

12+

Кирилл Ледовский

ПРЕДМЕТНО-ОРИЕНТИРОВАННОЕ МЫШЛЕНИЕ ПРИ ЦИФРОВОЙ ТРАНСФОРМАЦИИ ПРЕДПРИЯТИЯ

Как условие перехода к интеллектуальному предприятию



Методология

Граф знаний

Проектная практика

Том II • серия корпоративных изданий «Электронная фабрика»

Кирилл Ледовский

**Предметно-ориентированное
мышление при цифровой
трансформации предприятия**

«Издательские решения»

Ледовский К.

Предметно-ориентированное мышление при цифровой трансформации предприятия / К. Ледовский — «Издательские решения»,

Второй том серии «Электронная фабрика» посвящён предметно-ориентированному мышлению как условию перехода от обычной цифровизации к самоорганизующемуся интеллектуальному предприятию.

Содержание

Место второго тома в серии	7
Как читать второй том после первого тома серии	8
Оглавление	9
Часть I. Мировой контекст предметно-ориентированного мышления: цифровые двойники, графы знаний и самоорганизующееся интеллектуальное предприятие	10
Глава 1. Почему движение к интеллектуальному предприятию является мировым трендом	11
Глава 2. Digital Twin of Organization и цифровая нить предприятия	13
Глава 3. Графы знаний, онтологии и семантический слой предприятия	15
Глава 4. Process mining, object-centric process mining и фактические потоки деятельности	17
Глава 5. Самоорганизация, морфогенетическое развёртывание и организационная энтропия	19
Глава 6. Платформенные подходы: SAP, Palantir, Celonis и сдвиг к операционной онтологии	21
Глава 7. Среднее производственное предприятие как основное поле конкуренции	23
Глава 8. Почему предметно-ориентированное мышление становится условием самоорганизующегося интеллектуального предприятия	25
Библиография к Части I	27
Часть II «Почему ИИ предприятия не может рассуждать без предметного мира и иницилирующего ядра»	28
Глава 9. ИИ пришёл в предприятие и обнаружил пустоту предметного мира	29
Глава 10. Галлюцинация ИИ как управленческий симптом	34
Глава 11. Кантовская развилка: задать предмет и дать предмет	37
Глава 12. Авторский опыт: почему консультанты видят кнопки, но не видят предметы	41
Часть III «Почему процессов, документов, ERP-объектов и BPM/BPA- моделей недостаточно»	44
Глава 13. Процесс без предмета — процедурная оболочка	45
Глава 14. ERP-объект не равен бизнес-предмету	50
Конец ознакомительного фрагмента.	55

Предметно-ориентированное мышление при цифровой трансформации предприятия

Кирилл Ледовский

© Кирилл Ледовский, 2026

ISBN 978-5-0070-7614-2

Создано в интеллектуальной издательской системе Ridero

УДК 004.418:005.7

ББК 65.291.21с1

Кирилл Ледовский

Л39 Предметно-ориентированное мышление при цифровой трансформации предприятия

Единая редакция v07 — 2026. — ил. Настоящий второй том серии «Электронная фабрика» посвящён предметно-ориентированному мышлению как условию перехода от обычной цифровизации к самоорганизующемуся интеллектуальному предприятию. Первый том серии описывает управленческую рамку интеллектуального предприятия. Второй том рассматривает понятийный и методологический слой, без которого невозможно собрать цифровой двойник деятельности, граф знаний, цифровую нить и иницилирующее ядро предприятия.

Под самоорганизующимся интеллектуальным предприятием в этом издании понимается предприятие, в котором предметная модель, граф знаний, цифровая нить, фактические потоки данных, управленческие события и человеко-машинный контур образуют механизм устойчивого упорядочивания деятельности. Такая система не сводится к внедрению ERP, BPM/BPA, BI, электронного документооборота или корпоративного ИИ-помощника. Она требует отдельной методологии выделения бизнес-предметов, состояний, событий, носителей, источников, ролей и связей.

Первая часть второго тома вводит мировой контекст: цифровые двойники организации, цифровую нить, графы знаний, онтологии, process mining, object-centric process mining, платформенные подходы SAP, Palantir и Celonis, а также исследовательскую линию самоорганизации, морфогенетического развёртывания и организационной энтропии. Этот обзор показывает, что предметно-ориентированное мышление является не частной техникой ERP-аналитика, а необходимым основанием для нового класса корпоративных интеллектуальных систем.

Дальнейшее содержание второго тома раскрывает, почему предприятие не может устойчиво рассуждать о себе только через процессы, документы, ERP-объекты, СМК-формы, показатели и отчёты. За ними должны быть выделены управляемые бизнес-предметы: потребности, заказы, запасы, партии, обязательства, отклонения, рекламации, события качества, лимиты, состояния и переходы. Эти предметы образуют материал для табличной базы модели, графа знаний и последующего таксоно-онтологического метода.

Издание адресовано специалистам, которым требуется методологически строгий переход от корпоративной цифровизации к интеллектуализации предприятия: бизнес-архитекторам,

ERP-методологам, аналитикам, руководителям проектов, специалистам СМК, архитекторам данных, консультантам и исследователям корпоративных информационных систем.

© Кирилл Ледовский

Место второго тома в серии

Второй том серии занимает промежуточное место между управленческой картиной первого тома и профессиональной методикой третьей книги серии. Первый том отвечает на вопрос «куда идти»: к интеллектуальному предприятию, цифровому двойнику деятельности, иницирующему ядру и снижению организационной энтропии. Второй том отвечает на вопрос «как нужно думать»: через бизнес-предметы, состояния, события, носители и связи. Третья книга серии отвечает на вопрос «как это делать профессионально»: через таксоно-онтологический метод, табличную базу модели, граф знаний, источники и проверяемые проектные артефакты.

Поэтому эту книгу не нужно читать как отдельный справочник по ERP и не нужно читать как книгу только о работе с искусственным интеллектом. Это книга о способе различения деятельности. Если предприятие не умеет различать, чем оно управляет, оно не может устойчиво автоматизировать, контролировать, анализировать и развивать себя. Оно может иметь много систем, документов и отчётов, но оставаться несобранным.

Предметно-ориентированное мышление переводит внимание с вопроса «какой процесс нарисован?» на вопрос «какой предмет проходит через процесс и как меняется его состояние?». Оно переводит вопрос «какой документ создать?» в вопрос «какое событие деятельности или качества должно быть зафиксировано?». Оно переводит вопрос «какую кнопку нажать?» в вопрос «какой управляемый переход должен произойти в реальности предприятия?».

Второй том подготавливает важный вывод: иницирующее ядро нельзя собрать из одних процессных схем и нельзя получить автоматически вместе с ERP. Ядро собирается из предметно различённых элементов деятельности и становится действующим только тогда, когда эти элементы связываются с информационными потоками предприятия.

Как читать второй том после первого тома серии

Второй том следует читать в четырёх слоях. Первый слой — исследовательский: как мировой переход к цифровым двойникам, графам знаний, process mining, онтологиям и корпоративному искусственному интеллекту ведёт к самоорганизующемуся интеллектуальному предприятию. Второй слой — аналитический: как отличать бизнес-предмет от документа, ERP-объекта, процесса, формы, показателя и отчёта. Третий слой — методологический: какие элементы затем переходят в профессиональный таксоно-онтологический метод. Четвёртый слой — проектный: какие различия необходимы для последующей сборки инициирующего ядра.

Читателю не нужно сразу строить полную онтологию предприятия. На этом уровне важно научиться задавать правильные вопросы: что является предметом управления, в каком состоянии он находится, какое событие изменяет его состояние, где этот факт зафиксирован, кто владелец, какой источник подтверждает запись и как этот предмет связан с другими предметами.

Второй том не даёт полной технологии эксплуатации ядра. Эксплуатационный ввод инициирующего ядра, его связь с ERP, CMK, Excel, ЭДО, почтой, RPA-роботами, пользовательскими действиями и обратной связью раскрывается в отдельном практическом руководстве серии. Здесь фиксируется мыслительный фундамент: без предметного различения такое ядро невозможно.

Оглавление

Часть I. Мировой контекст предметно-ориентированного мышления: цифровые двойники, графы знаний и самоорганизующееся интеллектуальное предприятие

Переход к интеллектуальному предприятию следует рассматривать не как локальное обновление корпоративных ИТ-систем, а как часть более широкого мирового движения к новому способу организации производственных и управленческих систем. В этом движении соединяются несколько направлений: цифровой двойник организации, цифровая нить, графы знаний, онтологии, process mining, объектно-центрированный анализ процессов, корпоративные AI-платформы и исследования самоорганизующихся систем.

Важен экономико-технологический ракурс. Страны и корпорации, которые быстрее научатся превращать предприятия в интеллектуальные, получают преимущество не только в аналитике, но и в скорости реакции на отклонения, в снижении управленческих потерь, в качестве координации, в способности быстро переносить успешные операционные модели между площадками и в умении связывать человеческое решение с машинно читаемой структурой деятельности.

Особое значение имеют средние производственные предприятия. Они не обладают ресурсами крупнейших корпораций, но часто удерживают промышленную компетенцию территории: выпускают нишевую продукцию, создают рабочие места, формируют спрос на инженерные, ремонтные, транспортные, сервисные и малые подрядные структуры. В международной статистике малые и средние предприятия рассматриваются как основа занятости и создания стоимости: в странах ОЭСР они составляют около 99 процентов фирм и создают в среднем 50—60 процентов добавленной стоимости [14]; Всемирный банк также связывает SME-сектор с более чем половиной занятости в мире [15].

Из этого следует практический вывод: интеллектуализация средних производственных предприятий становится не второстепенной задачей, а вопросом промышленной устойчивости. Если предприятие остаётся на уровне разрозненных систем, ручных таблиц, слабо связанных регламентов и неустойчивого знания опытных сотрудников, оно проигрывает тем участникам рынка, которые быстрее строят цифровые двойники, графы знаний, фактические контуры наблюдения и управляемые механизмы обновления модели.

Эта часть вводит обзор направлений, которые формируют мировой контекст предметно-ориентированного мышления. Она не заменяет профессиональный метод, раскрываемый далее, но показывает, почему такой метод становится необходимым. Предметно-ориентированное мышление связывает мировой технологический сдвиг с практикой описания предприятия: чтобы цифровой двойник и граф знаний стали рабочими, предприятие должно различать собственные бизнес-предметы, их состояния, события, носители и связи.

Глава 1. Почему движение к интеллектуальному предприятию является мировым трендом

Мировой тренд к интеллектуальному предприятию складывается не из одного технологического направления. Его формируют одновременно корпоративный искусственный интеллект, цифровые двойники, платформенная аналитика, онтологии, process mining, имитационное моделирование, автоматизация решений и эксплуатационные контуры обратной связи. Отдельные элементы этого тренда могут называться по-разному, но общий вектор устойчив: предприятие всё чаще рассматривается как система, которую нужно не только автоматизировать, но и представить в виде связанной, проверяемой и обновляемой модели.

В традиционном подходе корпоративная информационная система фиксирует факты: документы, заказы, движения, статусы, платежи, остатки, операции, отчёты. Это необходимо, но недостаточно. Интеллектуальное предприятие требует не только фиксации факта, но и объяснения его места в общей структуре деятельности: какой предмет изменился, какая роль выполнила действие, какой источник подтверждает событие, какой риск возник, какой показатель должен измениться и какое решение требуется. Именно этот переход от регистрации к связанному смысловому управлению формирует новое поле конкуренции.

Gartner выделяет Digital Twin of an Organization как класс платформ, помогающих приоритизировать, планировать, моделировать, анализировать и масштабировать сложные инициативы [1]. Даже если конкретные платформы различаются по зрелости, сама постановка показывает смещение: объектом моделирования становится не только станок, изделие или производственная линия, но организация как система процессов, данных, ролей, решений и зависимостей.

Параллельно развивается представление о цифровой нити. Digital Twin Consortium определяет digital thread как двунаправленную, надёжную и взаимосвязанную информационную систему, которая связывает структуру, поведение, пространство, время и стадии жизненного цикла [2]. Для предприятия это означает принципиальный переход: важен не только отдельный цифровой двойник, но и трассируемая связь между требованиями, проектированием, производством, качеством, эксплуатацией, сервисом, финансовыми последствиями и управленческими решениями.

Этот сдвиг нельзя сводить к очередной версии ERP или BPM. ERP остаётся основой транзакционного и учётного слоя. BPM/BPA-системы сохраняют значение как средства описания процессов, ролей и регламентов. BI и аналитические платформы помогают видеть показатели. Однако интеллектуальное предприятие требует слоя, который связывает эти элементы в единую предметную структуру. Без такого слоя предприятие получает набор цифровых следов, но не получает устойчивой способности рассуждать о собственной деятельности.

Экономико-технологический смысл этого тренда состоит в том, что управление становится фактором производственной конкуренции. Если две компании выпускают сходную продукцию и используют сопоставимое оборудование, преимущество начинает смещаться к тому, кто быстрее видит отклонение, точнее понимает его причину, быстрее перестраивает цепочку действий, аккуратнее связывает качество с производством, финансы с обязательствами, склад с доступностью ресурса, а решение руководителя с фактическим состоянием деятельности.

Для средних производственных предприятий этот вопрос особенно острый. У крупных корпораций есть ресурсы для длительных программ цифровой трансформации, собственных платформ, центров компетенций и больших ИТ-команд. Среднее предприятие чаще вынуждено действовать быстрее и экономнее. Если оно не получает метод быстрого построения предметного ядра, оно оказывается между двумя рисками: либо остаётся в ручной управляемости, либо покупает отдельные системы без общего интеллектуального контура.

Поэтому движение к интеллектуальному предприятию нужно понимать как переход к новому способу сборки управленческой системы. Это не отказ от ERP, CMK, BI, ЭДО, BPM/BPA и RPA. Это изменение их места. Они становятся носителями фактов, представлениями, исполнительными и аналитическими контурами внутри более широкой модели. Центром становится предметно связанное ядро, которое удерживает бизнес-предметы, состояния, события, источники, роли и управленческие реакции.

Второй том серии раскрывает именно этот слой. Предметно-ориентированное мышление необходимо потому, что цифровой двойник организации и граф знаний не могут быть построены из одних документов, процессных схем и транзакционных записей. Им нужен предметный материал: что предприятие считает управляемой сущностью, какие состояния эта сущность проходит, где событие фиксируется и как оно связано с дальнейшим действием.

Глава 2. Digital Twin of Organization и цифровая нить предприятия

Понятие цифрового двойника долгое время связывалось прежде всего с инженерными объектами: изделием, станком, производственной линией, оборудованием, инфраструктурным объектом. Такой цифровой двойник описывает физическую систему, её параметры, поведение, состояние, историю и прогнозируемые изменения. Но логика цифрового двойника постепенно расширяется на организационные системы, где объектом моделирования становится не только физическая установка, а деятельность предприятия.

