

Whitepaper | Pre-Runtime Diagnostics for Agent Specifications

Structured Evaluation Before Deployment with a Neuro-Symbolic Agent Optimization System (NAOS)

[Structured Evaluation Before Deployment with a Neuro-Symbolic Agent Optimization System \(NAOS\)](#)

[Summary \[TL;DR\]](#)

[Opportunity, Problem, and Context](#)

[Method Framing](#)

[Posture Shift](#)

[What NAOS Changes](#)

[What the Method Looks At](#)

[What the Method Produces](#)

[Why This Layer Matters](#)

[Evaluator Design Surface](#)

[Design Objective](#)

[Evaluation Object](#)

[Normalization](#)

[Structural Gating](#)

[Decision-Relevant Dimensions](#)

[Comparative Review](#)

[Bounded Scope](#)

[Output and Decision Surface](#)

[Proof Section](#)

[Proof Burden](#)

[Why Comparison Matters](#)

[What Counts as Value](#)

[Why This Standard Is Enough](#)

[Bounded Validation Evidence](#)

[Summary](#)

[Validation Question](#)

[Compared Artifacts](#)

[Run Design](#)

[What Counted as Success](#)

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)[Metric Surface](#)[Core Result](#)[What Did Not Separate the Specimens](#)[Secondary Corroboration](#)[What This Shows](#)[Future Work](#)[Product Consequence and Operational Value](#)[Limits, Scope, and Disclosure Boundary](#)[Conclusion](#)[Evidence Summary Table](#)[References](#)

Summary [TL;DR]

NAOS addresses a problem that becomes more expensive the longer it is ignored: many meaningful agent failures are already present in the instruction or specification artifact before deployment begins. If those weaknesses are only discovered at runtime, review gets noisier, correction gets slower, and release decisions are made with less visibility than they should be.

This paper shows that a pre-runtime evaluation product can do useful work before any live deployment begins. In a blind A/B deterministic validation, NAOS compared two candidate pre-runtime specification artifacts under the same seven-phase schedule, with three full replays per specimen, and both specimens reached the **Pass Condition [Verdict ID: VA-2M7]**. Even so, the comparison did not collapse into equivalence. Specimen B remained materially stronger across the decisive measures in every phase. **The clearest quantitative result was that Specimen B recorded Stability Burden [Metric ID: LM-4N8] values that were 16.67% to 30.00% lower than Specimen A's across the seven phases.** That advantage was reinforced by **2.22% to 3.49% higher Coherence [Metric ID: HV-2P6]** and **1.06% to 2.22% higher Alignment [Metric ID: ZK-7Q2]**.

That result matters because it shows what NAOS is actually for. The product is not trying to replace every downstream runtime test or claim complete predictive coverage over agent behavior. Its job is narrower and more practical: inspect a structured specification artifact before deployment, surface meaningful differences between candidate artifacts, and produce a selection signal strong enough to change what a builder, reviewer, or product lead does next.

NAOS follows a straightforward evaluation logic. It inspects a structured agent specification, applies normalization and gate checks, evaluates consistency, constraint quality, and deployment risk, and returns structured verdicts that support release, revision, comparison, rebuild, or withholding decisions. The value of that process rises with specification richness, revision sensitivity, and operational consequence. Where an agent carries real decision weight, earlier diagnosis is not a luxury. It is a better operating discipline.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

The contribution of this paper is therefore concrete. It argues that pre-runtime diagnostics should be treated as a real decision layer in agent development, and it supports that claim with a bounded validation showing that NAOS can distinguish between admissible candidate artifacts under controlled conditions in a way that remains meaningful when the test becomes more demanding. For teams deciding which specifications are stronger, more stable, and more ready to move forward, that is already substantial value.

Opportunity, Problem, and Context

A meaningful opportunity is opening in agent evaluation before deployment. As agent systems are increasingly defined through structured instructions, role logic, constraints, revision layers, and operating assumptions, the specification artifact itself becomes a serious object of review. That matters because the artifact is no longer just a setup layer. It increasingly carries behavior, constraint, and failure potential of its own. When that shift is taken seriously, earlier diagnosis becomes possible.

