Architect: An Agent Architecture for Enterprise Process Transformation

Linc AI Research
founders@linc-ai.com

May 2026

Abstract

We introduce a benchmark for enterprise process-transformation work and present **Architect**, a specialized agent architecture targeting three task types that dominate consulting-grade engagements: (1) workflow replication from raw stakeholder inputs, (2) ROI-quantified improvement-opportunity discovery, and (3) future-state (*to-be*) process design under operating constraints. The benchmark consists of 13 task instances across 8 unique source workflows derived from real-world process-transformation engagements with enterprise customers (anonymized for confidentiality, not LLM-generated), and audited gold standards (cross-checked by a five-agent deliberative panel of Anthropic Claude Opus 4.7 and Sonnet 4.6 with web-search and web-fetch tool access, then human-reviewed before acceptance). We compare Architect against the strongest publicly available frontier-model baselines from Anthropic, OpenAI, and Google. On workflow replication, Architect achieves shippable output on 5/5 cases with zero hallucinated steps, versus 1/5 for the strongest single-call baseline. On opportunity discovery, Architect produces 88 valid opportunities against 40 for the strongest baseline (2.2×), with perfect coverage of leadership-flagged improvement themes. On to-be process design, Architect leads the strongest baseline by 0.32 in mean composite score (0.906 vs. 0.586 on a 0–1 scale) and is the only system to clear the shippability bar on every case. We introduce a *brief-engagement* metric that measures whether a rationale explicitly cites a numbered leadership priority; GPT-5 and Gemini 3.1 Pro score zero on this metric across all three to-be cases, exposing a structural gap between generic frontier models and a brief-aware agent architecture.

1 Introduction

Most public benchmarks for large language models measure capabilities that frontier laboratories optimize for directly: graduate-level reasoning (GPQA [1]), software engineering (SWE-bench [2]), mathematics (AIME), and exam-style knowledge (MMLU [3]). Enterprise process-excellence work has a different shape: the input is a heterogeneous collection of stakeholder interview transcripts, standard operating procedure (SOP) documents, regulatory framework references, and exported system extracts; the output is a structured deliverable that a process-excellence team or senior consultant would accept without significant rework. The dimension that determines whether an output is useful in practice is not raw correctness on any single metric, but *shippability*: would a senior reviewer accept this artifact and present it to leadership without intermediate revision?

Shippability is not benchmarked in the public LLM evaluation literature. We construct a benchmark that targets it directly, and we present an agent architecture, **Architect**, that achieves shippable output on consulting-grade enterprise transformation tasks.

Contributions.

1. A three-task benchmark for enterprise process transformation (workflow replication, opportunity discovery, to-be design) with audited gold standards (Section 3.4) covering 13 task instances across 8 unique workflows in Finance & Accounting, Compliance, Procurement (with cross-regional variants), Healthcare/Payer SI, Customer Support Ops, Engineering Operations, and People Operations.
2. A multi-agent audit methodology for gold-standard validation using mixed-tier models with web search and web fetch, producing fully traceable audit logs.
3. A composite scoring framework for to-be process design that captures shippability via six programmatic dimensions, including a *brief-engagement* metric that measures whether a system's change rationales explicitly cite stated leadership priorities.
4. A grounding-gated change-set coverage metric that requires per-step rationale grounding to count toward change capture, eliminating credit for ungrounded "best-practice" modifications.
5. Experimental results showing Architect outperforms all evaluated frontier-model baselines (Claude Opus 4.7, Claude Haiku 4.5, GPT-5, Gemini 3.1 Pro) by wide margins on every task and every measured dimension.

2 Related Work

Process mining. Automated process discovery from event logs is an established field, with commercial systems (Celonis, Signavio, UiPath Process Mining) and academic work [4] focused on log-driven analysis of structured transactional data. Recent work has begun exploring large language models for process-mining tasks [11, 12], though primarily on log abstraction and querying rather than consultative design synthesis. Our setting is complementary: the input is unstructured stakeholder narrative, SOPs, and documents rather than transactional logs, and the output is consultative process design rather than statistical discovery.