Digital Twin of Organization переносит идею цифрового двойника на организацию как систему. В этом подходе моделируются не только активы, но и процессы, роли, данные, приложения, решения, зависимости, ограничения, показатели и сценарии изменений [1]. Иными словами, организация становится объектом моделирования: можно анализировать, как изменение правила, ресурса, роли, спроса, поставщика или информационного потока повлияет на выполнение деятельности.

Для производственного предприятия это означает переход от частной оптимизации к системной. Можно отдельно улучшать склад, закупки, производство, качество или финансы. Но реальное конкурентное преимущество возникает тогда, когда предприятие понимает связи между этими контурами. Задержка поставки влияет на производственный план; изменение спецификации влияет на потребность, закупку и себестоимость; решение по несоответствию влияет на складскую доступность, отгрузку, обязательства и выручку. Цифровой двойник организации должен удерживать эти связи.

Цифровая нить является связующим принципом такого двойника. Её смысл состоит в трассируемой связности: от требования к проектному решению, от проектного решения к производственному исполнению, от исполнения к качеству, от качества к допуску, от допуска к отгрузке, от отгрузки к обязательству, от обязательства к финансовому результату. Определение Digital Twin Consortium подчёркивает именно многомерную связность по структуре, поведению, времени и жизненному циклу [2].

Без цифровой нити цифровой двойник организации превращается в набор локальных моделей. Отдельный процесс может быть описан, отдельный отчёт может быть настроен, отдельная интеграция может работать. Но если между ними нет трассируемой связи, предприятие не получает целостного механизма анализа последствий. Тогда каждое отклонение приходится разбирать вручную, через совещания, переписки и уточнения у опытных сотрудников.

Промышленное значение цифровой нити особенно заметно в дискретном производстве. Изделие, заказ, партия, маршрут, комплектующие, контроль качества, отклонение, рекламация и сервисное событие образуют цепочку управляемых предметов. Если цепочка разрывается, предприятие может видеть отдельные факты, но не видеть причинно-следственный маршрут. Цифровая нить восстанавливает этот маршрут в машинно читаемом виде.

В мировой литературе цифровые двойники всё чаще связываются с графовыми и онтологическими подходами. Для существующих производственных систем, особенно brownfield-среды, создание цифрового двойника требует объединения информации из разнородных источников. Исследования предлагают графовые представления для связывания таких источников и выявления шаблонов в структуре производственных систем [11]. Это близко к задаче предприятия, у которого данные уже распределены между ERP, CMK, Excel, электронным документооборотом, оборудованием, складскими системами и человеческими процедурами.

Важный вывод для предметно-ориентированного мышления состоит в следующем: цифровой двойник организации не может строиться только как модель процессов. Он требует устойчивых предметных узлов. Процесс показывает движение, но цифровая нить должна знать,

что именно движется: потребность, заказ, партия, обязательство, протокол, несоответствие, рекламация, лимит, платеж, производственное задание. Без предметов невозможно связать события в устойчивую нить.

Поэтому второй том вводит бизнес-предмет как основную единицу мышления. Бизнес-предмет нужен не для терминологической точности самой по себе. Он нужен как узел будущего цифрового двойника организации. Через него связываются процессы, документы, ERP-объекты, СМК-записи, показатели, события и решения. Именно такие предметы позволяют превратить цифровой двойник из набора схем в управляемую модель деятельности.

Глава 3. Графы знаний, онтологии и семантический слой предприятия

Второй крупный мировой контур связан с графами знаний и онтологиями. Если цифровой двойник организации отвечает на вопрос о моделировании деятельности как системы, то граф знаний отвечает на вопрос о способе представления этой системы в машинно читаемой форме. Граф знаний связывает сущности и отношения: предмет, состояние, событие, роль, документ, источник, показатель, риск, правило, решение, действие.

Семантические технологии имеют формальную основу. RDF задаёт модель представления информации в виде графа троек «субъект — предикат — объект» [3]. OWL предназначен для представления более сложного знания о вещах, группах вещей и отношениях между ними, причём такое знание может использоваться программами для проверки согласованности и вывода неявных связей [4]. Эти стандарты важны не потому, что каждое предприятие обязано сразу строить модель в RDF/OWL, а потому что они показывают зрелую мировую линию: знание должно быть представлено не только в тексте, но и в формальной структуре.

Для предприятия это означает переход от словаря к онтологическому слою. Словарь объясняет термины, но не показывает, как они связаны. Онтологический слой фиксирует классы предметов, их свойства, состояния, события, допустимые отношения и ограничения. Например, термин «партия» сам по себе мало что даёт. В онтологическом слое нужно указать, чем партия отличается от серии, запаса, производственного задания, выпуска, отгрузочной партии и объекта контроля качества.

Граф знаний предприятия может использовать различные технологические формы: таблицы, реляционные базы, графовые базы, RDF-хранилища, документные структуры, специализированные платформы. Но независимо от технического носителя он должен решать одну задачу: переводить деятельность предприятия из набора документов и записей в связанную сеть сущностей, отношений, источников и правил.

Именно здесь предметно-ориентированное мышление становится необходимым. Нельзя построить граф знаний предприятия, если неясно, какие предметы должны стать узлами графа. Процессная схема может дать действия, ERP может дать документы и регистры, СМК может дать процедуры и формы, ВІ может дать показатели. Но граф знаний требует нормализованного ответа: какие управляемые сущности стоят за этими носителями и как они связаны между собой.

Семантический слой особенно важен для корпоративного искусственного интеллекта. Большая языковая модель может формировать текст, объяснять связи и помогать пользователю задавать вопросы. Но если за ответом нет графа знаний, модель получает только фрагменты документов и статистические связи между словами. В этом случае результат зависит от качества контекста, но не получает устойчивой предметной опоры.

Платформенные решения крупных вендоров показывают тот же сдвиг. SAP развивает SAP Knowledge Graph как способ grounding корпоративного ИИ в бизнес-данных и контексте [7]. Palantir строит Ontology как слой единой операционной реальности, который связывает данные, действия, AI-команды и транзакционные обновления в неоднородной инфраструктуре [8]. Эти примеры не следует копировать как готовый метод для любого предприятия, но они показывают направление: корпоративный ИИ требует не только данных, но и семантического слоя.

В производственной среде граф знаний выполняет ещё одну функцию: он позволяет объединять инженерные, производственные, качественные и управленческие данные. Исследования цифровых двойников в производстве всё чаще используют knowledge graph architecture для объединения разнородных данных и поддержки интеллектуального моделирования про-

изводственных процессов [12]. Это особенно значимо для предприятий, где реальность распределена между конструкторской документацией, ERP, СМК, станочными данными, Excel-файлами и экспертным знанием сотрудников.

Таким образом, мировой контур графов знаний и онтологий подтверждает центральный тезис второго тома. Предприятие управляет не текстами и не экранными формами, а различными предметами, их состояниями и связями. Граф знаний является машинной формой этого различения. Предметно-ориентированное мышление даёт материал, без которого граф знаний остаётся технической оболочкой.

Глава 4. Process mining, object-centric process mining и фактические потоки деятельности

Третье направление мирового движения связано с process mining. Если классическое процессное моделирование показывает, как процесс описан или как он должен выполняться, process mining позволяет анализировать фактическое прохождение процесса по событиям, зафиксированным в информационных системах. Process Mining Manifesto подчёркивает возможность извлекать знания о процессах из event logs и использовать их для discovery, monitoring and improvement процессов [5].

Этот подход важен потому, что он меняет источник знания о процессе. В традиционном обследовании процесс описывается через интервью, регламенты, рабочие инструкции и экспертные сессии. Это необходимо, но недостаточно: реальное выполнение часто отличается от нормативного описания. Люди обходят маршруты, изменяют последовательность, выполняют действия параллельно, пропускают контрольные точки, задерживают операции, создают документы позже фактического события. Process mining позволяет увидеть часть этой фактической картины через цифровые следы.

Однако классический process mining сталкивается с ограничением, близким к проблеме второго тома. Процесс редко существует как один изолированный case. В реальном предприятии заказ клиента связан с производственным заказом, закупочной потребностью, поставкой, складским запасом, партией, качеством, отгрузкой и платежом. Такие объекты взаимодействуют между собой. Поэтому развивается object-centric process mining, где анализ строится не вокруг одной линейной последовательности, а вокруг взаимодействующих объектов и событий [6].

Object-centric process mining показывает, что современный анализ процессов движется к предметной логике. Исследования подчёркивают, что в объектно-центрированных процессах экземпляры процессов взаимодействуют через события, связывающие разные объекты [13]. Для предприятия это означает: нельзя устойчиво анализировать выполнение, если модель не различает предметы, которые участвуют в процессе. Заказ, партия, запас, поставка, рекламация и платеж не являются просто атрибутами одного дела; они образуют сеть взаимосвязанных предметов.

Из этого следует важное различие между процессной схемой и фактическим потоком деятельности. Процессная схема может описывать последовательность действий. Фактический поток деятельности включает события, документы, задержки, исключения, возвраты, изменения ответственных, неполные данные, ручные обходы и связи между несколькими предметами. Чтобы связать эти уровни, требуется предметная структура: какие события относятся к какому предмету и какой переход состояния они подтверждают.

Средние производственные предприятия часто имеют богатый фактический поток, но слабое формальное описание. Часть событий фиксируется в ERP, часть в Excel, часть в СМК-документах, часть в почте, часть в производственных журналах, часть в памяти сотрудников. При обычной цифровизации эти следы могут оставаться разрозненными. В интеллектуальном предприятии они должны быть связаны через бизнес-предметы и цифровую нить.

Process mining также показывает ограничение ручного описания бизнес-процессов. Если процесс описан один раз и затем не связан с фактическими событиями, он быстро теряет соответствие реальности. Предприятие продолжает изменяться: появляются новые исключения, поставщики, маршруты, регуляторные требования, способы контроля, ручные таблицы. Без механизма сопоставления описанной модели с фактическим потоком деятельность снова становится неустойчивой.

Поэтому предметно-ориентированное мышление должно учитывать оба уровня: нормативно описанный процесс и фактический поток. Бизнес-предмет связывает их. Он позволяет спросить: какой предмет должен пройти процесс согласно модели; какие события по этому предмету реально возникли; какие состояния подтвердились; какие носители использованы; где появились разрывы; какие действия должны последовать.

Эта логика напрямую готовит понятие самоорганизующегося интеллектуального предприятия. Самоорганизация здесь не означает отсутствие управления. Напротив, она требует структурированной среды, где новые события не просто накапливаются, а связываются с предметами, состояниями, источниками, правилами и управленческими реакциями. Process mining даёт фактический поток; предметно-ориентированная модель задаёт структуру, в которую этот поток может быть включён.

Глава 5. Самоорганизация, морфогенетическое развёртывание и организационная энтропия

Четвёртое направление связано с самоорганизацией и морфогенетическим развёртыванием. В сложных системах самоорганизация означает возникновение упорядоченных структур за счёт внутренних взаимодействий элементов и правил среды. Морфогенетическая инженерия рассматривает возможность проектировать системы, способные к «self-architecturing», то есть к формированию сложной структуры на основе заданных правил, локальных взаимодействий и программируемой самоорганизации [9].

Для предприятия такая аналогия требует аккуратного переноса. Предприятие не является биологической системой, но оно является сложной организационной системой: в нём есть множество участников, ролей, документов, потоков, правил, исключений, информационных систем, внешних требований, управленческих решений и повторяющихся событий. Если эти элементы не связаны, возникает организационная энтропия: неустойчивость, расползание смыслов, информационная асимметрия, неясные цели, неэффективные потоки и ошибочное распределение ресурсов. В организационной психологии и менеджменте management entropy связывается именно с хаотичным и неустойчивым состоянием организации [10].

Второй том вводит термин «самоорганизующееся интеллектуальное предприятие» для обозначения следующего уровня интеллектуализации. Интеллектуальное предприятие имеет цифровой двойник деятельности, граф знаний и человеко-машинный контур управления. Самоорганизующееся интеллектуальное предприятие дополнительно обладает механизмом, через который новые факты, события, статусы, записи, вопросы, разрывы и решения встраиваются в предметную структуру, уточняют её и поддерживают снижение организационной энтропии.

Ключевым элементом такого механизма является иницирующее ядро. Его нельзя понимать как один документ, одну базу или один программный модуль. Это связанная структура бизнес-предметов, состояний, событий, носителей, источников, ролей, правил, проверок и управленческих реакций. После ввода в информационные потоки предприятия ядро начинает выполнять структурирующую функцию: новые факты не просто складываются в архив, а получают место в модели.

Морфогенетический смысл этого процесса состоит в переходе от исходной структуры к развёрнутой организационной форме. В начале ядро может быть минимальным: набор ключевых предметов, процессов, таблиц, связей и источников. Но если оно правильно связано с потоками предприятия, оно становится механизмом дальнейшего развёртывания. Каждый новый подтверждённый факт уточняет состояние предмета; каждый GAP фиксирует границу незнания; каждое управленческое решение создаёт связь между отклонением и действием; каждая запись качества связывает процедуру с доказательством.

Такой подход отличается от внедрения ERP. ERP-проект обычно направлен на настройку прикладной системы, миграцию данных, запуск документов, обучение пользователей, регламенты и поддержку операций. Это необходимая и сложная работа. Но внедрение самоорганизующегося интеллектуального предприятия требует другого слоя: предметной модели, графа знаний, цифровой нити, правил приёма фактов, связи с источниками, управления версиями и контуров актуализации модели.

Отличие можно показать на примере несоответствия качества. В обычной цифровизации предприятие может завести форму, журнал, маршрут согласования и отчёт. В самоорганизующемся интеллектуальном предприятии несоответствие становится бизнес-предметом: у него есть источник выявления, состояние, владелец, связь с партией, документом контроля, решением, корректирующим действием, риском, показателем и последующим изменением про-

цесса. Тогда каждое новое несоответствие не просто регистрируется, а уточняет структуру знания о качестве.

Другой пример — производственная потребность. В обычной системе она может проявляться как строка плана, заявка, заказ, резерв или ручная таблица. В предметной модели потребность описывается как управляемый предмет со стадиями возникновения, подтверждения, обеспечения, дефицита, замены, переноса и закрытия. После ввода ядра в потоки предприятия факты по потребности начинают структурировать работу закупки, склада, производства и финансового планирования.

Поэтому самоорганизация в контексте предприятия не означает стихийность. Она означает проектируемую способность системы поддерживать собственную структуру через фактические события и обратную связь. Морфогенетическое развёртывание задаёт язык описания этого процесса: из исходного ядра разворачивается более сложная, связанная и проверяемая модель деятельности. Предметно-ориентированное мышление обеспечивает материал для такого развёртывания.

