Оценка планирования ИИ-агентов в аналитике контакт-центров

не клиентское обслуживание,
а аналитика контакт-центра

Олег Зельдин. Апекс Берг

Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027

SYDNEY, Australia, June 25, 2025

> **Более 40% проектов с ИИ-агентами будут отменены к концу 2027 года** из-за:
>
> - **“растущих затрат;”**
> - **“неясной бизнес-ценности;”**
> - **“недостаточного контроля рисков.** G gartner.com **”**

Reuters передал это как рыночный сигнал: крупные поставщики активно инвестируют в ИИ-агентов, но Gartner предупреждает, что многие проекты пока остаются ранними экспериментами или пилотами, часто движимыми ажиотажем, а не ясной экономикой. Reuters

Еще важная деталь: Gartner использовал термин **agent washing** — когда существующие чат-боты, помощники или RPA-решения переупаковывают как «агентный ИИ», хотя у них нет существенных агентных возможностей. Gartner оценивал, что из тысяч поставщиков агентного ИИ реально агентными являются около 130. G gartner.com

Источник:

Tool-Aware Planning in Contact Center AI: Evaluating LLMs through Lineage-Guided Query Decomposition

Varun Nathan, Shreyas Guha, Ayush Kumar

https://arxiv.org/abs/2602.14955

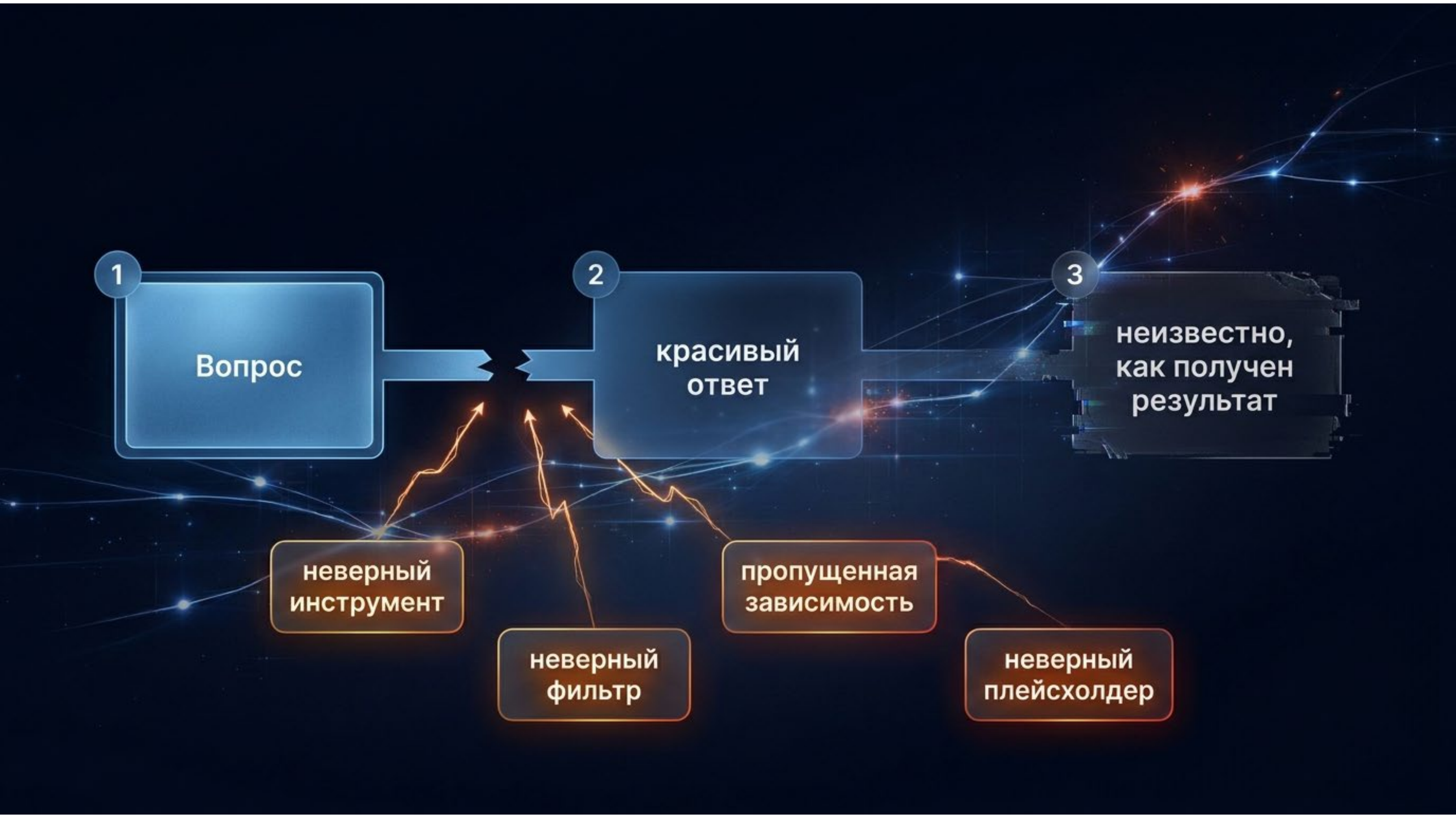
Область доклада

Агент обслуживания

ведет диалог, отвечает, передает оператору

Аналитический агент

работает в контуре руководителя/аналитика, понимает источники, фильтры, строит цепочки анализа


1
Вопрос
2
красивый ответ
3
неизвестно, как получен результат
неверный инструмент
неверный фильтр
пропущенная зависимость
неверный плейсхолдер

«Сравнить оценки контроля качества по критериям “профессионализм” и “процедуры решения проблемы” для нерешенных звонков, в которых тональность клиента изменилась с негативной на позитивную»

Оценка качества плана

покомпонентная и по сравнению с эталоном

Улучшение планирования

цикл исправления ошибок и сохранения истории

Сравнение моделей

тестирование 14 больших языковых моделей на планировании


«Ответ»
проверка результата
«План»
проверка метода
получения результата

Инструменты среды


языковая модель
для объединения
результатов (LLM)
расшифровки
разговоров (RAG)
структурированные
данные (T2S)

T2S / Snowflake

«counts» «trends» «QA metrics» «flags» «duration» «call drivers»

