

Платформенный подход к управлению на основе данных

В.Лемех | <https://8m.by>

1. Три подхода к сбору, упорядочению и визуализации данных

1) Ручной ввод (current manual entry)

Описание процесса

Сотрудник открывает форму, вводит данные из первичных документов или выгруженных файлов (Excel), заполняя строки/ячейки вручную. Контроль – визуальный/по замечаниям ответственных. Частые правки происходят в одном интерфейсе-таблице.

Архитектура данных / модель

Плоская структура: все атрибуты (поставщик, номенклатура, единица, цена, период, источник) собраны в одной записи без ссылок на справочники; отсутствуют уникальные идентификаторы и нормализация.

Валидация и качество

Почти полностью ручная валидация; слабая или отсутствующая проверка диапазонов, документов-оснований, исторических аномалий; дубликаты и ошибки обнаруживаются постфактум.

Производительность и масштаб

Подходит только для малого объёма (пилоты, единичные отчёты). При тысячах позиций – экспоненциальный рост ошибок и затрат времени.

Плюсы

- Низкая начальная стоимость внедрения.
- Подходит для тестовой работы и когда нет ERP/источника.

Минусы

- Ненадёжные данные, высокий риск ошибок.
- Невозможна автоматизация и массовая верификация.
- Невозможность построения корректных временных рядов и KPI.

Подходит для

- Пилотов, одноразовых сборов данных, экстренных случаев, когда нет доступа к системам источников.

2) Пакетный (принцип «упаковал файл из ERP/учётной системы»)

Описание процесса

Ответственная служба или модуль выгружает регулярные файлы (Excel/CSV/XML) из ERP/учётной системы; пакет загружается в платформу раз в сутки/неделя/месяц. В загрузке могут быть базовые преобразования (ETL) и частичная валидация.

Архитектура данных / модель

Средняя степень структурированности: в выгрузке присутствуют идентификаторы, но они зависят от источника – может потребоваться

сопоставление/нормализация по канонической модели платформы. История версий возможна, но чаще требуется хранение исходных пакетов.

Валидация и качество

Включает правило-пакеты: схемы XSD/JSON-Schema, базовые проверки типа и обязательных полей, но часто отсутствует глубокая бизнес-валидация (аномалии, сопоставление контрагентов). Версионирование – пакетное (вся выгрузка как срез).

Производительность и масштаб

Хорошо подходит для регулярных массовых загрузок; масштабируется при адекватном ETL/CDC-слое. Проблемы возникают с частотностью (realtime отсутствует) и синхронизацией мастер-данных.

Плюсы

- Значительное снижение ручного ввода; используемые данные исходят из учётной системы.
- Относительно простая автоматизация импорта.
- Быстрее обеспечить верификацию и агрегирование, чем при ручном вводе.

Минусы

- Зависимость от качества выгрузок ERP; возможны несовпадения справочников и дубли.
- Недостаточно гибкий для событийных изменений (оперативные цены, исправления).
- Трудности с одновременной обработкой нескольких источников и слиянием разных версий одной позиции.

Подходит для

- Организаций с развитой ERP, но без готовых API-интеграций; когда достаточно периодических обновлений (ежедневно/еженедельно).

3) Идеальная (полная интеграция с ERP – real-time / near-real-time)

Описание процесса

Платформа напрямую интегрирована с ERP, реестрами поставщиков, прайс-листами и другими источниками через API, событийную шину (CDC/Kafka) или ETL с короткой задержкой. Все записи имеют канонические идентификаторы и привязаны к мастер-данным.

Архитектура данных / модель

Нормализованная многослойная модель: выделены сущности (Supplier, Item, Unit, PriceRecord, Document, Project). Единая мастер-база (MDM), справочники синхронизированы, версии и audit trail встроены.

Валидация и качество

Сложные бизнес-правила реализованы централизованно (валидация на входе, контроль аномалий, ML-алгоритмы для выявления выбросов). Поддерживаются транзакционные и событийные гарантии, двухсторонняя синхронизация, документ-ориентированная проверка.

Производительность и масштаб

Высокая: поддержка десятков тысяч позиций, множественных поставщиков, историй цен. Позволяет оперативную аналитику и построение KPI в реальном времени.

Плюсы

- Источник истины – система (ERP/MDM), не человек.
- Поддержка корректного аудита, версионности и согласованных мастер-данных.
- Полноценная аналитика, дашборды, тревоги по аномалиям.
- Автоматизация процессов workflow (согласование, подтверждение документов).