Глава 6. Платформенные подходы: SAP, Palantir, Celonis и сдвиг к операционной онтологии

Платформенные решения крупнейших игроков показывают, что мировой рынок движется от набора приложений и отчётов к контекстным, онтологическим и процессно-интеллектуальным слоям. Эти решения различаются по архитектуре, назначению и коммерческой модели, но в них виден общий сдвиг: корпоративный ИИ должен опираться на бизнес-контекст, операционные связи и управляемую модель предприятия.

SAP Knowledge Graph позиционируется как слой, который помогает корпоративному AI работать с бизнес-данными и контекстом SAP, снижая риск недостоверных ответов и поддерживая бизнес-рекомендации [7]. Для методологии второго тома здесь важна не конкретная реализация SAP, а сам принцип: корпоративному ИИ требуется knowledge graph, связывающий данные с бизнес-смыслом.

Palantir Ontology развивает другую линию: единый операционный слой, в котором данные, операции, AI-команды и транзакционные обновления координируются поверх неоднородной инфраструктуры. Описание Palantir подчёркивает работу Ontology как масштабируемой архитектуры для Human+AI teams и как слоя, обеспечивающего согласованность и целостность в реальном времени [8]. Это близко к идее операционной онтологии предприятия: модель не только описывает сущности, но и связывает их с действиями.

Celonis развивает Process Intelligence Platform, объединяя process mining, AI и анализ фактических процессов [16]. Object-centric process mining в экосистеме Celonis также отражает переход от линейных process cases к нескольким взаимосвязанным объектам процесса [6]. Для предметно-ориентированной методологии это особенно важно: мировой рынок сам движется к пониманию, что процесс нельзя адекватно анализировать без объектов, которые в нём взаимодействуют.

Эти платформенные подходы нельзя трактовать как готовую универсальную методику для среднего производственного предприятия. Крупная платформа может быть недоступна по стоимости, избыточна по масштабу или неприменима без предварительной методологической подготовки. Но они показывают, какой слой становится предметом конкуренции: не просто транзакционная система, не просто отчётность, не просто процессная схема, а операционная модель, соединяющая данные, смысл, процессы, действия и AI.

Для предприятий среднего размера возникает практический вопрос: как получить методологический эквивалент такого слоя без многолетней программы и без зависимости от одной платформы. Ответ не сводится к покупке программы. Нужна технология сборки предметного ядра: выделение бизнес-предметов, описание состояний, связь с процессами, источниками, ERP-носителями, СМК-записями, показателями, цифровой нитью и графом знаний.

Этот вывод особенно важен для предприятий, работающих в нишевой продукции и дискретном производстве. Их конкурентоспособность часто зависит не только от оборудования, но и от способности управлять сложными связями между заказами, спецификациями, поставками, качеством, сроками, заменами, финансовыми ограничениями и персональной ответственностью. Если такие связи остаются в головах отдельных сотрудников, предприятие технологически уязвимо.

Платформенный тренд показывает: крупные игроки уже строят слои бизнес-контекста, онтологии, process intelligence и AI-операций. Следовательно, методология среднего предприятия должна отвечать не на вопрос «какую систему купить», а на вопрос «какую предметную структуру нужно собрать, чтобы предприятие могло перейти к такому классу управления».

Поэтому второй том не ограничивается критикой процессов, документов или ERP-объектов. Он готовит понятийный слой, который позволяет понимать мировые платформен-

ные решения без прямого копирования их архитектуры. Бизнес-предмет, состояние, событие, носитель и граф знаний становятся универсальными элементами, через которые предприятие может строить собственный путь к самоорганизующемуся интеллектуальному предприятию.

Глава 7. Среднее производственное предприятие как основное поле конкуренции

В обсуждении интеллектуального предприятия часто внимание смещается к крупным корпорациям, промышленным холдингам и технологическим платформам. Такой фокус понятен: у крупных компаний больше данных, ресурсов, публичных проектов и исследовательских программ. Однако с экономической точки зрения особенно важным полем применения становятся средние производственные предприятия, включая региональные и градообразующие производства.

Малые и средние предприятия в международной статистике рассматриваются как значимый источник занятости, добавленной стоимости и локального экономического развития. ОЭСР указывает, что SME-сектор составляет около 99 процентов фирм и формирует значительную часть занятости и добавленной стоимости [14]. Всемирный банк также характеризует SMEs как основу большинства экономик, на которую приходится около 90 процентов предприятий и более половины глобальной занятости [15]. Эти данные не описывают только промышленность, но они показывают масштаб зависимости экономики от предприятий среднего и малого размера.

Для промышленной территории среднее производственное предприятие выполняет не только функцию выпуска продукции. Оно поддерживает инженерные компетенции, потребность в ремонте, логистике, обслуживании, снабжении, локальном малом бизнесе, профессиональном обучении и занятости. Если такое предприятие теряет конкурентоспособность, последствия выходят за пределы его баланса. Снижается устойчивость экономической среды вокруг него.

Именно поэтому интеллектуализация средних предприятий имеет стратегическое значение. Если внешние конкуренты быстрее внедряют цифровые двойники, графы знаний, process intelligence и AI-контуры, они получают преимущество в скорости расчёта, адаптации, контроля качества, управления сроками, себестоимостью и исполнением обязательств. Среднее предприятие, остающееся на уровне ручной координации и фрагментарной цифровизации, постепенно теряет пространство для манёвра.

Эта проблема особенно заметна в дискретном производстве, приборостроении, машиностроении, производстве комплектующих, пищевой промышленности и других областях, где продукцию можно заменить поставками из других регионов или стран. Если внешний производитель быстрее перестраивает цепочку поставки, точнее управляет качеством и ниже держит операционные потери, локальное предприятие проигрывает не из-за одного плохого решения, а из-за более слабой системы управления деятельностью.

Крупные инфраструктурные и сырьевые предприятия обладают иной устойчивостью. Их положение определяется ресурсной базой, инфраструктурой, лицензиями, масштабом капитала и государственными ограничениями. Средние производственные предприятия чаще конкурируют по скорости, качеству, стоимости, гибкости, способности выполнять нестандартные заказы и удерживать технологическую дисциплину. Здесь интеллектуализация даёт не абстрактное преимущество, а прямой эффект управляемости.

Следовательно, вопрос о самоорганизующемся интеллектуальном предприятии нельзя рассматривать только как интерес крупных вендоров или академических лабораторий. Это вопрос технологического выживания среднего промышленного слоя. Если метод внедрения интеллектуального предприятия остаётся недоступным, дорогим или неопределённым, именно этот слой оказывается наиболее уязвимым.

Предметно-ориентированное мышление предлагает путь к снижению входного порога. Оно не требует сначала строить всю корпоративную платформу. Оно предлагает начать с раз-

личения управляемых предметов, их состояний, событий, носителей и связей. На этой основе можно собирать табличную базу модели, граф знаний, цифровую нить и инициирующее ядро, постепенно подключая ERP, CMK, Excel, ЭДО, RPA и фактические потоки.

Такой подход особенно важен потому, что внедрение самоорганизующегося интеллектуального предприятия отличается от внедрения ERP. ERP-проект имеет свои методики, этапы, роли, настройки, миграции и испытания. Интеллектуализация требует другого результата: не только запущенной системы, но и управляемого ядра, которое принимает факты, фиксирует разрывы, поддерживает модель, связывает предметы и возвращает в организацию управленческие действия. Это отдельное технологическое ноу-хау, которое необходимо описывать и развивать.

Глава 8. Почему предметно-ориентированное мышление становится условием самоорганизующегося интеллектуального предприятия

Предыдущие главы показали несколько мировых направлений: цифровой двойник организации, цифровую нить, графы знаний, онтологии, process mining, object-centric process mining, платформенные операционные онтологии, морфогенетическую самоорганизацию и организационную энтропию. Эти направления не совпадают друг с другом, но сходятся в одном пункте: предприятие должно быть представлено как связанная система сущностей, событий, отношений, источников и действий.

Именно здесь появляется роль предметно-ориентированного мышления. Оно отвечает на вопрос, какой материал должен попасть в такую систему. Если предприятие не различает бизнес-предметы, оно не сможет построить цифровую нить. Если оно не различает состояния, оно не сможет определить управленческий переход. Если не различает событие и документ, оно не сможет отделить факт от его носителя. Если не различает источник и интерпретацию, оно не сможет поддерживать доверие к графу знаний.

Предметно-ориентированное мышление также связывает цифровой двойник организации с process mining. Фактический поток событий сам по себе ещё не объясняет деятельность. Событие нужно отнести к предмету, состоянию, роли, процессу, носителю и последствию. Без этого event log остаётся технической записью. С предметной моделью он становится материалом для анализа фактического прохождения деятельности.

То же относится к корпоративному искусственному интеллекту. AI-помощник может работать с документами и текстами, но для управленческого применения ему требуется структурированный контекст: какие предметы существуют, как они связаны, какие состояния допустимы, какой источник подтверждает факт, где находится разрыв. Граф знаний и предметная модель позволяют заменить свободное угадывание смысла проверяемым контуром.

Морфогенетическое развёртывание предприятия также невозможно без предметного слоя. Исходное ядро должно содержать элементы, которые способны разворачиваться в более сложную структуру. Такими элементами являются не абстрактные фразы, а нормализованные бизнес-предметы, состояния, события, носители, источники и правила связности. Именно они позволяют новым фактам входить в модель и поддерживать её упорядоченность.

Термин «самоорганизующееся интеллектуальное предприятие» фиксирует результат этого перехода. Это предприятие, в котором интеллектуальность не ограничивается отчётами, чат-ботом или аналитической панелью. Оно имеет предметное ядро, цифровую нить, граф знаний и механизм включения новых событий в модель деятельности. Такая структура позволяет не только описывать предприятие, но и поддерживать его способность к управляемому обновлению.

Обычная цифровизация не решает эту задачу автоматически. Она может дать ERP, электронный документооборот, BI, регламенты, процессные схемы, интеграции, роботизацию и корпоративный поиск. Но если эти элементы не связаны через предметное ядро, они остаются набором сильных, но разрозненных инструментов. Самоорганизующееся интеллектуальное предприятие требует их соединения в единую предметно-событийную структуру.

Именно поэтому второй том начинается с мирового контекста. Задача не в том, чтобы добавить внешний обзор ради расширения книги. Задача в том, чтобы показать место предметно-ориентированного мышления в мировом сдвиге корпоративных систем. Оно является связующим слоем между управленческой целью интеллектуального предприятия и профессиональным методом построения таксоно-онтологической модели.

После этой части можно перейти к основной проблеме второго тома: почему искусственный интеллект предприятия не может устойчиво рассуждать без предметного мира. Галлюцинации корпоративного ИИ в этом контексте рассматриваются не как исходная тема, а как один из симптомов более общего разрыва: предприятие использует интеллектуальные технологии, не имея достаточно строгой предметной структуры собственной деятельности.

Библиография к Части I

- [1] Gartner. Market Guide for Digital Twin of an Organization Platforms. Published 20 November 2024. URL: <https://www.gartner.com/en/documents/5936107>
- [2] Digital Twin Consortium. The Definition of Digital Thread. URL: <https://www.digitaltwinconsortium.org/initiatives/the-definition-of-digital-thread/>
- [3] W3C. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, 25 February 2014. URL: <https://www.w3.org/TR/rdf11-concepts/>
- [4] W3C. OWL — Semantic Web Standards. URL: <https://www.w3.org/OWL/>
- [5] IEEE Task Force on Process Mining. Process Mining Manifesto. 2012. URL: <https://www.tf-pm.org/resources/manifesto>
- [6] Celonis. What is object-centric process mining? URL: <https://www.celonis.com/blog/what-is-object-centric-process-mining-ocpm>
- [7] SAP. SAP Knowledge Graph — AI with Business Context. URL: <https://www.sap.com/products/artificial-intelligence/knowledge-graph.html>
- [8] Palantir. Ontology. URL: <https://www.palantir.com/platforms/ontology/>
- [9] Doursat R., Sayama H., Michel O. Morphogenetic Engineering: Reconciling Self-Organization and Architecture. In: Morphogenetic Engineering: Toward Programmable Complex Systems. Springer, 2012.
- [10] Jia H. Introducing Entropy into Organizational Psychology. Behavioral Sciences, 2024, 14 (1), 54.
- [11] Braun D., Müller T., Sahlab N., Jazdi N., Schloegl W., Weyrich M. A graph-based knowledge representation and pattern mining supporting the Digital Twin creation of existing manufacturing systems. 2022.
- [12] Su C., Tang X., Jiang Q., Han Y., Wang T., Jiang D. Digital twin system for manufacturing processes based on a multi-layer knowledge graph model. 2025.
- [13] Galanti R., de Leoni M., Navarin N., Marazzi A. Object-centric Process Predictive Analytics. 2022.
- [14] OECD. SMEs and entrepreneurship. URL: <https://www.oecd.org/en/topics/policy-issues/smes-and-entrepreneurship.html>
- [15] World Bank. Small and Medium Enterprises (SMEs) Finance. URL: <https://www.worldbank.org/ext/en/topic/competitiveness/small-and-medium-enterprises-smes-finance>
- [16] Celonis. Process Intelligence, AI, and Process Mining. URL: <https://www.celonis.com/>
- [17] Doursat R. A review of morphogenetic engineering. Natural Computing, 2013.
- [18] Rodriguez M. A., Watkins J. H., Bollen J., Gershenson C. Using RDF to Model the Structure and Process of Systems. 2007.

Рассмотренный мировой контекст показывает, что интеллектуализация предприятия требует не только технологий, но и предметной структуры деятельности. Далее рассматривается внутренняя причина этой необходимости: корпоративный искусственный интеллект, цифровой двойник и граф знаний не могут устойчиво работать с предприятием, если оно не различает собственные бизнес-предметы, состояния, события, носители и источники. Поэтому следующий раздел переходит от мирового обзора к предметному основанию рассуждения предприятия о самом себе.

Часть II «Почему ИИ предприятия не может рассуждать без предметного мира и иницирующего ядра»

ИИ предприятия не должен сам угадывать, чем предприятие управляет. Если большая языковая модель получает только регламенты, загрузки, письма, протоколы, схемы процессов и отчёты, она вынуждена восстанавливать предметный мир по косвенным признакам. Иногда это даёт полезные догадки, но в управленческом контуре такая догадка слишком опасна.

Предметный мир должен быть задан предприятием. Не в виде одной таблицы и не в виде красивой схемы, а как проверяемая система различий: бизнес-предметы, состояния, события, носители, источники, роли, связи и ограничения. В дальнейшем такая система различий станет материалом иницирующего ядра цифрового двойника деятельности.

Поэтому в этой части важно увидеть: проблема галлюцинаций ИИ не сводится к техническому сбою модели. Она показывает более глубокую проблему предприятия — отсутствие предметной опоры, на которую можно было бы посадить рассуждение.