LLM benchmarks for structured outputs. Recent work has evaluated LLMs on structured-output tasks including code (SWE-bench [2], LiveCodeBench [8]), function-calling (BFCL [9]), and document understanding (DocVQA [10]). Our work differs in three respects: (1) the gold standards are derived from real engagement material and audited through multi-round deliberation rather than retrieved from a public ground-truth corpus, (2) the output schema is rich and consultative rather than syntactically constrained, and (3) the success criterion is shippability to a senior reviewer rather than exact match.

Agent architectures and multi-agent deliberation. General-purpose agent frameworks [5] provide tooling for agentic behavior, and multi-agent debate has been shown to improve reasoning quality [6, 7]. Linc Architect is specialized: it incorporates per-session decomposition, customer-stack awareness, anti-hallucination guardrails, and structural change-tagging tailored to process-transformation deliverables. The audit methodology applied to our gold standards (Section 3.4) draws on the multi-agent deliberation literature, using independent agents as a structured cross-check rather than as primary content generators.

LLM-as-judge. The use of LLMs to evaluate other LLMs' outputs is now established [13]. Our scoring framework currently uses heuristic detectors (substring overlap for grounding, regular expressions for brief engagement); replacing or supplementing these with LLM-judge layers is on the roadmap and discussed in our limitations.

DMAIC alignment. The three task types in our benchmark align with the first four phases of Lean Six Sigma's DMAIC methodology [14]: workflow replication maps to Define and Measure, opportunity discovery to Analyze, and to-be design to Improve. The Control phase is intentionally out of scope; statistical process control and governance routines remain human-owned.

3 Tasks and Benchmark Design

3.1 Workflow Replication

Input: Stakeholder interview transcripts (2–3 per case), SOP documents, and optionally extracted PDF and Excel evidence. **Output:** A structured workflow hierarchy where every step carries a name, description, inputs, outputs, dependency edges, stakeholder owner, and an SOP-vs-practice status tag in {`documented`, `partial`, `gap`, `inferred`}.

Scoring is per-step: recall and precision against the audited gold hierarchy, dependency-graph F_1, stakeholder Jaccard, and SOP-status accuracy. A composite *shippability* flag is binary: recall $\geq 0.80 \wedge$ precision $\geq 0.90 \wedge$ dep-$F_1 \geq 0.50$. The thresholds correspond to "would a senior reviewer accept this without rework?"

3.2 Opportunity Discovery

Input: The same source materials plus any leadership-stated constraints (forbidden replatform targets, out-of-scope categories). **Output:** An ROI-quantified portfolio of process-improvement opportunities, each with hours-per-occurrence, occurrences-per-week, FTE involvement, annual savings, implementation complexity, time-to-value, dependencies, and source-session attribution.

Scoring dimensions include: groundedness (whether claims trace to source evidence), specificity (operational detail level), ROI plausibility, theme coverage (whether leadership-flagged improvement themes appear in the portfolio), constraint compliance, and per-set yield (number of opportunities that pass every quality gate).

3.3 To-Be Process Design

Input: The current-state workflow hierarchy (output of Task 1), a leadership brief containing ordered improvement priorities and seven hard operating constraints, and the source materials. **Output:** A future-state workflow hierarchy where each step is tagged in {`retained`, `modified`, `added`, `deprecated`} with a source-grounded rationale and a traceable link to a named improvement opportunity.

Scoring is on six programmatic dimensions (see Section 5.2): change-set coverage (grounding-gated), retained-step preservation, change-rationale grounding, opportunity traceability, brief engagement, and structural coherence.

3.4 Audited Gold-Standard Construction

A skeptical reader reasonably asks: "did the authors shape the gold standards to flatter their own system?" To address this attack surface, every gold standard is *derived from* the engagement-derived case material (Section 3.5) and *audited* — not generated from scratch by AI. The audit uses a five-agent deliberative panel as a structured cross-check: each candidate gold standard is critiqued from multiple independent perspectives over three rounds, with unresolved disagreements explicitly flagged for human review before the gold is accepted as ground truth.

