





Business OS: эволюция от автоматизации к ядру бизнеса

Переход от автоматизации к Business OS

Ключевые тезисы:

-  Точечная автоматизация процессов (workflow) больше не даёт стратегического преимущества.
 -  Будущее за AI-агентами, работающими в едином контексте бизнеса.
 -  Нужен фундамент — **Business OS** (ядро бизнеса), а не разрозненные боты и сценарии.
 -  Ценность смещается от продажи автоматизаций к продаже **конкретных результатов** (закрытые лиды, сэкономленное время).
-

Почему автоматизация больше не работает

Workflow-автоматизация (если X, то Y) стала коммодити. Собрать цепочку действий может кто угодно за вечер. Это:

- **Пластырь:** ускоряет лишь один шаг, но не делает бизнес умнее.
- **Хрупкость:** ломается, как только реальность отклоняется от жёсткого сценария.
- **Без контекста:** бот не знает историю клиента, правила компании, риски и не понимает, что делать при нестандартном ответе.

Пример: Лид — это просто уведомление. Бот отправит письмо, но не знает, давали ли этому клиенту скидку раньше и кто за него отвечает.



Три типа систем: от чат-бота к агенту

1. **Чат-бот** (ChatGPT, Claude): пассивно ждёт запроса в окне.
2. **Автоматизация (workflow)**: жёсткий сценарий с условиями и ветвлениями.
3. **AI-агент**: *сам принимает решения, выбирает действия и выполняет их внутри вашей системы.*

Проблема современных агентов: они упёрлись не в модели, а в отсутствие контекста.

- У модели ограниченный бюджет внимания (большой контекст деградирует).
- У агентов нет бизнес-ориентиров, правил и доступа к истории решений.
- Системы хранят данные (скидка 15%), но не хранят **решения** (почему её дали? кто согласовал?).



Ключевой сдвиг: от Automation First к Context First

Старый подход (Automation First)

Центр всего — автоматизация
(триггер → действие → результат).

Отвечаем на вопрос: "Как ускорить этот шаг?".

Автоматизация — это цель.

Новый подход (Context First)

Центр всего — ядро бизнеса.

Начинаем с: **объекты, события, контекст, правила, решения.**

Автоматизация (и агенты) — это просто "**руки**", один из способов сделать что-то поверх ядра.

Вывод крупных платформ: Salesforce (Agent Force), HubSpot (Briz), ServiceNow, PWC, EY — все встраивают агентов **внутри** своих платформ и данных, а не продают их отдельно.

Итог: Если ты не строишь ядро, ты просто поставщик чат-ботов и скоро потеряешь ценность.

Что такое Business OS (ядро бизнеса)

Business OS — это не CRM, не чат-бот и не набор интеграций. Это слой, где бизнес хранит объекты, события, правила, решения и контекст так, чтобы и люди, и агенты работали из общей картины мира.

Архитектура ядра (снизу вверх):

1. **Контекст компании:** кто мы, что продаём, правила, цели.
2. **Память:** документы, история сделок, переписки.
3. **Агенты (вертикальные):** узкие специалисты (продажи, поддержка, финансы) со своими скиллами и правами.
4. **"Дирижёр":** распределяет задачи между агентами.
5. **"Руки" (Интеграции):** доступ к CRM, почте, ERP (1C, SAP). Без этого агент — просто болтовня.
6. **Governance:** права, логирование, эскалация на человека.
7. **"Приборная панель":** любой интерфейс для взаимодействия (дашборд, мессенджер). Его может создать и сам агент.

Ключевой слой — **"Решения"**: Логирование не просто факта ("скидка 15%"), а контекста решения — почему, кто согласовал, на каком основании.

Вертикальные агенты vs. Зоопарк автоматизаций

Вертикальный агент — это не универсальный помощник, а **"один агент — один сотрудник"** с чёткой ролью (квалификация лидов, проверка счетов, follow-up).