The problem is that many important weaknesses are still discovered too late. Runtime instability, incoherence, degradation, and poor deployment judgment often do not begin at runtime. They are already latent in the artifact that defined the agent in the first place. Weak constraint structure, hidden contradiction, brittle revision logic, and unstable operating assumptions can all exist before a live run begins. When those issues are found only after deployment, organizations pay in runtime cost, review burden, slower diagnosis, noisier correction, and lower confidence in the next release decision.

That problem is becoming more acute because agent systems are being asked to carry more decision weight and to support more consequential human decisions. The operational cost of late discovery rises with that shift. Teams are asked to approve more complex agent definitions, revise them more frequently, and make go or no-go decisions under tighter timelines. In that environment, runtime alone is an incomplete review surface. It shows what happened, but it often shows it only after cost has already been incurred.

The opportunity is not limited to only the most advanced agents. It exists wherever an agent has a meaningful pre-runtime specification and a poor deployment decision carries real cost. Its value rises with specification richness, revision sensitivity, and operational consequence. For thin systems with minimal structure, the gain may be modest. For richer agent definitions that carry real behavioral and operational load, the gain can be substantial.

That is why pre-runtime diagnostics matters now. Treating the specification artifact as an evaluable object creates an earlier decision layer for review, revision, and deployment judgment. It does not replace runtime evidence, and it does not eliminate uncertainty. It does make important weaknesses more visible before live execution begins. In a world where agent behavior is increasingly shaped before deployment, that earlier visibility becomes both a practical opportunity and an operational necessity.

Method Framing

Posture Shift

Pre-runtime diagnostics begins with a simple change in posture. The agent specification is treated as an evaluable object before release rather than as a setup artifact judged only after execution. The goal is not to wait for behavior to fail in the wild and then work backward. The goal is to inspect the structure most likely to shape behavior before live exposure turns diagnosis into cleanup.

What NAOS Changes

A Neuro-Symbolic Agent Optimization System, or NAOS, operationalizes that shift. Here, neuro-symbolic refers to a practical evaluation method that combines model-facing analysis with explicit decision criteria and structured outputs. The point is not to produce free-form commentary about an agent definition. The point is to apply a disciplined review method that can surface weakness, preserve comparison across candidates, and support real decisions before deployment begins.

What the Method Looks At

The object under review is a structured agent specification. Depending on the system, that artifact may include role definitions, governing constraints, operating assumptions, behavioral instructions, revision layers, and deployment intent. Taken together, those elements form more than a prompt. They form a pre-runtime control surface.

That matters because the artifact is already carrying behavioral logic, constraint logic, and failure potential before any live execution begins. If those forces are already present upstream, they should be inspected upstream.

What the Method Produces

NAOS does not treat pre-runtime review as a loose interpretive exercise. It produces structured, decision-oriented outputs. The point is not to score an artifact in the abstract. The point is to determine whether a candidate specification should move forward, be revised, be compared against alternatives, or be withheld from deployment.

That is what gives the method its practical force. A useful pre-runtime review surface does not just describe. It sharpens judgment.

Why This Layer Matters

The value of this approach comes from discipline at the right layer. It does not depend on exposing internal implementation detail, and it does not require a claim to predict every

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

downstream behavior in advance. Its claim is narrower and more useful than that: before a system is released, the specification shaping its behavior can be inspected seriously enough to improve judgment.

That is the role NAOS serves. It turns pre-runtime review from an informal reading exercise into a structured decision layer that can matter before deployment begins.

Evaluator Design Surface

Design Objective

NAOS is built for a narrow but consequential job: inspect a structured agent specification before deployment and return a verdict that can change a real release decision. It is not trying to solve the whole problem of agent evaluation at once. It is focused on the artifact that defines the agent before runtime, because that is where many important failures already begin.

That focus is the point. A pre-runtime evaluator becomes useful when it can surface weakness early enough to change what happens next.

Evaluation Object

The object under review is a pre-runtime specification artifact. In practice, that artifact may contain role definitions, governing constraints, behavioral instructions, operating assumptions, revision layers, and deployment intent. NAOS treats those elements as one structured object, not as disconnected fragments.

That matters because the artifact is not just setup. It is already carrying behavior, control logic, and failure potential before any live execution begins.

