ТЕХНОЛОГИЯ + ГОТОВЫЕ ПРОМТЫ

Требования к продукту с ИИ: от идеи до спецификации

ИИ-агент собирает ровно то, что вы описали, а всё, что не описали, додумывает сам — и обычно не так, как вы хотели. Требования — это способ договориться с агентом заранее: что строим, для кого, как поймём, что готово, и чего делать не нужно. Хорошая новость: писать их самому почти не придётся. ИИ вытянет требования из вас вопросами, найдёт дыры и оформит документ.

Второй мозг

Работает как бизнес-аналитик: расспрашивает вас, раскладывает ответы на цели, пользователей, сценарии и функции, предлагает то, о чём вы не подумали.

Критик

Ищет противоречия, размытые формулировки, забытые ошибки и граничные случаи: пустые поля, медленный интернет, повторное нажатие, «злой» пользователь.

Оформитель

Собирает всё в спецификацию для агента: документ требований, критерии приёмки, список экранов, план задач и файл правил AGENTS.md.

Как пользоваться раздаткой

- Начинайте с результата раздатки 1: карточки контекста и одностраничника. Если их нет, хватит описания идеи в 5–7 предложениях.
- Ведите шаги 0–5 в одном чате, а проверку (шаг 6) делайте в другой нейросети или новом чате.
- Заменяйте всё, что в **[квадратных скобках]**. Результат каждого шага сохраняйте в один файл — к концу он и станет спецификацией.
- Первая версия требований не бывает идеальной. Её задача — чтобы агент построил то, что можно показать людям, а не всё сразу.

Технология: 8 шагов

Требования отвечают на пять вопросов по порядку: **зачем** (цель и метрика), **для кого** (пользователи), **что** (сценарии и функции), **как проверить** (критерии приёмки) и **в каких рамках** (ограничения и качество). Если пропустить первые два, третий превратится в список хотелок.

ШАГ	ЧТО ДЕЛАЕМ	РОЛЬ ИИ	МЕТОД
0	**Интервью-выжималка**: ИИ расспрашивает вас	Второй мозг	Интервью бизнес-аналитика: вопросы по блокам, по одному
1	**Цель, метрика, пользователи**	Второй мозг	Цель SMART, персоны и роли, Job Stories
2	**Карта историй** и нарезка версий	Второй мозг	User Story Mapping (Джефф Паттон), «ходячий скелет» первой версии
3	**Приоритеты**	Второй мозг + критик	MoSCoW, модель Кано (базовое / ожидаемое / восхищающее)
4	**Истории и критерии приёмки**	Оформитель	«Как... я хочу... чтобы...», проверка INVEST, «Дано / Когда / Тогда» (BDD)
5	**Качество и ограничения**	Второй мозг	Нефункциональные требования (ISO 25010 простыми словами), 152-ФЗ
6	**Стресс-тест требований**	Критик	Граничные случаи, негативные сценарии, поиск противоречий
7	**Экраны и данные**	Оформитель	Список экранов и их состояний, модель данных словами, текстовый прототип
8	**Спецификация для агента**	Оформитель	Spec-driven development: спецификация → план → задачи; файл AGENTS.md

Плохое и хорошее требование

РАЗМЫТО — АГЕНТ ДОДУМАЕТ САМ	КОНКРЕТНО — МОЖНО ПРОВЕРИТЬ
«Удобный интерфейс»	Новый пользователь получает первый результат не больше чем за 3 действия, без инструкции
«Быстро работает»	Страница открывается на телефоне с мобильным интернетом за 2–3 секунды
«Можно сохранить результат»	Кнопка «Скачать PDF» на экране результата; файл называется по дате и имени проекта
«Защищённые данные»	Имена и телефоны не хранятся; в базе только обезличенный номер и ответы теста

Правило: если про требование нельзя сказать «выполнено / не выполнено» за минуту проверки, его нужно переписать. Это умеет делать промт 9.

ВТОРОЙ МОЗГ

0 Интервью-выжималка

Самый быстрый способ получить требования — не писать их, а ответить на вопросы. Промт превращает нейросеть в бизнес-аналитика. Важно, что вопросы идут по одному: так вы отвечаете подробно и не пропускаете блоки.