Глава 9. ИИ пришёл в предприятие и обнаружил пустоту предметного мира

ИИ вошёл в предприятие не как абстрактная технология будущего, а как вполне практический инструмент сегодняшнего дня. Сначала он появился в виде больших языковых моделей, затем — в виде корпоративных ассистентов, чат-ботов, помощников руководителя, интеллектуальных поисковиков по базе знаний, генераторов писем, протоколов, регламентов, аналитических справок и проектных документов. Очень быстро стало понятно, что предприятие хочет от ИИ не только красивого текста. Оно хочет от него ответов на управленческие вопросы. [8]

Предприятие спрашивает: почему сорван срок поставки? Что будет, если задержится закупка критичного материала? Как изменится себестоимость, если партия пойдёт другим маршрутом? Почему у нас есть остаток на складе, но производство всё равно сообщает о дефиците? Какой документ является основанием для следующего действия? Кто должен принять решение? Какие последствия возникнут в учёте, в производстве, в качестве, в деньгах, в обязательствах перед клиентом?

На первый взгляд кажется, что ИИ как раз для этого и нужен. Он умеет читать тексты, сопоставлять фрагменты, строить объяснения, отвечать на вопросы, формулировать выводы. Он может взять регламент, протокол совещания, выгрузку из ERP, письмо от поставщика, кусок инструкции и собрать из них связный ответ. Именно эта способность сначала производит сильное впечатление. Возникает ощущение, что предприятие наконец получило универсальный интерфейс к собственной сложности.

Но почти сразу приходит первое разочарование. ИИ отвечает уверенно, но не всегда правильно. Он может убедительно объяснить несуществующую связь между документами. Может назвать процесс, которого на предприятии в таком виде нет. Может принять технический объект системы за управленческий предмет. Может смешать плановую задолженность с фактической, заказ с договором, статус документа с состоянием бизнес-предмета, регламентную обязанность с реальной процедурой исполнения. Ответ выглядит стройно, но в нём нет опоры на предметную правду предприятия.

Эта проблема особенно опасна потому, что ошибка ИИ редко выглядит как грубая ошибка. Чаще она выглядит как гладкая управленческая формулировка. Текст написан уверенно, логика внешне выдержана, термины похожи на профессиональные, выводы кажутся разумными. Но при проверке выясняется, что модель не знала, какой именно предмет движется по процессу, в каком состоянии он находится, кто его владелец, какой носитель фиксирует изменение, какой документ только сопровождает действие, а какой действительно является основанием следующего шага.

Обычно это разочарование объясняют технически. Говорят: модель недостаточно обучена; промпт был плохо написан; данных было мало; данные были грязными; нужно подключить модель к базе; нужно дать ей больше контекста; нужно использовать более новую версию; нужно применить RAG, векторный поиск, агентный контур, классификатор документов, корпоративную базу знаний. Во всех этих объяснениях есть часть правды. Но они не исчерпывают проблему. [9]

Технический слой действительно важен. Плохой промпт ухудшает ответ. Неполные данные ограничивают рассуждение. Некачественный поиск по документам приводит к неверным фрагментам. Прямое подключение к ERP может дать больше фактов. Но даже если всё это улучшить, остаётся более глубокий вопрос: что именно ИИ должен считать предметом рассуждения?

Предприятие может дать ИИ тысячи документов и миллионы записей, но не дать ему предметного мира. Оно может открыть доступ к регламентам, протоколам, выгрузкам, отчётам, договорам, заявкам, заказам, актам и справочникам, но не объяснить, какие смысловые сущности за ними стоят. Тогда модель видит слова, записи, поля и документы, но не видит того, чем предприятие реально управляет.

Управленческая причина галлюцинаций в предприятии часто состоит не в том, что ИИ плохо думает, а в том, что предприятие не дало ему предметов для мышления. Оно дало тексты, формы, таблицы, отчёты, названия процессов, документы системы менеджмента качества, показатели и регламенты. Но оно не задало явную карту бизнес-предметов: что возникает, что изменяется, что передаётся, что фиксируется, что контролируется, что измеряется и что становится основанием для управленческого решения. [10]

У предприятия, как правило, много видимых слоёв. Есть регламенты, положения, инструкции, маршрутные карты, описания бизнес-процессов, организационная структура, роли, зоны ответственности. Есть ERP-система, где ведутся заказы, документы, справочники, регистры, отчёты, настройки, статусы, взаиморасчёты, остатки, движения и аналитики. Есть KPI, бюджетные формы, отчёты руководителя, управленческие панели, протоколы совещаний, корректирующие мероприятия. Есть СМК — система менеджмента качества, в которой хранятся процедуры, записи, формы, доказательства выполнения требований и результаты проверок.

На уровне управленческой поверхности предприятие выглядит насыщенным данными. Кажется, что здесь уже всё описано. Есть процесс закупки, процесс производства, процесс продажи, процесс сервисного обслуживания, процесс управления качеством. Есть документы, которыми эти процессы оформляются. Есть отчёты, которые показывают результат. Есть показатели, которые измеряют эффективность. Есть сотрудники, которые выполняют работу. Есть система, которая хранит записи.

Но именно здесь и обнаруживается пустота. Между процессом и документом, между документом и отчётом, между отчётом и решением часто не задан главный слой — слой бизнес-предметов. Нет явного ответа на вопрос: чем именно управляет предприятие в данном месте? Не просто какой документ создаётся, не просто какой пункт регламента выполняется, не просто какая кнопка нажимается, а какой предмет меняет состояние и почему это изменение имеет управленческий смысл.

Бизнес-предмет — это смысловая управляемая сущность предприятия. В этой книге это одно из базовых понятий. Бизнес-предметом может быть запас, потребность, заказ, обязательство, партия, отклонение, рекламация, производственная программа, сервисная заявка, доказательная запись, лимит, резерв, дефицит, маршрут исполнения, состояние оригинала первичного документа. Важно не само слово, а то, что за ним стоит управляемая сущность: она возникает, живёт во времени, меняет состояние, имеет владельца, носитель, связи, последствия и может стать основанием действия.

Чего часто нет у предприятия в явном виде? Нет реестра таких предметов. Нет описания их состояний. Нет переходов между состояниями. Нет различения, где предмет находится в ERP, где — в бумажной форме, где — в Excel, где — во внешнем кабинете, где — только в голове ответственного сотрудника. Нет связи между предметом и процессом. Нет связи между предметом и процедурой. Нет связи между предметом и ролью. Нет связи между предметом и KPI. Нет связи между предметом и нормативным требованием. Нет понимания, какие носители допустимы, какие являются временными, а какие должны стать управляемыми.

Документы не заменяют предметный мир. Документ может быть важным носителем, но он не равен предмету. Заказ клиента, акт, заявка, накладная, протокол, отчёт, форма СМК — всё это может фиксировать действие, решение, состояние или результат. Но документ не объясняет сам по себе, какой предмет через него проходит, в каком состоянии он был до доку-

мента, в какое состояние перешёл после него, кто отвечает за переход, что запускается дальше и какие последствия возникнут.

Предприятие часто исходит из ошибочного представления, что если есть документ, то есть и управление. Но документ может быть только следом. Он может появиться поздно, заполняться формально, служить доказательством для внешней проверки, но не управлять самим предметом. Бывает и наоборот: реальное управление давно произошло в переписке, в устной договорённости, в Excel-файле или в рабочем чате, а официальный документ появился позже как регистрация уже принятого решения. Для ИИ такая ситуация особенно опасна. Если он видит только документ, он может принять след за причину, форму за предмет, регистрацию за управление.

ERP также не заменяет предметный мир. В первом упоминании зафиксируем термин: ERP — Enterprise Resource Planning, класс корпоративных систем управления ресурсами предприятия. В этой книге ERP-система рассматривается как важный прикладной и учётный контур: она фиксирует значимые факты, документы, движения, статусы, расчёты, аналитику, права, настройки, маршруты и отчёты. Без такого прикладного и учётного ядра зрелое управление предприятием невозможно. [1]

Но ERP-объект не равен бизнес-предмету. Документ, справочник, регистр, отчёт, настройка, статус или аналитика могут быть носителями состояния предмета, но сами по себе они ещё не раскрывают предметную модель. Один бизнес-предмет может жить сразу в нескольких ERP-объектах. Один ERP-объект может участвовать в жизни нескольких предметов. Часть предмета может быть в ERP, часть — в регламенте, часть — в СМК, часть — во внешней системе, часть — в фактической практике предприятия. Поэтому прямой доступ LLM к базе ERP может помочь ответить на вопрос «что записано?», но не гарантирует ответа на вопрос «что это значит?».

Процессы тоже не заменяют предметный мир. Процессный подход необходим. Он дисциплинирует описание деятельности, показывает последовательность действий, роли, входы, выходы, контрольные точки и передачу результата. Без процессов предприятие легко распадается на набор отдельных операций и документов. Процессный язык — обязательный шаг к зрелому описанию деятельности.

Но процесс без предмета остаётся процедурной оболочкой. Он говорит, что кто-то что-то делает после чего-то и перед чем-то. Однако он не всегда отвечает, что именно меняет состояние. Если не задан предмет процесса, то процесс превращается в маршрут по действиям, а не в маршрут изменения управляемой сущности. Тогда ИИ может читать процесс, но не понимать его предметной логики. Он видит последовательность шагов, но не видит, что именно проходит через эти шаги, почему переход важен, где возникло отклонение и какое решение должно быть принято.

Именно здесь появляется вторая причина этой книги. Первая причина — ИИ и его галлюцинации в управленческом контуре. Вторая причина — осмысление работы над учебной концепцией по ERP-системам под методическим руководством Галины Ледовской. Эта работа была важным этапом: она помогала вводить человека в профессию, показывала логику ERP-системы, её прикладные контуры, управленческие возможности и методические основания.

Но у такой концепции была своя честная граница. Она не должна была становиться тяжёлой онтологической книгой. Её нельзя было перегружать графами знаний, LLM, цифровыми двойниками, доменными ядрами памяти, таксономиями бизнес-предметов и подробной предметной философией. Для входа в профессию это было бы преждевременно. Читателю нужно было сначала понять систему, её назначение, основные контуры, управленческую и прикладную логику. В этом смысле глубина была отсечена не случайно и не по слабости, а сознательно.

За кадром той концепции остался слой, который тогда ещё не был необходим в явном виде для массового входа в ERP-системы, но теперь становится необходимым. Это слой биз-

нес-предметов, предметной онтологии, графа знаний, LLM как интерфейса к графу, доменных ядер памяти и цифрового двойника деятельности предприятия. Раньше этот слой можно было держать как внутреннюю методологическую интуицию опытного архитектора. Теперь его нужно вынести наружу и описать. [7]

Причина проста: ИИ не умеет устойчиво рассуждать о предприятии, если предприятие не умеет различать собственные предметы. Пока опытный консультант, архитектор или руководитель держит предметные связи в голове, он может компенсировать неполноту документов, процессов и ERP-объектов. Он понимает, что за заявкой стоит потребность, за заказом — обязательство, за складским остатком — доступность ресурса, за статусом документа — степень готовности управленческого действия. Но LLM не обладает этой внутренней профессиональной памятью предприятия. Ей нужно дать её в явном виде.

Поэтому предметный слой стал необходим не как формальное дополнение, а как практическое условие перехода к интеллектуальному предприятию. Если предприятие хочет, чтобы ИИ не просто писал тексты, а помогал рассуждать о последствиях, отклонениях, рисках, маршрутах и решениях, оно должно дать ему не только документы и данные, но и предметную модель. Иначе модель будет достраивать связи сама. А там, где ИИ достраивает управленческий мир сам, начинается галлюцинация. [4]

Эта книга находится именно на пересечении двух линий. Первая линия — проблема ИИ-галлюцинаций в управлении предприятием. Вторая линия — осмысление предметного слоя ERP-систем после работы над учебной концепцией ERP-систем для обучения и профессионального входа. В точке их пересечения становится ясно: следующий шаг нельзя делать только через улучшение промптов, подключение баз и накопление документов. Нужна предметная опора.

Теперь эту мысль нужно связать с понятием иницирующего ядра. Ядро не является промптом, набором инструкций для модели или ещё одной базой документов. Ядро — это проверяемая смысловая структура, в которой уже различены предметы предприятия, их состояния, носители, источники и связи. Именно поэтому ИИ должен опираться не на поток несвязанных материалов, а на ядро как на организованную память деятельности.

Если такого ядра нет, модель вынуждена сама собирать предметный мир из фрагментов. Она видит слова, но не всегда видит управляемые сущности. Она видит документ, но не всегда отличает его от предмета. Она видит статус, но не всегда понимает, какое состояние предмета он подтверждает. Она видит процесс, но не всегда понимает, какой предмет проходит через этот процесс.

Предметно-ориентированное мышление нужно именно для того, чтобы эта пустота перестала быть пустотой. Оно готовит материал будущего ядра: не абстрактные термины, а различимые элементы деятельности, пригодные для дальнейшей проверки, сериализации и эксплуатации.

Финальная формула главы такова: бизнес-предмет — носитель смысла и результата; процесс — маршрут изменения состояния бизнес-предмета; ERP — система фиксации, контроля и анализа этих изменений; искусственный интеллект предприятия — способность рассуждать о предметах, их состояниях, связях и последствиях. LLM не является источником истины. LLM является интерфейсом к графу знаний. ERP-данные отвечают на вопрос «что записано?». Граф знаний отвечает на вопрос «что это значит, как предмет движется, почему он изменил состояние и что делать дальше?».

С этого и начинается книга. Не с отрицания документов, ERP или процессов. Они необходимы. Но без предметного мира они не дают ИИ устойчивой основы для рассуждения. Предприятие, которое хочет стать интеллектуальным, должно сначала научиться различать, называть и связывать то, чем оно управляет. Оно должно увидеть собственный предметный мир. И

только после этого ИИ перестаёт быть генератором убедительных текстов и становится интерфейсом к управляемому смыслу предприятия.

Глава 10. Галлюцинация ИИ как управленческий симптом

О галлюцинациях больших языковых моделей чаще всего говорят как о технической проблеме. Модель придумала факт. Модель неверно сослалась на источник. Модель связала два утверждения, которые в реальности не связаны. Модель уверенно ответила там, где должна была сказать: «данных недостаточно». Это привычное понимание галлюцинации LLM, и оно верно на своём уровне.

Технически галлюцинация выглядит как сбой вероятностного продолжения текста. Модель строит ответ, опираясь на языковые закономерности, контекст запроса, доступные фрагменты, статистические связи и внутренние представления. Если опоры недостаточно, она может достроить недостающий смысл. Для обычного пользователя это проявляется просто: ИИ отвечает складно, но ошибается.

