

# НАИВНАЯ ЦИФРОВИЗАЦИЯ

Когда вместо процессов мы цифровизируем  
действия человека



ASIC  
Book  
AI SNACKS

ВЕСЕЛИН БОЙЧЕВ

САНКТ-ПЕТЕРБУРГ • 2026

**Веселин Бойчев**

**"Наивная цифровизация"**  
**- Почему строительный**  
**бизнес не справляется с**  
**крутыми поворотами рынка.**

*<https://litres.ru/74492934>  
SelfPub; 2026*

**Аннотация**

Книга «Наивная цифровизация» — это честный, экспертный и написанный с отличным юмором ответ на вопрос, почему строительный бизнес не вписывается в крутые повороты рынка. Как оказалось, цифровая трансформация девелоперов часто носит лишь поверхностный характер. Застройщики оцифровали документацию, увлеклись 3D-визуализациями и другими «игрушками», но так и не дошли до главного — автономного анализа, управления и прогнозирования. А ведь именно автоматический анализ ключевых показателей корпорации лежит в основе точных прогнозов. Сегодня же, с появлением искусственного интеллекта (ИИ), это открывает прямую дорогу к автоматизированному управлению будущим в краткосрочной перспективе.

# Содержание

|                                                              |    |
|--------------------------------------------------------------|----|
| Содержание                                                   | 11 |
| Почему мы цифровизируем не то                                | 13 |
| Где математическая модель работает<br>безупречно             | 15 |
| Человек живёт в противоречиях и<br>конфликтах                | 17 |
| В чём беда ТИМ-идеологии                                     | 19 |
| Там, где формальной логики недостаточно                      | 21 |
| Альтернатива — автономный цифровой<br>двойник                | 23 |
| Автономность через повторяющиеся циклы                       | 24 |
| Пример 1. Термостат и Иван Иванович из<br>котельной          | 29 |
| Пример 3. Завод и незаменимый Сергей<br>Петрович             | 36 |
| Пример 5. Цифровая служба качества                           | 41 |
| Человек — не датчик, а маленькое<br>информационное агентство | 51 |
| Пример 1. Четыре фотографии одного<br>дефекта                | 52 |
| Факт сначала проходит проверку на<br>безопасность            | 55 |
| Пример 2. Компания, победившая                               | 56 |

|                                                          |    |
|----------------------------------------------------------|----|
| аварийность                                              |    |
| Усталость как метод обработки информации                 | 58 |
| Пример 3. Пятничный отчёт                                | 59 |
| Личная выгода редко приходит с табличкой                 | 61 |
| Пример 4. Прогноз продаж, который всем понравился        | 62 |
| Статус определяет вес факта                              | 64 |
| Пример 5. Молодой специалист и уважаемая колонна         | 65 |
| Отношения между подразделениями                          | 67 |
| Пример 6. Исчезнувшие окна                               | 67 |
| Желание избежать ответственности                         | 70 |
| Пример 7. Красная кнопка                                 | 70 |
| Добросовестность нельзя использовать как архитектуру     | 72 |
| Человек не виноват в том, что он человек                 | 74 |
| От управления добросовестностью — к управлению процессом | 76 |
| Неустойчивый элемент в устойчивой модели                 | 78 |
| Пример 1. Бетон, который решил немного подождать         | 83 |
| Задержка как психологическая анестезия                   | 85 |
| Пример 2. Письмо, которое созревало три дня              | 86 |
| Как улучшить показатель без улучшения результата         | 88 |

|                                                            |     |
|------------------------------------------------------------|-----|
| Пример 3. Отдел технической поддержки<br>победил обращения | 88  |
| Показатель становится мишенью                              | 90  |
| Пример 4. Проект, который никогда не<br>опаздывал          | 90  |
| Перенос ответственности как корпоративный<br>спорт         | 93  |
| Пример 5. Совещание о недостающем<br>кабеле                | 93  |
| Выиграть время — значит изменить<br>реальность             | 96  |
| Пример 6. Поставщик, который почти<br>успел                | 97  |
| Неофициальный маршрут быстрее<br>официального              | 99  |
| Пример 7. Лифт, о котором знали все                        | 99  |
| Психика не ломает систему — она к ней<br>приспосабливается | 102 |
| Почему цифровая модель не видит теневой<br>контур          | 104 |
| Как сделать теневой контур видимым                         | 106 |
| Два контура должны поменяться местами                      | 108 |
| Тень, которая всегда следует за моделью                    | 110 |
| Универсальный регулятор                                    | 114 |
| Пример 1. Начальник, который услышал<br>отставание         | 115 |

|                                                  |     |
|--------------------------------------------------|-----|
| Глаз начальника                                  | 118 |
| Пример 2. Склад, на котором всё было             | 119 |
| Совесть начальника                               | 121 |
| Пример 3. Партия изделий и один честный человек  | 122 |
| Герой как симптом слабой системы                 | 125 |
| Пример 4. Алексей Викторович спасает понедельник | 125 |
| Начальник как интеграционная платформа           | 128 |
| Пример 5. Четыре программы и один Николай        | 129 |
| Когда глаз становится бутылочным горлышком       | 131 |
| Пример 6. Директор и двести согласований         | 132 |
| Зависимость от личности                          | 134 |
| Пример 7. Когда Павел Андреевич ушёл             | 134 |
| Опасность доброго начальника                     | 137 |
| Цифровизация глаза начальника                    | 139 |
| Как превратить опыт в систему                    | 141 |
| Глаз, совесть и алгоритм                         | 143 |
| Не уничтожить начальника системой                | 145 |
| Пример 1. Счётная машина в кабинете директора    | 150 |
| Пример 2. Машина задаёт лишний вопрос            | 156 |
| Пример 3. Три правды об одном подшипнике         | 161 |



**Веселин Бойчев**

**"Наивная цифровизация"  
- Почему строительный  
бизнес не справляется  
с крутыми  
поворотами рынка.**

# ЭВОЛЮЦИЯ ОПЫТА И ТЕХНОЛОГИЙ

БОЛЕЕ 30 ЛЕТ ПРАКТИКИ ЦИФРОВОЙ ТРАНСФОРМАЦИИ СТРОИТЕЛЬСТВА

КАРТОЧКА №2



## БОЙЧЕВ ВЕСЕЛИН АТАНАСОВ

ПРЕДПРИНИМАТЕЛЬ,  
ЭКСПЕРТ И РАЗРАБОТЧИК



Путь длиной более 30 лет — от первых CAD-систем и Excel до автономных цифровых двойников и искусственного интеллекта.