ПРОМТ 1 · ИИ-АНАЛИТИК

Ты опытный бизнес-аналитик. Я делаю продукт: **[описание идеи или одностраничник из раздатки 1]**.
Твоя задача — собрать требования к первой версии. Проведи со мной интервью по блокам:
1) цель и как я пойму, что продукт работает; 2) кто пользователи и какие у них роли;
3) главный сценарий от входа до результата; 4) что пользователь вводит и что получает;
5) где и на чём он будет пользоваться (телефон, компьютер, офлайн); 6) какие данные храним;
7) что точно НЕ входит в первую версию; 8) ограничения: сроки, деньги, технологии.
Задавай по одному вопросу и жди ответа. Если ответ расплывчатый, переспроси и предложи 2–3 варианта на выбор. В конце каждого блока коротко подведи итог.

ВТОРОЙ МОЗГ

1 Цель, метрика, пользователи

Цель отвечает на вопрос «зачем», метрика — «как поймём, что получилось». Пользователей описываем не по возрасту и полу, а по ситуации: что человек пытается сделать, когда открывает продукт. Для этого удобен формат Job Story.

ПРОМТ 2 · ЦЕЛЬ И МЕТРИКА

По итогам интервью сформулируй:
1. Цель продукта в одном предложении: кому, какой результат, за какое время.
2. Метрику успеха первой версии: одна главная цифра и 2 вспомогательные. Пример: «7 из 10 тестовых пользователей дошли до результата без моей помощи».
3. Чем продукт НЕ является — 3–5 пунктов, чтобы агент не расширял задачу.

ПРОМТ 3 · РОЛИ И JOB STORIES

Опиши всех, кто будет пользоваться продуктом: **[например, ученик, учитель, администратор]**.
Для каждой роли:
— кто это и в какой ситуации он открывает продукт;
— 2–3 Job Story в формате «Когда **[ситуация]**, я хочу **[действие]**, чтобы **[результат]**»;
— что он может делать, а что ему запрещено (права доступа).
Отметь роль, без которой первая версия не имеет смысла. Остальные роли — кандидаты на «не сейчас».

ВТОРОЙ МОЗГ

2 Карта историй и нарезка версий

Список функций обманчив: в нём всё кажется одинаково важным. Карта историй Джеффа Паттона раскладывает продукт по пути пользователя. По горизонтали идут шаги пути — «хребет». Под каждым шагом в столбик — варианты, как его можно сделать, от простого к сложному. Затем проводим горизонтальную линию: всё, что выше неё, и есть первая версия.

	УЗНАТЬ О ПРОДУКТЕ	ПРОЙТИ ТЕСТ	УВИДЕТЬ РЕЗУЛЬТАТ	ПОДЕЛИТЬСЯ
Версия 1	Одна страница с кнопкой «Начать»	10 вопросов, по одному на экране	Итог текстом + 3 совета	Скопировать ссылку
Версия 2	Примеры результатов	Сохранение прогресса	График, PDF	Картинка для сторис
Потом	Отзывы, тарифы	Разные тесты	Личный кабинет, история	Сравнение с друзьями

Первая версия — «ходячий скелет»: каждый шаг пути есть, пусть самый простой. Это лучше, чем один идеально сделанный шаг и пустота вокруг.

ПРОМТ 4 · КАРТА ИСТОРИЙ

Построй карту пользовательских историй по методу Джеффа Паттона для главной роли: **[роль]**.
1. Хребет: 4–7 крупных шагов пути пользователя слева направо, от «узнал о продукте» до «получил результат и вернулся».
2. Под каждым шагом — 3–5 вариантов реализации, от самого простого к самому богатому.
3. Проведи линию первой версии: минимальный набор, при котором пользователь проходит путь от начала до конца.
Оформи таблицей: столбцы — шаги, строки — «Версия 1», «Версия 2», «Потом».

ПРОМТ 5 · СРЕЗАТЬ ЛИШНЕЕ

Посмотри на «Версию 1». Можно ли сделать её ещё меньше, чтобы собрать за 1–2 вечера с ИИ-агентом? Что можно временно заменить ручной работой, готовым сервисом или заглушкой (например, вместо регистрации — ссылка, вместо базы — файл)? Предложи урезанный вариант и перечисли, чем мы жертвуем.

