

AI Engineering: A Disciplinary Program for the Reliable Delivery of Intelligent Tasks

—Capability Structure, Reliability Relations, and Uncertainty Governance

Author: Liu Ming

Version: V1.0

Date: August 2026

Abstract

Large language models and their Agent systems can already generate code, plan tasks, invoke tools, and modify real engineering states, yet growth in capability has not been matched by corresponding improvements in the reliable delivery of complex tasks. Existing research has addressed related problems from the perspectives of model capability, context, retrieval augmentation, Agent architecture, software engineering, evaluation, reliability, and safety assurance. What remains insufficiently organized across these lines of work is a common question: how can the reliable delivery of intelligent tasks serve as a unifying axis connecting capability structure, task–capability matching, process observation, and engineering intervention?

Building on three prior studies—KSC, SCR, and Multi-Round Audit Convergence (MRAC)—this paper proposes an initial disciplinary program for AI Engineering. It defines the core object of AI Engineering as the **conditions, mechanisms, methods, and infrastructure through which intelligent systems achieve reliable delivery in open, complex tasks under persistent constraints**. Within this initial foundation, KSC serves as an abstract framework for intelligent capability and describes the capability structure required by intelligent tasks; SCR serves as a relational theory of delivery reliability, explaining the relationship among effective search space, coverage ability, and delivery reliability, and deriving Search Space Governance as a major engineering direction; MRAC serves as a process-observation mechanism for coverage state, providing second-order information through multi-round audit–repair trajectories when no complete external verifier is available.

Around the reliable delivery of intelligent tasks, AI Engineering must address at least three foundational questions that cannot be directly bypassed: **where capability limitations arise; under what conditions limited capability can produce reliable delivery; and, when coverage boundaries cannot be measured directly and completely, how process evidence can be used to determine whether a system is operating within a range it can handle reliably**. KSC, SCR, and MRAC respectively provide a capability coordinate system, a reliability relation, and a process-observation mechanism.

AI Engineering identifies capability boundaries and governance effects through the continuous accumulation of evidence. When tasks approach or exceed coverage boundaries, reliability can be recovered through comparison, rollback, and decomposition. One important long-term objective is to gradually construct project-level closed loops that can

sustain complex-project delivery with low human intervention within clearly defined task domains, while continuously expanding their coverage boundaries—that is, to approach Engineering AGI [5].

Keywords: AI Engineering; KSC; SCR; Search Space Governance; Multi-Round Audit Convergence; MRAC; Delivery Reliability; Engineering AGI; Uncertainty Governance

1. Introduction

The fundamental change introduced by generative AI is not merely the addition of a new class of probabilistic components to software systems. Intelligent systems capable of invoking knowledge, generating structures, planning, and acting are beginning to participate directly in engineering activity. AI can already interpret requirements, select solutions, modify code, invoke tools, and run tests, allowing it to undertake parts of the cognitive work previously performed by engineers.

Yet local correctness, a reasonable plan, or passing the current test suite is not sufficient to ensure that a system can complete tasks reliably under complex state and persistent constraints. The question therefore shifts from “Can the model generate a correct result?” to “Can the system continue to complete complex tasks under conditions in which failures are discoverable and state is recoverable?” Recent research on Agent reliability has likewise begun to measure consistency, robustness, predictability, and safety separately from single accuracy metrics, indicating that improvements in capability or accuracy do not automatically imply corresponding improvements in reliability [14].

This raises a problem that cannot be fully addressed by model-capability research or existing engineering tools alone:

In the absence of a complete external verifier, when task constraints cannot be fully formalized and system state continues to change, how can we observe, intervene in, and progressively improve the delivery reliability of intelligent systems?

This paper treats that question as a core research line of AI Engineering and proposes an initial disciplinary program on the basis of KSC, SCR, and MRAC. The complete arguments for these foundational contributions are provided in their respective papers. Here, the focus is on their positions within the research program and on the engineering paradigm, evidence requirements, research agenda, and long-term objectives that follow from them.

