АРТЁМ ВЕКТОР

АРХИТЕКТОР ИИ

Управление моделями,
сильные промпты
и создание новых решений

ПРОДВИНУТАЯ ПРАКТИКА



30 глав • Промпты • Задачи с решениями

Артем Вектор

Архитектор ИИ. Управление моделями, сильные промпты и создание новых решений

«Автор»
2026

Вектор А.

Архитектор ИИ. Управление моделями, сильные промпты и создание новых решений / А. Вектор — «Автор», 2026

Для тех, кто уже пользуется языковыми моделями и хочет проектировать более управляемые процессы: понимать желание человека, опираться на источники, считать то, что можно проверить, и вовремя передавать задачу специалисту. Книга читается как увлекательный нон-фикшн: у каждой главы есть живая сцена, примеры из учебной мастерской «Северный модуль» и понятный метод.Здесь нет универсальной фразы, которая заставит любую модель сделать что угодно. Есть способы поставить задачу, собрать контекст, ограничить полномочия, оценить качество и посчитать полную стоимость принятого результата. Шаблоны запросов встроены в главы. Для основной части программирование не нужно.Тридцать глав — от устройства моделей до экономики продукта и собственной методики. Рекомендуемый возраст: 16+. Независимое учебное издание под псевдонимом Артём Вектор. Не является официальным руководством разработчиков моделей. Учебные каталоги, персонажи и сценарии вымышлены.

© Вектор А., 2026
© Автор, 2026

Содержание

Об этом издании 5
Как работать с книгой 6
I. Основы и границы возможностей 7
Цепочка важнее одной модели 8
Сначала самое простое решение 9
Требование и способ — не одно и то же 10
Где остановиться и отдать человеку 11
Шаблон P01. Карта решения и границы автоматизации 12
Практикум. Архитектурный выбор 13
Самопроверка 14
Конец ознакомительного фрагмента. 15

Артем Вектор
Архитектор ИИ. Управление моделями, сильные промпты и создание новых решений

Об этом издании

Книга выходит под авторским псевдонимом Артём Вектор. Это независимое учебное издание. Оно не является официальным руководством OpenAI, xAI, Anthropic, Google или других упомянутых организаций. Названия продуктов и сервисов встречаются в тексте только для того, чтобы читатель мог узнать предмет разговора, и не означают партнёрства, одобрения или официального статуса.

Библиография в конце книги ведёт к оригинальным книгам, документации, исследованиям и видео. Полные переводы чужих книг и расшифровки чужих лекций в издание не включены. Русские названия источников даны для удобства навигации и могут отличаться от названий опубликованных переводов.

Все учебные каталоги, цены, заказы, письма клиентов, персонажи и результаты опытов в этой книге вымышлены. Мастерская «Северный модуль», которая будет встречаться почти в каждой главе, придумана как учебный стенд, а не списана с действующего бизнеса. Реальные тарифы, лимиты и возможности сервисов меняются. Прежде чем опираться на конкретную настройку в работе, сверяйтесь с действующей документацией того продукта, которым пользуетесь.

Упражнения по проверке устойчивости и безопасности предназначены только для собственного стенда и учебных площадок, без реальных секретов и без действий во внешней среде.

Рекомендуемая возрастная маркировка издания: 16+.

Как работать с книгой

Каждая глава начинается с короткой истории или сцены. За ней идёт принцип, потом — как им пользоваться и где обычно ошибаются. Так проще увидеть живую задачу, а не только схему.

В тексте встречаются готовые шаблоны запросов к модели. Квадратные скобки отмечают места, которые нужно заменить своими данными. Это заготовки, а не волшебные формулы: один и тот же шаблон в разных сервисах может сработать чуть иначе. Не копируйте шаблон вслепую — сначала впишите свою задачу, свои ограничения и свои источники.

У упражнений для самопроверки ответы собраны в конце книги. Практические задания разобраны рядом с условием. Сначала запишите свой вариант, и только потом смотрите разбор. В открытых задачах допустимы разные решения, если вы соблюли ограничения и показали, на чём держится вывод.