ВТОРОЙ МОЗГ КРИТИК

3 Приоритеты: MoSCoW и Кано

MoSCoW делит функции на четыре корзины: Must (без этого не работает), Should (важно, но можно позже), Could (приятно иметь), Won't (точно не сейчас). **Модель Кано** добавляет взгляд пользователя: базовые функции никто не хвалит, но без них ругаются; ожидаемые — чем лучше, тем довольнее; восхищающие — их не ждут, но запоминают. В первую версию идут все базовые и одна-две восхищающие.

ПРОМТ 6 · РАССТАВИТЬ ПРИОРИТЕТЫ

Возьми все функции из карты историй. Для каждой укажи:
— корзину MoSCoW (Must / Should / Could / Won't) и почему;
— тип по Кано (базовая / ожидаемая / восхищающая);
— трудоёмкость для ИИ-агента: низкая / средняя / высокая.
Проверь, что Must не больше 5–7 пунктов. Если больше — предложи, что перенести. Найди одну восхищающую функцию, которую можно сделать дёшево.

ОФОРМИТЕЛЬ

4 Истории и критерии приёмки

Каждую функцию из Must превращаем в пользовательскую историю и проверяем по **INVEST**: независимая, обсуждаемая, ценная, оцениваемая, небольшая, проверяемая. К истории добавляем критерии приёмки в формате **«Дано / Когда / Тогда»** — этот формат пришёл из BDD-разработки. По нему агент поймёт, что именно делать, а вы — что проверить.

Пример. Как ученик, я хочу видеть, сколько вопросов осталось, чтобы не бросить тест на середине.
Дано ученик на 4-м вопросе из 10. **Когда** он открывает экран вопроса, **тогда** вверху видно «4 из 10» и полоску прогресса на 40%.
Дано ученик ответил на последний вопрос. **Когда** он нажимает «Далее», **тогда** открывается экран результата, а не 11-й вопрос.

ПРОМТ 7 · ИСТОРИИ С КРИТЕРИЯМИ

Для каждой функции из Must напиши:
1. Пользовательскую историю: «Как **[роль]**, я хочу **[действие]**, чтобы **[ценность]**».
2. 2–4 критерия приёмки в формате «Дано / Когда / Тогда»: основной сценарий и хотя бы один случай с ошибкой.
3. Оценку по INVEST: какие буквы не выполняются и как это исправить (например, разбить историю на две).

ВТОРОЙ МОЗГ

5 Качество и ограничения

Функциональные требования описывают, **что** делает продукт. Нефункциональные — **каким** он должен быть: быстрым, понятным, безопасным. Новички их почти всегда забывают, а агент без них выбирает за вас: например, сделает сайт только для компьютера или сохранит данные там, где их видно всем.

Чек-лист: о чём договориться

- [ ] **Устройства:** телефон, компьютер, оба; какие браузеры.
- [ ] **Скорость:** за сколько секунд открывается и отвечает.
- [ ] **Понятность:** можно ли разобраться без инструкции.
- [ ] **Доступность:** крупный шрифт, контраст, работа с клавиатуры.
- [ ] **Язык** интерфейса и тон текстов.
- [ ] **Надёжность:** что будет при ошибке или пропаже интернета.
- [ ] **Где живёт продукт:** один файл, хостинг, нужен ли сервер.
- [ ] **Стоимость:** платные API и ключи, лимиты.

Данные и закон

- Если продукт собирает имя, телефон, почту или результаты тестов конкретных людей — это персональные данные. Для россиян действует **152-ФЗ**: нужно согласие на обработку, политика конфиденциальности, хранение на серверах в России.
- Первую версию проще делать **без персональных данных**: анонимно или с выдуманными данными.
- Ключи API и пароли не пишут в код и не отправляют в чат. Агенту нужно прямо запретить это делать.
- Дети — отдельная история: нужно согласие родителей.

ПРОМТ 8 · НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Для продукта **[краткое описание]** составь нефункциональные требования для первой версии по чек-листу: устройства и браузеры, скорость, понятность, доступность, язык, надёжность при ошибках, где размещается, безопасность и персональные данные (152-ФЗ), стоимость работы.
Каждое требование сформулируй так, чтобы его можно было проверить за минуту. Плохо: «быстрый». Хорошо: «открывается на телефоне за 3 секунды».
Отдельно перечисли, какие решения нужно принять мне, а не тебе.