Постоянное развитие методологии, технологий и практических решений в строительстве.



Фокус на реальных кейсах, практической пользе и измеримых результатах.

СОЗДАТЕЛЬ МЕТОДОЛОГИИ И ТЕХНОЛОГИЙ



**АСИК**

МЕТОДОЛОГИЯ.  
ТЕХНОЛОГИЯ. РЕЗУЛЬТАТ.



**ASIC DTW**

DIGITAL TWIN  
FOR THE REAL WORLD

### 1994 ПЕРВЫЕ ШАГИ В ЦИФРОВОМ ПРОЕКТИРОВАНИИ

FloorPlan 3D — один из первых практических проектов автоматизированного проектирования. Интеграция проектирования с Microsoft Excel и макросами еще в среде Windows 3.11.



**FLOORPLAN 3D**

### 1998–2007 МЕТОДИКИ УПРАВЛЕНИЯ И АВТОМАТИЗАЦИЯ

Разработка методик управления строительными проектами, сметных технологий и систем автоматизации планирования на основе Excel, VBA и баз данных.



### 2008 СОЗДАНИЕ МЕТОДОЛОГИИ АСИК

Разработка и внедрение собственной методологии управления инвестиционно-строительными проектами на основе кибернетического подхода.



**АСИК**

### 2010–2018 BIM: ОТ ГЕОМЕТРИИ К УПРАВЛЕНИЮ

Практические внедрения BIM-технологий. Переход от 3D (геометрия) к 4D/5D управлению сроками, ресурсами, стоимостью и качеством.



### 2019–2023 ЦИФРОВЫЕ ДВОЙНИКИ СТРОИТЕЛЬСТВА

Разработка и внедрение кибернетических цифровых двойников (Cyber Digital Twin). Интеграция данных факта, планов, отклонений и прогнозной аналитики.



### 2024–2026 ИИ, ТТК И ИНДУСТРИЯ 4.0 / 5.0

Интеграция искусственного интеллекта, типовых технологических карт (ТТК) и технологий Индустрии 4.0/5.0. Подготовка кадров, стандартизация знаний и автономное управление строительными проектами.



Я не наблюдал эволюцию цифровых технологий — я проходил её вместе с ними, от первых CAD-систем до автономных цифровых двойников с искусственным интеллектом.

*B. Boychev*



ОПЫТ  
**30+ ЛЕТ**



МЕЖДУНАРОДНЫЙ ОПЫТ  
Великобритания, Болгария,



ПРАКТИЧЕСКИЕ КЕЙСЫ  
**90+**



ФОКУС  
Результат.

# **Наивная цифровизация**

*Как цифровые инструменты оставили человека регулятором, контролёром и живым алгоритмом процесса*

Веселин Бойчев

Санкт-Петербург • 2026

# Содержание

Вступление 5

**Часть I. Первая наивность кибернетики 10**

Глава 1. Машина, которая обещала управлять 11

Глава 2. Человек как нестабильный элемент модели 25

Глава 3. Психика как теневой контур управления 39

Глава 4. Глаз и совесть начальника 54

**Часть II. Советский эксперимент 69**

Глава 5. Кибернетику разрешили — управление не отдали  
70

Глава 6. ОГАС и угроза объективного контура .88

Глава 7. Как большая система уменьшилась до рабочего  
места 105

Глава 8. Почему историческая ошибка сохранилась 121

**Часть III. Две цифровизации 136**

Глава 9. Цифровизация выполнения работ 137

Глава 10. Цифровизация управления процессом 152

Глава 11. Человек в двух архитектурах 171

Глава 12. Стандартизация или вечное ручное управление  
188

**Часть IV. Анатомия современной наивной цифро-  
визации 209**

Глава 13. ИТ-архитектура как цифровое штатное распи-  
сание 210

|                                                           |            |
|-----------------------------------------------------------|------------|
| Глава 14. Цифровой документ вместо цифрового состояния    | 230        |
| Глава 15. Человек как ручной API                          | 247        |
| Глава 16. Уведомление вместо управления                   | 256        |
| Глава 17. Дашборд и иллюзия контроля                      | 264        |
| <b>Часть V. Диагностика ИТ-бюджета</b>                    | <b>272</b> |
| Глава 18. Пять вопросов к цифровому проекту               | 273        |
| Глава 19. Признаки неправильного расходования бюджета     | 282        |
| Глава 20. Ложные показатели цифровой зрелости             | 288        |
| <b>Часть VI. Документооборот и граница управления</b>     | <b>302</b> |
| Глава 21. Что действительно цифровизирует документооборот | 303        |
| Глава 22. Документ согласован — процесс не завершён       | 312        |
| Глава 23. «Типовое ПО: документооборот»                   | 319        |
| <b>Часть VII. Выход из „наивной цифровизации“</b>         | <b>333</b> |
| Глава 24. Сначала процесс, потом программа                | 334        |
| Глава 25. Новая роль человека                             | 342        |
| Глава 26. Замкнутый цифровой цикл                         |            |
| <b>О чём эта книга</b>                                    |            |

# Почему мы цифровизируем не то

Цифровизация вошла в организации через рабочее место.

Сначала компьютер заменил пишущую машинку. Затем электронная таблица заменила журнал. Документооборот заменил папки, ВІМ — чертежи, дашборд — бумажный отчёт, а искусственный интеллект начал помогать сотруднику читать, считать, искать и формулировать.

Каждый новый инструмент делал отдельного человека быстрее.

Но организация в целом далеко не всегда становилась производительнее.

Чем больше появлялось программ, тем чаще сотрудник превращался в связующее звено между ними. Он переносил данные, сверял версии, объяснял расхождения, проверял FАСТ, искал причины отклонения и вручную запускал следующий этап.

