«Вертикальный AI» уже не значит то, что вы думаете

2026-05-26

К весне 2026 года в B2B AI закончился спор о форме. Если в 2024-м тезис «вертикальные AI-агенты надёжнее горизонтальных продуктов» был контрарной позицией нескольких аналитиков, то к началу 2026-го он стал дефолтом. Y Combinator в Request for Startups держит «vertical AI agents» отдельной категорией с 2024 года. Joanne Chen из Foundation Capital в апреле 2024-го прямо сформулировала тезис service-as-software: «Быстрорастущий «рынок услуг» объёмом в триллионы долларов ждёт своей очереди на софтверизацию» — AI-агенты заходят не на 350 млрд долларов SaaS, а на 4,6 триллиона долларов рынка услуг, через специализированные продукты под конкретные роли и отрасли. Sequoia в AI Ascent 2024 собрала эту же мысль в одну метафору: «генеративный AI второй акт» — про переход от инфраструктуры к прикладным вертикальным продуктам. Любой пост в венчурной ленте, любая презентация инвестору, любой питч в acceleration-программе начинаются с «мы строим вертикальный AI для…».

Когда тезис становится консенсусом, он перестаёт работать как фильтр. Под одним словом начинают помещаться несколько разных вещей. Так уже было с «облаком» в 2010-м: между арендой виртуальной машины и платформой для совместной работы оказалась пропасть, которая выяснялась только через несколько лет операций. Под «вертикальным AI» сейчас живут две принципиально разные модели бизнеса с разной экономикой и разной защитимостью. Эту разницу можно описать через один вопрос: что именно у клиента останется уникальным через полтора года работы с продуктом, чего нет у соседа из той же отрасли с тем же поставщиком.

Две вертикали под одной вывеской

Первая модель — то, что можно назвать **вертикальным продуктом**. Команда берёт горизонтальный класс задач (customer support, sales development, qualification, content operations) и оборачивает его в отраслевые декорации: интеграции с типовыми CRM и ERP для конкретного сегмента, готовые сценарии для двух-трёх частых процессов, glossary терминов отрасли, шаблоны коммуникаций. Внутри агента — стандартная управляемая оболочка от Anthropic, OpenAI, AWS или Google. Снаружи — обёртка «AI for finance», «AI for legal», «AI for trades». Sales-команда продаёт «вертикальный AI» — архитектурно это горизонтальный продукт в нишевой упаковке.

Вторая модель — **вертикальный операционный слой**. Здесь компания строит отдельный слой между фронтирной моделью и бизнес-процессом клиента: специфическое представление сущностей и событий отрасли, в котором

каждое понятие имеет однозначное определение и набор допустимых состояний; накопленные траектории решений (что сделал агент, что вернулось из реальности, как это исправил человек) на горизонте года и больше; кодифицированные регламенты отрасли, версионируемые по мере того, как меняется реальность. Внутри тоже используется фронтирная модель, но защищённое — не она.

Обе модели в публичных материалах называют себя одним словом. На презентации инвестору обе говорят: «мы знаем эту отрасль глубже, чем любой горизонтальный игрок». Но это разные «знаем глубже»: первая модель знает отрасль на уровне продакт-маркетинга, вторая — на уровне архитектуры данных, с артефактами, которые не воспроизводятся при смене поставщика.

Сравнение: вертикальный продукт против вертикального слоя

Параметр	Вертикальный продукт	Вертикальный операционный слой
Что такое продукт	Управляемая оболочка плюс отраслевая обёртка	Слой между моделью и бизнес-процессом, накапливающий артефакты
Какие интеграции	Готовые коннекторы к типовым CRM/ERP отрасли	Собственная схема, в которую отображаются данные клиента
Где живёт «знание отрасли»	В промптах, шаблонах, маркетинговых материалах	В представлении сущностей, истории решений, регламентах
Что воспроизводится после смены поставщика	Практически всё за 1–3 месяца	Восстанавливается за год и больше
Источник защитимости	Скорость выхода на рынок, branding	Накопленные траектории и схема отрасли
Чувствительность к управляемым оболочкам	Растворяется в них	Не пересекается с ними
Стоимость переключения для клиента	Низкая: миграция за квартал	Высокая: год работы придётся проходить заново

Почему «вертикальный продукт» не выживает фазы массовых оболочек

В 2024 году у вертикального продукта было оправдание. Управляемые агентные оболочки ещё не были массовыми, поэтому ценность «мы собрали для вас агента, оркестратор инструментов, память и аудит» была реальной — даже если базовая архитектура была одинаковой для всех вертикалей. Внешняя упаковка плюс инфраструктурная работа давали средний контракт выше, чем у голой подписки на API провайдера.