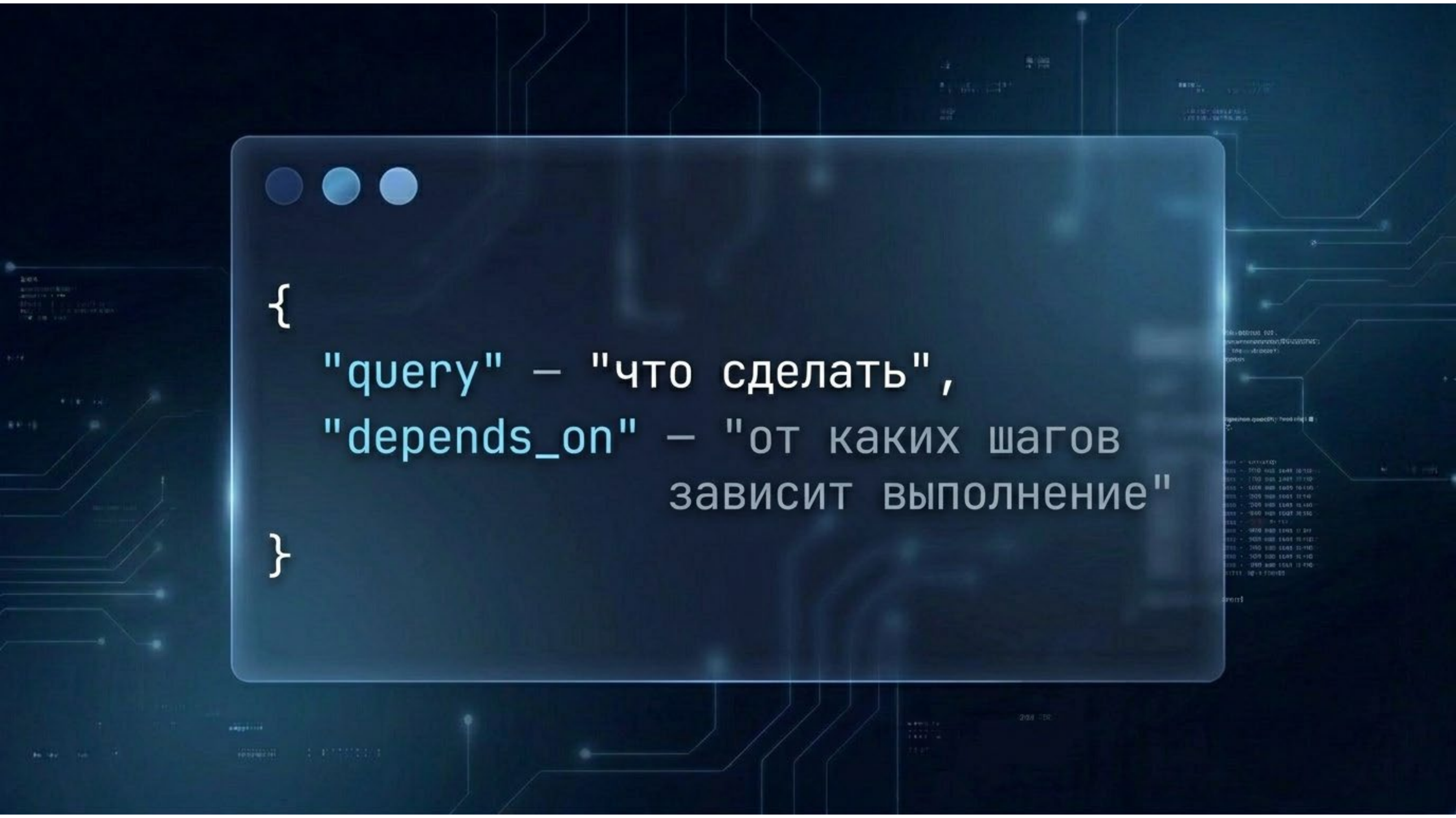
RAG / Transcripts

«текст разговоров» «темы» «формулировки» «sentiment внутри звонка»

LLM

синтез, переформатирование, извлечение ID, объединение результатов

Требования к структуре плана


{
"query" – "что сделать",
"depends_on" – "от каких шагов
зависит выполнение"
}

«2. Как устроен tool-aware plan»

«План — это не текстовое рассуждение, а структура: шаг, инструмент, инструкция, зависимости и исполнимый формат.»

Домен
контакт-центры

Задача
аналитические запросы

Инструменты
T2S/Snowflake, RAG/транскрипты, LLM-синтез

Фокус оценки
качество плана

Бенчмарк: 14 LLM, 600 запросов

$$P \equiv \langle s_k \rangle_{k=1}^{n}, s_k = (t_k, p_k, D_k)$$

Выход: JSON-план

Инструмент

Инструкция / prompt

Зависимости от предыдущих шагов

T2S / Snowflake

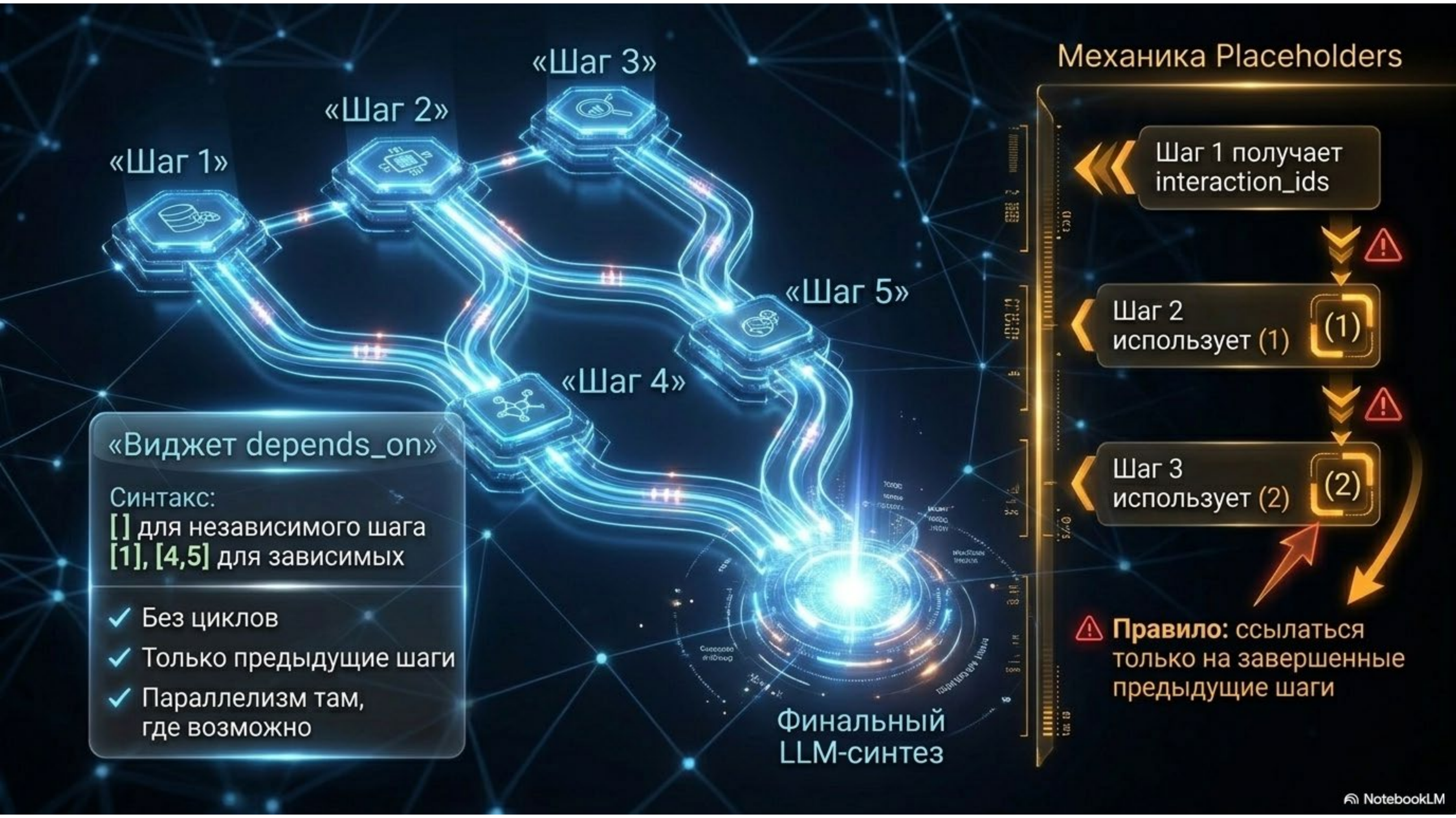
Структурированные данные: counts, counts, trends, QA metrics, flags, duration, call drivers