Так возникла наивная цифровизация — массовая автоматизация рабочих мест без изменения самой архитектуры управления.

Эта книга предлагает другой путь: цифровизировать не человека и не его отдельное действие, а повторяющийся процесс управления. Создавать не очередное программное рабочее место, а автономную цифровую модель — цифровой двойник, способный наблюдать изменение процесса, опреде-

лять тенденции, прогнозировать последствия, поддерживать коллективное решение, выполнять допустимое воздействие и проверять его результат.

# Где математическая модель работает безупречно

Математические модели особенно эффективны там, где объект управления не обладает собственной волей. В таких системах можно установить входные условия, определить зависимости, рассчитать результат и ожидать повторяемого поведения. Но производственная или управленческая организация состоит не только из машин, материалов и формальных операций. В неё вплетены люди, подразделения, договорные отношения, профессиональные позиции, ответственность, страх, опыт, усталость, статус и личный интерес.

Когда человек входит в математическую модель, он не становится ещё одной безвольной переменной. Он начинает понимать, как система его оценивает. Он приспособливает поведение. Защищает собственную роль. Оспаривает критерии. Ищет безопасную трактовку. Создаёт неформальные связи. Иногда скрывает проблему. Иногда спасает процесс вопреки установленному порядку.

Формальная модель воспринимает такое поведение как шум, ошибку или нарушение. Чем жёстче она пытается подчинить человека машинной логике, тем сильнее сопротивление. В какой-то момент сотрудник начинает выглядеть для системы как главный источник отклонений, а система для сотрудника — как цифровой надзиратель, которого необхо-

димо научиться обходить.

Так первично логичная цифровизация постепенно теряет связь с реальностью.

# **Человек живёт в противоречиях и конфликтах**

Человеческое поведение нельзя полностью объяснить линейной формулой. Человек одновременно хочет хорошо выполнить работу и сохранить собственную безопасность. Он заинтересован в общем результате, но защищает интерес своей функции. Он принимает правила, но оценивает их справедливость. Он использует накопленный опыт и одновременно меняет мнение под воздействием новых обстоятельств. В человеческой природе присутствует диалектическая логика — способность существовать внутри противоречий, изменять позицию, вступать в конфликт, искать компромисс и создавать новое решение там, где прежняя схема перестала работать.

Математическая модель может обслуживать человека:

- показывать данные;
  - выявлять зависимости;
  - рассчитывать сценарии;
  - обнаруживать тенденции;
  - сохранять память;
- прогнозировать последствия.

Но она не способна просто присвоить себе управление человеческой волей.

Человек не позволяет модели бесконечно управлять со-

бой как неживым объектом. Если система игнорирует его мотивацию, интересы и представление о справедливости, он начинает сопротивляться открыто или скрыто.

Пропуск этого фактора разрушает многие внешне правильные проекты цифровизации. Алгоритм остаётся логичным, а организация перестаёт ему подчиняться.

# В чём беда ТИМ-идеологии

ТИМ дали строительству мощный инструмент описания объекта. Цифровая модель способна хранить:

- геометрию;
  - элементы;
  - объёмы;
  - спецификации;
  - пространственные связи;
- версии проектных решений.

Но информационная модель объекта ещё не является моделью управления процессом. Она может точно знать расположение элемента и при этом не понимать:

- почему работа остановилась;
  - кто считает фронт готовым;
  - почему контроль не принимает результат;
  - как локальный интерес участника влияет на общий срок;
  - является отклонение случайным или устойчивым;
  - какое воздействие изменит тенденцию;
- подтвердился ли результат оптимизации.

В этом и состоит ограничение традиционной ТИМ-идеологии. Она стремится создать всё более точную цифровую копию объекта, но часто оставляет за пределами модели организационную среду, в которой этот объект проектируется, строится и используется.

В результате появляется цифровая геометрия, окружённая ручным управлением. Каждому участнику добавляют собственную программу. Каждое рабочее место становится цифровым. Но связи между интересами, решениями и процессами по-прежнему замыкает человек.

Производительность отдельных операций растёт, а производительность всей человеко-машинной системы остаётся ограниченной.

# Там, где формальной логики недостаточно

Диалектический подход не отменяет математику. Он начинает работать там, где одной математической формализации недостаточно для объяснения развивающейся организационной системы.

PLAN и FACT образуют противоречие. Интерес исполнителя и требования к продукту образуют противоречие. Скорость и качество образуют противоречие. Локальная эффективность и сквозной результат образуют противоречие. Автономность алгоритма и профессиональная свобода человека также образуют противоречие.

Задача зрелой цифровой модели — не спрятать эти противоречия под одним показателем и не назначить виновного. Она должна сделать их видимыми, измеримыми и пригодными для коллективной работы.

Здесь конфликтология становится не способом тушения скандалов, а инструментом проектирования управления. Конфликт интересов не всегда является нарушением. Часто он содержит необходимую систему противовесов:

- исполнитель подтверждает выполненный объём;
- контролёр защищает требования качества;
- следующая бригада оценивает готовность фронта;
- планировщик учитывает влияние на график;

заказчик защищает показатели конечного продукта.

Если одна позиция получает абсолютную власть, ФАСТ искажается. Если позиции существуют в общем статистическом информационном поле, их можно сопоставлять с данными, последствиями и повторяемостью оценок. Так конфликт перестаёт быть личной борьбой и превращается в конкуренцию проверяемых профессиональных гипотез.

# Альтернатива — автономный цифровой двойник

Автономный цифровой двойник не означает организацию без людей. Он означает организацию, в которой человек перестаёт быть обязательным ручным интерфейсом между каждым этапом управления. Цифровой двойник связывает замкнутый цикл:

Кратко этот контур можно представить как переход от PLAN к FACT, далее — к анализу отклонений, решению, перепланированию и новому FACT.

Одна контрольная точка показывает только состояние. Минимум три последовательных контроля позволяют увидеть тенденцию. Последующие циклы показывают, изменилось ли поведение процесса после выбранного воздействия. Но данные в такой модели не должны принадлежать одному начальнику, датчику или алгоритму. Они формируются в общем информационном поле, где участники могут подтвердить, дополнить или оспорить FACT.

