

# Конспект для обсуждения

Курс «Сигнал» · Модуль 1 · «Задача»

Подготовлено по транскрипту лекции. Формат — выдержка ключевых идей, а не пересказ.

## Зачем этот документ

Это не “сжатая стенограмма”, а рабочий handout для обсуждения лекции: с главными тезисами, спорными местами, вопросами для группы и идеями для практики.

## Главный вопрос лекции

**Как ставить задачу модели так, чтобы получить не убедительно выглядящий мусор, а результат, который можно использовать в работе и защищать перед другими.**

## Ключевые тезисы

1. Все, что генерирует модель, по умолчанию стоит считать slop’ом, даже если это выглядит убедительно и даже если местами это правда.
2. Главное преимущество человека перед моделью — не “ум вообще”, а контекст, опыт и понимание того, что в этой задаче считается хорошим результатом.
3. Качество ответа сильно зависит не только от модели, но и от того, насколько явно задан контракт на результат до начала генерации.
4. Контракт и проверка — это разные вещи: сначала мы описываем, что нужно получить, потом отдельно оцениваем, годится ли получившийся результат.
5. Работа с моделью — это не “нажал и получил”, а цикл из авторства, постановки задачи, редактуры, проверки и точных правок.
6. Даже сильная модель не отменяет необходимости проверки, а слабая модель резко сужает класс задач, где можно получить надежный результат.
7. Безопасность — часть постановки задачи: модель не понимает автоматически, какие данные чувствительные, и ей нельзя доверять это решение.

## Ключевые идеи

### 1. AI slop — не сбой, а базовый режим

**Суть:** Лекция предлагает жесткую исходную позицию: любой ответ модели нужно считать потенциально ненадежным. Особенно опасны быстрые, уверенные ответы без пруфов, потому что именно они сильнее всего создают иллюзию качества.

**Почему это важно:** Это сдвигает ожидание: проблема не в том, что “модель иногда ошибается”, а в том, что пользователь не должен по умолчанию принимать ее ответ за готовое знание.

**Пример из лекции:** В эксперименте одна и та же задача про конкурентный анализ дала четыре разных типа поведения: от уверенного быстрого ответа до отказа отвечать на слишком широкую постановку.

**Ограничение / риск:** Если этот тезис понять слишком буквально, можно скатиться в паранюю и потерять практическую пользу моделей вообще.

**Вопрос для обсуждения:** Где проходит граница между здоровым недоверием к модели и бесполезной тотальной перепроверкой всего?

## 2. Опыт человека — это не дополнение, а основной рычаг

**Суть:** Модель не знает, какие параметры в задаче важны именно для вас: где границы рынка, какие вопросы задаст руководитель, что нужно включить в ответ, а что можно игнорировать. Это знает человек, который находится в контексте задачи.

**Почему это важно:** Отсюда главный вывод лекции: сила пользователя не в том, чтобы “написать волшебный промпт”, а в том, чтобы перевести свой опыт в инструкцию для модели.

**Пример из лекции:** В конкурентном анализе именно человек понимает, важны ли цена, фичи, рынок РФ против глобального рынка, ожидания SEO и команды. Для модели это неочевидно, пока это не описано.

**Ограничение / риск:** Если человек сам плохо понимает задачу, он не сможет компенсировать это “хорошей моделью”.

**Вопрос для обсуждения:** Что важнее для качества результата: сила модели или зрелость постановки задачи со стороны человека?

## 3. Контракт на результат — это описание хорошего ответа до генерации

**Суть:** Контракт на результат — это способ заранее определить, как выглядит хороший результат, чтобы модель не целилась в “ну вроде сойдет”. В лекции он описан как перевод опыта пользователя в инструкцию для модели.

**Почему это важно:** Без контракта модель заполняет пробелы сама. С контрактом у нее хотя бы появляется шанс приблизиться к тому, что действительно нужно.

**Пример из лекции:** Для примера с конкурентным анализом в контракт входили: источники, формат ответа, ограничения, правила приемки и критерий готовности. После этого модели тратили больше времени, но результаты становились полезнее.

**Ограничение / риск:** Контракт не превращает слабую модель в сильную и не гарантирует отсутствие галлюцинаций.

**Вопрос для обсуждения:** Какие задачи в вашей работе уже сейчас можно было бы радикально улучшить просто за счет явного контракта на результат?

## 4. Нужно просить не документ, а результат

**Суть:** Один из самых сильных тезисов лекции: часто мы ставим модели ложную цель. Нам не нужен “документ”, “лендинг” или “код” как таковые; нам нужен эффект, решение, понимание, выбор, действие.

**Почему это важно:** Когда пользователь просит промежуточный артефакт вместо цели, модель может идеально попасть в форму и полностью промахнуться по содержанию.

**Пример из лекции:** В конкурентном анализе цель не “написать анализ”, а помочь принять решение. Значит, критерии должны быть завязаны на возможность понять рынок и сделать выбор, а не просто на объем текста.

**Ограничение / риск:** Не всегда легко быстро назвать настоящую цель задачи; иногда люди сами мыслят через артефакт, а не через решение.

**Вопрос для обсуждения:** Какие типичные задачи вашей команды сейчас сформулированы через артефакт, хотя должны быть сформулированы через результат?

## 5. Контракт и чек-лист — это два разных инструмента

**Суть:** Контракт нужен до генерации, чек-лист — после. Контракт определяет ожидания, чек-лист помогает оценить качество результата с разных точек зрения.

**Почему это важно:** Иначе легко смешать постановку задачи и приемку и получить хаос: “сделай хорошо” вместо понятных критериев.

**Пример из лекции:** Для чек-листа были названы отдельные критерии: полезность, точность, доказуемость, стиль, формат, риски. Это уже не инструкция модели, а рамка для оценки ответа.