В 2025–2026 годах эта база ушла. Anthropic Agent Skills, OpenAI Assistants и Responses API, AWS Bedrock AgentCore, Google Vertex AI Agent Builder — каждый из этих продуктов внутри 2025 года прошёл из beta в production: оркестрация инструментов, длинная память, политики безопасности, встроенные средства наблюдаемости теперь поставляются как часть облака. К концу 2025 года этот сдвиг стал очевидным: то, что было защитимой инженерией, стало доступной настройкой. Подробнее эта зависимость разобрана в «Управляемая обвязка как зависимость».

Foundation Capital в ноябре 2025-го прямо описала эту динамику в «When model providers eat everything»: провайдеры моделей «больше не продают только доступ к API — они агрессивно движутся вверх по стеку и превращаются из инфраструктурных компаний в продуктовые». Тезис фонда прямой: единственный способ построить устойчивый стартап на прикладном слое — система, в которой агенты непрерывно учатся на реальных взаимодействиях так, как лаборатории моделей воспроизвести у себя не могут.

Это означает, что вертикальный продукт остаётся без своей основной ценностной составляющей. Раньше он продавал «отраслевую упаковку плюс настроенную агентную среду». Теперь второе перестаёт быть товаром — у каждого крупного облачного клиента это уже подключено. Остаётся одна отраслевая упаковка, конкурентоспособность которой ограничена скоростью копирования. Промпты копируются за дни, glossary — за недели, коннекторы — за месяц. В практике публичных миграций между AI-вендорами видна та же цифра: гайды по уходу с Sierra или Decagon на конкурента (например, разбор миграции с Sierra) описывают перенос конфигурации в горизонт квартала. У вертикального продукта нет того, что не воспроизводится у конкурента в окне одного-двух кварталов.

При этом средний контракт у вертикального продукта в 2026-м продолжает падать. Корпоративный покупатель к этому моменту прошёл два-три цикла внедрения и научился оценивать стоимость собственными силами. Он сравнивает не «вашу платформу с конкурентом», а «вашу платформу с собственным внутренним инженером плюс управляемая оболочка плюс месяц на интеграцию». В этом сравнении вертикальный продукт проигрывает, потому что не предлагает ничего, что покупатель не может собрать сам — за один проект, а не за подписку на годы.

Почему вертикальный операционный слой работает иначе

Вертикальный операционный слой защищён по другому основанию. Внутри он опирается на ту же самую управляемую оболочку — но строит над ней независимый продукт из артефактов, которые не лежат в облаке провайдера.

Если смотреть на архитектуру работающих внедрений этой модели, прослеживаются три класса артефактов, которые и создают защиту.

Представление сущностей отрасли — первое из таких артефактов. Это не «модель данных под клиента», как её делают системы интеграции. Это словарь, в котором каждое понятие отрасли имеет фиксированное определение и набор допустимых состояний, согласованных по всем процессам компании. В сегменте проектных продаж B2B это означает, что «возможность», «контрагент», «стадия квалификации» и «эскалация» определены одинаково для всех агентов и не зависят от формата исходной системы. В сегменте операций со спецтехникой — что «заявка», «единица техники», «маршрут» и «возврат» описаны как событийный поток, а не как строки в таблице. Эту схему нельзя купить готовой — она формируется непосредственно на продуктивных данных клиента в первом цикле внедрения и проверяется только тем, что система работает.

Накопленные траектории — второй артефакт. Каждый запуск агента оставляет след: вызовы инструментов, последовательность шагов, результаты, корректировки от человека. На горизонте года этот след превращается в датасет, на котором базовая модель тонко настраивается под конкретный контекст, и который сам по себе становится отдельным активом. Этот тезис — отдельная статья: «Данные траекторий: как закрытый цикл становится единственным реальным moat». Чтобы новый поставщик воспроизвёл этот слой, ему нужно собрать те же месяцы логов с нуля. Платформенный провайдер тоже не имеет к нему доступа — траектории живут в инфраструктуре, к которой у клиента прямой контракт с поставщиком слоя.