RAG / Transcripts

Текст разговоров, темы, формулировки, sentiment внутри звонка

LLM

Синтез, переформатирование, извлечение ID, объединение результатов


«Шаг 1»
«Шаг 2»
«Шаг 3»
«Шаг 4»
«Шаг 5»
«Виджет depends_on»
Синтакс:
[ ] для независимого шага
[1], [4,5] для зависимых
✓ Без циклов
✓ Только предыдущие шаги
✓ Параллелизм там, где возможно
Финальный LLM-синтез
Механика Placeholders
Шаг 1 получает interaction_ids
Шаг 2 использует (1)
(1)
Шаг 3 использует (2)
(2)
Правило: ссылаться только на завершенные предыдущие шаги
NotebookLM

«Правило без повторной фильтрации»

Правильно

unresolved

Далее передается список ID

Неправильно

unresolved

unresolved (re-apply)

unresolved (re-apply)

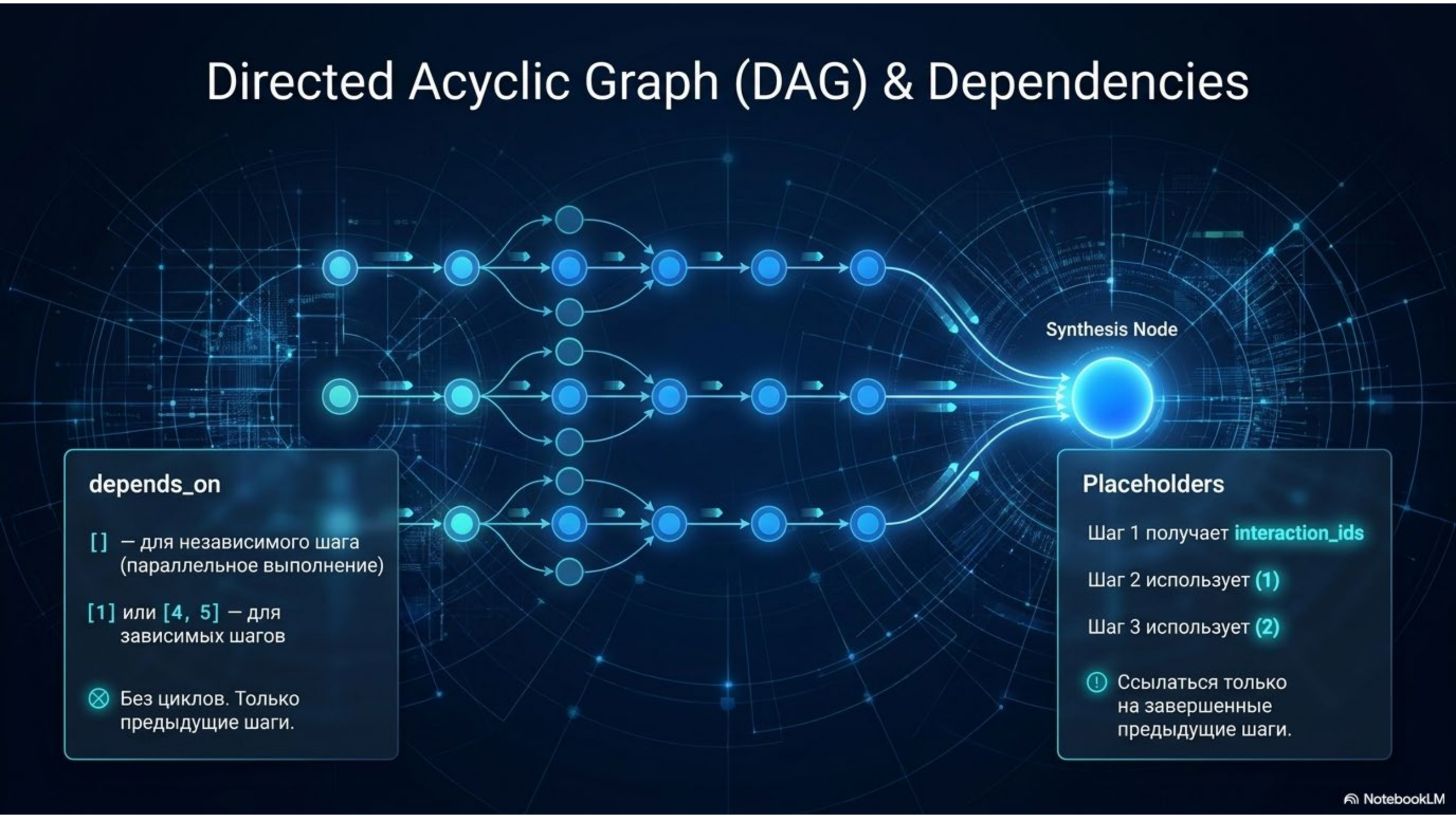
Риски: лишние join, задержка, рассинхронизация условий

Пример правильной декомпозиции

1. T2S unresolved interaction_ids
2. RAG sentiment negative→positive по (1)
3. LLM извлекает IDs из (2)
4. T2S QA resolution procedures по (3)
5. T2S QA professionalism по (3)
6. LLM сравнивает (4) и (5)


шаг 1 → шаг 2 → шаг 3
1
2
3
4
5
шаги 4 и 5 могут выполняться параллельно

Directed Acyclic Graph (DAG) & Dependencies



depends_on

`[]` — для независимого шага (параллельное выполнение)

`[1]` или `[4, 5]` — для зависимых шагов

⊗ Без циклов. Только предыдущие шаги.