В корпоративной среде эту проблему пытаются лечить техническими средствами. Улучшают промпт. Подключают поиск по документам. Добавляют векторную базу. Настраивают RAG. Ограничивают модель инструкциями. Подключают её к ERP, базе знаний, файловому архиву, регламентам, корпоративному portalу, протоколам совещаний. Всё это нужно. Но всё это не закрывает главный управленческий разрыв.

Проблема предприятия не сводится к тому, что ИИ иногда ошибается как языковая модель. Проблема глубже: предприятие часто само не имеет явно заданного предметного мира, о котором ИИ должен рассуждать. Модель получает тексты, документы, отчёты, выгрузки, фрагменты регламентов, но не получает управляемой предметной системы. Она видит следы деятельности, но не всегда видит сами предметы деятельности.

Предприятие — не набор текстов. Предприятие — предметная система. В нём существуют заказы, запасы, партии, потребности, обязательства, лимиты, маршруты, отклонения, рекламации, платежи, производственные задания, первичные документы, состояния качества, статусы готовности, управленческие решения. Эти сущности живут во времени, меняют состояние, проходят через роли, документы, ERP-объекты, СМК-записи, отчёты и управленческие решения.

Если эти предметы не заданы явно, ИИ вынужден восстанавливать их сам. Он видит в одном документе слово «заказ», в другом — «резерв», в третьем — «поступление», в четвёртом — «потребность», в пятом — «отклонение срока». Но если между ними не построена предметная связь, модель начинает достраивать её по вероятностной логике. Она может решить, что заказ клиента автоматически запускает закупку. Может предположить, что резерв означает фактическое наличие товара. Может принять плановую дату за обязательство. Может перепутать статус документа с состоянием управляемого предмета.

Именно в этом месте техническая галлюцинация превращается в управленческую. ИИ не просто ошибся в словах. Он построил ложную картину предприятия. Опасность здесь не в отдельной неточности, а в том, что ответ может выглядеть как управленчески зрелый. В нём будут роли, действия, документы, статусы, выводы, рекомендации. Но за этим не будет предметной модели.

В терминах этой серии галлюцинация часто возникает там, где нет иницилирующего ядра или где оно ещё не введено в информационные потоки предприятия. У предприятия могут быть документы, регламенты, базы, отчёты и даже корпоративная база знаний. Но если они не связаны через предметы, состояния, события и источники, они не образуют ядро. ИИ получает не собранную деятельность, а набор материалов, из которых он начинает достраивать картину сам.

Поэтому борьба с галлюцинациями не должна сводиться только к улучшению промптов и поисковых механизмов. Нужен более глубокий слой — предметная нормализация реально-

сти предприятия. Сначала предприятие должно сказать, какие предметы оно различает и как эти предметы живут в процессах, документах, ERP, CMK, KPI и данных. Только после этого языковая модель может становиться интерфейсом к управляемой памяти, а не генератором правдоподобных версий.

Убедительность ответа становится отдельным риском. Слабый ответ легко отвергнуть. Нелепая ошибка быстро бросается в глаза. Но хороший язык, правильная интонация, знакомые термины и уверенная структура создают видимость компетентности. Руководитель, аналитик или консультант может прочитать такой ответ и увидеть не ошибку, а готовую схему. Если при этом у предприятия нет собственной предметной проверки, галлюцинация проходит внутри управленческого контура.

Рассмотрим простой пример с процессом закупки. На предприятии есть потребность в материале. Она может возникнуть из производственного заказа, нормы запаса, заявки подразделения, прогноза продаж, аварийной потребности, сервисного обращения или решения руководителя. В ERP эта потребность может быть отражена разными объектами: заказом на обеспечение, заказом поставщику, заявкой на закупку, резервом, плановой потребностью, строкой отчёта, записью регистра. В реальности за всем этим стоит предмет: потребность в обеспечении.

Если предмет «потребность» не описан, ИИ может начать рассуждать только по документам. Он увидит заказ поставщику и решит, что потребность уже обеспечена. Он увидит остаток на складе и решит, что закупка не нужна. Он увидит заявку и решит, что она автоматически стала обязательством поставщика. Но управленчески это разные состояния. Потребность может быть выявлена, подтверждена, согласована, включена в план обеспечения, закрыта заказом поставщику, частично обеспечена, просрочена, заменена аналогом или снята. Без такой статусной карты ИИ будет говорить о закупке, но не будет понимать предмет закупочного управления.

Второй пример — ERP-объект. Пользователь спрашивает: «Можно ли по этому заказу запускать производство?» Если модель видит только документ «Заказ клиента», она может ответить формально: да, заказ есть, значит производство можно запускать. Но для предприятия этого недостаточно. Нужно понять состояние предмета: заказ согласован или нет, условия продажи подтверждены или нет, оплата требуется или нет, обеспечение материалов возможно или нет, есть ли ограничения по качеству, есть ли спецификация, есть ли маршрут, есть ли производственная мощность, есть ли запрет на изменение условий.

ERP-объект является важным носителем, но не равен бизнес-предмету. Документ «Заказ клиента» может быть одним из носителей предмета «клиентское обязательство» или «портфель отгрузки». Но сам предмет шире документа. Он включает условия исполнения, связь с обеспечением, связь с производством, финансовые ограничения, статус согласования, управленческий приоритет, риски и последствия. Если ИИ не различает документ и предмет, он будет рассуждать слишком прямо.

Третий пример — CMK-документ. Предприятие может загрузить в базу знаний процедуру контроля качества, форму протокола, инструкцию, регламент, чек-лист, акт несоответствия. ИИ прочитает эти документы и сможет пересказать их содержание. Но управленческий вопрос звучит иначе: какой предмет качества здесь управляется? Партия? Протокол контроля? Несоответствие? Разрешение на выпуск? Рекламация? Корректирующее действие? Доказательная запись?

Если предмет качества не задан, ИИ может смешать форму и событие. Он может считать, что наличие заполненного протокола означает пригодность продукции. Но протокол сам по себе может фиксировать как положительное решение, так и отклонение. Он может быть черновиком, может быть не согласован, может быть связан не с той партией, может не закрывать

нормативное требование. СМК-документ без предметной связи не даёт полной управленческой правды.

Отсюда следует главный вывод главы: промпт не заменяет предметную модель. Хороший промпт может заставить модель быть осторожнее. Он может потребовать ссылаться на источники, задавать уточняющие вопросы, не придумывать факты. Но промпт не создаёт предметы предприятия. Он не формирует состояния, связи, владельцев, носители, правила перехода и последствия. Он только управляет поведением модели в конкретном ответе.

Если предприятие хочет использовать ИИ в управлении, ему нужно дать ИИ не только текстовый контекст, но и граф знаний. Не в смысле красивой технологической моды, а в практическом смысле: нужно явно связать бизнес-предметы, их состояния, переходы, процессы, ERP-объекты, СМК-записи, документы, роли, показатели, нормативные основания и управленческие последствия.

Граф знаний не заменяет ERP. ERP фиксирует факты, документы, движения, остатки, статусы, расчёты и отчёты. Граф знаний отвечает на другой вопрос: что эти факты значат в предметной модели предприятия? Какой предмет изменил состояние? Что было причиной? Что стало следствием? Какой процесс должен продолжиться? Какая роль должна принять решение? Какой риск возник? Какой отчёт покажет последствие?

В этом смысле управленческая галлюцинация ИИ является зеркалом предприятия. Там, где предприятие не различило свои предметы, ИИ тоже их не различит. Там, где предприятие держит связи только в голове опытных сотрудников, модель будет пытаться угадать эти связи. Там, где документы не привязаны к предметам, ИИ будет принимать документы за предметы. Там, где процессы не имеют предметной опоры, ИИ будет пересказывать процедуры, но не понимать, что именно изменяется.

Поэтому борьба с галлюцинациями в enterprise-контуре начинается не с новой версии модели и не с более длинного промпта. Она начинается с вопроса: какие предметы предприятия должны быть заданы, чтобы о них можно было рассуждать? Какие из них уже проявлены в ERP? Какие живут в документах? Какие существуют только в практике? Какие заданы нормативно, но не встроены в управление? Какие имеют состояния, а какие пока описаны только словами?

Эта постановка выводит нас к более глубокой развилке. Предмет для ИИ и для управленческого мышления должен быть не только назван. Он должен быть задан и дан. Задан — через типовую модель, методологию, нормативный слой, классификаторы, best practice. Дан — через фактическую деятельность конкретного предприятия, его исключения, боли, маршруты, роли, документы, ограничения и способы работы.

Именно здесь возникает необходимость философской опоры. Не ради отступления от темы, а ради точного различения. Нельзя рассуждать о предприятии, если предметы не заданы понятиями. Но нельзя построить живую модель предприятия, если предметы не даны самой практикой. Эту развилку удобно объяснить через Канта. К ней и переходит следующая глава.

[3]

Глава 11. Кантовская развилка: задать предмет и дать предмет

На первый взгляд появление Канта в книге об ERP-системах, бизнес-предметах и интеллектуальном предприятии может показаться лишним. Кажется, что здесь достаточно практических слов: процессы, документы, справочники, регистры, показатели, граф знаний, LLM, доменная память. Зачем в эту прикладную область вводить философскую развилку?

Причина проста. Вопрос этой книги не только технический. Он касается самого способа управленческого мышления. Предприятие хочет, чтобы ИИ рассуждал о его деятельности. Но рассуждать можно только о чём-то. Нужно понять, откуда берётся предмет рассуждения. Мы его заранее задаём? Или предприятие само даёт нам его через свою деятельность? Можно ли взять типовую модель и считать, что предмет уже определён? Можно ли только поговорить с пользователями и считать, что предмет уже понят? Где проходит граница между методологией и реальностью предприятия?

Здесь и нужна кантовская опора. Не как академическое отступление, а как удобный инструмент различения. Известная формула Канта звучит примерно так: мысли без содержания пусты, созерцания без понятий слепы. Для этой книги важны обе половины. Если у нас есть только понятия, но нет материала предприятия, модель становится пустой. Если у нас есть только факты, интервью, документы и наблюдения, но нет понятий, модель становится слепой.

Переведём это на язык управления. Типовая модель, методология, нормативный слой и классификаторы дают предприятию понятия. Они помогают заранее различить заказ, запас, потребность, партию, обязательство, производственный заказ, НЗП, протокол качества, рекламацию, затраты, платеж, финансовый результат. Без этих понятий аналитик вынужден каждый раз заново угадывать, с чем он имеет дело.

Но одних понятий недостаточно. Конкретное предприятие даёт содержание. Оно показывает, как именно у него возникает потребность, кто фактически принимает решение, где происходит задержка, какие документы создаются формально, какие используются реально, какие исключения постоянно повторяются, какие предметы не названы, но управляются каждый день. Без этого содержания типовая модель остаётся правильной, но не живой.

Так возникает главная развилка главы: предмет должен быть задан и дан.

Задать предмет — значит ввести его в модель как управляемую сущность. Это делает методология. Это делает типовая логика ERP. Это делает нормативный слой. Это делают профессиональные классификаторы, справочники, проектные стандарты, типовые процессы, лучшие практики. Когда мы говорим «партия продукции», «потребность в материалах», «заказ клиента», «лимит бюджета», «несоответствие качества», мы не просто повторяем слова из предприятия. Мы задаём различения, без которых деятельность нельзя анализировать.

Заданный предмет обладает именем, границами, ожидаемыми состояниями, допустимыми переходами, ролями, носителями и связями. Например, предмет «потребность в материалах» можно задать как управляемую сущность, которая возникает из производственного плана, заказа, нормы запаса или заявки, затем проверяется, обеспечивается, резервируется, закупается, снимается или переносится. Даже если конкретное предприятие пока не называет это словом «потребность», сама предметная рамка уже позволяет увидеть соответствующую реальность.

Другой пример — «несоответствие качества». Его можно задать как предмет СМК: событие, в котором результат, материал, продукция, процесс или документ не соответствует требованию. У такого предмета есть источник выявления, описание, классификация, решение, ответственный, корректирующее действие, доказательная запись, статус закрытия. Если это

задано, предприятие уже не может свести несоответствие к устному замечанию или случайной записи в журнале.

Третий пример — «производственный заказ». Его можно задать как предметно-прикладную сущность, связывающую потребность выпуска, производственную программу, ресурсную спецификацию, этапы производства, материалы, мощность, НЗП, выпуск и себестоимость. В ERP этот предмет может иметь конкретный документный носитель. Но предмет шире формы документа: он задаёт управленческую логику исполнения.

Задать предмет обычно быстрее и дешевле, чем извлечь его из предприятия с нуля. Именно поэтому типовые модели, методики и best practice так ценны. Они не заставляют проект начинать с пустого листа. Они дают профессиональный язык. Они позволяют сразу спросить: где у вас возникает потребность? Как вы управляете резервом? Как фиксируете несоответствие? Кто принимает решение по партии? Где замыкается обязательство перед клиентом?

Но предмет нужно не только задать. Его нужно дать.

Для будущего иницирующего ядра это различие принципиально. Задать предмет — значит ввести его как смысловую единицу: назвать, ограничить, отличить от других предметов, определить его состояния и признаки. Дать предмет — значит связать его с носителями, фактами, документами, ERP-объектами, СМК-записями, источниками и событиями, через которые он становится проверяемым.

Ядро не может быть построено только на заданных понятиях и не может быть построено только на данных. Если есть только понятия, оно остаётся словарём. Если есть только данные, оно остаётся складом записей. Иницирующее ядро возникает тогда, когда заданные предметы получают проверяемые носители и начинают связывать деятельность предприятия.

Дать предмет — значит обнаружить, как он реально существует в конкретном предприятии. Это происходит через интервью, наблюдение, анализ документов, выгрузок, переписок, исключений, ручных таблиц, старых баз, неформальных маршрутов, управленческих конфликтов и повторяющихся ошибок. Предприятие даёт предмет не всегда в чистом виде. Часто оно даёт его через симптомы.

Например, предприятие может не говорить: «У нас плохо управляется предмет „дефицитный остаток“». Оно говорит иначе: «На складе что-то есть, но производство всё равно не может стартовать». Или: «Закупка говорит, что заказано, производство говорит, что не обеспечено». Или: «В отчёте остаток есть, а в цеху говорят, что материала нет». За этими фразами нужно увидеть предмет: обеспеченность, дефицит, резерв, доступность, назначение, потребность, замена, срок поступления.

Другой пример — предприятие может не называть предмет «доказательная запись качества». Оно говорит: «У нас протоколы есть, но при проверке их трудно найти». Или: «ОТК проверяет, но в ERP это никак не видно». Или: «Продукция отгружена, а решение по отклонению лежит в письме». Здесь предмет даётся через разрыв между процедурой, документом, ERP-объектом и доказательством.