Если хотите идти не по порядку, вот три коротких маршрута.

Маршрут «Управление поведением»: главы 2, 4–7, 12, 15, 26–27. Маршрут «Новые идеи»: главы 16, 20, 22, 28, 30. Маршрут «Собери помощника»: главы 1, 8–14, 18–19, 23, 25, 29. Для цельной картины лучше читать книгу последовательно: поздние главы опираются на привычки, которые появляются в начале.

Отдельного приложения с файлами промптов и программным практикумом у этого издания нет. Все нужные шаблоны встроены в главы. Для основной части книги программирование не требуется.

И ещё одна рабочая привычка, без которой остальное быстро расползается. Сохраняйте исходный запрос, данные, версию модели, ответ, способ проверки и, если важно, стоимость. Отличайте настоящий опыт от плана и от красивой демонстрации. Если приём сработал один раз, это повод проверить его ещё раз — а не объявлять универсальным законом.

I. Основы и границы возможностей
01 Что делает архитектор ИИ

Письмо пришло вечером, когда в учебной мастерской «Северный модуль» уже выключили станки. Клиент писал свободно и тепло: нужны два шкафа в прихожую, «как в прошлом году, только чуть уже, и чтобы точно прошли в дверь». В конце — просьба «побыстрее выставить счёт, мы уже почти определились».

Если отдать такое письмо целиком «умному помощнику» и ждать готового счёта, легко получить красивый текст с неверной шириной, выдуманным сроком и ценой, которой нет в каталоге. Беда не в том, что модель «плохо думает». Беда в том, что никто не решил, что именно должно произойти: понять желание, свериться с каталогом, посчитать, показать черновик человеку — или сразу отправить обещание клиенту.

Архитектор ИИ как раз занимается этой невидимой работой. Он превращает человеческую потребность в систему, которая даёт проверяемый результат. Его дело начинается раньше, чем выбор модели. Нужно понять, кто примет результат, какие ошибки ещё можно пережить, откуда берутся данные и чем весь процесс обойдётся. Фраза «сделай умного помощника» не задаёт ни продукт, ни способ проверки. А вот просьба «подготовь проект ответа клиенту по утверждённому каталогу и укажи, откуда взято каждое обещание» уже похожа на работу, которую можно сделать хорошо.

Цепочка важнее одной модели

Результат складывается не из одного вызова модели. Сначала есть задача человека. Затем — данные и правила предметной области: каталог, размеры, допуски, запреты. Потом модель, которая превращает этот контекст в текст или в предложение действия. Дальше программа: поиск, инструменты, память, права доступа, обработка ошибок. И наконец эксплуатация — наблюдение за качеством, расходами и изменениями.

Когда что-то идёт не так, почти всегда хочется сказать: «модель плохо думает». Часто причина лежит совсем в другом месте. В каталоге нет нужной ширины. Письмо двусмысленно. Программа не проверяет артикул. Человек машинально нажимает «отправить».

Сами слова тоже стоит развести. Модель можно вызвать через программный интерфейс — как инструмент, который продолжает текст или извлекает сведения. Приложение добавляет к ней экран, инструкции и дополнительные функции. Агент — это уже система, которая может сама выбирать следующие шаги и обращаться к инструментам. Оркестратор держит процесс в руках: передаёт задания, помнит, на чём остановились, и решает, когда продолжить, повторить шаг или остановиться.

Сначала самое простое решение

Если нужно сложить числа, достаточно вычисления. Если нужен точный артикул, начните с базы или с каталога. Если из свободного письма надо понять намерение человека, языковая модель оказывается к месту. Часто лучший продукт сочетает все три хода: модель понимает запрос, программа проверяет факты, интерфейс показывает понятный результат. Лишняя «интеллектуальность» оправдана только тогда, когда от неё есть измеримая польза, а не эффектный вид.