ПРОМТ 9 · ПЕРЕПИСАТЬ РАЗМЫТОЕ

Найди в наших требованиях все размытые формулировки: «удобно», «быстро», «красиво», «надёжно», «и т. д.», «по возможности», «при необходимости». Для каждой предложи проверяемую замену с цифрой или конкретным действием. Таблица: было | стало | как проверить.

КРИТИК

6 Стресс-тест требований

Требования пишут для счастливого пути: пользователь всё вводит правильно, интернет есть, сервер отвечает. Реальность другая. Критик проходит по требованиям глазами тестировщика и находит то, что сломается в первый же день. Лучше делать это в другой нейросети — так вы получите независимый взгляд.

ПРОМТ 10 · ГРАНИЧНЫЕ СЛУЧАИ

Ты опытный тестировщик. Вот требования к продукту: **[вставьте требования]**.
Для каждого сценария найди граничные и негативные случаи:
— пустые, слишком длинные и странные значения (эмодзи, пробелы, 0, отрицательные числа);
— повторное и быстрое нажатие кнопки, кнопка «Назад», обновление страницы;
— потеря интернета, медленный ответ, ошибка внешнего сервиса или нейросети;
— первый вход без данных и тысяча записей;
— пользователь делает не то, что задумано.
Для каждого случая предложи ожидаемое поведение одной фразой, чтобы добавить его в критерии приёмки.

ПРОМТ 11 · ПРОТИВОРЕЧИЯ И ДЫРЫ

Проверь требования как придирчивый разработчик, который будет их выполнять:
1. Противоречия: где одно требование мешает другому.
2. Дыры: какие экраны, переходы и состояния не описаны.
3. Двусмысленности: где можно понять по-разному. Приведи оба толкования.
4. 10 вопросов, без ответов на которые ты бы не начал работу.
Не исправляй сам — сначала спроси меня.

ПРОМТ 12 · ЗЛОЙ ПОЛЬЗОВАТЕЛЬ

Представь, что продуктом пользуется человек, который хочет сломать его, обмануть или получить чужие данные. Какие 5–7 способов он попробует? Для каждого — простое требование, которое закрывает дыру, и как его проверить. Не нужно уровня банка, только разумная защита для первой версии.

После стресс-теста вернитесь в основной чат: «Вот замечания критика: [вставьте]. Ответь на вопросы, которые можешь, и задай мне остальные. Потом обнови требования». Так требования проходят два круга без лишней работы.

ОФОРМИТЕЛЬ

7 Экраны и данные

Агенту проще строить по списку экранов, чем по абзацам текста. У каждого экрана есть не только «нормальное» состояние, но и **пустое** (данных ещё нет), **загрузка**, **ошибка** и **успех**. Если их не описать, агент сделает только нормальное.

ПРОМТ 13 · КАРТА ЭКРАНОВ

По требованиям составь список экранов первой версии. Для каждого:
— назначение одной фразой и кто его видит;
— что на экране: блоки, кнопки, поля;
— состояния: пусто, загрузка, ошибка, успех — что видит пользователь в каждом;
— куда ведёт каждая кнопка.
В конце нарисуй схему переходов между экранами стрелками в тексте: Главная → Вопрос → Результат.

ПРОМТ 14 · ДАННЫЕ ПРОСТЫМИ СЛОВАМИ

Опиши, какие данные хранит продукт, без технических терминов: какие «карточки» есть (например, пользователь, тест, ответ), какие поля в каждой, как они связаны, кто может их видеть и менять. Отметь персональные данные. Предложи, можно ли в первой версии обойтись без базы данных.

ПРОМТ 15 · КЛИКАБЕЛЬНЫЙ ПРОТОТИП

Сделай кликабельный прототип по карте экранов одним HTML-файлом: без логики и без сохранения, с выдуманными данными, все переходы по кнопкам работают. Нужно, чтобы я и 2–3 будущих пользователя прошли путь и сказали, понятно ли. Подпиши в углу каждого экрана его название.

Прототип — самая дешёвая проверка требований. Покажите его людям до разработки: правки в требованиях стоят минуты, правки в готовом продукте — часы.

ОФОРМИТЕЛЬ

8 Спецификация для ИИ-агента

Подход **spec-driven development**: сначала спецификация («что и зачем»), затем план («как»), затем список задач, а потом агент выполняет задачи по одной. Всё хранится файлами в папке проекта, чтобы агент перечитывал их, а не полагался на память чата.