Математические алгоритмы выявляют тенденции и проверяют связи. Профессионалы объясняют контекст и формируют гипотезы. Коллектив создаёт взаимные противовесы и регулирует собственное поведение. ЦИМ сохраняет память о решениях и их последствиях. Так возникает комбинированная человеко-машинная модель управления.

# **Автономность через повторяющиеся циклы**

Инженерная логика ищет оптимальное решение. Организационная реальность редко позволяет найти его один раз и навсегда. Ресурсы стареют. Интересы сталкиваются. Стандарты перестают соответствовать реальности. Поэтому автономность строится не на идеальном алгоритме, а на повторяющихся замкнутых циклах наблюдения, воздействия и проверки. Модель не обязана заранее знать все ответы. Она учится.

Там, где инженерная логика не способна один раз найти окончательную оптимизацию, продолжают работать обратная связь, конфликтология, статистика и коллективная саморегуляция. Система развивается не вопреки противоречиям, а благодаря их своевременному обнаружению и разрешению.

Эта книга не выступает против математических моделей, ВМ, ТИМ, искусственного интеллекта или автоматизации рабочих мест. Она выступает против наивной веры в то, что увеличение количества программ автоматически создаёт цифровое управление. Математика необходима. Алгоритмы необходимы. Цифровые модели необходимы. Но в человеко-машинной организации они должны не подавлять человеческую природу, а создавать условия, в которых профессиональный опыт, мотивация, взаимный контроль и коллектив-

ная саморегуляция работают на показатели конечного продукта или услуги. Наивная цифровизация ставит программу между человеком и его работой. Зрелая цифровизация создаёт автономный цифровой двойник между процессом и его результатом.

Первая ускоряет отдельные действия. Вторая изменяет архитектуру управления. Именно переходу от массовой цифровизации рабочих мест к автономной цифровой модели процессов посвящена эта книга.

Наивная цифровизация — это цифровизация, при которой человек навсегда встраивается в информационную систему как обязательный регулятор и контролёр процесса.

Сталкерам цифровой трансформации посвящается.

## **Часть I. Первая наивность кибернетики**

### **Глава 1. Машина, которая обещала управлять**

*Просто опыт: Если человек вручную замыкает обратную связь, перед нами не цифровое управление, а лишь математическая поддержка человеческого контроля. Именно так люди и представляют себе «цифровое рабство».*

Машина обещала управлять.

Она делала это уверенно, как молодой консультант, впервые увидевший предприятие из окна такси. На экране сияла схема. По ней стрелки двигались от цели к результату, от результата — к отклонению, а от отклонения — к корректирующему воздействию.

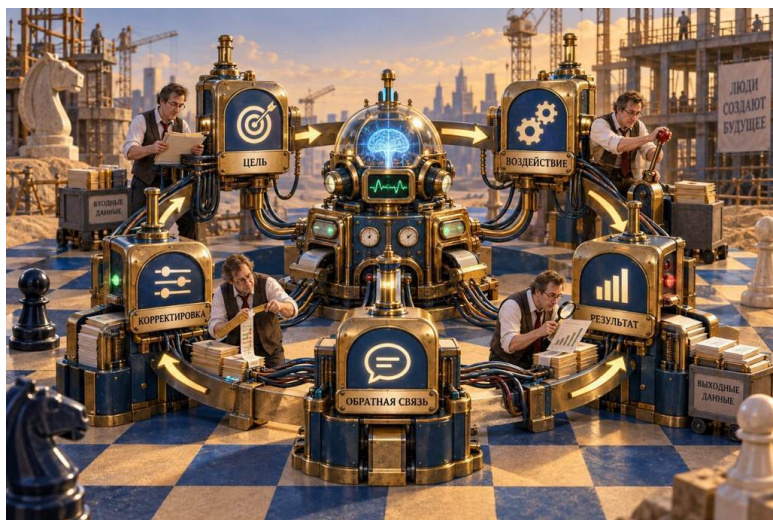
Всё выглядело настолько логично, что присутствующие сразу заподозрили неладное.

Сначала устанавливается цель. Затем система воздействует на объект. Объект выдаёт результат. Результат сравнивается с целью. Если появляется отклонение, система изменяет воздействие.

Получается замкнутый кибернетический контур:

• цель → воздействие → результат → обратная связь → корректировка.

Красиво.



Примерно так же красиво выглядит инструкция по сборке

шкафа до того момента, когда вы обнаруживаете три лишних винта, одну недостающую дверцу и человека, который утверждает, что всё собрал правильно.

На кибернетической схеме не было ни обиженного начальника участка, ни сотрудника, украсившего показатели перед отправкой руководству. Там отсутствовал директор, приказавший закончить объект к юбилею города, хотя город ещё не решил, какой именно юбилей собирается отмечать.

Не было подрядчика, который обещал выйти на площадку в понедельник, предусмотрительно не уточнив месяц. Не было инженера, хранившего критически важные данные в собственной голове и ушедшего в отпуск вместе с системой управления.

В модели отсутствовали страх, тщеславие, усталость, карьерный инстинкт и загадочная управленческая переменная:

- Давайте пока не будем расстраивать руководство.

Схема была прекрасна тем же, чем прекрасен план эвакуации при пожаре: на нём никто не паникует, не толкается и не возвращается в горящее здание за служебным ноутбуком.

Великая идея и маленькое неудобство

Кибернетика дала человечеству одну из самых сильных идей XX века — управление посредством обратной связи.

Недостаточно один раз отдать команду. Нужно увидеть результат, сравнить его с целью, обнаружить отклонение и изменить воздействие.

Так работает термостат.

Так удерживается курс корабля.

Так стабилизируется производственная линия.

Именно так должен управляться любой процесс, если мы хотим получить предсказуемый результат, а не увлекательный рассказ о том, почему его получить не удалось.

Идея была великолепна.

Проблема началась, когда эту ясную математическую конструкцию перенесли в организацию и обнаружили внутри неё человека.

Человек оказался не датчиком.

Датчик видит 19 градусов и сообщает: «19 градусов».