**Ограничение / риск:** Даже хороший чек-лист не поможет, если человек не готов реально читать и проверять результат.

**Вопрос для обсуждения:** Что в ваших задачах чаще всего ломается: сама постановка или приемка результата?

## 6. Роль человека меняется: автор → редактор

**Суть:** Лекция предлагает не отказываться от авторства, а сместить роль. Человек остается автором идеи и критерия качества, а модель становится исполнителем черновика. После этого человек снова входит в процесс как редактор.

**Почему это важно:** Это убирает фантазию, что модель “сама все поймет”, и возвращает ответственность за смысл пользователю.

**Пример из лекции:** В конкурентном анализе пользователь задает идею, рамку и критерии, а после генерации надевает “шапочку редактора” и проверяет, что получилось на самом деле.

**Ограничение / риск:** Если редакторского навыка нет, человек либо все принимает слишком легко, либо просто раздражается на результат и бросает.

**Вопрос для обсуждения:** Какие навыки редактора сейчас важнее всего для работы с LLM в вашей роли?

## 7. Когда не хватает компетенции, модель можно использовать для подготовки рамки

**Суть:** Если пользователь не знает, какие критерии качества вообще бывают в этом типе задач, модель можно сначала попросить собрать best practices, критерии и подходы — и только потом переходить к основной работе.

**Почему это важно:** Это делает модели полезными и за пределами текущей зоны компетенции, но без иллюзии, что можно перескочить проверку.

**Пример из лекции:** Предлагается сначала попросить модель помочь сформулировать критерии, контракт и чек-лист, а уже затем давать ей основную задачу. Для задач средней сложности спикер рекомендует минимум 2–3 подготовительных прохода до начала “настоящей” работы.

**Ограничение / риск:** Это повышает стоимость работы: путь через модель становится не короче “в один клик”, а более структурированным.

**Вопрос для обсуждения:** Готовы ли команды тратить дополнительное время на подготовку задачи, если это повышает надежность результата?

## 8. Безопасность — это часть зрелой работы с моделью

**Суть:** Модель сама не понимает, какие данные секретные или чувствительные. Поэтому пользователь обязан сам фильтровать, анонимизировать и минимизировать передаваемый контекст.

**Почему это важно:** Иначе риск не только в плохом ответе, но и в реальной утечке: коммерческой, юридической или персональной.

**Пример из лекции:** Спикер делит данные на публичные, внутренние, коммерческие и персональные и предлагает принцип минимизации: отправлять только то, без чего задача не решается, и по возможности обезличивать данные.

**Ограничение / риск:** Люди систематически недооценивают, сколько контекста уже достаточно, чтобы деанонимизировать компанию или кейс.

**Вопрос для обсуждения:** Какие типы данных в вашей компании сейчас сотрудники, скорее всего, уже сливают в модели, не считая это проблемой?

## Что здесь новое или спорное

- Радикальный тезис “все, что генерирует модель, — slop” полезен как дисциплина мышления, но может вызвать сопротивление из-за своей жесткости.
- Отказ модели отвечать на слишком широкую задачу подается как признак силы, а не слабости.
- Просьба к модели “сделай документ” трактуется как плохая постановка задачи; правильнее формулировать желаемый эффект, а не артефакт.
- Для серьезной задачи “первый заход” объявляется опасной рулеткой: перед основной работой нужны подготовительные проходы на формулировку критериев, контракта и чек-листа.
- Сильная модель может не только выполнять задание, но и находить ошибки в вашем эталоне, то есть проверять самого пользователя.

## Вопросы для группы

- В каких задачах вашей работы вы чаще всего принимаете убедительный ответ модели за полезный, не проверяя, отвечает ли он на реальную цель?
- Где у вас сегодня отсутствует контракт на результат, хотя именно из-за этого вы потом раздражаетесь на качество ответа?
- Какие критерии из чек-листа — полезность, точность, доказуемость, стиль, формат, риски — у вас чаще всего выпадают?
- В каких задачах “сильная модель + плохая постановка” уже приводила вас к ложному чувству качества?
- Какие данные вы бы точно запретили отправлять в внешние модели внутри своей команды?

## Что попробовать на практике

- Взять одну регулярную рабочую задачу и перед следующим запросом к модели явно прописать пять элементов контракта: источники, формат, ограничения, детализацию и критерии готовности.
- Для уже полученного ответа модели прогнать отдельный чек-лист из шести критериев: полезность, точность, доказуемость, стиль, формат, риски.
- В незнакомой доменной области сначала попросить модель помочь собрать критерии качества и best practices, и только потом переходить к выполнению основной задачи.

## Открытые вопросы

- Насколько далеко можно зайти за пределы своей компетенции, если опираться на контракт и проверку, но не иметь глубокого предметного знания?
- Где проходит граница между “достаточно сильная модель” и “недопустимо слабая модель” для конкретного класса задач?
- Можно ли формализовать хороший контракт на результат так, чтобы им массово пользовались команды, а не только отдельные продвинутые люди?
- Как practically отделять реальные источники от вторичного LLM-контента, если сама среда уже зашумлена?
- Где проверка должна оставаться у человека, даже если часть валидации можно отдать другой модели?

## Короткая выжимка в 5 строк

1. Модель по умолчанию генерирует не знание, а правдоподобный текст.
2. Пользователь выигрывает не моделью, а качеством постановки задачи.
3. Контракт на результат нужен до генерации.
4. Чек-лист нужен после генерации.
5. Без проверки и ограничений по данным работа с LLM быстро становится либо бесполезной, либо опасной.

Версия для чтения и обсуждения. Не является стенограммой лекции.