2. The Object of Study and the Disciplinary Gap in AI Engineering

2.1 From Generative Capability to Delivery Reliability

The quality of a single response, a benchmark score, or a success rate measured over a limited set of tasks can demonstrate that a system possesses some capability, but it is not sufficient to show that the system can undertake real-world complex projects. Constraints in

complex tasks are typically distributed across Specs, code, documentation, dependencies, historical decisions, and runtime environments; intermediate results change subsequent state; and local correctness does not automatically imply global correctness. Software engineering has long studied complexity, maintenance, and system evolution [7]. Research on machine-learning systems has further shown that data dependencies, configuration, coupling, and technical debt outside the model itself can create persistent system-level engineering risks [8][9]. These lines of work indicate that local model capability or local task performance cannot substitute for system-level engineering governance.

It is therefore necessary to distinguish among **generative capability, the ability to complete an individual task, and the ability to sustain reliable delivery over time**. AI Engineering primarily studies the third: how engineering governance can transform AI's existing generative and task-execution capabilities into sustainable, reliable delivery. SWE-bench has already constructed repository-level software-engineering tasks from real GitHub issues and their corresponding pull requests [10]. Research on long-task capability has further attempted to characterize the scale of tasks AI can complete autonomously by using the time required by human experts to complete them, showing that task scale itself is an important dimension for understanding system capability boundaries [15]. However, moving from the completion of individual tasks to long-running project-level closed loops still requires more complete governance of state, constraints, dependencies, recovery, and evidence.

Delivery reliability differs from one-shot correctness and is not a static success rate detached from conditions. It concerns whether a system can continuously produce acceptable results under a given task distribution and set of engineering conditions, and whether it can discover, isolate, repair, and recover from failures. AI Engineering therefore studies not only whether a system can complete a task, but also the conditions under which reliability emerges, why it changes, and whether engineering interventions can improve it.

2.2 Criteria for Disciplinary Formation Adopted in This Paper

This paper does not attempt to define universal minimum conditions for the formation of every new discipline. Instead, it adopts a set of criteria suitable for this research program: a relatively independent object of study; foundational relations capable of organizing key phenomena; actionable targets of intervention; external evidence that can accumulate over time; and a research path that allows foundational claims to be tested, revised, or replaced.

Under these criteria, delivery uncertainty in intelligent tasks and its governance already constitute an object of study that can be organized independently. Task formulation, context, engineering state, boundaries, verification, and execution processes provide intervention targets; task conditions, execution trajectories, failures, and real-world feedback provide sources of evidence. Whether this research direction can mature into a stable disciplinary system will still depend on whether its foundational frameworks can explain key problems and withstand continuing empirical examination.

2.3 Name, Boundaries, and Current Research Scope

The term “AI Engineering” was not first proposed in this paper. In 2021, the Carnegie Mellon University Software Engineering Institute (SEI) explicitly described AI Engineering as an emerging discipline and organized related research and practice around human-centered, robust and secure, and scalable AI [13]. This paper neither claims priority over the name nor suggests that relevant research did not previously exist. Instead, it proposes a more explicit research program organized around the reliable delivery of complex intelligent tasks as a unifying axis:

AI Engineering studies the conditions, mechanisms, methods, evidence, and infrastructure through which intelligent systems reliably complete real-world complex tasks.

The boundary of AI Engineering is determined not by application domain but by problem structure. Whenever AI participates as a cognitive or acting agent in the delivery of a complex goal, the task involves multiple steps, evolving state, persistent constraints, incomplete verification, or failure-recovery requirements, and reliability needs to be improved through external engineering structures, the problem can fall within the scope of AI Engineering. Model training alone, one-shot generation, routine functional integration, or activities that cannot form an observable delivery process do not automatically belong to this research scope merely because they use AI.

At present, software engineering is the primary research entry point for AI Engineering. Code, Specs, tests, version control, diff comparison, and rollback provide a relatively complete external foundation for governance and evidence. Treating software engineering as the main object of study in the early stage is a consequence of current research conditions, not a limitation on the long-term boundary of the discipline. Whether scientific research, product design, management, and other complex goals should fall within this research scope must be examined separately according to whether their goals, states, constraints, processes, and evidence can be externalized and governed.

3. Knowledge Sources, Levels, and Open Incorporation

3.1 Existing Engineering Foundations and New Problems Introduced by AI

AI Engineering does not begin from zero. Design science, software engineering, systems and safety engineering, and AI risk management have already provided substantial bodies of knowledge that can be inherited [6][7][11][12], offering mature perspectives on complex design, software complexity, system safety, and risk governance.