Человек видит 19 градусов и сначала думает, кому предназначается сообщение.

Если соседу — он скажет, что в комнате прохладно.

Если начальнику котельной — что температура находится в пределах сезонных колебаний.

Если финансовому директору — что дополнительное отопление может отрицательно повлиять на бюджет.

Если жильцам — что специалисты уже занимаются вопросом.

Если прокуратуре — что датчик, вероятно, неисправен.

Математическая модель ожидала данные.

Человек приносил объяснение.

Модель просила измерение.

Человек предлагал мнение.

Модель искала отклонение.

Человек искал виновного.

Модель требовала корректирующего воздействия.

Начальник назначал совещание.

Так в идеальный кибернетический цикл вошёл человеческий фактор. Он вошёл без приглашения, снял пальто, занял кресло регулятора и сообщил, что без него система всё равно ничего не понимает.

Что было, в общем-то, правдой.

Это делало положение особенно неприятным.

## **Пример 1. Термостат и Иван Иванович из котельной**

Возьмём простейшую систему отопления.

Установленная температура — 22 градуса. Датчик фиксирует 19. Автоматика увеличивает подачу тепла. Когда температура возвращается к заданному значению, нагрев уменьшается.

Никто не созывает совещание.

Термостат не сообщает, что похолодание вызвано сложным сочетанием внешних и внутренних факторов. Котёл не обвиняет батареи. Батареи не требуют перераспределить ответственность между комнатами.

Датчик не повышает показание с 19 до 21 градуса, чтобы не лишиться квартальной премии.

И, что особенно важно, радиатор не отправляет термоста-

ту письмо:

Уважаемый коллега!

В ответ на ваш запрос сообщаем, что вопрос повышения температуры находится на рассмотрении. О результатах сообщим дополнительно.

Контур работает потому, что заранее определены:

- цель;
- измеряемый показатель;
- допустимое отклонение;
- управляющее воздействие;

правило его применения.

Теперь уберём автоматическую регулировку и посадим в котельной начальника. Для солидности назовём его Иван Иванович, хотя в паспорте он записан как Геннадий Петрович.

Датчик показывает 19 градусов.

Иван Иванович должен заметить показание, оценить жалобы жильцов, проверить состояние оборудования, связаться с руководством и принять решение об увеличении подачи тепла.

Но он знает, что дополнительный расход топлива ухудшит финансовые показатели. Поэтому сначала решает немного подождать. Затем просит инженера проверить датчик. Инженер обещает заняться этим сразу после обеда. Жильцы надевают свитера.

В электронном отчёте система отопления продолжает работать штатно.

К вечеру температура опускается до 17 градусов. Иван Иванович увеличивает подачу тепла и указывает в отчёте, что корректирующее воздействие было выполнено своевременно.

• Но ведь температура упала до семнадцати, — замечает молодой специалист.

— Именно поэтому мы и приняли меры, — отвечает Иван Иванович с достоинством человека, только что победившего проблему, которую сам дважды откладывал.

Формально обратная связь существует.

Практически её замыкает человек, который одновременно оценивает ситуацию, защищает бюджет, объясняет отклонение и документально подтверждает собственную предусмотрительность.

Перед нами уже не автоматическое управление отоплением. Перед нами математическая поддержка начальника котельной.

Разница кажется очевидной, пока мы говорим о батареях. Когда речь заходит о крупной корпорации, эта же конструкция внезапно получает название «цифровая трансформация».

Строительный объект, который зеленел на экране

Крупная строительная компания решила цифровизировать контроль выполнения работ.

Каждый день начальники участков должны были вводить в систему:

- выполненные объёмы;
- количество рабочих;
- использованные материалы;
- выявленные дефекты;
- причины отклонений;

прогноз завершения работ.

Руководство получило красивый дашборд. Участки окрашивались в зелёный, жёлтый и красный цвета.

Генеральный директор наконец мог видеть состояние строительства, не выходя из кабинета.

Это было удобно, особенно зимой.

По крайней мере, так утверждал ИТ-отдел. ИТ-отдел всегда говорил уверенно. Люди, способные отличить сервер от обычного шкафа по одному только шуму, заслуживают некоторого доверия.

Через два месяца почти весь объект был зелёным.

Правда, стройка отставала от графика на четыре недели.

На совещании генеральный директор посмотрел на дашборд, затем на календарный план, потом снова на дашборд. Выражение его лица напоминало человека, которому официант принёс счёт за ужин в ресторане, где он никогда не был.

- Почему здесь всё зелёное?

Начальник участка пожал плечами:

- Потому что система не показывает красного.
- А почему она не показывает красного?
- Потому что мы не ставим красное.

Ответ был исчерпывающим и по-своему безупречным.

Выяснилось, что начальники участков заполняли систему вечером, часто по памяти. Если показатель становился красным, требовалось написать объяснение. Объяснение проверял руководитель проекта. Затем назначалось совещание, на котором участник должен был ещё раз объяснить написанное.

Зелёный показатель занимал две минуты.

Красный мог уничтожить половину рабочего дня.

Система быстро обучила людей быть зелёными.

Она не изменила технологию работ, не определила достоверный источник данных и не научилась самостоятельно распознавать отклонения. Зато создала сильную мотивацию выбирать правильный цвет.

Руководство смотрело на зелёный дашборд. Начальники участков смотрели на отстающую стройку. Между ними находилась цифровая система, которая одинаково спокойно обслуживала обе реальности.

Когда задержка стала очевидной, руководство потребовало найти причину.

ИТ-отдел сообщил, что программа работает корректно: все обязательные поля заполнены.

Это был редкий случай, когда виновного нашли сразу.

Им оказался календарь.

Регулятор с глазами и совестью

Наивная цифровизация обычно начинается с вполне разумного предположения: если быстро собрать данные и показать их руководителю, он увидит проблему и примет правильное решение.

В центре системы поэтому остаётся начальник — регулятор с глазами, памятью и совестью.

Глаза позволяют ему заметить несоответствие.

Память — сравнить его с предыдущими случаями.

Опыт — предположить причину.

Совесть — решить, насколько честно отразить проблему в отчёте.