Placeholders

Шаг 1 получает **interaction_ids**

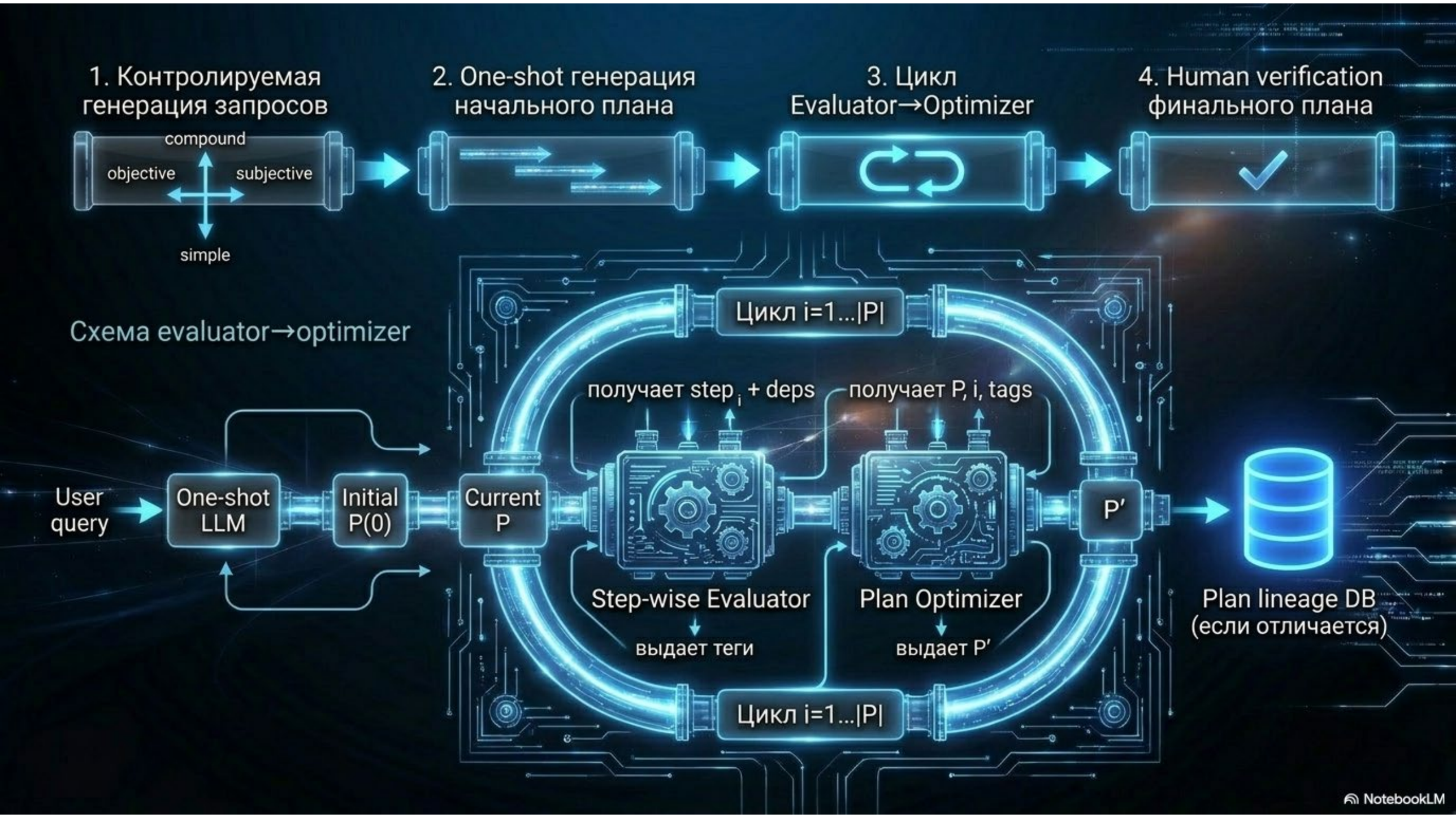
Шаг 2 использует **(1)**

Шаг 3 использует **(2)**

ⓘ Ссылаться только на завершенные предыдущие шаги.

Процесс получения эталона


запрос
первый план
(черновой
вариант за один
проход)
проверка
улучшение


1. Контролируемая генерация запросов
compound
objective
subjective
simple
2. One-shot генерация начального плана
3. Цикл Evaluator→Optimizer
4. Human verification финального плана
Схема evaluator→optimizer
Цикл i=1…|P|
получает step i + deps
получает P, i, tags
User query
One-shot LLM
Initial P(0)
Current P
Step-wise Evaluator
выдает теги
Plan Optimizer
выдает P′
P′
Plan lineage DB (если отличается)
Цикл i=1…|P|
NotebookLM