The continued validity of this mature knowledge in AI systems does not automatically make it a new contribution to AI Engineering. New AI-engineering problems arise only when AI, as a generative, probabilistic, and partially autonomous engineering participant, materially changes the original object of study, relationship, or conditions of applicability. For example, managing complexity through decomposition and hierarchy is an established engineering idea [6][7]. By contrast, how task or module boundaries alter the

effective search space faced by AI—and thereby move a current system from a state in which it cannot reliably cover the task to one in which it can—is a question that must be re-examined in the AI setting.

3.2 Different Levels of Disciplinary Knowledge

AI Engineering contains foundational frameworks and theories, local mechanisms, engineering methods, tools, and empirical evidence. All of these may affect reliable delivery, but they play different roles and should not be treated as foundational knowledge at the same level.

For a study to become a candidate foundational component of the discipline, it should at minimum directly concern the reliable delivery of intelligent tasks, reveal a new abstraction, relation, or mechanism introduced by the AI setting, have relevance across implementations, and allow external evidence to revise its scope of applicability. Local mechanisms, methods, tools, and experience remain important even when they do not meet these conditions; they simply occupy different levels of the knowledge structure.

3.3 An Open and Revisable Initial Foundation

The initial foundation consists of three types of components: KSC as an abstract framework for intelligent capability, SCR as a relational theory of delivery reliability, and MRAC as a process-observation mechanism for coverage state. Search Space Governance is an engineering derivation of SCR; MRAC L1–L5 further examines the levels of justification, evidence boundaries, and research roadmap of MRAC [4].

This foundation remains open. External frameworks, theories, or mechanisms need not use the terminology of this paper; if they explain the reliable delivery of intelligent tasks more effectively, they should be included in comparative evaluation. Future research may supplement or replace the capability framework, revise the SCR relation, or propose other foundational relations, engineering directions, or observation mechanisms. AI Engineering should remain centered on its object of study rather than restricting its sources of knowledge by author, terminology, or a particular technical route.

4. Three Components of the Initial Foundation

This section provides only the definitions and disciplinary positions of the three components that are necessary for the present program. Their detailed derivations, boundaries, and empirical propositions are developed in the corresponding independent papers.

4.1 KSC: An Abstract Framework for Intelligent Capability

KSC describes intelligent tasks through three progressively related capabilities [2]:

- **K (Knowledge):** information, experience, rules, facts, and representations that the system can effectively invoke in a task;

- **S (Structural Generation):** the capability to form reasoning, explanations, plans, designs, code, or other new structures from existing knowledge;
- **C (Constraint Maintenance):** the capability to keep structural generation subject to goals, boundaries, state, and constraints over time.

To avoid confusion with Coverage Ability in SCR, this paper uses **KSC-C** when a letter abbreviation is needed for Constraint Maintenance, and **SCR-C** for Coverage Ability in SCR.

The three can be understood as the existence of information, the formation of new structures from information, and control over structural generation. With K but without S, information cannot contribute to generation. With K and S but insufficient Constraint Maintenance (KSC-C), a system may produce many locally valid structures without ensuring that those structures remain appropriate to the current task. Constraint Maintenance does not refer only to memory over long contexts; it also includes goal selection, relevance judgment, boundary control, constraint coordination, error identification, and stopping conditions.

KSC is an abstract framework and coordinate system for intelligent capability, not a causal theory or operational guide, and it does not require one-to-one correspondence with internal model modules. It decomposes the vague statement that “AI capability is insufficient” into three points of analysis: Knowledge, Structural Generation, and Constraint Maintenance. Under the KSC analysis, AI has already demonstrated relatively strong capabilities in knowledge invocation and structural generation, while Constraint Maintenance in complex tasks still often needs to be supplemented by humans or external engineering structures [2]. This provides an important basis for human–AI division of labor and AI engineering governance. KSC describes the capabilities on which intelligent tasks depend, but it cannot by itself determine whether a given task can be delivered reliably; that also depends on task scale, engineering state, and governance conditions.