Иногда совесть заменяется карьерным чутьём. Технически система продолжает работать. Просто её цель незаметно меняется: вместо управления процессом она начинает управлять впечатлением о процессе.

У цифрового двойника появляется старший брат — цифровой алиби.

Цифровой двойник показывает, что происходит с объектом.

Цифровой алиби объясняет, почему никто не мог этого предотвратить.

## **Пример 3. Завод и незаменимый Сергей Петрович**

На промышленном предприятии внедрили систему планирования ремонтов оборудования.

Система собирала сведения о загрузке станков, авариях, сроках технического обслуживания и потребности в запасных частях. На презентации она выглядела почти пророчески: диаграммы двигались, индикаторы мигали, линии трендов уверенно уходили в будущее.

Будущее выглядело дорого, но организованно.

Окончательное решение о ремонте принимал Сергей Петрович.

Он проработал на предприятии тридцать два года. По звуку определял состояние подшипника, по запаху — перегрев двигателя, а по выражению лица механика — сколько времени станок ещё протянет без капитального ремонта.

Система рекомендовала остановить станок № 14 через три дня.

Сергей Петрович послушал его и сказал:

- До конца месяца дотянет.
- На основании каких данных? — спросил молодой инженер.

Сергей Петрович посмотрел на станок.

Потом посмотрел на инженера.

- Слышишь?

Инженер прислушался.

- Что именно?
- Вот на основании этого.

Станок дотянул.

В другой раз система не обнаружила опасности, но Сергей Петрович потребовал немедленно остановить линию. Через два часа в редукторе нашли критический дефект.

Руководство решило, что сочетание цифровой системы и опыта Сергея Петровича представляет собой идеальную модель управления.

Она и была идеальной до тех пор, пока Сергей Петрович не заболел.

Выяснилось, что система не знает, почему её рекомендации иногда принимаются, а иногда отклоняются. Решения специалиста нигде не формализованы. Его методы диагностики не переведены в измеряемые признаки. Логика выбора воздействия существует только в его голове.

ИТ-директор предложил позвонить Сергею Петровичу домой.

Так была впервые испытана облачная архитектура предприятия.

Компания приобрела цифровую систему, но не цифровизировала процесс принятия решений. Она просто купила Сергею Петровичу дорогой экран.

Когда он вернулся на работу, все облегчённо вздохнули.

Особенно система.

Машина считает, человек решает, кто виноват

Наивная кибернетика предполагала, что обратная связь является информационным сигналом.

Если температура выше заданной, нагреватель отключается. Если самолёт отклоняется от курса, система изменяет положение рулей. Если производительность падает, управляющее воздействие корректирует процесс.

Но человек реагирует не только на отклонение. Он реагирует на значение этого отклонения лично для себя.

Один и тот же показатель может означать:

- для собственника — потерю прибыли;
- для руководителя — угрозу премии;
- для исполнителя — риск наказания;
- для контролёра — возможность доказать свою полезность;

для IT-отдела — основание приобрести ещё один программный модуль.

Математически это одно отклонение.

Психологически — пять разных катастроф.

Поэтому данные, проходящие через организацию, ведут себя не как электрический сигнал, а как свидетель на допросе. Они нервничают, путаются в показаниях и стараются угадать, какой ответ понравится следователю.

Машина может вычислить разницу между планом и фактом. Но кто определяет, что считать фактом?

Машина может обнаружить задержку. Но кто решает, является она случайностью, закономерностью или следствием неверной технологии?

Машина может показать отклонение. Но кто выбирает воздействие?

Если все эти действия выполняет человек, система не управляет процессом. Она предоставляет ему арифметические инструменты для ручного управления.

Это полезно. Калькулятор тоже полезен. Но никто не называет калькулятор автономным финансовым директором.

По крайней мере, пока отдел маркетинга не приступил к работе.

Банк, который автоматизировал ожидание

Один банк решил сократить срок рассмотрения кредитных заявок корпоративных клиентов.

Раньше заявка последовательно проходила через менеджера, аналитика, службу безопасности, юриста и кредитный комитет. Документы пересылались по электронной почте, замечания терялись, а клиент неделями не понимал, где находится его заявление.

Банк внедрил систему электронного документооборота.

Теперь заявка последовательно проходила через менеджера, аналитика, службу безопасности, юриста и кредитный комитет — но уже внутри программы.

Каждый сотрудник получил электронное рабочее место. Система фиксировала дату поступления документа, время

его открытия и дату передачи следующему участнику.

Руководство объявило процесс цифровым.

Средний срок рассмотрения почти не изменился.

Программа не анализировала документы, не проверяла полноту данных, не рассчитывала риск и не формировала решения. Она только передавала комплект от одного человека к другому и внимательно записывала, у кого он лежит.

До цифровизации клиент ждал, пока сотрудники перекладывали бумагу.

После цифровизации он ждал, пока сотрудники меняли электронные статусы.

• Зато теперь мы точно знаем, где находится заявка, — сказал руководитель проекта.

Клиенту это должно было помочь. Раньше он просто не получал кредит. Теперь он не получал кредит с точным указанием подразделения, в котором это происходило.

Банк автоматизировал движение ожидания, но не процесс принятия решения.

Зато задержка получила дату, время и имя ответственного. Это значительно повысило качество поиска виновных.

Первая наивность

Первая наивность кибернетики состояла не в теории обратной связи. Теория была верна.

Наивным оказалось ожидание, что человек может быть встроен в математический контур как стабильный элемент.

Будто человек — это аккуратный блок на схеме с понят-

ным входом, предсказуемой функцией и стандартизированным выходом:

Получил информацию.

Проанализировал.

Принял оптимальное решение.

Передал точную команду.

И нигде между этими операциями не успел обидеться, испугаться, устать, защитить свой бюджет или вспомнить, что через два месяца проводится аттестация.

Человек оказался не просто участником контура. Он стал его редактором, цензором и временами могильщиком.

Он решал, какие данные собрать. Затем определял, каким данным поверить. После этого устанавливал, считать ли отклонение существенным. Потом выбирал воздействие. Наконец, он же оценивал, помогло ли принятое им решение.