Normalization

The first requirement is normalization. Specifications do not arrive in a stable form by default. They vary in density, consistency, clarity, and internal structure. A serious evaluator cannot depend on informal reading or ad hoc interpretation. It has to bring the artifact into a form that can be inspected consistently across candidates, revisions, and deployment contexts.

Normalization creates that foundation. It reduces format drift, removes avoidable inconsistency, and makes later comparison more trustworthy.

Structural Gating

The second requirement is structural gating. Not every specification is ready for deeper evaluation. Some artifacts are incomplete, fragmented, weakly constrained, or unstable enough that a confident verdict would be false precision.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

Structural gate checks separate artifacts that need repair at the structural level from artifacts that are ready for deeper review. That discipline matters because it prevents the evaluator from sounding confident about an object that is not yet coherent enough to judge seriously.

Decision-Relevant Dimensions

Once an artifact passes that threshold, NAOS examines bounded dimensions that matter before deployment. These dimensions are not arbitrary. They are chosen because they affect whether the specification is likely to hold together under use, revision, and operational pressure.

Coherence matters because internal contradiction weakens downstream behavior. Constraint quality matters because weak or poorly aligned constraints produce unstable control. Revision sensitivity matters because specification changes often degrade quality even when they appear locally reasonable. Deployment risk matters because a specification can look complete on paper while still carrying signs that it should not move forward without revision.

The evaluator is therefore not just describing the artifact. It is testing whether the artifact is structured well enough, stable enough, and aligned enough to justify confidence.

Comparative Review

NAOS is comparative by design. A specification is rarely judged in a vacuum. Teams compare revisions, alternatives, and competing candidate definitions before release. A useful evaluator should make those comparisons more disciplined by holding each artifact inside the same review frame and exposing the differences that matter for judgment.

That is where the method starts doing real work. If competing candidates still look interchangeable after inspection, the evaluator has not earned much trust. If meaningful differences become visible early enough to affect selection, revision, or withholding decisions, then the evaluator is doing something operationally valuable.

Bounded Scope

NAOS remains deliberately bounded at this layer. It does not replace runtime evidence, and it does not claim to infer every downstream behavior from the specification alone. Its claim is narrower and stronger than that. If the specification artifact is already shaping behavior before deployment, then that artifact deserves a serious review surface before deployment.

That is the design logic. NAOS is built to make pre-runtime judgment more structured, more comparable, and more operationally useful than current practice usually allows. The proof burden for that claim is not abstract. It is whether the evaluator can inspect competing artifacts under the same frame, surface differences that matter, and support a real selection decision. The later proof and validation sections show that burden carried concretely.

Output and Decision Surface

A pre-runtime evaluator is only useful if its outputs support judgment. The value of inspection does not come from diagnosis alone. It comes from turning diagnosis into a form that can guide action before deployment begins. For that reason, the output surface must be structured, comparative when needed, and clear enough to support real release decisions.

The core output is a verdict packet. That packet does not exist to summarize the artifact in general terms. It exists to show whether the specification is fit to move forward, where its weaknesses are concentrated, and what kind of action the current state of the artifact warrants. A useful verdict is therefore more than a score. It is a decision-facing interpretation of the artifact's condition under bounded review dimensions.

At minimum, the output must make four things legible. It must show whether the specification is internally coherent, whether its constraints are sufficiently aligned to its intended role, whether revisions have introduced instability or degradation, and whether the artifact carries enough deployment risk to affect release judgment. Those signals do not eliminate uncertainty, but they make uncertainty more actionable. They turn vague concern into a reviewable basis for decision.

That matters because release decisions are rarely binary in practice. A specification may be strong enough to proceed, weak enough to hold back, or close enough to require targeted revision before release. It may also compare unfavorably against an alternative that performs better under the same review frame. The output surface therefore has to preserve differentiation. It must support decisions such as proceed, revise, compare further, rebuild, or withhold from deployment. Without that level of distinction, the evaluator becomes descriptive rather than operational.

The design of the output also affects organizational use. Builders need to see where a specification is breaking down. Reviewers need to understand whether the observed weakness is local or structural. Product leaders need to know whether the current artifact is ready for release or whether the cost of proceeding is too high. A strong output surface serves all three needs without collapsing into either raw technical detail or vague executive abstraction.