4.2 SCR: A Relational Theory of Delivery Reliability

SCR describes a core engineering relationship in AI tasks through three variables [3]:

- **Effective Search Space (Search Space, S):** the set of relevant interpretations, decisions, paths, implementations, and failure possibilities that a system must confront under a given task formulation, engineering state, and governance conditions;
- **Coverage Ability (C; denoted SCR-C in this paper):** the combined ability of the model, Agent, tools, context, and verification mechanisms to explore, eliminate, verify, and repair the relevant space;
- **Delivery Reliability (R):** the ability of a system to continuously produce acceptable deliverables and handle failures under a given task distribution and set of engineering conditions.

The effective search space is not the context window and does not need to be fully enumerable. Coverage Ability (SCR-C) is likewise not a fixed global property of a model, but a system property relative to a specific task. SCR first expresses a directional relationship: holding other conditions approximately constant, expansion of the effective search space generally tends to reduce delivery reliability, while stronger coverage ability generally

tends to increase it. Engineering systems can improve reliability either by strengthening coverage ability or by shrinking, partitioning, making the effective search space explicit, or stabilizing it so that existing capability can complete the task more reliably.

SCR explains why a system may possess relevant capabilities and still fail to deliver, and why the same model can perform differently under different engineering states. It connects capability research to engineering outcomes and derives Search Space Governance as a major engineering direction. KSC and SCR operate at different analytical levels: KSC provides a capability abstraction, whereas SCR explains the relationship among task conditions, system capability, and delivery outcomes.

4.3 MRAC: A Process-Observation Mechanism for Coverage State

Many complex tasks lack a complete verifier. A result that passes tests only demonstrates that it has passed the current tests; a single audit that finds no issue does not prove the absence of shared blind spots. AI Engineering therefore also needs to use changes across repeated inspection, repair, and re-execution to obtain second-order information that a single result cannot provide.

MRAC proposes such a process-observation mechanism: through repeated audit–repair rounds, it observes whether newly identified valid issues tend to decrease, whether repaired issues recur, and whether issue discovery gradually stabilizes [1]. When the task object, key constraints, and audit protocol remain stable, the audit findings are valid, and the process has not drifted away from the target, the continued appearance of high-impact issues strongly indicates that the current system still cannot achieve reliable coverage within that task boundary. Convergence provides supporting evidence stronger than a single success, but it cannot be directly equated with correctness or complete coverage.

MRAC does not replace testing, Eval, or real-world operational feedback. Instead, it turns multi-round issue trajectories into evidence for whether to continue, stop, roll back, or decompose the task again. Its non-convergence diagnosis, convergence evidence, convergence after decomposition, and recursive research roadmap are developed further in the MRAC L1–L5 study [4].

5. Core Engineering Paradigm: Search Space Governance

5.1 From Capability Competition to Task–Capability Matching

Once generative AI enters engineering, engineering systems must govern not only deterministic programs but also the many possible paths intelligent agents confront in semantic interpretation, solution selection, state modification, and verification.

Humans cannot realistically specify every internal path of AI step by step, nor should AI be reduced to a tool capable only of executing fully deterministic instructions. A more feasible direction is to govern external conditions: make viable paths easier to discover, eliminate unproductive paths earlier, preserve critical constraints over time, and identify and reorganize tasks when they exceed current coverage ability.

This paper refers to this class of activity as **Search Space Governance**. It is not a fourth foundational contribution parallel to the three components above, but a direct engineering derivation of SCR as the framework moves from relational explanation to intervention.

5.2 Governance Targets and Two Major Engineering Paths

Search Space Governance primarily addresses four classes of objects. Task governance covers goals, scope, constraints, and acceptance. State governance covers code, documentation, data, environment, dependencies, and historical decisions. Process governance covers task decomposition, ordering, checkpoints, rollback, and human intervention. Evidence governance covers tests, audits, trajectories, diffs, and failure classification.

Delivery reliability can be improved along two major engineering paths: strengthening coverage ability or governing the effective search space. Model scaling, reasoning, tool use, and Agent improvements primarily expand the range that the system can handle. Specs, modularization, engineering-state governance, task decomposition, and rollback primarily constrain or restructure the range the system needs to handle. A specific method may affect both sides; its actual effect should be judged from task conditions and external evidence rather than fixed as a one-to-one mapping to a variable.