В этой книге мы будем собирать помощника для вымышленной мастерской «Северный модуль». Он читает письмо клиента, уточняет размеры, подбирает позиции из учебного каталога и готовит проект предложения. Каталог, цены и все примеры придуманы. Помощник может готовить черновики и считать стоимость по тестовым данным. Он не должен выдумывать наличие товара, сам обещать доставку или отправлять коммерческое предложение без предусмотренной проверки человеком. Эти границы стоит записывать дважды: в требованиях и в программных проверках.

Требование и способ — не одно и то же

Продвинутый уровень начинается с простой различения. «Использовать пять агентов» — это способ. «Сократить время подготовки предложения, не увеличивая число ошибочных обещаний» — это требование. Способ можно заменить, цель при этом сохранить. Если команда не умеет описать результат отдельно от инструмента, она начинает украшать демонстрацию вместо того, чтобы делать полезную работу.

Полезно составить карту решений. Что вообще нужно определить? Какие данные для этого требуются? Кто имеет право подтвердить вывод? Какое действие последует и насколько дорого его отменить? В мастерской решение «клиент просит два модуля» можно поручить извлечению текста. Решение «конструкция выдержит указанную нагрузку» требует инженерных данных и проверки. Решение «отправить предложение» относится к полномочиям приложения и человека. Одно и то же слово «решение» прячет три разных механизма.

Затем опишите инварианты — условия, которые должны сохраняться при любом маршруте. Цена имеет источник и версию. Количество положительно. Отсутствие срока не превращается в обещание. Изменение заказа отменяет старое согласование. Инвариант тем и отличается от пожелания, что его нарушение можно заметить. Фраза «будь внимательным» ничего не даёт проверяющей программе. А правило «не отправлять предложение с неизвестным сроком без явной отметки» уже можно увидеть в работе.

Где остановиться и отдать человеку

Автоматическое выполнение, предложение действия и передача человеку — это три разные степени свободы. Выбор зависит от цены ошибки и от качества проверки. Разложить заметки по папкам можно доверить программе раньше, чем обещание клиенту. При этом кнопка «подтвердить» сама по себе ещё не делает процесс надёжным: человек может нажать её по привычке. На экране подтверждения должны быть видны существенные изменения, источники и всё, что осталось неизвестным.

Наконец, архитектуру стоит оценивать сразу на трёх бюджетах: деньгах, времени и ошибках. Допустим, ответ нужен примерно за минуту, на обработку отведены условные небольшие деньги, а неподтверждённая цена недопустима. Тогда бесконечный разговор между агентами ломает первые два бюджета, а быстрый уверенный ответ без каталога — третий. Архитектурный выбор — это управляемый компромисс, а не соревнование, у кого больше деталей.

Шаблон P01. Карта решения и границы автоматизации

Разбери задачу: [описание процесса]. Получатель результата: [роль]. Доступные данные: [перечень]. Для каждого решения укажи: входные сведения; исполнителя (модель, программа или человек); проверку результата; последствие ошибки и способ отмены. Затем предложи минимальный процесс и три измеримых условия приёмки. Отдели требования от выбранной реализации. Если данных для решения недостаточно, назови конкретный пробел.

Практикум. Архитектурный выбор

Помощник должен разобрать письмо, подобрать артикул, рассчитать стоимость и отправить счёт. Есть утверждённый каталог, но нет сведений о наличии. Разработчик предлагает дать модели доступ к почте и поручить весь процесс одним запросом. Спроектируйте минимальный вариант и объясните, где должна проходить граница автоматизации.

Разбор. Модель извлекает намерение и параметры. Поиск возвращает кандидатов из каталога, программа проверяет артикулы и считает сумму. Отсутствие сведений о наличии сохраняется как неизвестное значение. Результат — черновик предложения, а не подтверждение, что товар можно отгрузить. Отправка остаётся отдельной операцией после предусмотренной проверки. Критерии такие: ни одного неизвестного артикула; арифметика совпадает с расчётным модулем; если источника наличия нет, в ответе явно нет обещания отгрузки. Для учебного опыта достаточно файлов и имитации отправки. Добавлять почтовый доступ до этих проверок — значит увеличивать полномочия, а не качество понимания.

Самопроверка

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, купив полную легальную версию на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.