Третий пример — предприятие может не осознавать предмет «управленческое обязательство по сроку». Оно говорит: «Менеджер обещал клиенту, производство не знало, закупка не успела, потом всё срочно переносили». Формально есть заказ, письма, план, дата отгрузки. Но предметом является не отдельный документ, а обязательство, которое должно пройти через продажи, обеспечение, производство, склад, отгрузку и финансы.

Нормативный слой является третьим источником. Он не равен типовой модели и не равен фактической практике. Он задаёт требования, которые предприятие обязано учитывать, даже если внутри оно пока не умеет их правильно называть. Это могут быть требования бухгалтерского и налогового учёта, требования СМК, отраслевые стандарты, требования прослеживаемости, требования безопасности, требования к медицинским изделиям, ГОЗ, договорные обязательства, правила документооборота, требования к первичным документам.

Нормативный слой часто выявляет предметы, которые предприятие не выделяло само. Например, предприятие может считать протокол контроля просто формой. Нормативный подход заставляет увидеть в нём доказательную запись. Предприятие может считать партию просто количеством на складе. Нормативный слой качества и прослеживаемости заставляет увидеть в партии объект контроля, идентификации, допуска, отклонения и ответственности. Предприятие может считать договор просто юридическим файлом. Нормативный и управленческий слой показывают в нём источник обязательств, графиков, условий, рисков и признания фактов.

Поэтому нельзя брать только типовую модель. Типовая модель задаёт предметы, но не знает конкретной жизни предприятия. Она не знает, где сотрудники обходят систему, где документ создаётся поздно, где Excel сильнее ERP, где старый процесс сохраняется после внедрения, где руководитель принимает решение вне маршрута, где регламент написан, но не работает. Если ограничиться типовой моделью, можно получить красивую архитектуру без реальной силы. [12]

Нельзя брать только интервью. Интервью даёт живой материал, но оно всегда ограничено опытом говорящего. Пользователь рассказывает о своей зоне, своей боли, своей привычке и своём языке. Он может не видеть соседний контур. Он может путать документ и предмет. Он может считать нормой то, что является обходом системы. Он может не знать нормативного требования. Он может не различать управленческое состояние и технический статус. Если строить модель только из интервью, она станет слепком текущего хаоса.

Нужно движение навстречу. Методология идёт к предприятию со стороны заданных предметов: вот какие управляемые сущности должны быть различены. Предприятие идёт к методологии со стороны данных предметов: вот как эти сущности реально живут, ломаются, передаются, скрываются, искажаются, фиксируются или не фиксируются. Типовая модель задаёт вопрос. Предприятие даёт материал ответа. Нормативный слой проверяет обязательность. Граф знаний соединяет всё это в управляемую структуру.

Именно здесь возникает практическая роль графа знаний. Он не является просто схемой связей. Он становится местом, где заданное и данное соединяются. В графе можно показать, что предмет «партия продукции» задан типовой моделью производства и склада, дан фактической практикой предприятия через выпуск, перемещение, контроль качества и отгрузку, а нормативно связан с требованиями прослеживаемости, учёта и доказательных записей. Такой предмет уже пригоден для рассуждения.

Связь с ИИ становится прямой. LLM нельзя просто просить «разобраться в предприятии», если предприятие не подготовило предметную опору. Модель должна получать не хаотичный набор текстов, а подготовленный предметный слой: какие предметы существуют, как они определены, где они проявляются, какие состояния имеют, какими документами фиксируются, какие нормативные основания действуют, какие роли отвечают и какие последствия возникают.

Тогда ИИ перестаёт быть устройством свободного достраивания смысла. Он становится интерфейсом к предметной модели. Он может объяснять, сопоставлять, проверять гипотезы, строить сценарии, искать разрывы, формулировать вопросы к предприятию, помогать в стресс-тестировании. Но источником истины остаётся не сама LLM. Источником становится связанная предметная модель, где предмет задан методологически и дан фактически.

Эта развилка важна и для проектной работы. Аналитик не должен приходить на предприятие как переписчик интервью. Но он не должен приходить и как носитель готовой истины, который просто накладывает типовую схему. Его работа находится между двумя движениями. Он должен знать, какие предметы искать. И он должен уметь увидеть, как предприятие даёт эти предметы через свою практику.

После этой философской постановки нужно вернуться к земле. Потому что главная трудность не в том, чтобы красиво произнести слова «задать» и «дать». Главная трудность в том,

что на реальных проектах консультанты, пользователи и руководители часто видят не предметы, а кнопки, документы, формы и привычные действия. Поэтому следующая глава переходит от Канта к авторскому опыту: почему консультанты видят кнопки, но не видят предметы.

Глава 12. Авторский опыт: почему консультанты видят кнопки, но не видят предметы

После философской развилки нужно вернуться к практике. Кант помогает различить, что предмет должен быть задан и дан. Но на проекте внедрения ERP-системы эта развилка редко выглядит философски. Она проявляется проще и грубее: люди видят экранную форму, кнопку, документ, отчёт, настройку, но не видят бизнес-предмет, ради которого всё это существует.

Мой опыт обучения консультантов, аналитиков и пользователей ERP-систем показывает одну устойчивую проблему. Человек может довольно быстро научиться открывать разделы, создавать документы, заполнять поля, проводить операции, формировать отчёты, искать ошибки заполнения. Но это ещё не означает, что он понимает управление. Он может знать, где нажать, и не понимать, что именно изменилось в предприятии после нажатия.

Именно отсюда появляется феномен «волшебных кнопочек». Пользователь спрашивает: какую кнопку нажать, чтобы запустить производство? Какой документ сделать, чтобы списать материал? Где поставить флаг, чтобы заработал учёт серий? Какой отчёт открыть, чтобы увидеть остатки? Где включить адресный склад? Как провести реализацию? Как закрыть месяц?

Эти вопросы не плохие. Они естественные. ERP-система сложна, и без прикладной навигации работать невозможно. Проблема начинается тогда, когда кнопка воспринимается как объяснение. Как будто если мы нашли команду, документ или настройку, мы уже поняли управленческую сущность. На самом деле кнопка только запускает действие в системе. Она не объясняет, какой предмет управляется, в каком состоянии он был до действия, в какое состояние перешёл после него и какие последствия возникли.

Кнопка не объясняет управление. Документ не объясняет управление. Даже отчёт сам по себе не объясняет управление. Они показывают прикладной носитель, но не обязательно раскрывают предметную логику. Можно оформить документ перемещения, но не понять предмет «доступность запаса». Можно создать заказ на производство, но не понять предмет «партия запуска». Можно включить адресное хранение, но не понять, что меняется в управлении складским пространством, отбором, размещением, пересчётом и доступностью товара.

Процессный подход стал важным шагом вперёд по сравнению с кнопочным взглядом. Когда мы начинаем описывать процесс, мы уже не ограничиваемся отдельной операцией. Мы видим последовательность действий, роли, входы, выходы, контрольные точки, документы, переходы ответственности. Это намного зрелее, чем обучение по принципу «откройте раздел, нажмите кнопку, заполните поле».

Процессный подход позволяет увидеть, что действие не существует отдельно. Закупка связана с потребностью. Поступление связано с заказом поставщику. Производство связано с обеспечением материалами. Выпуск связан с себестоимостью. Отгрузка связана с заказом клиента, складом, взаиморасчётами и выручкой. Сервис связан с обращением клиента, запасными частями, работами, документами реализации, внутренним потреблением и возвратами.

Но и процесса недостаточно. Процесс может быть описан формально правильно, но всё равно остаться процедурной оболочкой. В нём будет указано, кто что делает, в какой последовательности и каким документом. Однако главный вопрос останется нераскрытым: какой предмет проходит через процесс? Что именно меняет состояние? Ради какого результата процесс выделен? Какой предмет является главным, какие предметы обеспечивающими, какие предметы переходят дальше?

Если процесс не имеет предметной опоры, он легко превращается в длинный текст о действиях. Такой текст можно согласовать, положить в регламент, использовать для обучения. Но при попытке построить граф знаний, доменное ядро памяти или ИИ-ассистента он начнёт

распадаться. Модель увидит действия, но не увидит управляемые сущности. Она не поймёт, что процесс — это маршрут изменения состояния бизнес-предмета.

Именно этот слой частично остался за кадром учебной концепции по ERP-системам, подготовленной под методическим руководством Галины Ледовской. Это не было ошибкой. Та концепция решала другую задачу. Она вводила в профессию, давала целостное понимание системы, её контуров, возможностей, прикладной логики и методической структуры. Её нельзя было перегружать онтологией, графами знаний, LLM и цифровыми двойниками.

Для учебного входа важно сначала освоить систему как систему. Нужно понять разделы, подсистемы, документы, процессы, настройки, управленческие задачи, связи оперативного и учётного контуров. Если сразу дать начинающему консультанту полную предметную онтологию, можно перегрузить его до того, как он научится уверенно стоять на прикладной земле.

Но сегодня ситуация изменилась. ИИ обострил то, что раньше можно было компенсировать опытом. Опытный консультант мог держать предметные связи в голове. Он мог понимать, что за документом стоит предмет, что за отчётом стоит состояние, что за настройкой стоит управленческое различие. Но LLM не держит эту профессиональную интуицию сама. Ей нужно дать явную предметную модель. Поэтому теперь предметный слой нужно раскрыть отдельно.

Покажем это на внедренческом примере с адресным складом. На кнопочном уровне вопрос звучит так: где включить адресное хранение и какие документы использовать? На процессном уровне вопрос уже лучше: как товар принимается, размещается, отбирается, перемещается, пересчитывается и отгружается? Но предметный уровень задаёт ещё более точный вопрос: какими предметами управляет адресный склад?

Адресный склад управляет не только документами. Он управляет запасом, местом хранения, ячейкой, доступностью товара, заданием на размещение, заданием на отбор, складской зоной, упаковкой, партией, серией, качеством, расхождением, пересчётом, фактическим наличием, резервом, ошибочной ячейкой, статусом выполнения складской операции. Если эти предметы не названы, адресный склад воспринимается как набор сложных экранов и правил. Если названы — он становится понятным управленческим контуром.

Например, «ячейка» — это не просто поле в системе. Это предмет складского управления. У неё есть вместимость, назначение, зона, состояние, доступность, периодичность пересчёта, связь с товарами, ограничения размещения, история операций. «Задание на отбор» — не просто документ. Это предмет исполнения складской операции: он возникает из потребности отгрузки или перемещения, назначается исполнителю, выполняется, может быть частично выполнен, может выявить расхождение. «Расхождение пересчёта» — не просто цифра. Это предмет контроля достоверности складского остатка.

Как только мы начинаем видеть предметы, оказывается, что они идут цепочками. Потребность в материале порождает обеспечение. Обеспечение связано с заказом поставщику. Заказ поставщику связан с поступлением. Поступление связано с складской приёмкой. Приёмка связана с размещением. Размещение связано с доступностью запаса. Доступность связана с резервом. Резерв связан с производством или отгрузкой. Производство связано с выпуском. Выпуск связан с партией. Партия связана с качеством. Качество связано с разрешением на использование или отгрузку. Отгрузка связана с выручкой, обязательством и финансовым результатом.

Предприятие управляет не отдельными кнопками и не отдельными процессами, а движением предметов через такие цепочки. Именно поэтому предметно-ориентированное мышление становится необходимым. Оно позволяет видеть не только отдельную операцию, но и то, как один предмет вызывает другой, как состояние одного предмета становится условием перехода другого, как ошибка в начале цепочки превращается в финансовый или качественный риск в конце.

Для ИИ это критично. Если модель видит только документы, она может пересказать документы. Если видит только процессы, она может описать последовательность действий. Если видит только ERP-объекты, она может назвать формы и статусы. Но чтобы рассуждать о предприятии, ей нужно видеть предметные цепочки: что изменилось, где изменилось, почему это важно, какой следующий предмет должен быть создан, какая роль должна действовать, какой отчёт покажет риск, какое решение требуется.

Задача этой книги — ввести читателя в такой способ мышления. Не отменить кнопки. Не отменить процессы. Не заменить ERP абстрактной онтологией. А показать, что прикладная система, процессный подход, СМК, учёт, KPI и ИИ становятся связными только тогда, когда под ними явно задан предметный слой.

Интонация здесь должна быть ответственной. Предметно-ориентированное мышление — не модная надстройка и не красивый термин. Это дисциплина. Она требует не путать документ и предмет, процесс и маршрут изменения состояния, ERP-объект и управляемую сущность, отчёт и смысл, промпт и модель. Она требует называть предметы, описывать состояния, фиксировать переходы, связывать носители, проверять источники, строить граф знаний.

После первых четырёх глав мы можем сделать главный предварительный вывод. ИИ пришёл в предприятие и обнаружил пустоту предметного мира. Галлюцинация ИИ оказалась не только технической ошибкой, но и управленческим симптомом. Предмет нужно задавать и давать. А практический опыт показывает: пока консультант видит только кнопки, документы и процессы, он ещё не видит предприятие как предметную систему.

Следующая часть должна разобрать это подробнее. Нужно показать, почему процессов, документов и ERP-объектов недостаточно. Не потому, что они плохи или устарели. Наоборот, они необходимы. Но без бизнес-предмета они не дают устойчивой основы для интеллектуального предприятия. С этого начинается часть III. Именно поэтому недостаточно научить консультанта видеть кнопки, документы и даже процессы. Нужно научить его видеть будущие элементы ядра. Когда консультант описывает потребность, заказ, партию, несоответствие, рекламацию, событие качества или обязательство, он должен понимать, что это не просто слова в интервью. Это потенциальные узлы цифрового двойника деятельности.

Такой взгляд меняет практику. Аналитик перестаёт спрашивать только «какой документ создаётся?» и начинает спрашивать: какой предмет изменил состояние, где это изменение зафиксировано, кто владелец записи, какой источник подтверждает факт, какие последствия возникли и какой следующий процесс должен получить этот предмет. С этого момента предметно-ориентированное мышление становится не теорией, а подготовкой к сборке ядра.

Часть III «Почему процессов, документов, ERP-объектов и BPM/BPA-моделей недостаточно»

К привычным опорам предприятия — процессам, документам, ERP-объектам, СМК и КРІ — нужно добавить ещё одну важную категорию: средства бизнес-моделирования и BPM/BPA-системы. Они полезны, потому что помогают фиксировать процессы, роли, документы, регламенты, организационную структуру, показатели и процессные связи. Но они также имеют границу.

Эта книга не критикует какой-либо конкретный продукт или конкретного разработчика. Речь идёт о классе инструментов. BPM/BPA-система может хорошо поддерживать процессное описание, но сама по себе не гарантирует наличие самостоятельного предметного слоя предприятия. Можно иметь аккуратный процессный репозиторий и всё равно не знать, какой бизнес-предмет является предметом управления, какие у него состояния, где его носители и как он связан с цифровой нитью.