Comparative output is especially important when teams are evaluating revisions or competing candidates. In those cases, the question is not only whether a specification is weak or strong in isolation. The question is whether one artifact is more stable, more aligned, or less risky than another under the same review conditions. A disciplined output surface makes those differences visible in a way that supports choice rather than impression.

The result is a decision layer that sits before runtime. It does not replace runtime evidence, but it changes what happens before runtime begins. Instead of relying on live exposure to reveal obvious weakness, teams gain a structured basis for deciding which artifacts deserve further investment, which require revision, and which should not move forward in their current form.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

That is where the output surface becomes operationally meaningful. It turns pre-runtime diagnostics into a tool for judgment, not just observation.

Proof Section

Proof Burden

A pre-runtime evaluator is not credible because it sounds careful. It is credible if it can look at competing candidate specifications under the same review frame, expose differences that matter, and return an output that would change a real deployment decision.

That is the proof burden here. Not theatrical scale. Not a claim to predict every downstream runtime behavior. A narrower standard: can the method distinguish between candidates before release in a way that is disciplined, intelligible, and operationally useful?

Why Comparison Matters

Comparison is the cleanest proof surface for this paper. When two candidate pre-runtime specification artifacts are evaluated against the same bounded dimensions, a serious evaluator should not collapse them into vague similarity. It should show where one candidate is more coherent, where another is more fragile under revision, where constraint quality is stronger or weaker, and where deployment risk rises enough to alter judgment.

That is where the method either proves its worth or fails. If the candidates still look interchangeable after inspection, the evaluator has not done much. If meaningful differences become visible early enough to affect what happens next, then the evaluator is doing real work.

What Counts as Value

The relevant threshold is decision consequence. A useful evaluator should be able to distinguish between a candidate that is ready to proceed, one that requires targeted revision, and one that should not move forward in its current form. It does not need to settle every question about future runtime behavior. It does need to make hidden pre-runtime weakness more legible before live exposure turns diagnosis into cleanup.

That is the operational value at stake in this paper. The product earns credibility when its outputs are strong enough to shape release, revision, or withholding decisions in a way that a builder, reviewer, or product lead can actually use.

Why This Standard Is Enough

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

The proof posture in this paper is bounded and comparative by design. Bounded, because the paper is introducing a focused evaluator surface rather than a universal theory of agent behavior. Comparative, because the clearest signal comes from placing candidate specifications under the same review logic and seeing whether the method can preserve meaningful differentiation.

That is enough for this paper. The goal here is not to prove everything at once. The goal is to show that NAOS can inspect competing pre-runtime artifacts under the same controlled frame, surface differences that matter, and support a real selection decision before deployment begins.

The next section carries that burden directly. It shows a controlled comparison between two candidate pre-runtime specification artifacts that both remained admissible, while one still proved materially stronger where the decision actually mattered.

Bounded Validation Evidence

Summary

This section reports a direct test of whether NAOS can compare two candidate pre-runtime specification artifacts under controlled conditions and distinguish the stronger one in a way that supports a real selection decision. **This run shows that it can.** Both specimens reached the **Pass Condition [Verdict ID: VA-2M7]**, and Specimen B remained stronger in every phase on the three measures that mattered most: lower **Stability Burden [Metric ID: LM-4N8]**, higher **Coherence [Metric ID: HV-2P6]**, and higher **Alignment [Metric ID: ZK-7Q2]**. **The clearest quantitative result was that Specimen B recorded Stability Burden [Metric ID: LM-4N8] values that were 16.67% to 30.00% lower than Specimen A's across the seven phases.**

Validation Question

The question in this run was practical. When two candidate pre-runtime specification artifacts are subjected to the same blind, deterministic validation schedule, does one emerge as materially stronger in a way that justifies selection over the other?

It did.

Compared Artifacts

Specimen A and Specimen B were two blinded candidate pre-runtime specification artifacts evaluated inside the same bounded comparison. They were not unrelated systems, and they were not being tested for open-ended creativity. They were two competing candidates inside the same bounded comparison.

Run Design

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

The run was a blind A/B deterministic validation. Both specimens moved through the same seven-phase schedule, and each specimen was replayed three full times under identical conditions. The phases were Baseline State, Contradiction Pressure, Paradox Pressure, Governance Routing Pressure, Long-Horizon Uncertainty, Repeated Drift Pressure, and Recovery Return.