This classification is intended to organize engineering interventions for reliable delivery. Safety, responsibility, authorization, cost, and other concerns may remain independent governance objectives; their relationships with search space and coverage ability require further study.

5.3 Engineering Governance Under Uncertainty

AI Engineering does not require the elimination of all uncertainty. Instead, it seeks to keep uncertainty within a governable range. Critical constraints need to be maintained, failures need to be discoverable, and tasks beyond current coverage ability need to be decomposable, rolled back, or handed off. Reliable delivery capabilities that have already been established should also be preserved and improved as engineering state and governance experience accumulate.

Long-term engineering reliability does not arise because every AI action becomes deterministic. It arises under a different condition: as project complexity grows, the effective search space faced by new tasks remains controllable, and failures remain within engineering boundaries in which they can be discovered and recovered from. This constitutes the independent engineering value of AI Engineering relative to an approach focused solely on model capability.

6. An Evidence-Driven AI Engineering Process

6.1 Why Evidence Has a Foundational Role

Neither effective search space nor coverage ability can currently be measured completely. A single benchmark, one successful run, or one static capability conclusion is therefore insufficient to determine the reliable operating boundary of a system.

AI Engineering cannot wait until every variable has been completely formalized before research begins. A more realistic path is to establish directional relationships first, then progressively identify their conditions of validity, strength of effect, and failure boundaries through evidence accumulated across tasks, systems, and long-running projects. Evidence is not merely an additional validation step after foundational claims are proposed; it is a basic mechanism through which disciplinary knowledge continues to develop.

6.2 From Single-Path Execution to Comparable Processes

When a task approaches or exceeds the coverage boundary of the current system and no complete verifier exists, a complex task may not be completed reliably along a single predetermined path. The system may need to compare candidate paths, reject solutions with insufficient support, roll back erroneous states, or decompose the task again so that task boundaries once more match current coverage ability.

These actions do more than handle failure; they also produce new engineering information. Eliminating a path indicates that, under current conditions, there is insufficient basis for continuing to pursue it. A rollback indicates that the current state is not worth preserving. Renewed convergence after task decomposition supports the judgment that the original task boundary was mismatched with coverage ability. AI engineering systems oriented toward reliable delivery therefore need not only to enable Agents to execute, but also to make candidate paths comparable and rejectable, erroneous states recoverable through rollback, and validated results preservable.

6.3 Evidence Directions and Current Boundaries

AI engineering evidence needs to answer four basic classes of questions: whether changes in reliability are stably associated with a particular condition or governance intervention; within what boundaries a system or method remains effective; whether a change should be attributed to the model, Agent, task, or governance; and whether a conclusion transfers across tasks, projects, and teams. Failures, non-convergence, path elimination, and real defects often reveal capability boundaries more clearly than a single success.

Continuously testing engineering judgments against evidence is a basic research direction of AI Engineering, not a completed unified standard. This paper does not presuppose an evidence Schema, a composite score, or a fixed protocol. Specific metrics and infrastructure should develop progressively from real projects and long-term evidence.

7. The Structural Necessity of the Foundational Questions

This section concerns the structural necessity of the problem positions themselves. Once AI Engineering takes the reliable delivery of intelligent tasks as its research objective, at least three consecutive foundational questions arise naturally and cannot be directly bypassed. This section does not claim that any specific framework is irreplaceable, nor does it claim that the set of foundational questions is complete.

7.1 Deriving at Least Three Necessary Questions from the Goal of Reliable Delivery

AI Engineering centers on the reliable delivery of complex tasks by intelligent systems. This objective itself constrains what foundational research must answer; the disciplinary questions are not reverse-engineered from existing theories.

The first is the **capability-boundary problem: what capabilities an intelligent system possesses, where its limitations arise, and which limitations need to be supplemented by humans or external systems**. Engineering governance is necessary in the first place because intelligent systems have finite capability, and their capabilities are not uniformly strong. Without a stable way to describe capability structure, it is difficult to explain why AI fails in certain tasks or to determine what human–AI division of labor and engineering governance need to supplement.

Capability description alone, however, cannot imply reliable delivery. This produces the second question: **under given capability conditions, what kinds of task and engineering conditions can produce reliable delivery?** The same system may exhibit substantially different delivery reliability under different task scales, effective search spaces, engineering states, and governance methods. Knowing only “what capabilities the system has” is therefore insufficient; it is also necessary to explain under what conditions finite capability is enough to complete a task reliably.

