Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Как писать сложные промпты: фреймворк CRF

Превращаем запрос в ТЗ для модели: роль, контекст, задача, ограничения, формат ответа

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Сотрудники ставят модели вопрос вместо ТЗ: «проверь договор», «напиши текст». Ответ выходит усреднённым, его переписывают по пять-шесть раз, часть ошибок утекает в документы и письма, а компания делает вывод «ИИ бесполезен». Мы закрываем это регламентом: промпт становится версионируемым артефактом с ролью, контекстом, задачей, ограничениями и машинно заданным форматом вывода,...

Почему это важно бизнесу

- Переписывание ответов съедает часы юриста и бухгалтера — оплаченное время уходит в правку черновиков
- Уверенно выдуманная цифра или формулировка в договоре стоит дороже любой экономии на проверке
- Без фиксированного формата ответ нельзя загрузить в учётную систему — данные снова вбивают руками
- Разрозненные личные промпты не масштабируются: ушёл сотрудник — ушло и качество ответов
- Загрузка персональных данных и коммерческой тайны в облачную модель — необратимый инцидент

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

5 блоков

Role, Context, Task, Constraints, Format — обязательный каркас любого рабочего промпта

наш стандарт CRF

3-5

Примеров в few-shot: рекомендованный диапазон, задаёт тон и структуру точнее словесных описаний

раздел «Multishot prompting» офиц. документации

4

Максимум точек кэширования (cache_control) в запросе — больше API не принимает

раздел «Prompt caching» офиц. документации

5 мин

TTL кэша ephemeral по умолчанию; для длинных сессий включаем опцию 1 час

раздел «Prompt caching» офиц. документации

5 уровней

output_config.effort: low, medium, high (по умолчанию), xhigh, max — рутина на low, аудит дого...

раздел «Effort» офиц. документации

20-30

Эталонных задач в регресс-наборе: каждую правку промпта прогоняем на них

наш стандарт

РЕАЛИЗАЦИЯ 1 ИЗ 3

Шаблон CRF в XML-разметке и библиотека промптов

Что настраиваем

Стандартные задачи юриста, бухгалтерии и продаж; системная часть запроса + карточка задачи

Как мы это делаем

1. Каркас пишем тегами: <role>, <context>, <task>, <constraints>, <format> — модель однозначно разделяет инструкцию и данные
2. Стабильную часть (роль, регламент, глоссарий) выносим в system, переменную — в messages: порядок рендеринга tools → system → messages
3. В Constraints жёстко: объём в знаках, запрет вводных оборотов, «опирайся только на приложенные документы, недостающее помечай как неизвестное»
4. Формат отдаём не словами, а JSON Schema через output_config.format (устаревший параметр output_format не используем); для инструментов — strict: true в самом описан…
5. Шаблоны храним в git с версией и changelog: правка промпта проходит ревью, как правка кода

РЕЗУЛЬТАТ

Ответ приходит сразу в целевой структуре и загружается в учётную систему без ручного разбора. Один и тот же шаблон даёт сопоставимый результат у любого сотрудника, а не только у автора промпта.

КЛЮЧЕВОЙ НЮАНС

Assistant-prefill как способ задать формат на актуальных моделях возвращает 400 — формат задаём схемой в output_config.format или инструкцией в system.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Декомпозиция: план → шаги → сборка, вместо монолита

Что настраиваем

Длинные задачи: аудит договора, разбор регламента, выжимка из пакета документов

Как мы это делаем

1. Первый вызов — только план работ и список проверяемых пунктов; ничего не пишем «сразу набело»
2. Каждый пункт плана исполняем отдельным запросом со своим Constraints; результаты собираем на своей стороне
3. Рассуждение включаем параметром thinking типа adaptive, глубину задаём output_config.effort; фиксированный budget_tokens на актуальных моделях отклоняется с ошибкой…
4. Стабильный префикс (регламент, приложенные документы) помечаем cache_control ephemeral и проверяем usage.cache_read_input_tokens; префикс короче минимума (512-4096…
5. Для сложных логических задач держим англоязычный вариант шаблона как запасной

РЕЗУЛЬТАТ

Цепочка коротких шагов даёт проверяемый промежуточный результат: видно, где именно модель ошиблась, и переделывается один шаг, а не весь ответ. Кэш префикса заметно снижает стоимость повторных прогонов по одному документу.

КЛЮЧЕВОЙ НЮАНС

Если cache_read_input_tokens стабильно ноль — в начале префикса стоит текущая дата, UUID или несортированный JSON; их убираем в хвост запроса.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Контур проверки и защита данных

Что настраиваем

Все промпты, работающие с договорами, отчётностью и перепиской клиентов

Как мы это делаем

1. Регламент: персональные данные, реквизиты и коммерческая тайна в облачные модели не уходят — обезличиваем или работаем в закрытом контуре
2. Требуем ссылок на исходник: документы передаём отдельными document-блоками с citations enabled, ответ без цитаты считаем черновиком
3. Citations и output_config.format в одном запросе несовместимы (ошибка 400) — сценарий с цитатами и сценарий со строгой схемой разводим по разным вызовам
4. Все цифры, сроки и правовые формулировки визирует человек — это отдельный пункт регламента, а не рекомендация
5. Держим регресс-набор из 20-30 задач с эталонными ответами и прогоняем его после каждой правки шаблона

РЕЗУЛЬТАТ

Ошибки ловятся до того, как попадут в документ клиента. Правка промпта перестаёт быть лотереей: видно, улучшила она результат или сломала соседний сценарий.