ПРОМТ 16 · СОБРАТЬ SPEC.MD

Собери всё, что мы обсудили, в один файл spec.md:
1. Цель и метрика. 2. Пользователи и роли. 3. Главный сценарий. 4. Функции версии 1 (Must) с историями и критериями «Дано / Когда / Тогда».
5. Не входит в версию 1. 6. Экраны и состояния. 7. Данные. 8. Нефункциональные требования. 9. Открытые вопросы.
Только требования, без выбора технологий. Кратко, однозначно, нумерованные пункты, чтобы на них можно было ссылаться.

ПРОМТ 17 · ПЛАН И ЗАДАЧИ

Прочитай spec.md. Предложи технический план для первой версии: самый простой стек под мои ограничения **[например: без установки, один HTML-файл / бесплатный хостинг]** и почему. Затем разбей работу на 6–12 задач. Каждая задача — до 1 часа работы агента, заканчивается результатом, который можно открыть и проверить, и ссылается на пункты spec.md. Сохрани как tasks.md с галочками.

ПРОМТ 18 · ПРАВИЛА ДЛЯ АГЕНТА (AGENTS.MD)

Составь файл AGENTS.md — правила, которые агент читает перед каждой задачей:
— что за проект и где лежат spec.md и tasks.md;
— делать одну задачу за раз и отмечать её в tasks.md;
— не добавлять функции, которых нет в spec.md; при сомнении спрашивать;
— не хранить ключи и пароли в коде, не использовать реальные персональные данные;
— после каждой задачи писать, что сделано и как проверить;
— язык интерфейса и стиль текстов: **[ваши требования]**.

Папка проекта

```
мой-проект/
├ AGENTS.md — правила для агента
├ spec.md — что и зачем
├ tasks.md — список задач с галочками
└ prototype.html — прототип экранов
```

Готово, если

- [ ] Все пункты Must прошли критерии «Дано / Когда / Тогда».
- [ ] Проверено на телефоне и компьютере.
- [ ] Ошибки и пустые состояния выглядят по-человечески.
- [ ] Метрику из шага 1 можно посчитать.

Шпаргалка: требования на одной странице

Если времени мало, заполните этот шаблон сами или отдайте ИИ промт 19. Даже такой короткий документ в разы точнее, чем «сделай мне приложение для...».

ПРОМТ 19 · ТРЕБОВАНИЯ ЗА 15 МИНУТ

Задай мне по очереди вопросы и заполни шаблон требований на одну страницу:
ЦЕЛЬ: ... МЕТРИКА: ...
ПОЛЬЗОВАТЕЛЬ: кто, в какой ситуации.
ГЛАВНЫЙ СЦЕНАРИЙ: 3–6 шагов от входа до результата.
ВЕРСИЯ 1 (обязательно): 3–7 функций, у каждой 1–2 критерия «Дано / Когда / Тогда».
НЕ ДЕЛАЕМ: ...
ЭКРАНЫ: ...
ДАННЫЕ: что храним, есть ли персональные данные.
ОГРАНИЧЕНИЯ: устройства, где размещаем, сроки, бюджет.
ОТКРЫТЫЕ ВОПРОСЫ: ...
Потом найди 5 самых слабых мест в заполненном шаблоне.

ПРОМТ 20 · ОБНОВИТЬ ТРЕБОВАНИЯ ПОСЛЕ ПОКАЗА

Я показал прототип или первую версию **[N]** людям. Вот что они делали и говорили: **[заметки]**. Раздели замечания на ошибки, непонятные места и новые хотелки. Ошибки и непонятное — что поменять в spec.md и tasks.md. Хотелки — разложи по MoSCoW; большинство, скорее всего, уйдёт в «не сейчас».

Частые ошибки

- **Сразу просить код.** Без требований агент строит «среднее» приложение, а не ваше.
- **Всё в Must.** Первая версия раздувается и не доезжает до людей.
- **Нет «не делаем».** Агент любит добавлять лишнее: регистрацию, админку, тёмную тему.
- **Только счастливый путь.** Ломается на первом пустом поле.
- **Требования только в чате.** Чат забывает, файл spec.md нет.

Усилители

- «Задавай вопросы по одному».
- «Предложи 2–3 варианта и рекомендуй один».
- «Сформулируй так, чтобы можно было проверить».
- «Что я забыл описать?»
- «Ссылайся на номера пунктов spec.md».
- «Не исправляй сам — сначала спроси».