Even after such a reliability relation has been established, engineering practice still faces a third question: **when coverage boundaries cannot be measured directly and completely, how can we determine whether the system is currently operating within a range it can handle reliably?** Many real-world complex tasks lack complete verifiers, while effective search space and coverage state are also difficult to measure directly. Additional evidence must therefore be obtained from execution, verification, and repair processes.

The three questions form a continuous progression: **capability description cannot directly imply reliability, so the conditions for reliable delivery must be studied; the reliability relation, in turn, cannot be observed directly and completely, so process observation is required**. This logic can be summarized as:

Finite capability → conditional reliability → incomplete observability

Capability boundaries, conditions for reliable delivery, and observation of coverage state are therefore not three categories artificially constructed to accommodate existing theories. They are at least three necessary questions that emerge step by step from the disciplinary goal of “reliable delivery.”

7.2 Positions of the Current Foundational Components

Foundational Question	Current Foundational Component	Current Answer
How should capability boundaries be described?	KSC	Provides a capability coordinate system, locating limitations in Knowledge, Structural Generation, and Constraint Maintenance and clarifying human-AI complementarity
When can finite capability produce reliable delivery?	SCR	Explains the relationship among effective search space, coverage ability, and delivery reliability
How can coverage state be judged?	MRAC	Uses multi-round audit-repair trajectories to obtain second-order evidence about coverage state

Within this initial foundation, KSC, SCR, and MRAC correspond respectively to capability boundaries, conditions for reliability, and engineering observation. Together they form a continuous structure from capability description, to reliable delivery, to process judgment.

7.3 What Is Necessary Is the Problem Position, Not a Specific Theory

The disciplinary objective constrains the questions that must be answered, not the answers that must be adopted. Future research may propose a capability framework more effective than KSC, revise or replace the SCR relational theory, or develop coverage-state observation mechanisms more effective than MRAC. As long as AI Engineering continues to study the reliable delivery of finite intelligent systems, capability limitations, conditions for reliable delivery, and observation of coverage state will still need to be answered in some form.

KSC, SCR, and MRAC come from the same line of research and have strong structural consistency; for that reason, they also require more independent research for comparison and examination. Whether future foundational contributions should be incorporated into AI Engineering should not depend on whether they inherit the terminology of this paper, but on whether they answer the foundational questions more effectively, explain reliable delivery better, and remain subject to continuous examination by external evidence.

8. Research Agenda and Engineering AGI

8.1 Initial Knowledge Domains

The initial knowledge domains of AI Engineering include at least: capability structure and system boundaries; tasks, constraints, and effective search space; governance of knowledge, context, and engineering state; module boundaries, task decomposition, and dependency orchestration; testing, Eval, audit, and real-world operational observation; roll-back, recovery, and human-AI responsibility boundaries; delivery reliability and reproducible evidence; and project-level closed loops and Engineering AGI.

These domains are not a final curriculum. They constitute an initial scope for organizing research around reliable delivery, allowing work on models, Agents, software engineering, evaluation, and organizational governance to collaborate around a common class of problems.

8.2 Core Research Directions

Foundational research first needs to test the applicability of KSC, SCR, and MRAC and compare them with alternative frameworks, theories, and observation mechanisms. Engineering research needs to identify the conditions under which methods such as Specs, context, RAG, engineering-state governance, modularization, task decomposition, and roll-back improve reliability. Infrastructure research should support the long-term accumulation of task conditions, engineering states, execution trajectories, failures, and delivery outcomes.

Empirical research on SCR must take particular care to avoid definitional circularity: effective search space, coverage ability, and delivery reliability should, as far as possible, be operationalized using mutually independent proxy metrics, intervention variables, or experimental designs before the relationships among them are tested. MRAC requires continuing research into how audit protocols, model capability, shared blind spots, repair drift, and external validation affect the interpretive strength of convergence.

The research focus should expand from “whether the system has ever done the task correctly” to “under what conditions the system can do it reliably,” and should progressively produce evidence that is comparable across tasks, systems, and projects.