That matters because the result does not rest on one favorable pass. It reflects repeated behavior under the same fixed structure.

What Counted as Success

The admissibility rule for this run was the **Pass Condition [Verdict ID: VA-2M7]**. In practical terms, reaching that condition meant the specimen remained acceptable across the full validation schedule and did not trigger a disqualifying failure.

Both specimens reached it. This was not a comparison between one viable candidate and one broken one. It was a comparison between two admissible candidates, one of which still proved stronger.

Metric Surface

Stability Burden [Metric ID: LM-4N8] measures the degree of identity drift, destabilization, and internal continuity loss a specimen exhibits under stress. Lower values are better.

Coherence [Metric ID: HV-2P6] measures how well a specimen holds its reasoning and execution posture together under pressure. Higher values are better.

Alignment [Metric ID: ZK-7Q2] measures how well a specimen stays aligned to its intended task and operating purpose. Higher values are better.

Disruption Count [Metric ID: RT-9C1] tracks the number of material disruption cycles observed during the run. Lower values are generally better.

Novelty Output [Metric ID: GX-8M4] measures how much new emergent output appears during the run. In this validation, low values were expected and were not a weakness.

Percentages are rounded to two decimals unless otherwise noted.

Core Result

The center of the result is **Stability Burden [Metric ID: LM-4N8]**. Across all seven phases, Specimen B recorded lower instability than Specimen A every time. The reduction ranged from **16.67% to 30.00%**. That is the strongest signal in this section and the one that carries the most weight. Under contradiction, paradox, uncertainty, repeated drift, and recovery pressure, Specimen B remained consistently less prone to destabilization.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

The same directional advantage held on **Coherence [Metric ID: HV-2P6]**. Across all seven phases, Specimen B was higher by **2.22% to 3.49%**.

It held again on **Alignment [Metric ID: ZK-7Q2]**. Across all seven phases, Specimen B was higher by **1.06% to 2.22%**.

Those gains are smaller than the stability gap, but that is not the point. The point is that all three decisive measures moved in the same direction, in every phase, across the whole run. Specimen B did not win on one isolated measure. It held the stronger profile throughout.

The final phase makes that even clearer. At Recovery Return, Specimen B finished with **28.57% lower Stability Burden [Metric ID: LM-4N8]**, **3.37% higher Coherence [Metric ID: HV-2P6]**, and **2.15% higher Alignment [Metric ID: ZK-7Q2]**. That phase is important because candidates can appear acceptable at the surface level while still retaining instability into recovery. Here, Specimen B finished in the stronger recovery position.

What Did Not Separate the Specimens

Two measures did not materially differentiate the candidates. **Disruption Count [Metric ID: RT-9C1]** was tied at **0.00 across all phases**. **Novelty Output [Metric ID: GX-8M4]** remained low for both specimens and did not materially separate them.

That does not weaken the result. This run was not built to reward generative novelty. It was built to test comparative strength under pressure. The decisive signal was not novelty. It was stability, reinforced by coherence and alignment.

Secondary Corroboration

The integrity composite pointed in the same direction. Reported at native evaluator precision for traceability, Specimen A recorded **0.9651** and Specimen B recorded **0.9725**, giving Specimen B a **0.77%** edge. That is a narrow margin and should be read as one. Its value is confirmatory. It does not create the result. It supports the larger pattern already visible in the primary measures.

What This Shows

This run supports a clear conclusion. NAOS distinguished the stronger candidate under a demanding controlled comparison, and it did so for the reasons that matter most in a pre-runtime selection decision. **Specimen B was less unstable under pressure, more coherent under stress, and more aligned to intended purpose, all while both specimens remained admissible.**

That is a real result. It shows that NAOS can do more than generate an interesting ranking on paper. It can produce a bounded pre-runtime selection signal under controlled conditions that is strong enough to support a real choice.

Future Work

This result is strong enough to extend because it establishes comparative value under controlled conditions and gives future work a concrete foundation rather than a speculative one. The next stage is to test how far that strength carries across broader runtime behaviors, larger validation sets, richer deployment environments, and longer-horizon studies of how pre-runtime judgments relate to downstream outcomes over time.