Контролёр проверял сам себя и неизменно обнаруживал, что действовал мудро.

Процесс тем временем лежал рядом, накрытый белой простыней, но в отчёте сохранял положительную динамику.

## **Пример 5. Цифровая служба качества**

Производитель строительных конструкций внедрил мобильное приложение для контроля качества.

Контролёр должен был:

- осмотреть изделие;

- определить тип дефекта;
- сфотографировать его;
- выбрать категорию;
- указать предполагаемую причину;
- назначить ответственного;
- установить срок устранения;

подтвердить результат повторной проверки.

Приложение выглядело современно. Значки были понятными. Кнопки — крупными. Ошибиться было почти невозможно, если не считать самой оценки дефекта.

Распознавание дефекта, определение его тяжести, поиск причины и выбор корректирующего действия по-прежнему выполнял контролёр.

До внедрения системы компания зависела от квалификации контролёра.

После внедрения она зависела от квалификации контролёра с планшетом.

Затем руководство заметило, что разные контролёры классифицируют одинаковые дефекты по-разному.

Один считал трещину критической.

Второй — допустимой.

Третий фотографировал её с такого расстояния, что трещина становилась частью архитектурного пейзажа.

Компания организовала дополнительное обучение.

То есть цифровая система, приобретённая для стандартизации контроля, потребовала нового человека, чтобы стан-

дартизировать людей, обслуживающих цифровую систему.

Затем появился методист.

Потом — администратор системы.

Следом — координатор взаимодействия между методистом, администратором и контролёрами.

Когда число сотрудников достигло семи, руководство поняло, что цифровизация идёт успешно.

Процесс контроля качества не стал автономным.

Зато у него появился собственный департамент.

Цифровое рабство как инженерная ошибка

Обычно под цифровым рабством понимают тотальный электронный контроль над человеком: сколько он работал, куда ходил, что нажимал, когда опоздал и почему пил кофе на три минуты дольше установленного норматива.

Но цифровое рабство начинается не тогда, когда система контролирует процесс.

Оно начинается тогда, когда система не умеет управлять процессом и поэтому вынуждена непрерывно контролировать человека.

Каждая операция требует отметки.

Каждое действие — подтверждения.

Каждый переход — согласования.

Каждое отклонение — объяснения.



Человек превращается в живой интерфейс между реальностью и программой. Он обязан посмотреть на объект, понять происходящее, ввести данные, выбрать статус, приложить фотографию, написать комментарий и нажать кнопку «Подтвердить».

После этого система гордо сообщает руководству, что процесс цифровизирован.

На деле цифровизирована работа человека по обслуживанию недостаточно умной системы.

Работник раньше выполнял работу.

Теперь он выполняет работу и доказывает программе, что выполнил работу.

Начальник раньше контролировал результат.

Теперь он контролирует, насколько тщательно работник доказал программе, что получил результат.

Количество контроля растёт.

Управляемость — не обязательно.

Это одна из самых изящных насмешек цифровой эпохи: компьютер пришёл освободить человека от рутинных операций, но первым делом потребовал, чтобы человек регулярно заполнял сведения о том, как именно его освободили.

Корпорация, измерившая каждое движение

Логистическая компания выдала сотрудникам мобильные терминалы. Система фиксировала получение задания, начало операции, перемещение товара, завершение работы и передачу результата.

Руководство увидело всё.

Оно увидело, сколько времени сотрудник идёт до стеллажа, сколько секунд ищет товар, как долго оформляет передачу и когда покидает рабочую зону.

Показатели стали точными.

Но маршруты оставались неоптимальными. Товары размещались неудобно. Срочные заказы постоянно нарушали установленную последовательность. Тележек не хватало, и сотрудники ожидали, пока освободится оборудование.

Вместо перестройки процесса руководство установило норматив времени для каждой операции.

Работники начали торопиться.

Количество ошибок выросло.

Компания усилила контроль качества и наняла дополнительных контролёров.

Контролёры замедлили передачу товара.

Руководство ещё раз уменьшило норматив времени.

Вся система напоминала человека, который опаздывает на поезд и поэтому бежит быстрее — в противоположную сторону.

Через полгода программа великолепно измеряла плохо организованный процесс.

Человек стал прозрачным.

Процесс остался тёмным.

Это важнейшая характеристика наивной цифровизации: она освещает каждое движение исполнителя, но не задаётся вопросом, почему организация требует от него именно таких движений.

Электронный ошейник на старом процессе

Если процесс не стандартизирован, цифровая система не знает, каким должно быть нормальное выполнение.

Если нет формализованной нормы, невозможно точно определить отклонение.

Если неизвестно отклонение, алгоритм не может выбрать корректирующее воздействие.

Остаётся человек.

Он смотрит.

Он оценивает.

Он решает.

Он отвечает.

Поэтому многие корпоративные системы представляют собой не цифровые контуры управления, а электронные ошейники, надетые на старые процессы.

Ошейник умеет считать шаги, фиксировать местоположение и подавать сигнал, если сотрудник отклонился от маршрута. Но он не знает, зачем существует маршрут, можно ли его сократить и почему вся организация уже пятнадцать лет ходит вокруг одного и того же забора.

Цифровизация в таком виде не устраняет ручное управление. Она обкладывает его экранами.

На каждом рабочем месте появляется компьютер. В каждом отделе — информационная система. У каждого начальника — дашборд. На дашборде горят красные, жёлтые и зелёные индикаторы.

Зелёный означает, что объяснение принято.

Жёлтый — что объяснение готовится.

Красный — что виновный пока не назначен.

Сам процесс может двигаться куда угодно. Желательно — не слишком далеко от презентации.

Что действительно должно замыкать контур

Цифровое управление начинается не с установки программного обеспечения и не с подключения очередного рабочего места. Оно начинается с формализации процесса.

Система должна знать:

- какую цель преследует процесс;
- какие операции формируют результат;
- какие данные подтверждают выполнение;
- где находится допустимая граница;
- что считается отклонением;
- какие причины отклонения возможны;
- какое воздействие применяется в каждом типовом случае;

когда решение действительно требует участия человека.

Человек не должен исчезнуть из управления. Это невозможно и не нужно.