Поэтому вопрос этой части шире: почему никакой отдельный слой — процесс, документ, ERP-объект, СМК-форма, КРІ или BPM/BPA-модель — не заменяет предметный мир. Все эти слои нужны, но они должны быть связаны в иницилирующем ядре.

Глава 13. Процесс без предмета — процедурная оболочка

Часть II закончилась на простой, но неприятной мысли: предприятие может иметь ERP, регламенты, процессы, отчёты, KPI, СМК и даже корпоративного ИИ, но при этом не иметь явно заданного предметного мира. В таком случае ИИ начинает достраивать связи сам, консультант видит кнопки, пользователь видит документы, руководитель видит отчёты, но устойчивого управленческого рассуждения не возникает.

Теперь нужно последовательно разобрать привычные опоры предприятия. Первая из них — процессный подход. Это важная опора. Без процессов предприятие распадается на отдельные действия. Нельзя управлять закупкой, производством, продажами, сервисом, качеством, платежами или закрытием месяца, если не понимать, кто что делает, в какой последовательности, на каком основании, с каким входом и с каким результатом.

Поэтому эту главу нельзя читать как критику процессного подхода в плохом смысле. Процессный подход не устарел. Он необходим. Он дисциплинирует описание деятельности, выводит предприятие из хаоса устных поручений, позволяет назначать роли, фиксировать границы ответственности, строить регламенты, готовить инструкции, проектировать автоматизацию и обучать сотрудников.

Процессный подход даёт несколько сильных вещей.

Во-первых, он показывает последовательность действий. Предприятие перестаёт смотреть на операцию как на одиночное событие. Поступление товара оказывается связано с заказом поставщику, заказ поставщику — с потребностью, потребность — с производственным планом или продажами, а производство — с обеспечением, складом, качеством, выпуском и себестоимостью.

Во-вторых, процессный подход показывает роли. Становится видно, где действует менеджер, где кладовщик, где плановик, где бухгалтер, где технолог, где ОТК, где руководитель, где экономист, где специалист по качеству. Это важно, потому что ERP-проект без ролевой структуры быстро превращается в набор абстрактных документов, которые «кто-то должен создать».

В-третьих, процессный подход показывает входы и выходы. У процесса появляется начало и конец. На входе может быть заявка, потребность, заказ, рекламация, план, несоответствие, договорное условие. На выходе — созданный заказ, обеспеченный запас, выпущенная продукция, закрытая рекламация, проведённый платеж, сформированный отчёт, принятое решение.

В-четвёртых, процессный подход позволяет увидеть точки контроля. Если процесс описан, можно спросить: где проверяется правильность данных? Где согласование? Где контроль качества? Где подтверждение факта? Где ручная операция? Где риск задержки? Где точка передачи между подразделениями? Где создаётся доказательная запись?

В-пятых, процессный подход даёт основу для автоматизации. ERP нельзя настраивать только по перечню пожеланий. Нужно понимать процесс: какие документы возникают, кто их создаёт, в какой момент, какие данные должны быть заполнены, какие статусы используются, где нужна блокировка, где нужен отчёт, где нужна интеграция, где нужна инструкция.

Всё это делает процессный подход обязательным элементом зрелого внедрения. Но именно здесь возникает первая граница. Процессный подход хорошо описывает движение деятельности, но сам по себе не всегда отвечает на вопрос: что именно движется?

Можно описать процесс закупки. Можно указать, что инициатор формирует заявку, руководитель согласует, закупщик выбирает поставщика, поставщик подтверждает срок, склад принимает товар, бухгалтерия отражает документы. Это похоже на полноценное описание. Но если не задан предмет процесса, остаётся неопределённость: чем мы управляем? Заяв-

кой? Потребностью? Заказом поставщику? Дефицитом? Обязательством поставщика? Запасом? Сроком обеспечения?

На практике именно здесь часто появляется процедурная оболочка. Процесс вроде бы описан. В нём есть шаги, роли, документы, действия, согласования. Но непонятно, какой бизнес-предмет проходит через этот процесс и какое состояние меняется. Такой процесс можно прочесть, но трудно использовать как основу для графа знаний, для ИИ, для стресс-тестирования, для поиска разрывов и для точной ERP-привязки.

Процесс без предмета похож на маршрут без груза. Мы видим дорогу, повороты, пункты передачи, участников движения, но не знаем, что именно перевозится, в каком состоянии оно было в начале, что с ним должно стать в конце и почему этот маршрут вообще нужен. Управленчески это слабая конструкция.

Эта граница сохраняется и тогда, когда процесс нарисован в специализированной BPM/ВРА-системе. Нотация может быть правильной, дорожки могут быть выстроены, роли названы, документы привязаны, контрольные точки отмечены. Но если процесс не показывает предмет-драйвер и переходы его состояния, он остаётся процессной оболочкой. Он сообщает, как движется деятельность, но не всегда сообщает, что именно в деятельности меняется.

Для иницилирующего ядра процесс важен не сам по себе, а как траектория изменения предмета. Ядро должно понимать не только порядок шагов, но и то, какой предмет вошёл в процесс, в каком состоянии, что с ним произошло, какой носитель это подтвердил, кто принял результат и куда предмет перешёл дальше.

Предмет процесса — это смысловая управляемая сущность, которая проходит через процесс и ради изменения которой процесс существует. Это может быть заказ клиента, потребность, запас, партия, производственная программа, производственный заказ, протокол контроля, несоответствие, заявка на платеж, платежное обязательство, договорное условие, первичный документ, управленческое решение.

Предмет процесса не всегда совпадает с документом. В процессе закупки документом может быть «Заказ поставщику», но предметом может быть «обеспечиваемая потребность». В процессе производства документом может быть «Заказ на производство», но предметом может быть «производственная партия» или «производственное обязательство». В процессе качества документом может быть «Протокол контроля», но предметом может быть «состояние пригодности партии». В процессе платежей документом может быть «Заявка на расходование денежных средств», но предметом может быть «платежное обязательство».

Если предмет не назван, процесс начинает притворяться полным. В тексте появляются слова «оформить», «проверить», «согласовать», «передать», «создать», «провести», «закрыть», но остаётся неясно, что именно изменилось в управлении. Документ создан — но предмет перешёл в новое состояние или нет? Согласование выполнено — но обязательство возникло или только подготовлено? Отчёт сформирован — но решение принято или только появились данные для решения?

Поэтому рядом с предметом процесса нужно всегда видеть состояние предмета. Состояние — это управленчески значимое положение предмета в конкретный момент. Предмет «потребность» может быть выявлен, подтверждён, отклонён, включён в обеспечение, закрыт заказом поставщику, частично обеспечен, просрочен, снят, заменён аналогом. Предмет «производственный заказ» может быть подготовлен, утверждён, обеспечен материалами, запущен, выполняется, приостановлен, выпущен, закрыт. Предмет «несоответствие» может быть выявлено, зарегистрировано, классифицировано, принято к разбору, закрыто корректирующим действием или признано несущественным.

Если в процессе нет состояний, процесс превращается в список действий. А список действий плохо показывает управление. Можно выполнить действия и не получить результата. Можно создать документ, но не изменить состояние предмета. Можно провести совещание, но

не принять решение. Можно оформить заявку, но не создать обязательство. Можно поставить статус, но не обеспечить фактическое выполнение.

Следующий ключевой элемент — пусковая точка. Процесс не должен начинаться просто потому, что «так принято» или «кто-то открыл документ». У зрелого процесса есть событие или состояние, которое запускает движение предмета. Пусковая точка отвечает на вопрос: что произошло с предметом или вокруг предмета, из-за чего процесс должен стартовать?

Например, процесс обеспечения материалов может запускаться появлением подтверждённой потребности. Не просто тем, что кто-то захотел купить материал, а тем, что потребность прошла минимальную проверку: материал нужен, количество определено, срок известен, источник потребности понятен, обеспечение через склад невозможно или недостаточно. Тогда процесс закупки имеет предметную пусковую точку.

Процесс управления несоответствием может запускаться фактом выявленного отклонения. Не просто заполнением формы, а возникновением состояния: результат контроля не соответствует требованию, партия не может быть автоматически допущена, требуется решение. Тогда процесс качества не является бумажной процедурой. Он начинается с предметного события.

Процесс платежа может запускаться наступлением платежного обязательства. Не просто созданием заявки, а тем, что предприятие признало необходимость платежа по договору, счёту, графику, решению, налоговому обязательству или внутреннему лимиту. Тогда платежный процесс не сводится к согласованию формы. Он управляет движением обязательства к исполнению.

Ещё один элемент — handoff-точка, точка передачи. В обычном процессном описании её часто воспринимают как «передали следующему сотруднику». Но в предметном подходе handoff-точка означает больше. Это момент, когда предмет или его состояние передаётся из одного контура ответственности в другой.

Например, закупка передаёт складскому контуру не просто документ поступления, а предмет «поступивший запас», который должен быть принят, размещён, идентифицирован, возможно проверен по качеству, сделан доступным или помещён в ограниченный статус. Склад передаёт производству не просто материалы, а обеспеченность производственного задания. Производство передаёт складу не просто выпуск, а партию продукции в определённом состоянии. ОТК передаёт производству или складу не просто протокол, а решение о пригодности, ограничении или несоответствии.

Если handoff-точка не описана предметно, возникает типовой разрыв: один контур считает, что свою работу выполнил, а следующий контур не может работать. Закупка говорит: «Мы заказали». Склад говорит: «У нас нет доступного остатка». Производство говорит: «Материала нет». Бухгалтерия говорит: «Документы не отражены». Руководитель говорит: «Почему заказ стоит?» Формально действия выполнены, но предмет не перешёл в нужное состояние.

Рассмотрим пример с запасами. В плоском процессном описании можно написать: «Сформировать потребность, оформить заказ поставщику, принять товар, разместить на складе, выдать в производство». Такой процесс выглядит понятным. Но если посмотреть предметно, то внутри него живёт несколько связанных предметов: потребность, заказ поставщику, ожидаемое поступление, фактическое поступление, складской запас, доступный запас, резервированный запас, запас в карантине, запас, переданный в производство.

Проблема часто возникает именно из-за смешения этих предметов. Пользователь видит, что товар «есть на складе», и считает, что производство обеспечено. Но предмет «складской остаток» не равен предмету «доступный для производства запас». Товар может быть резервирован под другой заказ, не пройти входной контроль, находиться не в той ячейке, числиться на удалённом складе, иметь неподходящую серию, быть заблокированным, не иметь нужной аналитики или быть учётным остатком без фактического наличия.

Если процесс описан только как последовательность действий, такая разница теряется. Если процесс описан через предметы и состояния, становится видно: цель процесса не просто «принять товар», а перевести потребность из состояния «не обеспечена» в состояние «обеспечена доступным запасом с нужными характеристиками, количеством, сроком и качественным статусом».

Тогда и управление запасами перестаёт быть набором складских операций. Оно становится маршрутом предметных переходов: потребность выявлена, обеспечение запланировано, заказ размещён, поступление ожидается, товар принят, качество подтверждено, запас доступен, резерв создан, материал передан в производство, потребность закрыта. Такой маршрут уже можно анализировать, автоматизировать, контролировать и объяснять ИИ.

Второй пример — производство. Формально процесс можно описать так: «Создать производственный заказ, обеспечить материалами, выполнить операции, выпустить продукцию, передать на склад, закрыть заказ». Это правильный процедурный каркас. Но предметно здесь возникает более сложная картина.

Что является главным предметом процесса? Производственный заказ? Производственная партия? Выпуск? НЗП? Производственное обязательство перед продажами? Ответ зависит от уровня описания. На одном уровне главным предметом может быть производственная программа. На другом — производственный заказ. На третьем — партия продукции. На четвёртом — состояние НЗП. На пятом — готовность изделия к передаче на склад или контролю качества.

Если этого различия нет, процесс производства начинает путать управление заданием, управление материалами, управление операциями, управление качеством и управление себестоимостью. В тексте всё может называться «производство», но для ERP и ИИ это разные предметные слои. Материальная обеспеченность — один предмет. Выполнение операций — другой. Выпуск — третий. НЗП — четвёртый. Производственная себестоимость — пятый. Качество партии — шестой.

Предметный взгляд позволяет задать вопрос точнее: что именно должно измениться в результате каждого этапа? Производственный заказ должен перейти из состояния «подготовлен» в состояние «запущен». Материальная обеспеченность должна перейти из состояния «не подтверждена» в состояние «достаточна для запуска». Операция должна перейти из состояния «запланирована» в состояние «выполнена». Партия должна перейти из состояния «в производстве» в состояние «выпущена». Качество должно перейти из состояния «не подтверждено» в состояние «разрешено к использованию» или «требует решения». Себестоимость должна быть рассчитана и включена в управленческий результат.

Именно поэтому процесс без предмета плохо автоматизируется. Автоматизация требует точных носителей: документов, справочников, регистров, статусов, аналитик, маршрутов, прав, отчётов. Но эти носители должны быть привязаны к предмету. Если предмет не определён, команда проекта начинает настраивать форму вместо управления. Система получает документы, но не получает ясной логики состояний. Потом появляются вопросы: почему документ проведён, но процесс не завершён? Почему отчёт есть, но решение не принято? Почему операция выполнена, но предмет не передан дальше?

Процесс без предмета плохо читается и человеком. Пользователь видит длинный регламент, но не понимает, что является результатом. Он выполняет шаги, но не видит, какой предмет должен перейти в какое состояние. Руководитель смотрит на процесс и не видит управляемого результата. Консультант пытается привязать процедуру к ERP, но не понимает, какой объект ERP-системы должен быть носителем предметного перехода.

Для ИИ это ещё опаснее. LLM хорошо читает процедурные тексты. Она может выделить шаги, роли, действия, документы, условия. Но если в тексте не задан предмет процесса, модель будет достраивать его сама. Она может решить, что главным предметом является документ,

хотя на самом деле документ только фиксирует состояние. Она может принять промежуточное действие за результат. Она может не увидеть handoff-точку. Она может не понять, где процесс должен остановиться из-за отклонения.

В результате ИИ начнёт уверенно описывать процесс, но не сможет устойчиво рассуждать о его последствиях. Он скажет, что закупка завершена, потому что заказ поставщику создан. Но предмет «потребность» ещё не обеспечен. Он скажет, что производство можно запускать, потому что производственный заказ сформирован. Но предмет «материальная обеспеченность» не подтверждён. Он скажет, что партия готова к отгрузке, потому что выпуск отражён. Но предмет «качество партии» ещё не получил разрешающее состояние.

Поэтому в предметно-ориентированной методологии процесс должен описываться не только через действия, но и через предметный переход. У процесса должен быть главный предмет. У предмета должно быть входное состояние. Должно быть целевое состояние или результат. Должны быть обеспечивающие предметы. Должна быть пусковая точка. Должны быть handoff-точки. Должны быть носители фиксации: ERP-объекты, документы, записи СМК, отчёты, внешние файлы, инструкции, роли.