«INCORRECTTOOL»
Неверный инструмент

«COMPLEXPROMPT»
Слишком сложный prompt

«MULTITOOLPROMPT»
Один prompt пытается использовать несколько инструментов

«REPEATEDDETAIL»
Повторяющаяся деталь

«INCORRECTPROMPT»
Неверный prompt

«NOCHANGE»
Менять не нужно

Правила цикла

Если P″=P
→ Следующий шаг

Если P″≠P
→ Сохранить в lineage

Длина плана не изменилась
→ Следующий шаг

Длина изменилась
→ Перепроверить тот же индекс

Остановка: полный проход без изменений ИЛИ max_passes=4

Плюсы offline-цикла

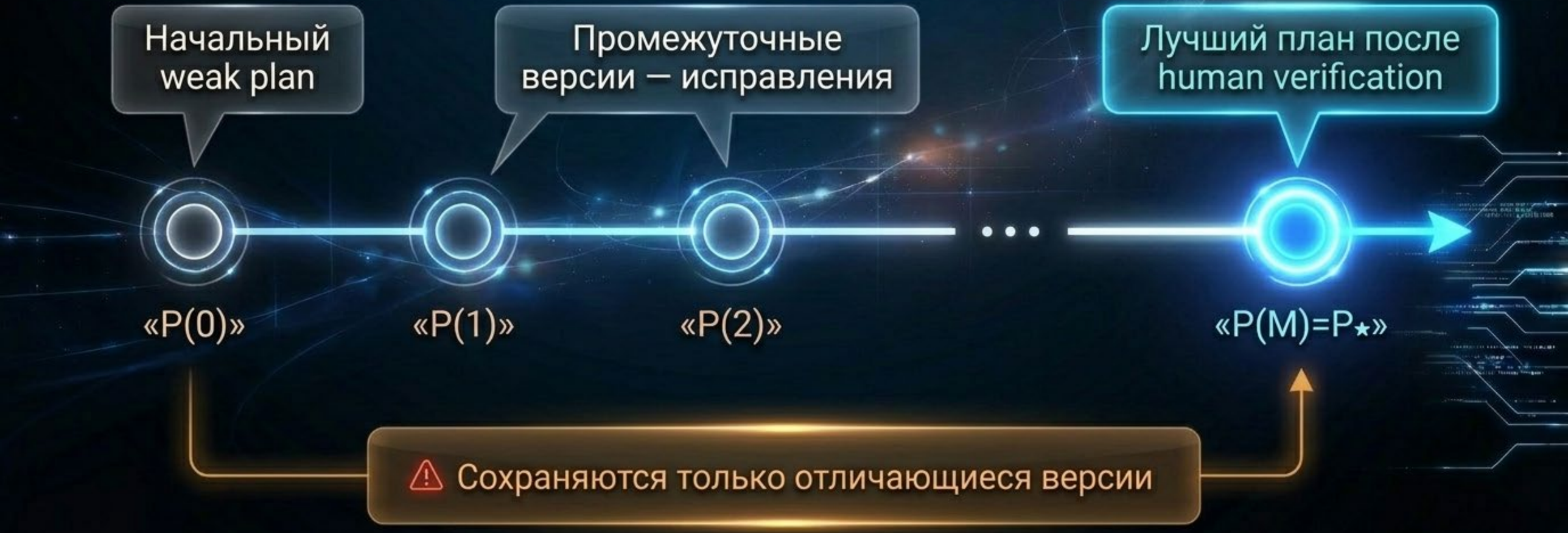
- Масштабируемая кураторская проверка
- Нет runtime-затрат
- Видимая история исправлений

Ограничения метода

- Инструменты не запускаются
- SQL не проверяется фактически
- Пустые join и race conditions не измеряются

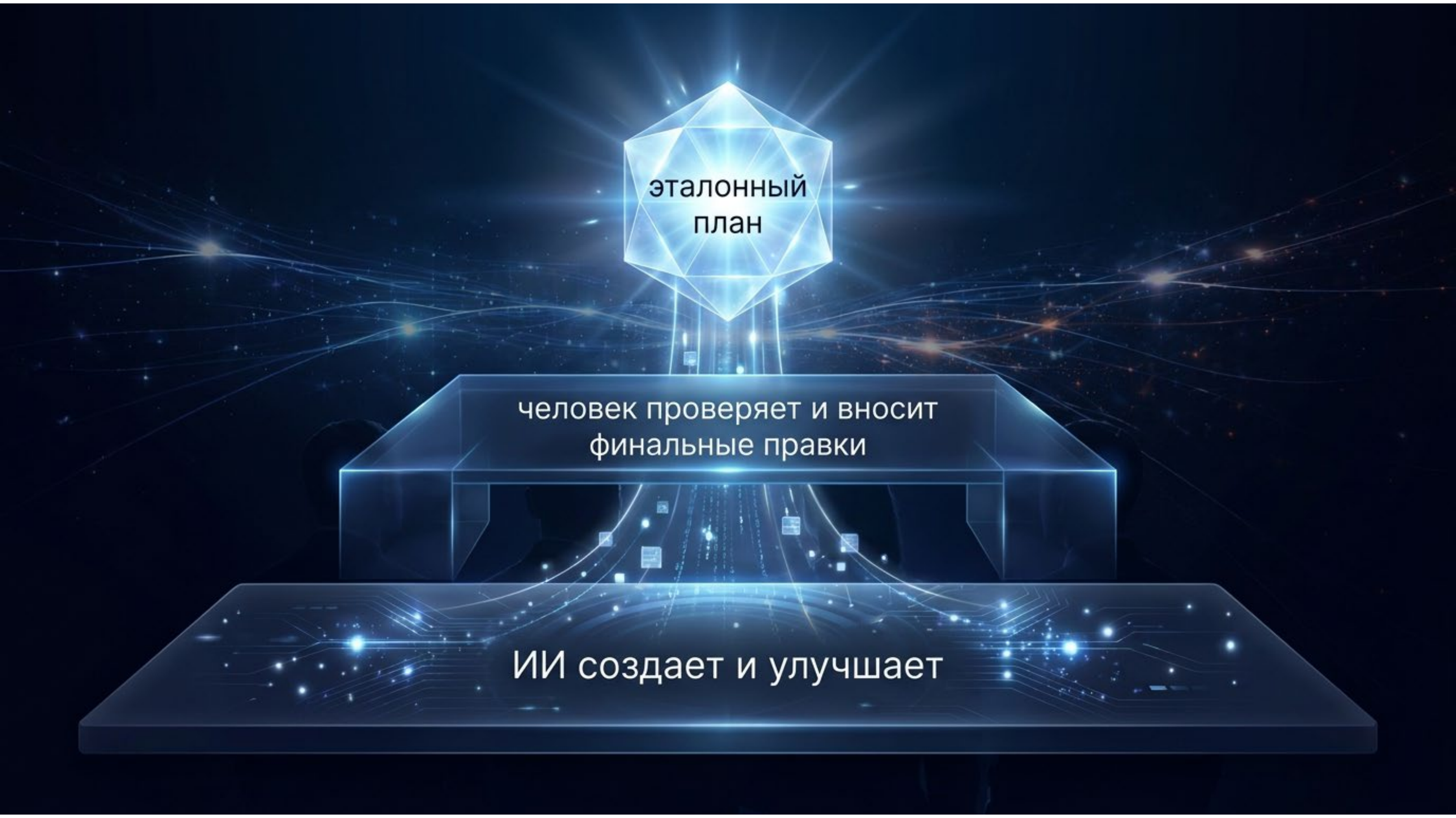
3. Lineage: как план исправляет сам себя

Тезис: Lineage хранит версии плана и показывает, где агент ошибался: инструмент, промпт, зависимость, формат, избыточность.



