (Gao et al., 2024; Kirk et al., 2023; Zhang, Z., et al., 2024).

4.5 Side Effect Scope Failures

Side effect scope failures are when something undesirable happens outside of the scope of the immediate execution goal. They range from state changes that have been made without permission, duplicate transactions, unwanted service interactions or violations of organizational access limits. Failure of these is especially important as they impact external systems and business processes. Modern safety systems thus suggest a continuous validation of rights of access, logging of actions and rollback options to limit the impact on operations (Wang et al., 2025; Chen, Z., et al., 2025). The relationship between the five proposed taxonomy dimensions is shown in a conceptual model in **Figure 2.**

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

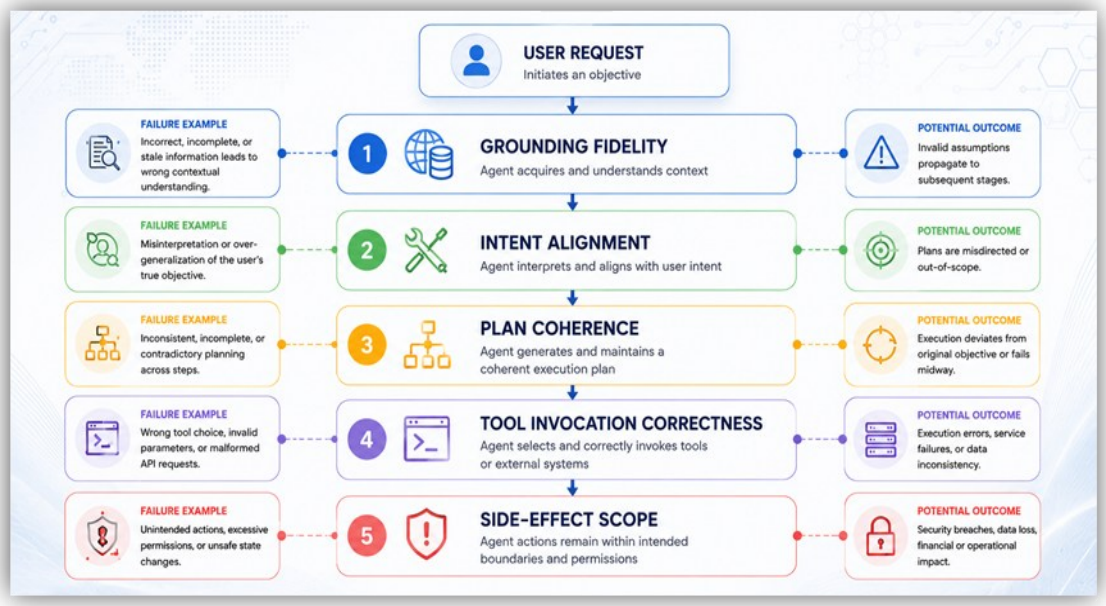

Figure 2. Conceptual flow of the proposed production LLM agent failure taxonomy
Source: Developed by the authors based on the synthesized literature

As shown in **Figure 2**, production failures often spread in a chain of events through the various operational levels. This means that problems that occur at the beginning of the process have a higher risk of problems later in the process.

4.6 Cross Dimensional Relationships

The proposed taxonomy takes into account the fact that production failures are not independent. Inadequate grounding can lead to misinterpreted intent, which leads to poor planning choices, followed by wrong tools being invoked and resulting in unintended operations. In the same way, small retrieval errors can be exacerbated by planning errors into major production errors. Recent studies on automated failure attribution, uncertainty aware planning, and distributed multi-agent reliability confirm that understanding the interactions between failure dimensions is key to ensuring operational resilience, as opposed to analyzing them individually (Zhang, S., et al., 2025; Zheng et al., 2024; Zhang, L., Zhai, et al., 2025).

5. DISCUSSION

5.1 Practical Implications

The proposed taxonomy offers a unified terminology that could enhance the reporting, monitoring, debugging, and governance of autonomous LLM deployments. Observed symptoms correlate with probable operational causes, which engineering teams can use to better prioritize root cause investigations and improve monitoring efforts. In addition, the taxonomy facilitates the development of structured reliability engineering practices for production AI, alongside taking account of new methods in AI-assisted diagnostics and intelligent operational management (Maheshkar, 2025; Chandrasekaran, 2025; Chen, X., et al., 2025). For the sake of demonstration only, there are representative engineering benefits summarized in **Table 5** for the taxonomy.