Минусы

- Высокие первоначальные инвестиции: интеграция, MDM, изменение процессов.
- Требуется зрелая ИТ-инфраструктура и организационная готовность менять бизнес-процессы.
- Зависимость от качества исходных систем; нужны договорённости о data contracts.

Подходит для

- Крупных ведомств/предприятий, где важна оперативная аналитика, контроль расходов и соблюдение регуляторных требований.

Сравнительная таблица (ключевые критерии)

Критерий	Ручной	Пакетный (ERP dump)	Интеграция (API/MDM)
Источник истины	Сотрудник	ERP (срез)	ERP/MDM (события)
Уникальные идентификаторы	Нет	Частично (зависят от ERP)	Да (канонические ID)
Нормализация	Нет	Частичная	Полная
Валидация	Ручная	Пакетная, базовая	Режим realtime + сложные правила
Аудит/версионность	Нет	Пакетное хранение	Полный audit trail
Масштабируемость	Низкая	Средняя	Высокая
Скорость обновлений	Медленно	Периодически	Near-real-time
Стоимость внедрения	Низкая	Средняя	Высокая
Время на возврат инвестиций (ROI)	Низкое/возможно отрицательное	Умеренное	Высокое (в долгой перспективе)
Готовность к дашбордам/BI	Плохо	Удовлетворительно	Отлично

2. Система, основанная на ручном вводе данных, не является цифровой платформой

Система, основанная на ручном вводе данных в электронные формы, по своей сути не является «цифровой платформой», а представляет собой её антитезу – цифровой регистратор. Это различие носит не терминологический,

а фундаментальный характер, определяющий ценность, масштабируемость и стратегическую роль системы.

1. Система как «регистратор»

Основные характеристики:

- 1) Однонаправленность: поток данных идет в одном направлении: от человека к системе. Система является конечной точкой, а не узлом обмена. Она не взаимодействует с другими системами для обогащения или верификации данных.
- 2) Человек как источник истины: система слепо доверяет данным, введенным пользователем. Она не способна самостоятельно проверить их корректность, сопоставив с эталонными данными из учётных систем (ERP). Таким образом, «источником истины» становится человек со всеми присущими ему рисками ошибок, опечаток и искажений.
- 3) Изолированность: система представляет собой «цифровой силос» – изолированное хранилище, оторванное от реальных бизнес-процессов предприятия. Данные в ней существуют в отрыве от своего источника, что делает невозможным построение сквозного аудита и анализа.
- 4) Ценность на уровне оцифровки: единственная ценность, которую создает такая система, – это перевод бумажного процесса в электронный формат. Она не создает новой информации, не генерирует знаний и не оптимизирует процессы. Это просто оцифрованный бланк.

2. Платформа как «экосистема»

Цифровая платформа функционирует по принципиально иным законам:

- 1) Многонаправленность: платформа является центральным интеграционным хабом. Данные поступают в неё из ERP-систем, синхронизируются со справочниками (MDM), обогащаются из внешних источников и передаются в аналитические инструменты (BI). Она управляет потоками, а не просто их принимает.
- 2) Система как источник истины: платформа сама становится верифицированным и доверенным источником данных. Она автоматически валидирует информацию при поступлении, обеспечивает её консистентность и целостность. Решения принимаются на основе данных платформы, потому что она является гарантом их качества.
- 3) Связность: платформа связывает разрозненные сущности (поставщиков, номенклатуру, документы, цены) в единую, нормализованную модель. Она создаёт контекст, позволяя отслеживать историю изменений, строить сложные зависимости и видеть полную картину.
- 4) Ценность на уровне генерации знаний: платформа – это фундамент для создания новой ценности. На её основе работают автоматизированные workflow, алгоритмы выявления аномалий, системы предиктивной аналитики и интерактивные дашборды, которые превращают сырые данные в управленческие инсайты.

Принцип	Система-Регистратор	Платформа-Экосистема (целевое состояние)
Источник истины	Человек	Система
Роль системы	Пассивное хранение	Активная оркестрация и аналитика
Поток данных	Однонаправленный (ввод)	Многонаправленный (интеграция)
Архитектура	Изолированный силос	Центральный хаб
Результат	Статический снимок данных	Динамическая модель для принятия решений

3. Дорожная карта

1. Фаза 0 – подготовка (1-2 мес):

- Определить каноническую модель данных (минимальный набор сущностей: Supplier, Item, Unit, PriceRecord, Document, Period).
- Сформировать schema-contracts (XSD/JSON-Schema).
- Выработать политики доступа и audit-requirements.