Lineage: Эволюция и Контроль Версий Плана

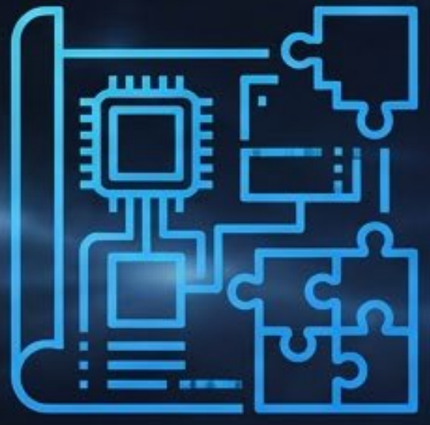
«Lineage хранит версии плана и показывает, где агент ошибался: инструмент, промпт, зависимость, формат, избыточность»


эталонный
план
человек проверяет и вносит
финальные правки
ИИ создает и улучшает

ПРОЦЕСС АНАЛИЗА ЗВОНКОВ

Оценка качества планов

«Два способа оценки качества»



Покомпонентная оценка: 7 критериев

Единая оценка: сравнение с эталоном, точность, полнота, F1

Компонентная оценка

Effectiveness (70 баллов):
Корректность и исполнимость плана

- Tool–Prompt Alignment (20)
- Format Correctness (20)
- Step Executability / Atomicity (15)
- Query Adherence (15)

$$SCORE(P) = 100 \cdot \sum w_k m_k$$

Efficiency (30 баллов):
Оптимальность маршрута

- Dependencies (10)
- Redundancy (10)
- Tool-Usage Completeness (10)

Effectiveness (70 баллов):
Корректность и исполнимость плана

Tool–Prompt Alignment (20)

Format Correctness (20)

Step Executability / Atomicity (15)

Query Adherence (15)

$$SCORE(P) = 100 \cdot \sum w_k m_k$$

«Компонент 1: формат плана»

ANALYTICAL FRAMEWORK | SYSTEM ARCHITECTURE v4.5

КРИТЕРИЯ

Формат

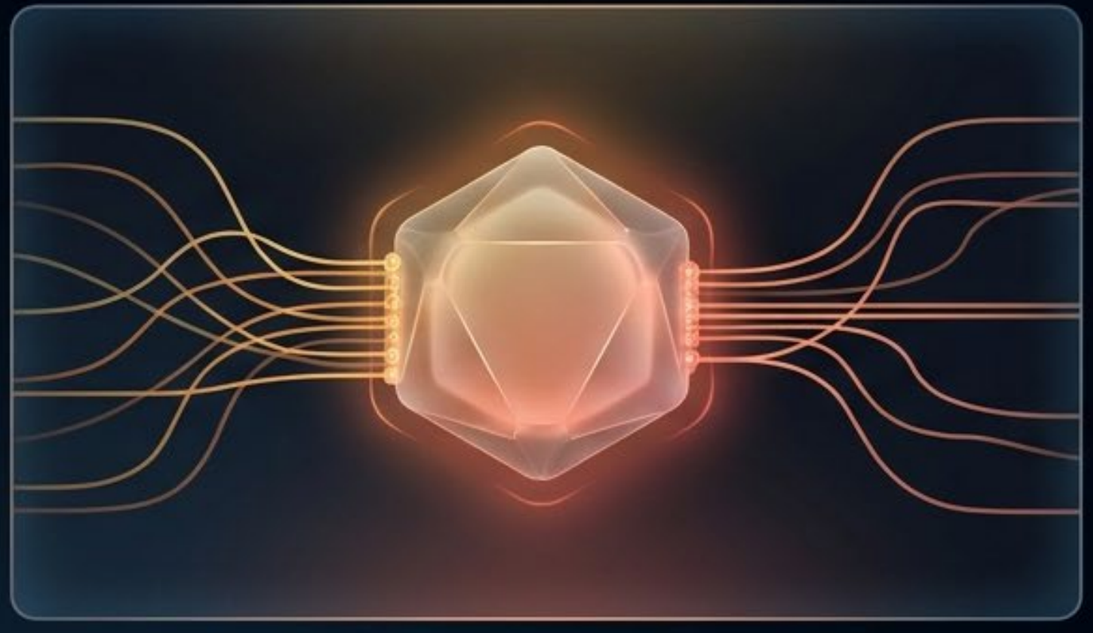
Можно ли план прочитать и выполнить как структуру?

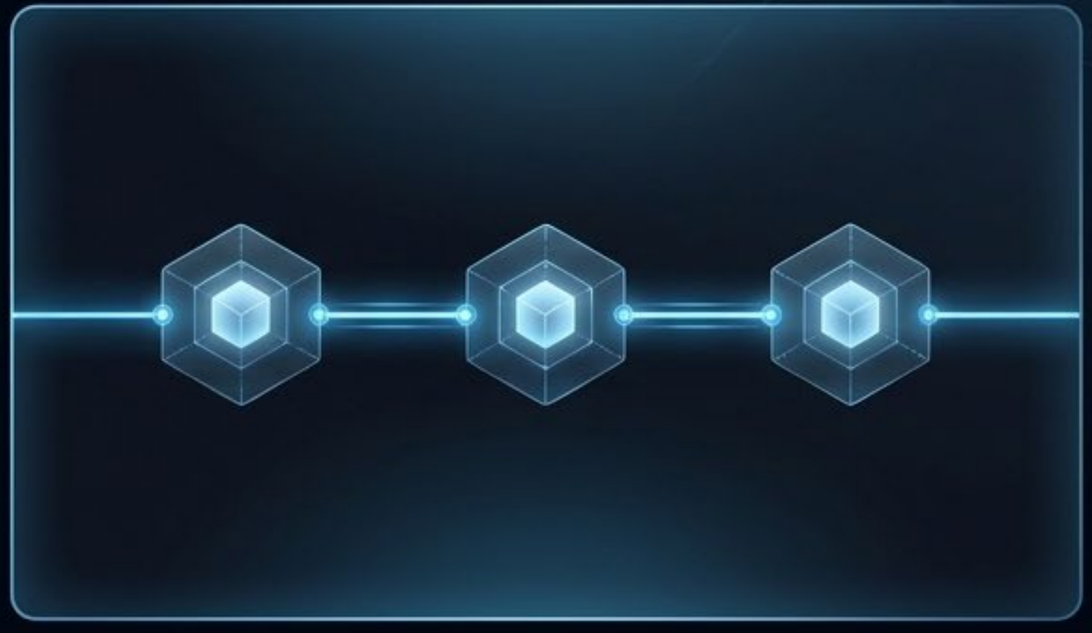
«Компонент 2: соответствие инструмента заданию»