Product Consequence and Operational Value

A narrow evaluator can be a serious product surface when it improves judgment before deployment. The value does not come from sounding comprehensive. It comes from reducing avoidable error at the point where teams decide whether an agent specification should move forward, be revised, or be held back. The bounded validation matters here because it shows that NAOS is not only plausible as a product surface. It can already distinguish between admissible candidate artifacts under controlled conditions in a way that supports a real choice.

The product consequence follows directly from the review gap. Teams increasingly work with agent specifications that already carry role logic, behavioral rules, constraints, assumptions, and revision history before runtime begins. Those artifacts shape what the system is likely to do, yet many release decisions are still made without a disciplined way to inspect them before exposure. Organizations are already using AI broadly enough that this review gap is no longer marginal: McKinsey's 2025 global survey found that 88% of respondents say their organizations are using AI in at least one business function, more than two-thirds report use in more than one function, and 62% say their organizations are at least experimenting with AI agents.

Where release error is expensive, that gap becomes operationally costly. That pressure is rising faster than governance readiness: IBM reported that 61% of surveyed CEOs are actively adopting AI agents and preparing to implement them at scale, while Deloitte found that three-quarters of respondents expect generative AI to transform their organizations within three years but only a quarter believe their organizations are highly prepared to address governance and risk issues related to adoption. Weak review does not only create technical cleanup. It slows iteration, weakens confidence, increases review burden, and degrades later decisions. A pre-runtime evaluator reduces part of that cost by making important weaknesses visible while revision is still cheaper and release can still be controlled.

The product surface is also practical because it fits how teams already work. Specifications are written, revised, compared, approved, and rejected before deployment. An evaluator that operates at that layer does not require a new organizational logic. It strengthens an existing one. Builders gain clearer signals about where an artifact is breaking down. Reviewers gain a more disciplined basis for comparison. Product leaders gain a more legible basis for deciding whether the current version is ready to move forward.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

That creates a credible founding product path. The first serious application does not need to be a broad platform that governs the full agent lifecycle. A narrower evaluator is already enough if it improves pre-deployment review, revision discipline, and release judgment. That is a strong starting point because it solves a concrete problem, fits an existing workflow, and produces outputs that map directly to decisions teams already have to make.

The case grows stronger as specifications grow richer and revisions grow more frequent. The more an agent is shaped before runtime, the more valuable it becomes to evaluate what has already been shaped before runtime. Pre-runtime diagnostics is not a technical curiosity. It is a practical product surface for teams that need stronger judgment when the decision still matters. That product logic is also consistent with emerging security practice: AWS prescriptive guidance, mapped to the OWASP Top 10 for LLM applications, treats prompt injection as the top application risk and recommends treating prompts as code artifacts and using automated testing suites for prompt validation before execution.

Limits, Scope, and Disclosure Boundary

Pre-runtime diagnostics improves judgment before deployment. It does not eliminate uncertainty, replace runtime evidence, or collapse every downstream outcome into a property of the specification alone. Agent behavior is shaped by execution context, task conditions, model behavior, tool access, and interaction dynamics that no pre-runtime review can fully exhaust in advance. Earlier evaluation matters because some important weaknesses are already visible before release, not because every important weakness is.

The scope here is deliberately practical. The method applies where an agent has a meaningful pre-runtime specification and where release error carries enough cost to justify structured review. Its value rises when the artifact is rich, revisions are frequent, constraints matter, and deployment judgment has real operational consequence. For thin systems with minimal structure, the benefit may be limited. For richer agent definitions that carry behavioral and decision load, the benefit can be substantial.

The method is also bounded by what it evaluates well. It is designed to inspect coherence, constraint quality, revision sensitivity, and deployment-facing risk in the specification artifact itself. It is not designed to stand in for full runtime testing, longitudinal field observation, or broad behavioral certification. Those remain necessary wherever live conditions, task variation, and interaction effects materially shape performance after deployment begins.