8.3 From Reliable Delivery to Engineering AGI

The near-term objective of AI Engineering is to improve the delivery reliability of individual intelligent tasks under explicit conditions. A further objective is to extend that reliability across tasks and stages, and throughout continuously evolving project-level closed loops. This paper adopts the working definition of Engineering AGI proposed in prior work [5]: **one important long-term objective** is for AI, with low human intervention, to continuously sustain the closed-loop execution of complex projects that would otherwise require long-term collaboration by an ordinary human team, thereby approaching Engineering AGI.

Engineering AGI is better understood as a continuously expanding coverage boundary than as a single endpoint describable only as “achieved” or “not achieved” [5]. A system may first establish low-human-intervention project-level closed loops in bounded, clearly defined task domains, and then progressively expand the types of tasks, project scale, duration, and openness of the environment it can handle. Establishing such a closed loop in a restricted task domain can be regarded as a local approximation to Engineering AGI; it does not mean that AGI in the general sense has already been achieved.

After a closed loop has been established, engineering research must still improve reliability, reduce unproductive paths, repeated audits, and rollbacks, lower time, compute, and organizational costs, and transfer closed-loop capability to a broader range of problems.

KSC describes the capability structure required by the closed loop, SCR describes the task boundary that can be reliably covered, and MRAC provides process signals for discovering and expanding that boundary. Here, low human intervention or unattended operation means that project execution no longer depends on continuous human participation; it does not exclude humans from continuing to govern goals, authorization, responsibility, and high-risk boundaries.

9. Adjacent Disciplines, Limitations, and Directions for Development

9.1 Relationship with Adjacent Disciplines

AI Engineering does not replace machine learning, software engineering, systems engineering, MLOps, Agent research, AI safety, human–AI collaboration, reliability research, or related fields. Instead, it organizes a shared question on top of these bodies of work: how can existing capabilities and engineering methods produce reliable delivery in real-world complex tasks?

AI Engineering is therefore better understood as a cross-disciplinary research layer connecting these fields, rather than being reduced to “how to use current AI coding tools.” Tools will change rapidly; task search spaces, coverage boundaries, Constraint Maintenance, failure discovery, and reliable delivery are more stable objects of study. Adjacent work on Agent reliability and long-task capability can also serve as evidence sources for this framework [14][15].

9.2 Current Limitations

This paper remains at the stage of proposing a disciplinary program rather than presenting a mature discipline, and it has the following limitations:

1. Effective search space is difficult to enumerate or measure precisely, while coverage ability depends heavily on the task, engineering state, and protocol and should not be compressed into a single global metric;
2. The effective search space, coverage ability, and delivery reliability in SCR still require mutually independent proxy metrics and intervention designs to avoid circular explanation;
3. KSC, SCR, and MRAC mainly originate from the same research line and require more independent research to compare their components, relations, and boundaries of applicability;
4. Existing experience comes primarily from AI software development, and other domains may not be able to reuse the same governance methods directly;
5. MRAC and other process-observation mechanisms depend on verification capability, and shared blind spots among verifiers may create false stability.

9.3 Cross-Domain Extension

Cross-domain extension does not depend on the name of the domain. It depends on whether goals, states, constraints, processes, and evidence can form governable external structures. As independent evidence accumulates, related research can gradually expand into scientific research, management, and other open-ended intelligent tasks, while testing whether domains beyond software engineering involve different boundaries of verification, value, and responsibility.

10. Conclusion

AI Engineering treats delivery uncertainty in intelligent tasks and its governance as an independent object of study. It examines how finite intelligent capability can be transformed into reliable outcomes under complex state, persistent constraints, and incomplete verification.

Around this objective, at least three foundational questions arise that cannot be directly bypassed: how capability boundaries should be described; under what conditions finite capability can produce reliable delivery; and how process evidence can be obtained when coverage state cannot be measured directly. KSC, SCR, and MRAC respectively provide a capability framework, a reliability relation, and a process-observation mechanism. Search Space Governance further translates the reliability relation into engineering intervention, making the governance of tasks, state, and process a second major path alongside the enhancement of model capability.

This remains an initial foundation that requires continuous testing and revision through evidence across models, Agents, projects, and teams. Its long-term direction is to extend reliable delivery from individual tasks to sustained project-level closed loops, progressively expanding the range in which AI can deliver complex projects with low human intervention and continuously approaching Engineering AGI.

