






# Claude Code: продвинутые техники для работы с AI-агентом

---

## Как мы используем Claude Code в Anthropic

### Ключевые тезисы:

-  **Агенты становятся умнее** — модели развиваются, позволяя агентам выполнять более сложные задачи.
  -  **Меняйте подход к работе** — чтобы получить максимум от Claude Code, нужно адаптировать свои привычки.
  -  **Позвольте Claude «интервьюировать» вас** — модель лучше извлекает требования, чем вы их формулируете.
  -  **Переходите от Markdown к HTML** — HTML-файлы более информативны и удобны для человека.
  -  **Встраивайте верификацию в артефакт** — делайте код проверяемым нативно, чтобы агент мог тестировать его самостоятельно.
- 

## Эволюция работы с агентами

### Почему меняется подход?

Модели становятся мощнее, что позволяет агентам работать дольше над сложными задачами. Однако это увеличивает расход токенов, если агент действует неверно. Ключ — **сместить верификацию в начало процесса**, сделав спецификацию более богатой и понятной.

### Горький урок (The Bitter Lesson)

*Идея Ричарда Саттона:* вкладывать силы в `hardcode`-ограничения системы менее эффективно, чем просто увеличивать объём данных и вычислительных мощностей. Аналогично с Claude: **модель лучше извлекает скрытые требования**, чем вы их формулируете.



## Этап 1: Пусть Claude задаёт вопросы

**Плохой промт:** «Сделай лучше».

**Хороший промт:** Определяет область интересов, но не навязывает конкретное решение, поощряя Claude к диалогу.

**Как это работает:**

1. В промте явно указывается использовать инструмент `ask user question`.
2. Claude начинает интерактивный опрос, уточняя детали (аудитория, функции и т.д.).
3. В результате создаётся более полная и точная спецификация без необходимости заранее всё детализировать.



**Рекомендации по настройке Claude Code:**

- Используйте **Auto mode** — это значительно упрощает работу.
- Настройте параметр **Effort** — рекомендуется `high` или `max`.
- Попробуйте **Fast mode** — дороже, но идеально для быстрой итерации по спецификациям.



## Этап 2: От Markdown к HTML-спецификациям

**Проблема Markdown:** Длинные файлы (200+ строк) плохо читаемы. Трудно представить итоговый вид интерфейса.


**Преимущество HTML:**

- **Большая информационная плотность** — больше данных в удобной форме.
- **Визуальная эргономика** — можно сразу оценить дизайн.
- **Интерактивность** — можно делать скриншоты и давать по ним обратную связь.

**Пример (приложение для разделения счетов):**

Claude может сгенерировать несколько HTML-макетов в разных стилях (брутализм,

Токуо fintech и др.). Это позволяет **наглядно выбрать направление и дать точные правки**, особенно с улучшенной vision-моделью Opus 4.7.

 **Совет:** Регулярно делайте скриншоты для обратной связи с Claude. Часто проще показать, что «что-то не так», чем описать это словами.

## **Этап 3: Нативная верификация для агентов**

**Цель:** Встроить возможность проверки прямо в артефакт (код), чтобы агент мог самостоятельно запускать тесты.

**Пример с To-Do приложением (React):**

1. **Публикация состояния в DOM:** Компоненты явно выводят свои данные (например, `data-verify-unit="total-done"`) в атрибуты HTML.
2. **Агент читает контракт:** Claude может прочитать это состояние напрямую из DOM, не вникая во внутренности React.
3. **Запуск верификации:** На основе этих данных запускаются проверки (инварианты, фикстуры).

**Три способа запуска верификации:**

1. **Для человека:** Через читабельную дашборд в браузере.
2. **Агент-центричный:** Claude самостоятельно запускает проверки из браузера (например, через Playwright MCP).
3. **Headless/CI:** Запуск через CLI командой (например, `bun verify`).

**Запись доказательств:**

Процесс верификации можно записывать как видео-клипы, сохранять их (например, в S3) и делиться с командой в качестве доказательства работоспособности.



## Итоговые рекомендации

1. **Доверяйте модели:** Позвольте Claude извлекать требования через диалог.
2. **Специфицируйте в HTML:** Переходите от длинных Markdown-файлов к наглядным HTML-макетам.
3. **Встраивайте верификацию:** Делайте состояние приложения и контракты проверки доступными прямо в DOM для агентов.
4. **Используйте Orus 4.7:** Новая модель с улучшенным vision-модулем идеально подходит для такой работы.
5. **Экспериментируйте с Fast mode:** Для быстрого прототипирования и итераций по спецификациям.

**Главный вывод:** Будущее за агент-нативными артефактами, где код не только работает, но и содержит в себе всю необходимую информацию для своей же автоматизированной проверки.