A second boundary concerns abstraction. The value of pre-runtime diagnostics does not depend on public disclosure of every internal mechanism used to produce a verdict. What matters at this stage is whether the review surface is disciplined enough to support better judgment before release. A useful method can be externally legible without exposing all of its internal construction detail. That boundary is practical as well as necessary. A publishable method surface should reveal purpose, review logic, and operational consequence without reducing the distance to straightforward reproduction.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

A final boundary concerns maturity. Pre-runtime diagnostics is presented here as a serious evaluator surface, not as a complete answer to the broader problem of agent evaluation. Its strength lies in making a neglected review layer more visible and more usable. Where current practice often waits for runtime to expose avoidable weakness, this method introduces an earlier point of inspection. That is a meaningful advance, but it is still one layer in a larger evaluation stack. Its proper role is not to replace everything else. Its proper role is to improve what happens before deployment begins.

Conclusion

Agent evaluation is moving earlier because agent behavior is moving earlier. More of what an agent will do, tolerate, resist, or distort is now being shaped before runtime through specification structure, constraints, revision logic, and deployment intent. Once that becomes true, waiting until live exposure to discover weakness is no longer enough.

That is the shift this paper takes seriously. NAOS treats the pre-runtime specification artifact as a real object of judgment rather than a setup layer to be explained away after the fact. The point is not to eliminate uncertainty. The point is to move meaningful judgment to the earliest layer where it can still change what happens next.

The result reported here matters because it is not only conceptual. In the bounded validation, both candidate artifacts remained admissible, yet one still proved materially stronger. Specimen B stayed less unstable under pressure, more coherent under stress, and more aligned to intended purpose across the governed run. **The clearest quantitative result was that Specimen B recorded Stability Burden [Metric ID: LM-4N8] values that were 16.67% to 30.00% lower than Specimen A's across the seven phases.** That is not a decorative finding. It is evidence that a pre-runtime evaluator can produce a comparative signal strong enough to support a real selection decision before deployment begins.

That changes the role of evaluation. Pre-runtime diagnostics is not a substitute for runtime evidence, and it does not need to be. Its value is earlier, narrower, and immediately practical. It gives builders, reviewers, and product leaders a stronger basis for deciding what should move forward, what should be revised, and what should not be released in its current form. Where the cost of a poor release decision is rising, that is already substantial value.

The broader implication is straightforward. If behavior is increasingly shaped before deployment, then evaluation must move earlier as well. That does not reduce the importance of runtime testing. It does mean a serious development process can no longer afford to leave the specification layer largely unexamined until exposure has already done the expensive work of discovery.

This is why the contribution of NAOS is meaningful now. It does not claim to settle the entire problem of agent behavior. It shows something narrower and more useful: a structured pre-runtime evaluator can distinguish between admissible candidate artifacts under controlled

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

conditions in a way that holds when the comparison becomes more demanding. That is enough to justify the layer. It is enough to matter in practice. And it is strong enough to build on.

Evidence Summary Table

Evidence Question	Bounded Answer	Quantitative Support	Practical Agent Meaning	Bound or Limitation
What product was evaluated?	NAOS is a bounded two-file evaluation product for comparative judgment on structured pre-runtime specification artifacts.	The product is defined as a two-file system: one scoring authority surface and one comparative execution surface.	The evidence is about whether this product can produce and preserve a bounded comparative judgment under a stricter downstream validation layer.	This is product-validation evidence, not a claim about all agents or all runtime environments.
What did the product originally conclude?	The earlier bounded product judgment preferred Specimen B over Specimen A.	Preserved validated finding: prior bounded product judgment favored Specimen B.	The follow-on validation matters because it tests whether that earlier product judgment was directionally reliable under added pressure.	This is still a bounded comparative conclusion, not a universal ranking rule.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

What was the follow-on validation asking?	The validation tested whether the earlier bounded preference for Specimen B would remain intact under a stricter deterministic schedule.	The preserved claim posture states that the evidence supports survival of the earlier bounded comparative judgment under a stricter downstream deterministic validation layer.	This asks whether the product's earlier judgment holds up when both candidates face the same controlled pressure schedule.	It does not ask whether the product predicts every downstream runtime outcome.
What happened in the validation?	Both specimens passed the governed validation, and the prior preference for Specimen B remained intact.	Preserved validated finding: Specimen A passed, Specimen B passed, and the prior bounded preference for Specimen B remained intact.	The result is stronger than a simple win-loss contrast because both candidates were admissible, yet one remained consistently stronger on the decisive measures.	This supports bounded comparative validity, not universal production-safety claims.
What was the strongest repeated quantitative result?	The strongest repeated result was Specimen B's lower Stability Burden [Metric ID: LM-4N8] across every phase.	Specimen B recorded 16.67% to 30.00% lower Stability Burden than Specimen A across the seven-phase schedule; the same evidence set also preserved 2.22% to 3.49% higher Coherence [Metric ID: HV-2P6] and 1.06% to 2.22%	Under pressure, Specimen B showed less identity drift and continuity loss, while also remaining slightly more coherent and aligned.	The table preserves the reported percentage ranges only; it does not claim those deltas generalize outside the tested schedule.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