КЛЮЧЕВОЙ НЮАНС

Промпт усиливает эксперта, но не заменяет его: там, где сотрудник не может проверить ответ, автоматизацию не внедряем.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

✕ Вопрос вместо задачи

«Проверь договор» не задаёт ни роли, ни критериев. Добавляем Role и Task с одним чётким действием — и получаем разбор вместо отписки.

✕ Монолит на пять задач сразу

В одном запросе смешаны анализ, вывод и оформление. Разбиваем на цепочку вызовов: план, затем по одному пункту, затем сборка.

✕ Пустой блок Constraints

Без рамок модель добирает объём водой. Задаём предел в знаках, запрет вводных фраз и правило «только из приложенных данных».

✕ Формат описан прозой

«Выведи таблицей» трактуется свободно. Описываем структуру JSON Schema через output_config.format, тогда поля и типы гарантированы.

✕ Prefill для навязывания формата

Приём из старых моделей: на актуальных версиях префилл ассистента возвращает 400. Заменяем структурированным выводом или инструкцией в system.

✕ Дата и UUID в начале system

Любой изменённый байт в префиксе сбрасывает кэш: cache_read_input_tokens падает в ноль. Волатильные данные переносим за последнюю точку кэширования.

✕ Ставка на temperature

На моделях поколения 4.7 и новее temperature, top_p и top_k отклоняются с ошибкой 400. Стабильность даёт структура промпта, схема вывода и уровень ef…

✕ ПДн и тайна в облачном чате

Отправленное в чужой сервис не отзывается. Обезличиваем данные до отправки, чувствительные сценарии выносим в закрытый контур.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Один шаблон CRF на отдел: роль, контекст, задача, ограничения, формат
- Правило «только из приложенных данных» в каждом промпте
- Обязательная проверка человеком всех цифр и правовых формулировок
- Запрет на загрузку ПДн и коммерческой тайны в облачные сервисы

НОРМАЛЬНО

- Библиотека промптов в git с версиями и ревью изменений
- Few-shot: 3-5 эталонных примеров вместо описания стиля словами
- Декомпозиция длинных задач на цепочку коротких вызовов
- Формат ответа задан схемой через output_config.format

ХОРОШО

- Регресс-набор 20-30 задач, прогон после каждой правки шаблона
- Кэширование стабильного префикса с контролем cache_read_input_tokens
- Подбор effort по задаче: low на рутину, high и xhigh на аудит, max — где цена ошибки…
- Логирование модели, версии промпта и параметров каждого вызова

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] В промпте явно заданы все пять блоков: Role, Context, Task, Constraints, Format?
- [ ] Задача сформулирована одним действием, а не списком из пяти разных задач?
- [ ] В Constraints есть предел объёма и запрет на выдумывание фактов?
- [ ] Формат ответа описан схемой (output_config.format), а не фразой «выведи таблицей»?
- [ ] Приложены 3-5 примеров нужного стиля вместо словесного описания тона?
- [ ] Стабильная часть вынесена в system, а волатильные данные — в конец запроса?
- [ ] Кэш реально работает: cache_read_input_tokens больше нуля на повторных вызовах?
- [ ] Есть регресс-набор задач, на котором проверяется каждая правка промпта?
- [ ] Назначен человек, который визирует цифры, сроки и правовые формулировки?
- [ ] Закреплён запрет на передачу ПДн и коммерческой тайны в облачные модели?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Собираем библиотеку CRF-промптов под ваши процессы и ведём её версии в git
- Переводим ответы модели в структурированный JSON под загрузку в 1С и CRM
- Настраиваем кэширование префикса и уровни effort — снижаем стоимость прогонов
- Строим регресс-набор эталонных задач и контроль качества правок промптов
- Пишем регламент работы с ИИ: что нельзя отправлять и кто визирует результат

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 Prompt engineering overview (docs.claude.com — 2026)

02 Use XML tags to structure your prompts (docs.claude.com — 2026)

03 Multishot prompting (few-shot examples) (docs.claude.com — 2026)

04 Extended thinking: adaptive thinking и output_config.effort (docs.claude.com — 2026)

05 Prompt caching (cache_control, TTL, breakpoints) (docs.claude.com — 2026)

06 Structured outputs (output_config.format, strict) (docs.claude.com — 2026)

07 Citations (document-блоки) (docs.claude.com — 2026)

08 Шаблон CRF-промпта из библиотеки ITfresh (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.