Кодифицированные регламенты — третий артефакт. В каждой отрасли есть негласные правила, которые меняются раз в квартал и не описаны ни в одном документе. Вертикальный слой фиксирует эти правила как версионируемые описания поведения агента: какие случаи эскалируются человеку, какие автоматизируются, какие требуют двойного подтверждения. Через год у клиента есть аудируемая история собственной операционной модели — артефакт, который сам по себе стоит дороже годовой подписки, потому что заменяет внутреннюю работу по регулярному пересмотру SOP.

Эти три артефакта в отдельности доступны любому горизонтальному игроку. В сочетании, с продуманной архитектурой и со временем накопления — нет. Именно это сочетание — а не «отрасль в маркетинге» — отделяет вторую модель от первой.

Когда разница между двумя моделями становится видной?

Разница между двумя моделями становится очевидной не в момент покупки, а через полтора года после первого внедрения. В первый период оба продукта выглядят одинаково: и тот, и другой настраивают агента, подключают данные, запускают первые автоматизации. Покупатель сравнивает скорость старта, удобство интерфейса, ценник. На этом этапе вертикальный продукт может выигрывать — у него понятная упаковка, быстрый онбординг, агрессивный sales.

Через год разрыв проявляется. У клиента вертикального продукта набор кейсов с метриками сэкономленных часов и закрытых тикетов — типичный набор, который воспроизводим у любого конкурента. У клиента вертикального слоя — собственный датасет траекторий, собственная отраслевая схема, собственная история регламентов. Первый клиент при появлении более дешёвого аналога меняет поставщика за квартал. Второй не меняет, потому что миграция уничтожит накопленные артефакты, и год работы придётся проходить заново. Это и есть пересборка экономики, описанная в «Service-as-software: как агенты переписывают формулу выручки».

В 2026 году большая часть «vertical AI» компаний — это первая модель. Они существуют, продаются, привлекают раунды, но в их публичной коммуникации почти ничего не сказано об артефактах. Они говорят о вертикали в категориях продакт-маркетинга («мы знаем финтех», «мы понимаем юристов»), а не в категориях архитектуры данных. Если в материалах компании не описано, чем именно защищено её положение через 18 месяцев — она в первой модели.

Sierra и Decagon — два примера компаний второй модели, которые проговаривают логику открыто. Bret Taylor, сооснователь Sierra, в подкасте Sierra формулирует принципиальную позицию: «software that gets better the more you use it» — слой накапливает контекст клиента, и именно этот контекст определяет качество ответа, а не базовая модель. Outcome-based pricing в этой логике становится естественным продолжением: вендор берёт деньги за разрешённый кейс, потому что качество разрешения опирается на собственный накопленный артефакт, а не на универсальные возможности модели. Decagon в публичных материалах строит ту же конструкцию вокруг «agent operating procedures» — версионируемого описания того, как компания работает с клиентом, которое и есть её продукт, а не отчёт о работе агента.

Публичная экономика этой модели подтверждается цифрами. По независимым разборам прайсинга Sierra, годовые контракты выходят на диапазон «200–350 тысяч долларов в первый год» при setup fee 50–200 тысяч, Decagon в публичных обзорах стартует от порядка «95 тысяч в год» в нижнем сегменте. Эти цены реалистичны не потому, что поставщик «вертикален», а потому что внутри сидит операционный слой, выход из которого по тем же гайдам миграции занимает отдельный проект, а не переключение подписки.

Граница тезиса: где обе модели не работают одинаково

Разница между двумя моделями имеет две границы.

Первая граница — слой инфраструктуры. Векторные базы, системы наблюдаемости, инструменты управления промптами, шлюзы безопасности — здесь горизонтальный продукт остаётся правильной формой, потому что у инфраструктуры по построению нет «отрасли». Pinecone, Weaviate, LangSmith работают по модели «один продукт — все вертикали», и это правильная для них стратегия. Защита у них устроена иначе: через сетевой эффект разработчиков и стоимость переключения инфраструктуры, а не через накопленные операционные артефакты. Описанное здесь различие касается прикладного слоя — продуктов, которые встраиваются в рабочий процесс компании, а не в стек разработчика.

Вторая граница — сверх-широкие потребительские категории, где сам процесс универсален. Календарь, почта, заметки, базовая помощь в коде — это категории, в которых вертикализация не даёт преимущества, потому что у пользователя нет «отрасли». Горизонтальный AI-копилот для письменной коммуникации или для управления личным временем — рабочая модель. Но это потребительский слой; в B2B-сегменте подобных категорий мало, и они быстро поглощаются крупными платформами.

Между этими двумя границами лежит остальной B2B-рынок. Здесь разница между двумя моделями становится критерием отбора, а не вопросом стиля.