Round 1. Four independent drafts: *Process Engineer* (Claude Opus 4.7), *Domain Expert* (Claude Opus 4.7 with WebSearch on APQC Process Classification Framework, ITIL, COSO,

and industry-specific frameworks), *Skeptic* (Claude Sonnet 4.6), and *Researcher* (Claude Sonnet 4.6 with WebFetch on cited references).

Round 2. Cross-critiques: each panelist critiques the other three drafts with severity tags in $\{\texttt{blocker}, \texttt{important}, \texttt{nit}\}$, producing $4 \times 3 = 12$ critique documents.

Round 3. Synthesis: Claude Opus 4.7 receives the four drafts, all twelve critiques, and the original source material, and produces the final gold hierarchy plus an audit log listing every resolved disagreement, every unresolved blocker, and an explicit constraint-compliance walkthrough.

Limitation disclosure. The panel uses 100% Anthropic models (Opus 4.7 and Sonnet 4.6 in the panel roles). Architect runs on Claude Opus 4.7. We acknowledge this as an attack surface: the gold standards could in principle exhibit Anthropic-specific stylistic preferences that flatter Anthropic-based systems. The naive baseline cohort spans three vendors (Anthropic, OpenAI, Google), but a fully bias-free evaluation would require replicating the panel methodology with cross-vendor agents. We treat this as a roadmap item rather than a resolved concern.

3.5 About the Data

Cases enter the benchmark via a two-step process. The validity of the benchmark anchors on Step 1 (human authoring from real engagement experience); the role of AI is confined to Step 2 (audited gold-standard synthesis from the human-authored materials).

Step 1 — Case authoring (humans). Linc team members who have worked directly on enterprise process-transformation engagements author the case materials by hand: stakeholder interview transcripts, SOP documents, and (for to-be design) the leadership brief. Transcripts reproduce stakeholder language patterns, decision rhythms, and pain-point articulations observed in real engagements; SOPs reflect actual procedural structure documented in customer environments; leadership briefs mirror the constraint language and improvement-priority phrasing real CFOs and VPs use. *No language model writes the case materials.* The procurement case reproduces the actual cycle-time bottlenecks, vendor-master duplication patterns, and exception-handling flows observed in indirect-procurement engagements; the HR-onboarding case reproduces the Day-1 access gaps and buddy-assignment failure modes; the engineering-to-operations handoff case reproduces the runbook-quality and escalation-back patterns observed in platform-team engagements. Identifying details (company names, specific transaction volumes, named individuals, distinguishing internal tooling) are anonymized during authoring under the fictional "Meridian Dynamics" umbrella; the structural realities — workflow shapes, system integrations (Coupa, NetSuite, AuditBoard, Workiva, Greenhouse, Workday, ServiceNow, Datadog, PagerDuty), and compliance constraints (SOX, PCAOB AS 2201, EU VAT directives, Japan Reiwa-5 invoice reform, IRS TIN-Match) — are preserved. The anonymization step is itself human-authored, not LLM-generated paraphrase.

Step 2 — Gold-standard synthesis from cases (AI panel + human audit). The human-authored case materials are then handed to a five-agent deliberative panel (Section 3.4) that drafts candidate gold-standard answers for the benchmark task — the "correct" workflow hierarchy for replication, the opportunity portfolio for discovery, or the to-be design. The panel deliberates over three rounds with cross-critique and explicit unresolved-blocker tracking; the synthesized gold plus a per-case audit summary is reviewed by a human reviewer before the gold is accepted as ground truth. The benchmark therefore does not test whether a language model can reproduce content another language model originated — it tests whether a system can reproduce structured answers to questions about human-authored, engagement-grounded material.