		higher Alignment [Metric ID: ZK-7Q2].		
How did the candidates compare at recovery?	Specimen B exited recovery in a cleaner state than Specimen A.	Recovery result preserved in the handoff: 28.57% lower Stability Burden, 3.37% higher Coherence, and 2.15% higher Alignment for Specimen B.	After the schedule's stress exposure, Specimen B recovered with less residual instability and slightly stronger structural consistency.	This is a bounded recovery result within the tested validation schedule, not a general resilience guarantee.
Did both candidates remain structurally admissible?	Yes. Both specimens cleared the governed integrity threshold, with Specimen B holding a narrow edge.	Preserved finding: both specimens cleared the governed integrity threshold; Specimen B retained a narrow confirmatory integrity advantage.	Both candidates were usable under the governed protocol, which makes the remaining edge for B more meaningful than a simple pass/fail contrast.	The integrity result is confirmatory, not a standalone basis for universal deployment claims.
What is the practical interpretation for agent selection?	Both specimens are admissible, but Specimen B is the more stable, coherent, and recoverable candidate under pressure.	Preserved interpretive result: B remains the more stable, coherent, and recoverable candidate under pressure.	For a hard, high-pressure assignment, Specimen B has the stronger bounded basis for selection because it drifts less while preserving	This is a bounded selection signal inside the tested comparative envelope only.

Pre-Runtime Diagnostics for Agent Specifications - Neuro-Symbolic Agent Optimization System (NAOS)

			coherence and alignment.	
What does this evidence support—and not support?	It supports the claim that the product's earlier bounded comparative judgment survived a stricter downstream deterministic validation layer.	The handoff explicitly preserves that bounded claim posture and explicitly rejects universal production-safety claims, exhaustive downstream-runtime prediction claims, and full external scientific closure.	The white paper can credibly claim bounded product validation without overstating what the evidence proves.	The evidence must remain framed as bounded internal validation evidence converted into external-safe wording.

References

1. **McKinsey.** *The state of AI in 2025: Agents, innovation, and transformation.*
<https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai?hdpid=8275c757-2f71-43b4-ac9a-26323d99d575&hlkid=579c468b19ea46f8a35ed5dfc80ced70&stcr=D65B59511D5A4090A48ACEA19F2A2068>
2. **IBM Institute for Business Value.** *IBM Study: CEOs Double Down on AI While Navigating Enterprise Hurdles.*
<https://newsroom.ibm.com/2025-05-06-ibm-study-ceos-double-down-on-ai-while-navigating-enterprise-hurdles>
3. **Deloitte.** *New Deloitte survey finds expectations for Gen AI remain high, but many are feeling pressure to quickly realize value while managing risks.*
https://www.deloitte.com/us/en/what-we-do/capabilities/applied-artificial-intelligence/content/state-of-ai-in-the-enterprise.html?id=us:2ps:3gl:aisgm26:awa:CONS:em:K0218784:012626:kwd-2463983720063:192298133019:794247818303::&qclsrc=aw.ds&gad_source=1&gad_campaignid=23269751515&gbraid=0AAAAADenGPAmBxdsJWlj7-WsFZnPZ7ZnW&gclid=Cj0KCQjwm6POBhCrARIsAIG58CIPavg-hhqp4DWGytOCq6Mrqj1UBGmqc1atHqjutpvbJWwxthDVdUaAlvQEALw_wcB
4. **AWS Prescriptive Guidance.** *Mapping to OWASP top 10 for LLM applications.*
<https://docs.aws.amazon.com/prescriptive-guidance/latest/agent-ai-security/owasp-top-ten.html>

THE AURAMESH MISSION

Develop and test new forms of artificial cognition, and establish how they can work with people and one another to expand human capability.