Инструмент соответствует задаче

«Компонент 3: исполнимость шага»

Шаг можно реально выполнить





один шаг содержит сразу
несколько разных задач

Компонент 4: соответствие исходному вопросу

План отвечает именно на заданный вопрос

нерешенные звонки

≠

все звонки с негативной тональностью

$$SCORE(P) = 100 \cdot \sum w_k m_k$$

Efficiency (30 баллов):
Оптимальность маршрута

Dependencies (10)

Redundancy (10)

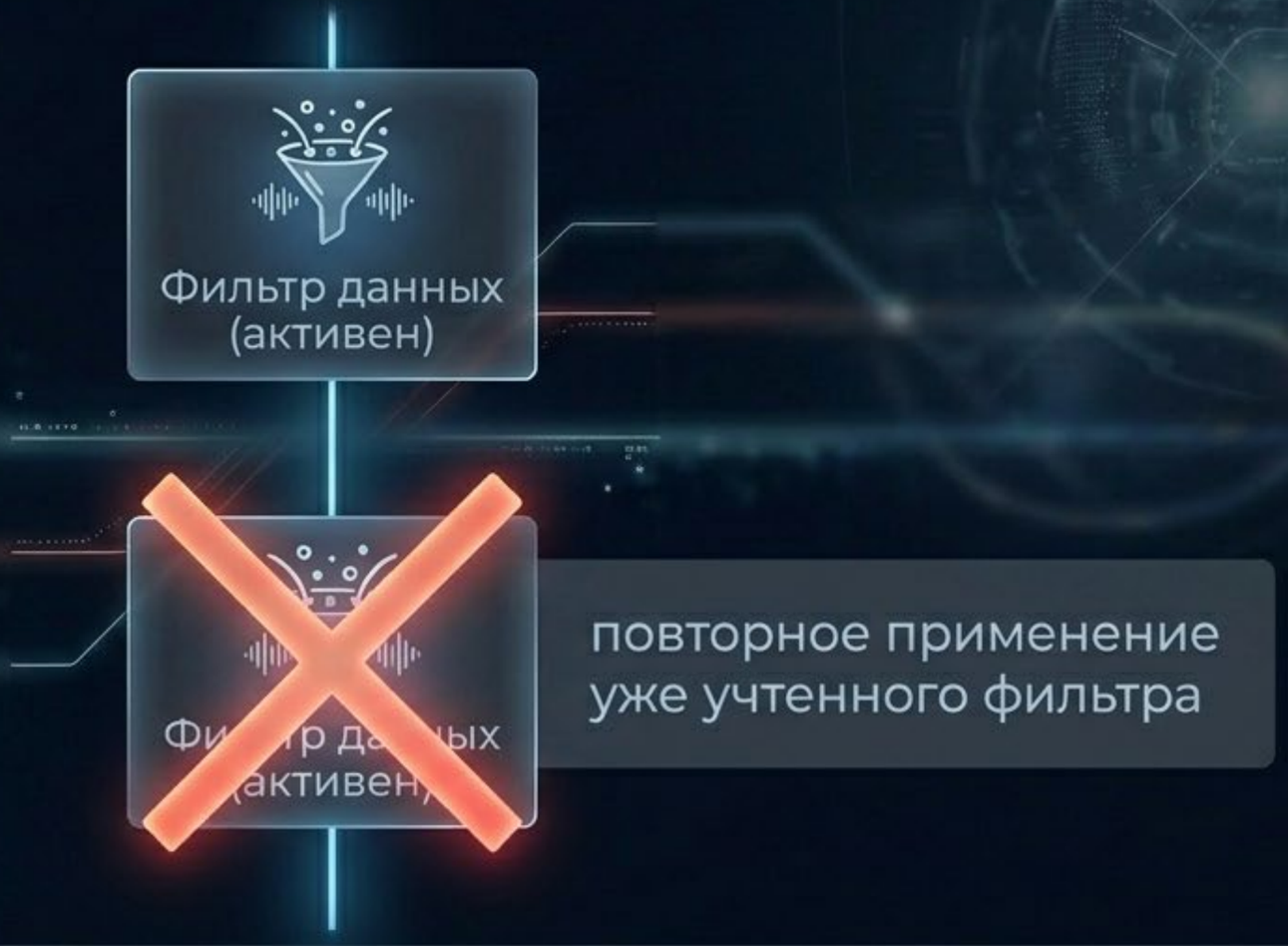
Tool-Usage Completeness (10)

«Компонент 5: корректность зависимостей»

Зависимости между шагами заданы правильно

Компонент 6: отсутствие избыточности

Нет лишних и повторяющихся действий



«Компонент 7: полнота использования инструментов»

Использованы все необходимые источники и инструменты

«Как из компонентов получается общая оценка»

«Единая оценка всего плана: точность, полнота и F1»

Оценка планов ИИ-агентов: Две ключевые метрики

ТОЧНОСТЬ (PRECISION)

«Какая доля из предложенного плана действительно правильная?»

Показывает, насколько шаги, сгенерированные ИИ, соответствуют эталонному (идеальному) плану.

Высокая точность может маскировать то, что ИИ предложил слишком мало шагов, боясь ошибиться.

ПОЛНОТА (RECALL)

«Какую часть обязательных шагов модель смогла покрыть?»

Показывает, сколько необходимых (критических) действий из эталонного плана ИИ **не пропустил**.

Высокая полнота может маскировать то, что ИИ добавил много лишнего "мусора" (галлюцинаций), лишь бы ничего не забыть.

F1-мера: Балансировка качества планирования

Защитный механизм, наказывающий ИИ за экстремальные перекосы в логике.

$$F1 = \frac{(2 \times \text{Точность} \times \text{Полнота})}{(\text{Точность} + \text{Полнота})}$$

Практический пример расчета агента:

- **Точность: 0.8** (80% предложенных шагов верны)
- **Полнота: 0.5** (Но ИИ нашел лишь 50% от всех нужных шагов)

F1-мера: $\frac{2 \times 0.8 \times 0.5}{0.8 + 0.5} = \frac{0.8}{1.3} \approx \mathbf{0.62}$

Вывод: Несмотря на высокую Точность (0.8), итоговая оценка снижена (0.62) из-за просадки в Полноте.

⚖️ Прагматичный смысл метрики

F1-мера не позволяет ИИ-агенту выглядеть «хорошим» за счет читерства по одному показателю. Она объективно наказывает за два сценария:

[Сценарий А]
ИИ предложил мало шагов, но все правильные (Точность ↑, Полнота ↓).

[Сценарий Б]
ИИ не упустил ничего, но добавил массу лишних, ненужных действий (Полнота ↑, Точность ↓).