2. Фаза 1 – MVP гибридного режима (3-6 мес):

- Развернуть платформу с поддержкой пакетных загрузок (CSV/XML) + простой формы ручного ввода как fallback.
- Реализовать ETL-процесс, базовую валидацию и реестр справочников.
- Настроить дашборды для основных KPI (средняя цена, медиана, отклонения).

3. Фаза 2 – переход к интеграции (6-18 мес):

- Постепенно подключать ERP через API/CDC для ключевых контрагентов.
- Внедрить MDM и синхронизацию справочников.
- Добавить бизнес-валидации и workflow согласования.

4. Фаза 3 – оптимизация и расширение (18+ мес):

- Реализовать event-driven архитектуру, ML-анализ аномалий, SLA-мониторинг.
- Расширить интеграции на поставщиков и внешние прайс-агрегаторы.
- Полная автоматизация выгрузки регуляторных отчетов.

Этап 0. Аналитика и подготовка (1-2 месяца)

Цель: определить архитектурную модель, стандарты обмена и минимальный состав данных для пилота.

№	Задача	Роли	Оценка, чел.ч
0.1	Анализ текущих форм, потоков данных и источников (ERP, Excel, бухгалтерия)	Аналитик данных, бизнес-архитектор	120
0.2	Формирование канонической модели данных (Supplier, Item, PriceRecord, Document, Period)	Data Architect	80
0.3	Разработка схем (XSD / JSON-Schema) и спецификаций полей	Data Architect, аналитик	60
0.4	Разработка концепции MDM-реестра и правил идентификации записей	Data Architect	40

№	Задача	Роли	Оценка, чел·ч
0.5	Подготовка технического задания для MVP (описание API, ETL, UX, аудит)	Бизнес-аналитик, системный архитектор	100
Итого по этапу			≈400 чел·ч (около 1,0–1,5 чел·мес)

Этап 1. MVP гибридной платформы (3-6 месяцев)

Цель: создать первую работающую версию – ручной интерфейс + загрузка XML/CSV из ERP, базовая валидация и отчётность.

№	Задача	Роли	Оценка, чел·ч
1.1	Проектирование и настройка базы данных (нормализованная модель)	Data Engineer	100
1.2	Разработка backend-модулей для приёма файлов (REST API, XML parser, schema-validation)	Backend Developer	180
1.3	Создание интерфейса ручного ввода и верификации (web-UI, форма, статусы)	Frontend Developer, UX-дизайнер	200
1.4	Разработка базового ETL-процесса (загрузка, дедупликация, логирование ошибок)	Data Engineer	120
1.5	Реализация системы прав и ролей (ответственный, проверяющий, админ)	Backend Developer	80
1.6	Настройка простых дашбордов (ценовые ряды, аномалии, средние значения)	BI-разработчик	120
1.7	Тестирование, пользовательские сценарии, документация	QA-инженер, аналитик	140
1.8	Внедрение пилотной инстанции (деплой, DevOps, CI/CD)	DevOps Engineer	80
Итого по этапу			≈1 020 чел·ч (≈2,5-3 чел·мес)

Этап 2. Интеграция с ERP и MDM-реестром (6-12 месяцев)

Цель: перейти от пакетных загрузок к автоматическому обмену данными, внедрить мастер-данные и централизованный аудит.

№	Задача	Роли	Оценка, чел·ч
2.1	Разработка и согласование API-контрактов с ERP-системами (REST/Soap/ETL)	Системный архитектор, интегратор	200
2.2	Реализация MDM-реестра (поставщики, номенклатура, единицы измерения)	Data Engineer, архитектор	160
2.3	Настройка синхронизации справочников и механизмов CDC	Data Engineer	140
2.4	Разработка бизнес-валидаций (диапазоны, тренды, документы-основания)	Backend Developer, аналитик	180
2.5	Внедрение расширенного аудита и версионности записей	Backend Developer	120
2.6	Расширение UI для просмотра истории изменений и статусов согласования	Frontend Developer	160
2.7	Настройка интеграции с BI-платформой (Power BI / Metabase / Superset)	BI-разработчик	120

№	Задача	Роли	Оценка, чел·ч
2.8	Обучение пользователей, документация, пилотирование в трёх организациях	Методист, аналитик	160
Итого по этапу			≈1 240 чел·ч (≈3-4 чел·мес)

Этап 3. Масштабирование и оптимизация (12-18 месяцев)

Цель: переход на событийную архитектуру, добавление предиктивной аналитики и мониторинга качества данных.