Scale. Per-case scale ranges from 20–27k characters for smaller cases to 60–90k characters

for multi-stakeholder cases. Every case has 2–3 stakeholder transcripts plus SOPs; one case (compliance audit) additionally ships an extracted PDF policy document and an Excel audit-log export to test document-evidence ingestion. The benchmark is intentionally not released publicly because the underlying engagement work is customer-confidential.

4 Architect: System Description

Linc Architect is an AI agent architecture purpose-built for enterprise process work. It consumes the inputs a senior consultant would receive at the start of a transformation engagement and produces three categories of structured output: a reconstructed current-state workflow hierarchy, a portfolio of ROI-quantified improvement opportunities, and a future-state redesign that honors stated operating constraints.

Architect builds on Claude Opus 4.7 as its underlying language model. The architectural contribution is not the choice of model but the structured prompt scaffolding around it. For brevity we summarize the key components:

- **Per-session decomposition.** Multi-stakeholder cases are decomposed by stakeholder before hierarchy synthesis, preventing single-call bias toward one stakeholder's vocabulary.
- **Customer-stack awareness.** Architect's prompt structure includes the customer's stated stack and forbidden-replatform targets, with constraint-violation guards built into the generation procedure.
- **Step-granularity prompting.** For replication, Architect operates at step granularity (12–30 fine-grained nodes per workflow) rather than coarse workflow granularity (4–8 nodes), enabling matched granularity with consultant-grade deliverables.
- **Anti-fabrication guardrails.** Architect's prompt explicitly enumerates the anti-hallucination requirements and a self-audit step that occurs before the output is returned.
- **To-be-specific structure.** For to-be design, Architect's prompt enforces "anchored evolution, not from-scratch": the current-state hierarchy is treated as the anchor, and changes are tagged differentially with explicit rationale requirements.

5 Experimental Setup

5.1 Baselines

We compare Architect against the strongest publicly available frontier models from Anthropic (Claude Opus 4.7, Claude Haiku 4.5), OpenAI (GPT-5), and Google (Gemini 3.1 Pro). The four-baseline cohort is held constant across all three results to maintain cross-vendor symmetry: one top model per vendor (Opus for Anthropic, GPT-5 for OpenAI, Gemini 3.1 Pro for Google) plus Haiku 4.5 as the low-tier Anthropic comparator. Each baseline receives identical source materials in a single API call with structured-output schema enforcement; the baseline prompts request the same output structure as Architect but contain no scaffolding for per-session decomposition, constraint walkthrough, or grounding requirements.

We additionally evaluate three chat-application baselines (Claude.ai, ChatGPT, Gemini.app) that simulate the experience of pasting source materials directly into a consumer chat interface without API-level structured-output enforcement; results from this cohort are referenced briefly in the discussion but not the main results tables.

5.2 Scoring Framework for To-Be Design

The to-be design composite score is a weighted mean of six programmatic dimensions:

Dimension	Weight	Description
Change-set coverage	0.25	Fraction of gold non-retained changes captured with correct change-type **and** a source-grounded rationale (grounding-gated).
Rationale grounding	0.25	Fraction of non-retained steps whose rationale traces to source material or to the to-be brief.
Retained preservation	0.15	Fraction of gold-retained steps the system correctly kept retained.
Opportunity traceability	0.10	Fraction of non-retained steps with a populated `source_opportunity_id` field.
Brief engagement	0.10	Fraction of non-retained steps whose rationale explicitly cites a numbered leadership priority from the brief.
Structural coherence	0.10	Dependency-graph F_1 over the matched-step subgraph.
Constraint compliance	0.05	1.0 minus the fraction of hard-constraint families violated.

Table 1: Per-dimension weights for the to-be design composite score.

Shippability for to-be is a binary flag: change-set coverage $\geq 0.60 \wedge$ retained preservation $\geq 0.80 \wedge$ constraint compliance $= 1.0$.