Table 5 Operational Benefits of the Proposed Taxonomy

Application Area	Expected Contribution
Incident management	Standardized failure reporting and diagnosis
Production monitoring	Earlier detection of operational anomalies
Governance	Improved accountability and risk management
System development	Better integration of mitigation strategies
Research	Consistent benchmarking and comparative evaluation

Source: Developed by the authors

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

The taxonomy continues beyond theoretical classification, and can provide practical guidance for production engineering, operational governance and continual system improvement, as shown in **Table 5.**

5.2 Research Implications

The taxonomy provides a shared framework for analyzing the reliability of production LLM agents in various fields from a research point of view. The current studies tend to focus on one capability at a time, while the proposed approach would include these aspects in a comprehensive view of operations. This standardization can help in conducting repeatable benchmarking, validation and comparison of reliability and enable the development of automated reliability measures in the future.

5.3 Limitations

The suggested taxonomy incorporates insights from the latest literature and production experiences, but LLM ecosystems remain dynamic and independent. This study doesn't account for all failure categories because emerging agent architectures, adaptive reasoning techniques, and new deployment scenarios could bring new forms of failure. Thus, in the future, empirical testing over large-scale industrial deployments will be necessary to further develop and extend the taxonomy and to quantify the evaluation of it as production AI systems evolve.

6. CONCLUSION

The rise of production LLM agents has revolutionized intelligent automation with their ability to reason, plan, invoke tools and make decisions in multiple steps for a wide range of app domains. These capabilities have also led to complicated operational failure modes, not covered by traditional software reliability models for deterministic systems. This study filled this gap by proposing a structured taxonomy of LLM agent failure that breaks down production LLM agent failures into five complementary dimensions: grounding fidelity, tool invocation correctness, plan coherence, intent alignment and scope of side-effects. The taxonomy provides a systematic mapping of failures in relation to the observable symptoms, root causes, detection signs, and mitigation measures of each failure type, creating a common conceptual understanding of the operational behavior of autonomous agents.

The proposed taxonomy adds to the existing work on the reliability and operational safety of LLMs, which includes studies in software reliability engineering, AI safety, agent collaboration, failure attribution, uncertainty aware planning, and production observability (Chen, X., et al., 2025; Cui et al., 2024; Gao et al., 2024; Jin et al., 2024; Maheshkar, 2025; Pan et al., 2023; Tran et al., 2025; Wang et al., 2025; Zhang, L., Jia, et al., 2024; Zhang, S., et al., 2025). The proposed framework offers an integrated operational view of the challenges and problems of reliability, which facilitates easier standardized reporting of incidents, root cause analysis, sharing of knowledge in different organizations, and more uniform evaluation of the behaviour of production LLM agents, as opposed to existing studies that mainly focus on individual reliability issues. The taxonomy thus has some practical use for researchers, developers and organizations looking to augment the robustness, transparency and governance of autonomous AI systems.

REFERENCE

1) Chandrasekaran, B. (2025). Enhancing Efficiency in Manufacturing through Automated AI Agent-Based Bearing Defect Diagnostic System (No. 2025-01-0158). SAE Technical Paper. https://doi.org/10.4271/2025-01-0158
2) Chen, X., Lei, Y., Li, Y., Parkinson, S., Li, X., Liu, J., Lu, F., Wang, H., Wang, Z., Yang, B., Ye, S., & Zhao, Z. (2025). Large Models for Machine Monitoring and Fault Diagnostics: Opportunities, Challenges, and Future Direction. Journal of Dynamics, Monitoring and Diagnostics, 4(2), 76–90. https://doi.org/10.37965/jdmd.2025.832
3) Chen, Z., Chen, J., Chen, J., & Sra, M. (2025). Standard Benchmarks Fail Auditing LLM Agents in Finance Must Prioritize Risk. arXiv preprint arXiv:2502.15865. https://arxiv.org/pdf/2502.15865v2
4) Cui, T., Wang, Y., Fu, C., Xiao, Y., Li, S., Deng, X., ... & Li, Q. (2024). Risk taxonomy, mitigation, and assessment benchmarks of large language model systems. arXiv preprint arXiv:2401.05778. https://arxiv.org/pdf/2401.05778
5) Gao, J., Gebreegziabher, S. A., Choo, K. T. W., Li, T. J. J., Perrault, S. T., & Malone, T. W. (2024, May). A taxonomy for human-llm interaction modes: An initial exploration. In Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (pp. 1-11). https://doi.org/10.1145/3613905.3650786