References

Note: References [1]–[5] are prior studies by the author of this paper; [5] provides the working definition of Engineering AGI adopted here. The remaining references are included to establish relevant engineering traditions, the existing AI Engineering context, and the research background on Agent reliability and long-task capability.

- [1] Liu, M. (2026a). *Multi-Round Audit Convergence (MRAC)* [Preprint]. Zenodo. doi:10.5281/zenodo.21804751
- [2] Liu, M. (2026b). *KSC: A Minimal Framework for Understanding AI Capability and Human–AI Collaboration* [Preprint]. Zenodo. doi:10.5281/zenodo.21834579

- [3] Liu, M. (2026c). *Search Space Governance in AI Engineering* [Preprint]. Zenodo. doi:10.5281/zenodo.21902420
- [4] Liu, M. (2026d). *MRAC from L1 to L5: Evidence Boundaries and Research Roadmap* [Preprint]. Zenodo. doi:10.5281/zenodo.21917629
- [5] Liu, M. (2026e). *An Engineering Definition of AGI: Complex-Project Closed-Loop Capability as an Assessment Criterion* [Preprint]. Zenodo. doi:10.5281/zenodo.21927623
- [6] Simon, H. A. (1969). *The Sciences of the Artificial*. MIT Press.
- [7] Brooks, F. P., Jr. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *Computer*, 20(4), 10–19. doi:10.1109/MC.1987.1663532
- [8] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems*, 28.
- [9] Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H. C., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (pp. 291–300). doi:10.1109/ICSE-SEIP.2019.00042
- [10] Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., & Narasimhan, K. (2024). SWE-bench: Can Language Models Resolve Real-World GitHub Issues? In *The Twelfth International Conference on Learning Representations (ICLR 2024)*.
- [11] Leveson, N. G. (2012). *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press.
- [12] National Institute of Standards and Technology. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. doi:10.6028/NIST.AI.100-1
- [13] Software Engineering Institute. (2021). *AI Engineering: An Emergent Discipline for Human-Centered, Robust and Secure, and Scalable AI* [Brochure]. Carnegie Mellon University.
- [14] Rabanser, S., Kapoor, S., Kirgis, P., Liu, K., Utpala, S., & Narayanan, A. (2026). *Towards a Science of AI Agent Reliability*. Proceedings of the 43rd International Conference on Machine Learning (ICML 2026). arXiv:2602.16666.
- [15] Kwa, T., West, B., Becker, J., et al. (2025). *Measuring AI Ability to Complete Long Software Tasks*. Advances in Neural Information Processing Systems (NeurIPS 2025). arXiv:2503.14499.

Appendix A: Core Terminology

Term	Notation	Meaning
AI Engineering	—	The study of the conditions, mechanisms, methods, evidence, and infrastructure through which intelligent systems reliably complete real-world complex tasks
Knowledge	KSC-K	Information and experience that a system can effectively invoke in a task
Structural Generation	KSC-S	The capability to form new structures such as reasoning, plans, designs, and code from knowledge
Constraint Maintenance	KSC-C	The capability to keep generation subject to goals, boundaries, state, and constraints over time
Effective Search Space	SCR-S	The relevant interpretations, paths, implementations, and failure possibilities that must be confronted under the current task and engineering conditions
Coverage Ability	SCR-C	The combined ability of models, Agents, tools, and governance systems to explore, eliminate, verify, and repair the relevant space
Delivery Reliability	SCR-R	The ability to continuously produce acceptable deliverables and handle failures under a given task distribution and set of engineering conditions
Search Space Governance	—	Intervention in tasks, state, process, and evidence so that search space and coverage ability are brought back into alignment
Multi-Round Audit Convergence	MRAC	A mechanism for observing whether issue discovery stabilizes through multi-round audit-repair trajectories
Second-order Observation	—	State information obtained from changes across repeated verification, repair, or execution that cannot be provided by a single result
Engineering AGI	—	AI continuously sustaining the closed-loop delivery of complex projects, with low human intervention, that would otherwise require long-term collaboration by an ordinary human team; establishing such a closed loop in a restricted task domain can be regarded as a local approximation to Engineering AGI