Grounding-gated change-set. A naïve change-set metric that simply counts matched gold non-retained steps rewards over-tagging: a system that proposes 20 changes without grounding the rationale captures more raw matches than a system that proposes 11 carefully-grounded changes, even though the former's output is operationally unusable. We define grounding-gated change-set coverage as the fraction of gold non-retained changes captured with both (a) correct change-type and (b) a per-step grounded rationale. Per-step grounding is checked via substring overlap of the rationale against the source material and the brief, with explicit reference markers (e.g., "Brief priority #1") as a strong positive signal.

Brief engagement. Architect's prompt structure produces rationales that explicitly cite stated leadership priorities (e.g., "Brief priority #1 directly…"); baseline systems typically do not. We detect explicit citation using a precompiled regular expression matching patterns including "brief priority", "priority #n", "leadership priority", and "stated priority". The metric is the fraction of non-retained steps whose rationale matches the pattern.

6 Results

6.1 Result 1: Workflow Replication (5 cases × 5 systems)

System	Vendor	Ship rate	Total caught	Hallucinated
Linc Architect	Linc	**5/5**	**92/102**	**0**
Claude Opus 4.7	Anthropic	1/5	71	2
GPT-5	OpenAI	1/5	72	3
Claude Haiku 4.5	Anthropic	0/5	72	4
Gemini 3.1 Pro	Google	0/5	62	1

Table 2: Workflow replication results across 5 cases. Architect achieves 90% step recall with zero hallucinations; the strongest baseline by composite (Claude Opus 4.7) catches 71 of 102 gold steps and ships 1/5 cases. Shippability is recall ≥ 0.80, precision ≥ 0.90, dependency-$F_1 \geq 0.50$.

6.2 Result 2: Opportunity Discovery (5 cases × 5 systems)

System	Valid opportunities	Theme coverage
Linc Architect	**88**	**1.00**
Claude Opus 4.7	40	0.44
Claude Haiku 4.5	37	0.77
GPT-5	34	0.24
Gemini 3.1 Pro	27	0.49

Table 3: Opportunity discovery results across 5 cases. Architect yields 2.2× more valid opportunities than the strongest baseline by yield (Claude Opus 4.7 at 40) and achieves perfect leadership-theme coverage. Claude Haiku 4.5 is the strongest baseline by theme coverage (0.77), surfacing a dissociation between yield and coverage in single-call baselines.

6.3 Result 3: To-Be Process Design (3 cases × 5 systems)

System	Composite	Change-set*	Retained	Grounding	Brief eng.†	Ship
Linc Architect	**0.906**	**0.74**	**0.98**	**1.00**	**0.87**	**3/3**
Claude Opus 4.7	0.586	0.42	0.55	0.50	0.36	0/3
Claude Haiku 4.5	0.485	0.32	0.69	0.34	0.20	0/3
Gemini 3.1 Pro	0.401	0.06	0.88	0.08	0.00	0/3
GPT-5	0.349	0.00	0.74	0.02	0.00	0/3

Table 4: To-be process design results, mean across 3 cases (procurement, engineering-to-operations handoff, HR onboarding). $\star$ Grounding-gated change-set: a change must have both correct change-type and per-step grounded rationale to score. † Brief engagement: fraction of non-retained steps whose rationale explicitly cites a numbered leadership priority.

Case	Architect	Best baseline	Gap	Best baseline system
Procurement P2P	**0.891**	0.527	+0.364	Claude Opus 4.7
Engineering → Ops handoff	**0.926**	0.722	+0.204	Claude Opus 4.7
HR onboarding	**0.901**	0.509	+0.392	Claude Opus 4.7
Mean across 3 cases	**0.906**	0.586	**+0.320**	Claude Opus 4.7

Table 5: Per-case to-be design composites. Architect leads on every case; per-case gaps to the strongest baseline range from +0.20 to +0.39 on the 0–1 composite scale. Architect is the only system to clear the shippability bar on every case.