Тогда процесс перестаёт быть процедурной оболочкой. Он становится маршрутом изменения состояния бизнес-предмета.

Эта формула важна для всей книги:

процесс — это маршрут изменения состояния бизнес-предмета.

Не просто маршрут действий. Не просто последовательность документов. Не просто регламент. Не просто схема согласования. А именно маршрут, по которому предмет предприятия проходит от одного управленчески значимого состояния к другому.

Из этой формулы следует практическое правило: если в описании процесса невозможно назвать предмет, его входное состояние, целевое состояние и точки передачи, процесс ещё не готов для интеллектуального предприятия. Он может быть полезен как первичное описание. Он может помочь обучить пользователя. Он может служить регламентной заготовкой. Но он ещё не стал предметной основой для ERP, графа знаний и ИИ.

Следующая глава переходит от процесса к ERP. Даже если предмет в процессе найден, в ERP-системе он фиксируется не напрямую, а через прикладные носители: документы, справочники, регистры, отчёты, настройки, статусы, аналитики. Поэтому нужно разобрать вторую ошибку: ERP-объект не равен бизнес-предмету.

Глава 14. ERP-объект не равен бизнес-предмету

В предыдущей главе мы зафиксировали формулу: процесс — это маршрут изменения состояния бизнес-предмета. Но в реальном проекте процесс почти сразу упирается в ERP. Нужно создать документ. Выбрать справочник. Заполнить реквизиты. Провести операцию. Получить движение по регистрам. Проверить отчёт. Настроить статус. Разобраться с аналитикой. Назначить права. Открыть рабочее место.

На этом этапе возникает вторая типовая ошибка: бизнес-предмет начинают отождествлять с ERP-объектом.

На первый взгляд это кажется естественным. Если в системе есть документ «Заказ клиента», значит предметом является заказ клиента. Если есть справочник «Номенклатура», значит предметом является номенклатура. Если есть регистр остатков, значит предметом является остаток. Если есть отчёт по задолженности, значит предметом является задолженность. Если есть статус документа, значит он и описывает состояние предмета.

Но это слишком прямое мышление. Оно удобно для обучения интерфейсу, но опасно для построения предметной модели предприятия.

ERP-объект — это прикладной или технический носитель. В ERP-системе таким носителем может быть документ, справочник, регистр, отчёт, настройка, статус, аналитика, рабочее место, реквизит, табличная часть, обработка, печатная форма, обменное сообщение, внешний идентификатор. Все эти объекты важны. Без них система не работает. Но они не равны бизнес-предметам.

Бизнес-предмет — это смысловая управляемая сущность предприятия. Он возникает, изменяется, передаётся, фиксируется, контролируется, измеряется или становится основанием для решения. ERP-объект может фиксировать его состояние, хранить часть его данных, запускать его переход, отображать его в отчёте или быть носителем действия над ним. Но предмет шире своего носителя.

Если этого не различать, методология начинает строиться от объектов ERP-системы, а не от управляемой реальности предприятия. Проектная команда открывает конфигурацию, видит список документов и справочников и пытается из них вывести бизнес-модель. Такой путь кажется рациональным: раз ERP уже устроена определённым образом, значит нужно описать именно эти объекты. Но в результате предприятие начинает мыслить не своими предметами, а формами системы.

Это приводит к нескольким искажениям.

Первое искажение: документ принимают за предмет. Например, говорят: «Наш предмет — заказ клиента». Но нужно уточнить, о чём речь. О документе «Заказ клиента» как объекте ERP-системы? О клиентском обязательстве? О планируемой отгрузке? О коммерческом соглашении? О резерве под клиента? О будущем денежном потоке? О производственной потребности, которая возникла из заказа? Один и тот же термин может обозначать разные предметные слои.

Второе искажение: статус документа принимают за состояние предмета. Документ может быть проведён, согласован, закрыт, отменён, помечен на удаление, находиться в определённом прикладном статусе. Но состояние бизнес-предмета может быть другим. Документ проведён, а потребность ещё не обеспечена. Заказ согласован, а клиентское обязательство ещё не исполнено. Производственный заказ создан, а материалы не доступны. Партия выпущена, а качество не подтверждено.

Третье искажение: отчёт принимают за предметную модель. Отчёт показывает выбранный срез данных. Он может быть очень полезным, но сам по себе не объясняет, какой предмет он отражает, какой жизненный цикл стоит за цифрой, какие состояния были пройдены, какие

переходы произошли и какие решения должны последовать. Отчёт отвечает на вопрос «что видно в этом разрезе?», но не всегда отвечает на вопрос «что происходит с предметом?».

Четвёртое искажение: регистр принимают за управленческую сущность. Регистр хранит движения, остатки, обороты, сведения. Но сам по себе он является способом фиксации. Управленческий предмет может проявляться в нескольких регистрах одновременно. Остаток, резерв, себестоимость, задолженность, доступность, обеспечение, НЗП — всё это может быть связано с регистрами, но не сводится к ним.

Пятое искажение: настройку принимают за методологию. В ERP есть множество параметров, флагов, вариантов учёта, правил заполнения, схем обеспечения, способов отражения, настроек складов, серий, качества, планирования. Но включить настройку не значит построить предметное управление. Настройка только разрешает или меняет поведение системы. Нужно ещё понимать, какой предмет после этого появляется, какие состояния нужно контролировать, кто отвечает и какие последствия возникают.

Поэтому формула главы звучит жёстко:

ERP-объект не равен бизнес-предмету.

Для будущего ядра ERP-объект является одним из самых сильных носителей предметного состояния, но не самим предметом. Документ, справочник, регистр, статус или движение могут фиксировать важный факт. Однако ядро должно удерживать более широкую связь: какой бизнес-предмет представлен этим объектом, какие состояния он проходит, какие внешние и внутренние источники его подтверждают, какие процессы и роли зависят от этого состояния.

Именно поэтому нельзя говорить, что внедрение ERP автоматически создаёт иницирующее ядро. ERP может дать транзакционную дисциплину, историю, статусы, маршруты, права и отчётность. Но предметное ядро возникает только тогда, когда ERP-объекты связаны с бизнес-предметами, процессами, СМК, KPI, источниками и цифровой нитью.

ERP-объект является носителем, следом, отражением, технической реализацией или точкой фиксации бизнес-предмета.

Эта формула особенно важна для ERP-систем, потому что система богата прикладными объектами. Она умеет фиксировать заказы, закупки, продажи, складские операции, производство, финансы, регламентированный и управленческий учёт, планирование, обеспечение, ремонты, бюджетирование, НСИ, взаиморасчёты, статусы, движения, отчёты. Но богатство прикладной структуры не отменяет вопроса: какие бизнес-предметы через эту структуру управляются?

Рассмотрим первый важный случай: один бизнес-предмет может быть реализован несколькими ERP-объектами.

Возьмём предмет «клиентское обязательство». В ERP он может проявляться через заказ клиента, соглашение, договор, резерв товара, график оплаты, плановую дату отгрузки, состояние обеспечения, документ реализации, отгрузочное распоряжение, задолженность, платёж, претензию, возврат, отчёт по выполнению заказов. Если смотреть только на документ «Заказ клиента», предмет окажется искусственно сужен.

Клиентское обязательство не исчерпывается созданием заказа. Оно может иметь коммерческую сторону, складскую сторону, производственную сторону, финансовую сторону, договорную сторону, сервисную сторону. В одном случае важна цена и скидка. В другом — срок. В третьем — резерв. В четвёртом — производство под заказ. В пятом — оплата до отгрузки. В шестом — рекламация после поставки. Всё это один предметный узел, но несколько ERP-носителей.

Другой пример — «партия продукции». В ERP партия может проявляться через выпуск продукции, складской остаток, серию, характеристику, назначение, документ передачи, складской ордер, регистр себестоимости, документ контроля качества, решение о пригодности, отчёт по остаткам, отчёт по движениям, документ реализации, рекламацию. Если принять

за партию только складской остаток, потеряется качество. Если только выпуск — потеряется движение. Если только себестоимость — потеряется прослеживаемость. Если только серия — потеряется экономическая сторона.

Третий пример — «потребность в материалах». Она может быть видна через план производства, заказ на производство, спецификацию, заказ на обеспечение, заказ поставщику, резерв, складской остаток, отчёт о дефиците, график поступления, распоряжение на перемещение, замену материала. Предмет «потребность» проходит через несколько ERP-объектов, и ни один из них по отдельности не раскрывает её полностью.

Из этого следует важное правило: если один предмет реализован несколькими ERP-объектами, граф знаний должен связывать эти объекты вокруг предмета. Нельзя оставлять их разрозненными. Иначе ИИ или аналитик увидит отдельные документы и отчёты, но не увидит управленческую сущность, которая через них проходит.

Теперь рассмотрим обратный случай: один ERP-объект может обслуживать несколько бизнес-предметов.

Документ «Заказ клиента» может одновременно быть носителем нескольких предметов. Он фиксирует коммерческое намерение клиента, плановую отгрузку, источник потребности в товаре или производстве, будущую выручку, возможный резерв, основание для обеспечения, элемент портфеля продаж, элемент план-факт анализа, источник обязательства по сроку. Один документ, несколько предметов.

Документ «Заказ поставщику» может быть носителем предмета «обеспечение потребности», предмета «обязательство поставщика», предмета «ожидаемое поступление», предмета «будущий кредиторский долг», предмета «риск срыва срока», предмета «цена закупки», предмета «условие поставки». Если проектная команда видит только документ закупки, она может не заметить, что этот документ связан с несколькими управленческими смыслами.

Документ «Выпуск продукции» может одновременно фиксировать предмет «результат производства», предмет «партия продукции», предмет «изменение НЗП», предмет «появление готового запаса», предмет «основа для расчёта себестоимости», предмет «событие для контроля качества». Один документ может быть точкой пересечения производства, склада, качества и учёта.

Регистр остатков может обслуживать предмет «фактический складской запас», предмет «доступный запас», предмет «резерв», предмет «запас под назначение», предмет «запас в ограниченном статусе». Если не различить эти предметы, пользователь будет видеть цифру и делать неправильный вывод. Он скажет: «Остаток есть». Но управленческий вопрос должен звучать иначе: какой это остаток, кому он доступен, в каком состоянии, под какое назначение, с каким качественным статусом и можно ли его использовать?

Отчёт по задолженности может обслуживать предмет «дебиторская задолженность», предмет «платёжное обязательство клиента», предмет «риск кассового разрыва», предмет «просрочка», предмет «лимит кредитования», предмет «финансовое поведение клиента». Если смотреть только как на отчёт, можно увидеть сумму. Если смотреть предметно, возникает управленческая картина.

Теперь нужно сформулировать, что именно делает ERP.

ERP фиксирует след изменения состояния предмета.

Это не умаляет ERP. Напротив, это показывает её настоящую силу. ERP-система ценна не тем, что в ней много документов и справочников, а тем, что она позволяет фиксировать изменения предприятия в связанной прикладной форме: документы создают движения, движения меняют остатки и обороты, статусы ограничивают действия, аналитики задают разрезы, отчёты показывают состояние, настройки определяют поведение, права задают ответственность.

Но ERP фиксирует то, что в неё внесено и что настроено как значимое. Она не обязана сама раскрывать весь предметный смысл. Если проект не задал предметы, ERP будет честно хранить документы, регистры, статусы и отчёты, но связность смысла останется в голове людей или будет утеряна.

Возьмём пример с заказом клиента. Пользователь открывает заказ и видит строки номенклатуры, количество, цену, дату, контрагента, соглашение, склад, условия оплаты, состояние отгрузки, обеспечение, резерв, статус. На уровне ERP-объекта это документ с реквизитами и табличной частью. На предметном уровне это узел нескольких управляемых сущностей.

Первый предмет — клиентское обязательство. Предприятие обещает что-то поставить на определённых условиях. Второй предмет — плановая выручка. Заказ может стать основанием для финансового ожидания. Третий предмет — потребность в обеспечении. Если товара нет, заказ создаёт потребность в закупке, производстве или перемещении. Четвёртый предмет — резерв или назначение запаса. Пятый предмет — риск исполнения срока. Шестой предмет — будущая дебиторская задолженность или предоплата. Седьмой предмет — управленческий приоритет клиента.

Если модель предприятия построена только от документа «Заказ клиента», все эти предметы сольются. Тогда ИИ может сделать опасно упрощённый вывод: заказ есть, значит обязательство существует и может быть исполнено. Но в реальности заказ может быть предварительным, не обеспеченным, без оплаты, с неподтверждённой датой, с дефицитной номенклатурой, с запретом отгрузки, с нерешёнными условиями по договору.

Предметный анализ требует спросить: в каком состоянии находится каждый предмет, связанный с этим ERP-объектом? Клиентское обязательство подтверждено или нет? Обеспечение возможно или нет? Резерв создан или нет? Финансовое условие выполнено или нет? Срок реалистичен или нет? Риск исполнения принят или нет? Только после этого можно рассуждать управленчески.

Теперь пример с партией. В ERP партия может выглядеть как серия, характеристика, назначение, строка выпуска, остаток на складе, движение по регистру, объект контроля качества, основание списания, связь с себестоимостью. Но предмет «партия» нельзя свести к одному из этих носителей.

Партия как складской предмет отвечает на вопросы: где находится, сколько осталось, в какой ячейке, доступна ли, зарезервирована ли, перемещена ли, отгружена ли. Партия как предмет качества отвечает на другие вопросы: прошла ли контроль, есть ли протокол, есть ли несоответствие, разрешено ли использование, есть ли ограничения, есть ли прослеживаемость. Партия как учётный предмет отвечает на вопросы стоимости: какая себестоимость, как она сформирована, как списывается, как влияет на финансовый результат. Партия как регуляторный предмет может отвечать на вопросы идентификации, срока годности, сертификации, рекламаций, обязательных записей.

Если проект строит модель только от ERP-объекта, он может выбрать один носитель и потерять остальные проекции. Например, увидеть партию только в складе и не увидеть её в качестве. Или увидеть только в качестве и не увидеть в себестоимости. Или увидеть только как серию, но не как управляемый жизненный цикл от выпуска до отгрузки и возможной рекламации.

Отсюда следует принцип: ERP-объект должен быть включён в предметную модель как носитель, а не как окончательное определение предмета.

Это особенно важно для методологии внедрения. Если проект начинается с перечня объектов ERP-системы, есть риск подмены. Главный предмет процесса подменяется документом. Роли начинают жить отдельно от предмета. Процедуры становятся шаблонными. ERP-привязка строится по догадке. Прикладной слой теряет связь с реальностью. В итоге команда знает,

какие документы нужно создать, но не понимает, какой управленческий переход должна обеспечить система.

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.