№	Задача	Роли	Оценка, чел·ч
3.1	Внедрение событийной шины (Kafka / RabbitMQ) для realtime-обмена	DevOps, архитектор	200
3.2	Разработка ML-модуля выявления аномалий в ценах и динамике	Data Scientist	160
3.3	Интеграция с внешними источниками (прайс-агрегаторы, гос-реестры)	Backend Developer	120
3.4	Внедрение SLA-мониторинга и дашбордов качества данных	BI-разработчик, DevOps	100
3.5	Оптимизация производительности БД и ETL-процессов	Data Engineer	100
3.6	Обновление пользовательского интерфейса, адаптация под новые формы	UX, Frontend Developer	140
Итого по этапу			≈820 чел·ч (≈2 чел·мес)

Сводная оценка проекта

Этап	Ориентировочная трудоёмкость	Длительность	Основные роли
0. Аналитика и подготовка	~400 ч	1–2 мес	Аналитик, архитектор
1. MVP гибридной платформы	~1 020 ч	3–6 мес	Backend, Frontend, Data Engineer, QA
2. Интеграция с ERP и MDM	~1 240 ч	6–12 мес	Интегратор, архитектор, BI, аналитик
3. Масштабирование и оптимизация	~820 ч	12–18 мес	DevOps, Data Scientist, BI
Итого по проекту	≈3 480 чел·ч (~8–9 чел·мес)	≈18 месяцев	—

Комментарий по ресурсам и организационной модели

1) Команда ядра проекта (core-team, 6-7 человек):

- 1 Системный архитектор / руководитель проекта
 - 2 Data Engineer (2 чел.)
 - 3 Backend Developer
 - 4 Frontend Developer / UX
 - 5 BI-разработчик
 - 6 QA / аналитик
- (+ временные привлечение: методист, data scientist, DevOps)

- 2) **Оптимальная организационная модель:** смешанная – ядро при ОАО «ЦНИИТУ» + интеграторы ведомств, с контрактами данных и регламентом API.
 - 3) **Экономическая эффективность:**
 - 1 Окупаемость наступает на этапе 2 – при переходе от ручного ввода к регулярным ERP-пакетам.
 - 2 После этапа 3 платформа становится «self-sustaining»: данные поступают автоматически, аудит встроен, аналитика строится без ручных действий.
-

4. Конкретный вывод относительно платформенности

- 1) **Необходима платформа с гибридной способностью** – стартовать с пакетного подхода + упрощённого ручного интерфейса, одновременно проектируя каноническую модель данных и MDM. Так можно быстро уменьшить ручной труд и при этом подготовить дорожную карту для полной интеграции.
 - 2) **Полная интеграция (идеальная)** – это целевое состояние, где платформа становится источником истины для ведомственной отчётности и аналитики. Однако переход требует инвестиций, управленческих изменений и договорённостей о data contracts с владельцами ERP.
 - 3) **Если бюджет/ресурсы ограничены**, лучше отказаться от «вечного» поддержания «плоских» форм и сосредоточиться на: (1) единой модели данных; (2) базовой автоматической валидации; (3) реестре справочников; (4) пакетных загрузках с чёткими схемами и логами – это даст максимальный эффект при умеренных затратах.
 - 4) **Для устойчивой платформенности** необходимы: API-first подход, версионизируемые схемы данных, централизованное управление справочниками (MDM), audit trail и workflow-механизмы
-

5. Интеллектуальный центр платформы: зачем нужен и что делает

Платформа, способная лишь собирать и показывать данные, не является цифровым двигателем управления, а представляет собой дорогостоящий регистратор – технологическую оболочку без когнитивного содержания. Без методического ядра любая система, даже идеально интегрированная с ERP и MDM, быстро превращается в хранилище графиков и дашбордов, где каждое новое визуальное представление снижает вероятность принятия взвешенного решения. Чтобы превратить поток цифр в механизм управления, платформа должна обладать интеллектуальным центром – совокупностью формализованных оценочных функций, имеющих экономический смысл, единицу измерения и управляемый рычаг воздействия. Этот центр выступает тем самым «органом мышления» системы, без которого платформа остаётся пассивной инфраструктурой.

5.1. Ошибка агрегирования и иллюзия аналитики

Классическая практика платформенного проектирования строится по схеме: «выгрузили 200 показателей, построили индекс, получили интегральную оценку». На первый взгляд это выглядит логично, но математически бесплодно.