7 Analysis

7.1 Brief Engagement as a Discriminator

The brief-engagement metric exposes a sharp structural gap between Architect and frontier-model baselines. GPT-5 and Gemini 3.1 Pro score zero on this metric across all three to-be cases: their rationales never cite stated leadership priorities by name, even when the underlying change tangentially addresses one. Claude Haiku 4.5 averages 20% citation; Claude Opus 4.7 averages 36%. Architect averages 87%. This is not a model-capability gap (Architect uses Claude Opus 4.7 as its underlying model) but a prompt-architecture gap: structured citation of priorities is something a single-call baseline does not naturally produce, even from the same underlying model.

7.2 The Conservative-Versus-Aggressive Trade-off

Single-call baselines propose more raw changes per case (typically 13–25) than Architect (11–19). The mechanism is aggressive labeling of supposedly-stable steps as modified: across the cohort, baselines refactor 12–45% of the steps the brief did not ask to change. Every spurious modification is operational debt the implementation team pays in change-management and retraining time. Architect's "anchored evolution, not from-scratch" prompt structure produces conservative change-set output paired with high grounding and high retained-step preservation. The grounding-gated change-set metric (Section 5.2) captures the correct trade-off: a redesign with 11 grounded changes is materially more shippable than one with 20 changes where half are ungrounded best-practice fluff.

7.3 Constraint Compliance is Universal but Brittle

Every system in the to-be cohort passes the hard-constraint check (constraint compliance = 1.0). This finding is initially surprising and warrants careful interpretation. We attribute the universal pass rate to two factors: (1) the briefs state constraints prominently and explicitly, so baselines avoid the most obvious violations (proposing platform replacements, hiring net-new headcount); (2) our constraint-detection heuristic uses substring matching with a 60-character negation window, which catches the lexicon-anchored violations but cannot detect implicit constraint drift (e.g., proposing a "governance owner" role without explicitly using flagged language). A more semantically aware constraint-violation detector, possibly using an LLM judge with constraint-by-constraint walkthrough, is on the methodological roadmap.

7.4 DMAIC Mapping

For readers fluent in Lean Six Sigma, the three task types align with the first four phases of DMAIC: workflow replication covers Define (workflow scope) and Measure (baseline metrics, SOP-vs-practice gap tagging); opportunity discovery covers Analyze (root causes, waste categories); and to-be design covers Improve (countermeasure design, future-state mapping). The Control phase is intentionally out of scope; statistical process control, audit-grade documentation, and governance routines remain human-owned for principled reasons. The benchmark covers the work that typically consumes the first three to six weeks of a Black Belt project before countermeasure design begins.

8 Limitations

Panel vendor concentration. The panel uses 100% Anthropic models. We mitigate via cross-vendor naive baselines and explicit limitation disclosure (Section 3.4). A replication with cross-vendor panel agents is planned.

Anonymization and case scope. Cases are anonymized under the fictional "Meridian Dynamics" name to protect customer confidentiality, and stakeholder transcripts are rewritten to remove identifying language (Section 3.5). The underlying workflow structures, system integrations, exception modes, and compliance constraints are sourced from real engagement work — not LLM-generated — but the anonymization step necessarily smooths some of the messiness present in unredacted production materials (off-topic stakeholder asides, contradictory historical SOPs, partial system-extract corruption). The benchmark thus tests reasoning under realistic-but-cleaned conditions; performance on raw production materials may differ on the long tail.

Heuristic-based scorers. Our brief-engagement detector, grounding detector, and constraint-violation lexicon are pattern-based. False positives and false negatives are possible. We mitigate by computing across multiple cases and reporting means with per-case detail. A judge-LLM-based scoring layer is on the roadmap for v2.

Three to-be cases. The to-be cohort is smaller (3 cases) than the replication and opportunity-discovery cohorts (5 cases each) because each to-be case requires an authored buyer brief and a separately audited gold standard. We will expand to 6–8 to-be cases as the case set grows.

Benchmark, not pilot. This is an AI-capability benchmark on engagement-derived cases. The relevant test for a customer pilot is whether the system performs comparably on the customer's own workflows. Pilot success criteria deliberately set thresholds below benchmark numbers so customer environments need not match the benchmark's anonymized-but-real conditions to count as success.