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
Peer-Reviewed | Open Access | International Journal
https://ijetrm.com/

6) Jin, H., Huang, L., Cai, H., Yan, J., Li, B., & Chen, H. (2024). From llms to llm-based agents for software engineering: A survey of current, challenges and future. arXiv preprint arXiv:2408.02479. https://arxiv.org/pdf/2408.02479
7) Kirk, H. R., Vidgen, B., Röttger, P., & Hale, S. A. (2023). Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback. arXiv preprint arXiv:2303.05453. https://arxiv.org/abs/2303.05453
8) Maheshkar, J. A. (2025). Bridging the Gap: A Systematic Framework for Agentic AI Root Cause Analysis in Hybrid Distributed Systems. Acta Scientiae, 26(1), 228-245. https://doi.org/10.13140/RG.2.2.32368.52485
9) Pan, L., Saxon, M., Xu, W., Nathani, D., Wang, X., & Wang, W. Y. (2023). Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. arXiv preprint arXiv:2308.03188. https://arxiv.org/abs/2308.03188
10) Pternea, M., Singh, P., Chakraborty, A., Oruganti, Y., Milletari, M., Bapat, S., & Jiang, K. (2024). The rl/llm taxonomy tree: Reviewing synergies between reinforcement learning and large language models. Journal of Artificial Intelligence Research, 80, 1525-1573. https://doi.org/10.1613/jair.1.15960
11) Rivkin, D., Hogan, F., Feriani, A., Konar, A., Sigal, A., Liu, S., & Dudek, G. (2023). Sage: smart home agent with grounded execution. arXiv preprint arXiv:2311.00772. https://arxiv.org/pdf/2311.00772
12) Tran, K. T., Dao, D., Nguyen, M. D., Pham, Q. V., O'Sullivan, B., & Nguyen, H. D. (2025). Multi-agent collaboration mechanisms: A survey of llms. arXiv preprint arXiv:2501.06322. https://arxiv.org/pdf/2501.06322
13) Wang, K., Zhang, G., Zhou, Z., Wu, J., Yu, M., Zhao, S., ... & Liu, Y. (2025). A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. arXiv preprint arXiv:2504.15585. https://arxiv.org/pdf/2504.15585
14) Weber, I. (2024). Large language models as software components: A taxonomy for llm-integrated applications. arXiv preprint arXiv:2406.10300. https://arxiv.org/pdf/2406.10300
15) Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., ... & Wang, C. (2023). Autogen: Enabling next-gen llm applications via multi-agent conversation. arXiv preprint arXiv:2308.08155. https://arxiv.org/abs/2308.08155
16) Xia, Y., Jazdi, N., & Weyrich, M. (2024, September). Enhance FMEA with large language models for assisted risk management in technical processes and products. In 2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-4). IEEE. https://doi.org/10.1109/ETFA61755.2024.10710996
17) Zhang, L., Jia, T., Jia, M., Wu, Y., Liu, A., Yang, Y., ... & Li, Y. (2024). A survey of aiops for failure management in the era of large language models. arXiv preprint arXiv:2406.11213. https://arxiv.org/pdf/2406.11213
18) Zhang, L., Zhai, Y., Jia, T., Huang, X., Duan, C., & Li, Y. (2025, June). Agentfm: Role-aware failure management for distributed databases with llm-driven multi-agents. In Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (pp. 525-529). https://dl.acm.org/doi/epdf/10.1145/3696630.3728492
19) Zhang, S., Yin, M., Zhang, J., Liu, J., Han, Z., Zhang, J., ... & Wu, Q. (2025). Which agent causes task failures and when? On automated failure attribution of llm multi-agent systems. arXiv preprint arXiv:2505.00212. https://doi.org/10.48550/arXiv.2505.00212
20) Zhang, Z., Cui, S., Lu, Y., Zhou, J., Yang, J., Wang, H., & Huang, M. (2024). Agent-safetybench: Evaluating the safety of llm agents. arXiv preprint arXiv:2412.14470. https://doi.org/10.48550/arXiv.2412.14470
21) Zheng, Z., Feng, Q., Li, H., Knoll, A., & Feng, J. (2024). Evaluating uncertainty-based failure detection for closed-loop LLM planners. arXiv preprint arXiv:2406.00430. https://arxiv.org/pdf/2406.00430