Плохая архитектура (Зоопарк):

- Каждому агенту — своя память и правила. Быстрый "вау-эффект", затем хаос.
- 10 агентов = 10 фрилансеров без менеджера. Нет единой картины.

Правильная архитектура:

- **Одно ядро** (объекты, события, память, правила) внизу.
- **Много вертикальных агентов** сверху, работающих из **общего контекста**.

Пример работы: Лид пришёл → агент за 2 минуты исследовал компанию, оценил профиль, подготовил персонализированный ответ, поставил follow-up и создал задачу. 40 минут ручной работы превратились в 2.



Баланс автономности: Агент предлагает, человек решает

Цепочка всегда контролируема:

1. **Агент предлагает действие.**
2. **Ядро записывает предложение.**
3. **Человек одобряет** (рисковые решения: цены, договоры, деньги — всегда за человеком).
4. **Инструмент исполняет.**
5. **Ядро логирует результат.**

Уровни автономности задаются явно: от подсказок (0) до выполнения мелких безопасных действий по правилам (3). **Ядро — источник правды. Человек — хозяин рискованных решений.**



Как начать: минимальное ядро "на коленке"

Не нужно быть корпорацией-гигантом. Минимальное ядро можно собрать на базе:

1. **Таблица объектов** (Notion, Airtable, Google Sheets): лиды, клиенты, сделки, задачи.
2. **Лог событий**: кто, когда, что сделал, результат.
3. **Лог решений**: скидки, исключения, согласования, обоснования.
4. **Файл правил**: что агенту можно самому, что требует согласования.
5. **Инбокс агента**: агент создаёт предложения, человек их одобряет/отклоняет.
6. **Ежедневный бриф**: что изменилось, где застряло.
7. **Один вертикальный агент** для одного ключевого процесса (например, follow-up).

Старт: начинаем с одного процесса, охваченного ядром. Чем больше процессов в ядре — тем больше "сотрудников"-агентов можно "нанять".

Модель внедрения: Forward Deployed Engineer (FDE) + AI

Проблема классических подходов:

- **Коробочное решение** ("вот ваш GPT") — не работает.
- **Разработка с нуля** — дорого, не все готовы платить.

Решение (по модели Palantir, но дешевле):

- **Forward Deployed Engineer (FDE)**: инженер приходит **внутрь бизнеса клиента**, работает с реальными данными и процессами.
- **AI-агенты как мультипликаторы**: делают черновую работу (интеграцию, генерацию сценариев), а **человек-инженер курирует и закрывает "последнюю милю"** (доменное знание).

- **Технически:** приносится **полуфабрикат ядра** с готовыми коннекторами (CRM, ERP, мессенджеры). Ценность — не в коннекторах (их много), а в **онтологии и сценариях** поверх них.
-

Self-Improving System и модель оплаты

Система учится:

- Ядро накапливает контекст, события, решения.
- Агенты на этом учатся, уточняют правила, расширяют сценарии.
- **Lining-слой** анализирует результаты (маржа, сегменты) и обновляет плейбуки. Система сама себя улучшает в заданных человеком границах.

Принципиально новая модель оплаты:

- Платёж не за настройку **workflow** или часы работы.
 - Платёж за **конкретный результат**: закрытые лиды, сокращённое ручное время, обработанные обращения.
 - **Риск смещается на исполнителя**, клиенту легче сказать "да". Это честный подход, проповедуемый Y Combinator для AI-компаний.
-

Выводы

- Автоматизация делает действия, ядро бизнеса хранит понимание.
- Агенты становятся по-настоящему полезными только при работе поверх **Business OS**.
- **Агент без ядра** — это умный сценарист. **Агент поверх Business OS** — это часть операционной системы компании.
- Интерес будущего — не в точечных ускорениях, а в сборке **"мозга компании"** (ядра) и настройке вертикальных агентов для работы в нём.