9 Discussion and Conclusion

We present a benchmark and an agent architecture targeting enterprise process-transformation work. Across three task types and 13 task instances, Linc Architect achieves shippable output where the strongest publicly available frontier-model baselines do not, by margins ranging from 5/5 vs. 1/5 on shippable replication cases (1.30× on raw step recall: 92 vs. 71 of 102 gold steps for Claude Opus 4.7) to 2.2× on valid-opportunity yield in discovery to a mean composite gap of 0.32 on a 0–1 scale for to-be design.

The methodological contribution is the recognition that shippability is a multi-dimensional concept that single-metric benchmarks fail to capture. A workflow reconstruction with 90% recall but with 5 hallucinated steps is not shippable; a to-be design with 80% change-set coverage but with two violations of stated constraints is not shippable; a to-be design with 20 ungrounded "improve-efficiency" rationales is not shippable. Our scoring framework targets shippability directly, gates change-set credit on grounding, and surfaces brief-engagement as a sharp discriminator between specialized agent architectures and generic frontier-model usage.

The key system contribution is the demonstration that prompt-architecture matters as much as model choice. Architect uses the same Claude Opus 4.7 model that powers our strongest baseline (single-call Opus); the 0.32 mean composite gap on to-be design comes entirely from the agent architecture surrounding the model, not from model capability differences. We interpret this as evidence that the next gains in enterprise AI deliverables will come from specialized agent architectures rather than from model-scale increases alone.

Reproducibility and Availability

Methodology is summarized in this paper. Per-case artifacts, panel deliberation logs, generator implementations, and scoring code are available to customers during pilot engagement. Contact: `founders@linc-ai.com`.

References

[1] Rein, D., Hou, B.L., Stickland, A.C., Petty, J., Pang, R.Y., Dirani, J., Michael, J., and Bowman, S.R. *GPQA: A Graduate-Level Google-Proof Q&A Benchmark*. arXiv preprint arXiv:2311.12022, 2023.

[2] Jimenez, C.E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?*. ICLR, 2024.

[3] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. *Measuring Massive Multitask Language Understanding*. ICLR, 2021.

[4] van der Aalst, W.M.P. *Process Mining: Data Science in Action*. Springer, 2nd edition, 2016.

[5] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A.H., White, R.W., Burger, D., and Wang, C. *AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation*. arXiv preprint arXiv:2308.08155, 2023.

[6] Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., and Mordatch, I. *Improving Factuality and Reasoning in Language Models through Multiagent Debate*. ICML, 2024.

[7] Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., and Shi, S. *Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate*. arXiv preprint arXiv:2305.19118, 2023.

[8] Jain, N., Han, K., Gu, A., Li, W., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. *LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code*. arXiv preprint arXiv:2403.07974, 2024.

[9] Yan, F., Mao, H., Ji, C.C.-J., Zhang, T., Patil, S.G., Stoica, I., and Gonzalez, J.E. *Berkeley Function Calling Leaderboard*. `https://gorilla.cs.berkeley.edu/leaderboard.html`, 2024.

[10] Mathew, M., Karatzas, D., and Jawahar, C.V. *DocVQA: A Dataset for VQA on Document Images*. WACV, 2021.

[11] Berti, A., Schuster, D., and van der Aalst, W.M.P. *Abstractions, Scenarios, and Prompt Definitions for Process Mining with LLMs: A Case Study*. arXiv preprint arXiv:2307.02194, 2023.

[12] Berti, A., Kourani, H., Hafke, H., Li, C.-Y., and Schuster, D. *Evaluating Large Language Models on Process Mining Tasks*. arXiv preprint arXiv:2403.06749, 2024.

[13] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., and Stoica, I. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. NeurIPS, 2023.

[14] George, M.L. *Lean Six Sigma: Combining Six Sigma Quality with Lean Production Speed*. McGraw-Hill, 2002.