

# Профессиональная работа с Claude

Практический гид для больших задач, которые не помещаются в одну сессию · 4-5 минут чтения

---

Главная ошибка перехода с «болталки с ИИ» к серьёзной работе — пытаться сделать всё в одном бесконечном чате. На больших задачах побеждает не тот, кто пишет хитрые промпты, а тот, кто грамотно устроил вокруг Claude рабочий процесс: проект, фазы, передачу контекста, версии. Ниже — практический минимум, который сразу даёт «профессиональный» результат.

## 1. Заведите Project под каждую крупную задачу

Project — это контейнер для связанных чатов: общая база знаний (загруженные файлы), системная инструкция и общий для всех чатов контекст. Любая задача, которая не закроется одним чатом, должна жить в проекте. Туда сразу загружайте: ТЗ, релевантную документацию, примеры желаемого результата, глоссарий терминов, ссылки на внешние стандарты. После этого вы перестаёте каждый раз объяснять контекст заново.

## 2. Системная инструкция — один раз и надолго

В «Project instructions» зафиксируйте то, что будет верно для всех чатов внутри проекта:

- **Роль:** «Ты старший аналитик по X», «Ты редактор технических текстов».
- **Цель проекта:** что вы вместе делаете и для какой аудитории.
- **Формат ответов:** язык, длина, тон, обязательные секции.
- **Анти-паттерны:** «не давай дисклеймеры», «не предлагай альтернатив без запроса», «не извиняйся за длину».
- **Иерархия источников:** приложенные файлы важнее общих знаний; при конфликте — спрашивай.

Чем точнее инструкция, тем меньше нужно повторять в каждом сообщении — и тем стабильнее результат от чата к чату.

## 3. Помните про окно контекста

У каждого чата есть лимит памяти. Когда он переполняется, Claude начинает забывать ранние детали, путать версии, повторяться, терять стиль. Признаки, что пора закрывать чат и открывать новый:

- Ответы становятся водянистыми и общими.
- Claude путает, какая версия документа актуальна.
- Приходится напоминать одно и то же дважды.
- Качество явно упало без видимой причины.

## 4. Декомпозиция: один чат — одна фаза

Не сваливайте большую задачу в один диалог. Режьте по фазам и каждой выделяйте отдельный чат: **исследование** → **план** → **исполнение** → **ревью**. В чате исследования вы собираете материал и инсайты. В чате плана — согласовываете структуру и принимаете решения. В чате исполнения — пишете, считаете, кодите. В чате ревью — критикуете и шлифуете. Так контекст каждого чата остаётся «чистым» и заточенным под одну цель.

## 5. Handoff-документ — мост между сессиями

Главный профессиональный приём для многосессионной работы. Перед закрытием чата попросите:

*«Сделай краткий handoff: цель → принятые решения → текущее состояние → открытые вопросы → следующие шаги. Так, чтобы другой Claude в новом чате продолжил без потерь.»*

Сохраните этот документ как файл в Project. Новый чат начинайте словами: «Прочитай handoff.md и продолжи с шага N». Это даст эффект непрерывной работы даже через дни и недели.

## 6. Артефакты и версии

Длинные документы, код и таблицы Claude отдаёт как артефакты. Они удобно правятся на месте, но дисциплина важнее удобства: сохраняйте ключевые версии локально или в Project отдельными файлами (*v1\_draft, v2\_after\_review, v3\_final*). Перед серьёзной правкой критичного документа сначала уточните, что именно меняете, и только потом просите переписать — иначе легко потерять удачные формулировки.

## 7. Память (Memory) не заменяет структуру

Если у вас включена функция Memory, Claude помнит факты о вас между чатами — это хорошо для предпочтений (стиль, тон, любимые форматы). Но для рабочих задач память не заменяет project-инструкции и handoff-файлы: она хранит выжимки, а не полный контекст. Опирайтесь на структуру, а память пусть работает фоном.

## 8. Инструменты — просите явно

Claude сам решает, когда искать в вебе, выполнять код или строить диаграмму. Если вам нужна свежая информация, расчёт или визуализация — скажите прямо: «поищи в вебе...», «посчитай в Python...», «построй диаграмму...». Явный запрос значительно точнее, чем надежда, что Claude угадает.

## 9. Итерации вместо «сделай идеально с первого раза»

Профессиональный паттерн — не один большой промпт, а короткий цикл:

- Получили черновик.
- «Оцени свой ответ по шкале 1–10 по критериям А, В, С. Где самые слабые места?»
- «Перепиши, усилив три самых слабых пункта.»

- Повторите 1-2 раза.

Claude обычно хорошо критикует свои же работы — используйте это как встроенный редактор.

## 10. Собирайте свою библиотеку промптов

Удачные промпты сохраняйте отдельно — в Notion, Obsidian или файлом в Project. За пару недель у вас наберётся десяток шаблонов, экономящих часы: «промпт для ревью кода», «структурированное summary встречи», «red-team аргументы против моей идеи», «handoff в новый чат». Это и есть ваше личное преимущество перед теми, кто каждый раз формулирует с нуля.

---

### Чеклист перед стартом большой задачи

- Создан Project с базой знаний и системной инструкцией.
- Задача разбита на фазы (research → plan → execute → review).
- Для каждой фазы заведён отдельный чат.
- В конце каждого чата сохранён handoff-документ.
- Ключевые версии артефактов выгружены в Project как файлы.
- Удачные промпты добавлены в личную библиотеку.

*Освойте эти десять пунктов — и Claude перестанет быть «умным собеседником» и станет полноценным напарником на проектах, которые длятся неделями.*