Результат: F1 показывает реальное, сбалансированное качество мышления агента.


«One-shot judge»
Extremely Good >95%
Very Good >85%
Good >75%
Acceptable >60%
Bad >45%
Very Bad >30%
Extremely Bad ≤30%
Метрики: Precision, Recall, F1 на уровне шагов
Не парсится JSON → Extremely Bad
«Треугольник контроля»
Кандидатный план P
Human-verified gold plans
Структурное сравнение
Меньше оценки по впечатлению
Рубрика оценки
Эталонный план P*
NotebookLM

Эксперимент и его результаты


600
запросов
контакт-центров
Train: 20
(конструирование промптов)
Validation: 80
(настройка оценщиков, Judge, step-wise evaluator, optimizer)
Test: 500
(бенчмарк моделей)
Публичный синтетический анонимизированный набор (200 запросов)
1
Claude-3-7-Sonnet
Claude-3-5-Haiku
Claude-Sonnet-4
2
Nova-Pro / Lite / Micro
3
Llama 1B / 3B / 70B
Llama4-Maverick-17B
4
GPT-4o
GPT-4o-mini
GPT-4.1-Nano
o3-mini
Режимы тестирования: with lineage / without lineage
NotebookLM

«Как был устроен эксперимент»

600
запросов

14
моделей

2
режима подсказки

«Результат 1: даже сильные модели не дают стабильного эталона»

лучший результат по единой оценке.

«Результат 2: по компонентам лучше, но ошибки остаются»

84,8

лучший общий результат по покомпонентной оценке.

Lineage prompting effect

Step Executability улучшилась у 9 из 14, но общий балл часто выше без lineage

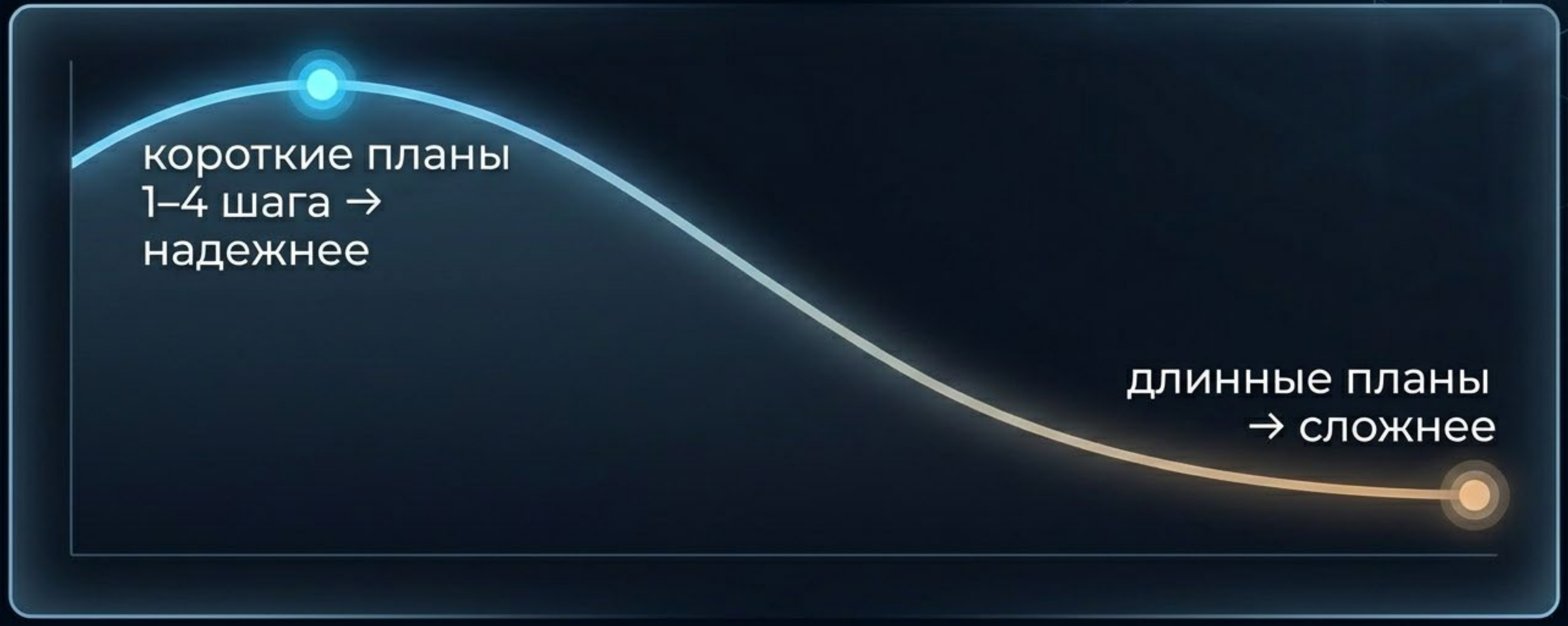
(on 14 models)

Самые слабые метрики

Metric	with lineage	vs without lineage
Tool Usage Completeness ⚠	48,74	71,12
Tool–Prompt Alignment ⚠	64,36	69,10
Format ⚠	70,43	74,26

Note: Lower scores indicate weaker performance.
Colors denote critical failure zones.

«Результат 3: длинные и составные планы сложнее»



«Результат 4: история планов помогает не всегда»

«Главный вывод эксперимента»

«Качество планирования — отдельная задача. Она еще не решена автоматически


Snowflake/T2S:
цифры, статусы, QA,
длительность,
флаги
Запросы,
требующие
обоих
источников
RAG/Transcripts:
смысл разговора,
формулировки,
причины, sentiment
Tool–Prompt Alignment
Format
Step Executability
Query Adherence
Dependencies
Redundancy
Tool-Usage Completeness
AI-агенту нужен не только prompt,
а проверяемая операционная модель
Исполнимый план
Эталонные планы
Lineage исправлений
Метрики допуска
Убедительный ответ VS Проверяемый маршрут
NotebookLM

«Практический вывод: начинать с контроля планов»

карта инструментов

эталонные запросы

оценка планов

короткие задачи

история исправлений

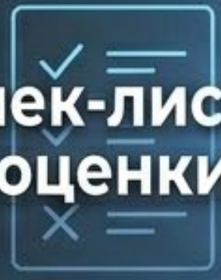
«Что можно внедрить у себя»

30–50 вопросов

Этап 1

эталонные планы

Этап 2



чек-лист оценки

Этап 3

07

пилот

Финальный этап

«Где нужен осторожный подход»

сложные запросы

длинные планы

управленческие решения по людям

Качественная ИИ-аналитика начинается с качественного плана