Пусть $x_i(t)$ – наблюдаемые параметры бизнес-процессов (цены, сроки, объёмы, доли). Попытка свести их в единый индекс по формуле:

$$I = \sum_i w_i x_i$$

не имеет физического смысла, поскольку величины x_i различны по размерности и направлению воздействия. Производная такого индекса по любому рычагу управления не определена, а значит, управление на его основе невозможно. Согласно теореме инвариантности агрегатов, если вектор показателей содержит величины разной размерности, то линейная свёртка с произвольными весами не сохраняет смысл при изменении единиц измерения. Следовательно, большинство «индексов зрелости» и «интегральных KPI» являются метками порядка, а не количественными оценками.

В эконометрических терминах это ведёт к спурионной (ложной) регрессии: на коротких исторических рядах коэффициент детерминации R^2 может быть высоким, но прогнозная ошибка растёт лавинообразно. Платформа демонстрирует «красивые тренды», не содержащие управляющего смысла.

5.2. Содержание и структура интеллектуального центра

Интеллектуальный центр – это не алгоритмический модуль, а методологический контур, задающий экономическую интерпретацию данных. Каждая оценочная функция F_j строго описана и включает:

- экономическую сущность – например, «стоимость владения запасом» или «риск недопоставки»;
- единицу измерения (руб./год, руб./тонну, % потерь и т. д.);
- допустимый диапазон значений (≥ 0 , \leq нормативного уровня);
- управленческий рычаг – параметр, изменение которого вызывает количественное изменение F_j (размер партии, срок контракта, структура авансов).

Каждая функция привязана к объектам канонической модели MDM – контракту, поставщику, SKU, проекту. При изменении рычага платформа пересчитывает изменение F_j в денежном выражении, предоставляя пользователю измеримый экономический эффект. Это принципиально отличает интеллектуальную платформу от визуализатора данных: вместо команды «посмотрите на график» система выдаёт конкретное управленческое действие – например: «уменьшите размер партии до 400 единиц – экономия 1,3 млн руб. при сохранении уровня сервиса».

Рассмотрим типичный бизнес-запрос: «Снизить закупочные расходы на 7 %». Обычная аналитика превращает этот запрос в дашборд со

«средневзвешенной ценой», но цена неуправляема: она отражает рынок, а не решения компании.

Интеллектуальный центр платформы строит задачу иначе, декомпозируя изменение совокупных затрат:

$$\Delta TC = \varepsilon_{price} + \varepsilon_{volume} + \varepsilon_{TCO},$$

где ε_{TCO} – скрытые издержки (хранение, устаревание, валютный риск). Именно этот компонент содержит реальные рычаги управления. Функция полной стоимости владения F_{TCO} определяется в руб./год и связана с каждым контрактом. Оптимизационный модуль (например, MILP-модель) решает задачу $\min \sum F_{TCO}$ при ограничениях на уровень сервиса и бюджет, получая портфель контрактов, а не картинку с трендом.

Только построение денежной эквивалентности для каждой функции делает агрегат управляемым и интерпретируемым. Такой подход устраняет логическую ошибку агрегирования.

5.4. Формализация и внедрение

Интеллектуальный центр платформы возникает на нулевом этапе проектирования платформы. Пока инженеры описывают схемы данных (schema-contracts, XSD, JSON-Schema), аналитики формируют дерево ценностных функций, где каждая ветвь имеет стоимостную интерпретацию:

- на этапе MVP API платформы должно возвращать не средние цены, а значения функций F_{TCO} , $F_{Efficiency}$ и т. п. по каждому объекту MDM.

- на этапе интеграции справочник «Критерии оценки» становится самостоятельной сущностью в MDM, синхронизируемой с ERP.

- на этапе оптимизации подключаются модули ML-прогнозов и RL-политик, формирующие предложения: какие контракты закрыть, какие перезаключить, какие параметры изменить.

Так реализуется замкнутый контур управления, где данные проходят полный цикл: данные → оценочные функции → оптимизация → управленческое действие → обратная связь.

5.5. Экономический смысл интеллектуализации платформы

Интеллектуальный центр превращает поток данных в систему количественного управления, где результат выражается в деньгах и реализуется в ERP одним действием. Он объединяет экономическую методологию, математическое моделирование и архитектуру данных в единую когнитивную инфраструктуру. Платформа без такого центра лишь констатирует проблемы, но не предлагает выхода. Когда же внутри неё действует интеллектуальное ядро, система становится инструментом действия: она сама обнаруживает отклонения, оценивает их последствия и предлагает управленческие сценарии, выраженные в стоимостных показателях.