Как отличить вертикальный продукт от операционного слоя?

Для фаундера ранней стадии, выбирающего рынок: одного «выбираем вертикаль» в 2026 году недостаточно — ключевой вопрос в том, какие именно артефакты накопятся у клиента в первый год и какой из них не воспроизводится при смене поставщика. Если ответ — «у нас будут логи», «у нас будут промпты», «у нас будут шаблоны» — это первая модель. Если ответ — представление отрасли, собранное на проде, плюс траектории решений, плюс кодифицированные регламенты — это вторая.

Для фаундера на стадии масштабирования, у которого продукт уже работает: вторая ось расширения — не «соседняя индустрия». Соседняя индустрия для второй модели — это нулевой год накопления, со старта. Расширение в ту же индустрию по соседним слоям процесса сохраняет накопленные артефакты и держит экономику. Прыжок в соседнюю вертикаль возвращает компанию в зону конкуренции с управляемыми оболочками платформ, где у неё больше нет основания для премиальной цены.

Для исполнительного руководителя, выбирающего поставщика: вопрос покупки в 2026-м — не «у кого лучше модель» и не «у кого красивее интерфейс». Полезный тест — попросить поставщика описать, что именно у клиента будет уникального через 18 месяцев работы, чего не было в день подписания контракта. Если ответ — общие фразы про «лучшее знание отрасли» — продукт в первой модели, и его себестоимость в долгую проиграет внутреннему инженеру плюс управляемой оболочке. Если ответ — конкретные артефакты с

описанием того, как они хранятся и в каком формате принадлежат клиенту — продукт во второй модели.

Для совета директоров и инвестора: метрика, которую стоит спрашивать в 2026-м, — не «какой у нас MRR» и не «насколько мы вертикальны в маркетинге», а «какие операционные артефакты собрали наши клиенты у себя, и сколько времени уйдёт у конкурента, чтобы их воспроизвести». Это прокси для второй модели и единственная защита, которая работает в фазе массовых оболочек.

Главное

- «Вертикальный AI» как тезис в 2026 году стал консенсусом, но потерял разрешающую способность. Под одним словом теперь живут две принципиально разные модели бизнеса.
- Вертикальный продукт — это управляемая оболочка плюс отраслевая упаковка. В фазе массовых оболочек у него нет того, что не воспроизводится у конкурента за 1–3 месяца.
- Вертикальный операционный слой — это представление сущностей отрасли, накопленные траектории решений и кодифицированные регламенты. Эти артефакты собираются годами на проде и не покупаются готовыми.
- Большая часть «vertical AI» компаний 2026 года — это первая модель, продающая себя как вторая. Разница становится очевидной через полтора года, когда первый клиент меняет поставщика, а второй — нет.

FAQ

Почему различие между двумя моделями не видно сразу? Потому что в первые месяцы работы оба продукта делают похожие вещи: подключают данные, настраивают сценарии, запускают первые автоматизации. Артефакты второй модели накапливаются медленно — представление сущностей собирается на проде, траектории накапливаются по мере объёма решений. Через полтора года у клиента второй модели есть слой, у клиента первой — нет. До этого момента продукты выглядят одинаково.

Можно ли построить вертикальный операционный слой поверх управляемой оболочки провайдера? Можно и нужно. Управляемая оболочка снимает инфраструктурную задачу: оркестрацию, память, аудит. Слой строится поверх неё как отдельный продукт — представление данных, история решений и регламенты живут в собственной инфраструктуре поставщика или клиента, а не у платформы. Главное условие — чтобы артефакты были архитектурно отделены от провайдера и переносимы.

Не значит ли это, что любая «vertical AI» компания со временем превратится в вертикальный слой? Нет. Большая часть остаётся в первой модели — потому что сборка артефактов требует архитектурного решения с самого начала, а не «потом, когда будут ресурсы». Если компания не закладывала отдельный слой представления и накопления траекторий в момент запуска

продукта, то через три года у неё накопятся только промпт-сниппеты и интеграционные коннекторы, не более.

Как покупатель в 2026 году отличает одну модель от другой до подписания контракта? Через прямой вопрос: что именно у клиента будет уникального и собственного через 18 месяцев работы. Если поставщик отвечает в категориях продукта — «у вас будет настроенный агент», «у вас будут готовые сценарии» — это первая модель. Если в категориях артефактов — «у вас будет собственная отраслевая схема в открытом формате», «у вас будет история всех решений с экспортом», «у вас будет аудируемая версия регламентов» — это вторая.