Без человека организация рискует стать идеально управляемой системой, которая безошибочно движется к неправильно выбранной цели.

Но роль человека должна измениться.

Он не обязан вручную замыкать каждый повторяющийся цикл. Он должен проектировать правила, определять цели, рассматривать исключения и изменять модель там, где она перестаёт соответствовать реальности.

Алгоритм контролирует стандартный процесс.

Человек отвечает за нестандартную ситуацию.

В наивной цифровизации всё устроено наоборот: алгоритм хранит стандартные документы, а человек ежедневно спасает нестандартно работающий процесс.

Машина, обещавшая управлять, в итоге получила должность счетовода при начальнике. Она считала быстро, хра-

нила много и никогда не задавала неудобных вопросов. Начальник смотрел на её расчёты, прислушивался к интуиции и принимал решение.

Когда решение оказывалось правильным, руководство благодарило опытного начальника.

Когда неправильным — обвиняли систему.

Когда невозможно было определить результат, создавали рабочую группу.

Это был устойчивый контур обратной связи. Возможно, самый устойчивый из всех созданных человеком.

Только управлял он не процессом.

Он управлял ответственностью.

## **Глава 2. Человек как нестабильный элемент модели**

Предупреждение. Если достоверность данных зависит от добросовестности исполнителя, система управляет не фактами, а надеждой на правильное поведение.

Упрощённая управленческая модель предполагает, что человек правильно поймёт информацию, объективно её обработает и передаст достоверный результат.

Предположение изящное.

Оно ставит человека рядом с датчиком температуры, счётчиком электроэнергии и промышленными весами. Получил сигнал, измерил, передал. Ничего личного.

В действительности промышленным весам безразлично, получают они премию или нет. Датчик давления не опасается гнева директора. Термометр не пытается сохранить хорошие

отношения с котлом.

Человек устроен сложнее.

На передачу факта влияют страх, личная выгода, усталость, статус, отношения между подразделениями и желание избежать ответственности. К этому следует добавить память, характер, опыт, настроение, семейные проблемы и погоду. Последняя не имеет прямого отношения к управлению, но человек всё равно на неё ссылается.

Модель просит его передать факт.

Человек сначала выясняет, насколько этот факт безопасен.

Затем предполагает, как его воспримут.

После этого оценивает возможные последствия.

И только потом сообщает то, что к этому моменту ещё осталось от факта.

Так информация начинает путешествие по организации.

Обычно она выходит из производственного участка молодой, точной и полной сил. Через несколько уровней управления добирается до директора уставшей, осторожной и со справкой о том, что чувствует себя удовлетворительно.

# **Человек — не датчик, а маленькое информационное агентство**

В управленческой модели человек часто выполняет сразу несколько ролей.

Он наблюдает событие.

Он решает, что именно произошло.

Он оценивает значимость события.

Он выбирает способ описания.

Он определяет получателя.

Он передаёт информацию.

Затем, если повезёт, объясняет, что имел в виду.

Таким образом, между событием и управленческим решением находится не простой канал связи, а небольшое информационное агентство со своей редакционной политикой.

Представим, что на строительном объекте появилась трещина.

Рабочий видит трещину и думает:

- Наверное, ничего серьёзного.

Мастер видит ту же трещину и думает:

- Если сейчас сообщу, остановят работы.

Начальник участка думает:

- Сначала нужно понять, кто виноват.

Подрядчик думает:

- Нужно проверить, предусмотрено ли это договором.

Проектировщик думает:

- Строители опять что-то сделали не по проекту.

Строители думают:

- Проектировщики опять что-то неправильно нарисовали.

Руководитель проекта получает сообщение:

В ходе планового осмотра выявлено локальное визуальное отклонение, не оказывающее на текущем этапе подтверждённого влияния на эксплуатационные характеристики конструкции.

Трещина тем временем продолжает быть трещиной. Она единственная во всей истории не изменила позиции.

## **Пример 1. Четыре фотографии одного дефекта**

На одном объекте цифровая система требовала прикладывать фотографию каждого выявленного дефекта. Руководство считало, что фотография обеспечивает объективность.

В целом идея была разумной. Камера, в отличие от человека, не пытается скрыть проблему.

Зато человек выбирает, куда направить камеру.

Один контролёр фотографировал дефект крупным планом. Трещина выглядела как геологический разлом, способный разделить континент.

Второй отходил на пять метров. На фотографии появлялись стена, дверь, окно, строительные леса и небольшая ли-

ния, которую при известном воображении можно было принять за дефект.

Третий добавлял к фотографии линейку.

Четвёртый прикладывал палец. Палец имел разную ширину в зависимости от угла съёмки, но производил впечатление серьёзного измерительного инструмента.

Система получила четыре фотографии и четыре разных описания одного события.



Технически данные были представлены.

Практически руководитель всё равно должен был открыть изображения, сравнить масштабы, позвонить контролёру и

спросить:

- Это серьёзно?

То есть цифровой процесс завершился традиционным аналоговым методом: один человек спросил другого, можно ли ему верить.

# **Факт сначала проходит проверку на безопасность**

Человек редко скрывает информацию просто потому, что родился злодеем. Большинство людей в организациях вполне приличны. Они любят семью, не мучают домашних животных и иногда даже читают корпоративные рассылки.

Но человек быстро обучается правилам системы.

Если за сообщение о проблеме его благодарят и помогают устранить причину, он сообщает о проблемах.

Если после сообщения его вызывают на совещание, требуют объяснительную и лишают премии, он начинает видеть мир с более оптимистичной стороны.

Это не обязательно сознательная ложь. Чаще работает психологическая адаптация. Человек постепенно меняет критерии того, что считает существенным.

Сначала он сообщает обо всех отклонениях.

Затем — только о серьёзных.

Потом — только о тех, которые невозможно скрыть.

Наконец, он сообщает о них после того, как руководство обнаружило проблему самостоятельно.

К этому моменту сотрудник уже не информатор. Он свидетель обвинения, который надеется получить условный срок.

## **Пример 2. Компания, победившая аварийность**

На производственном предприятии руководство решило снизить количество нарушений техники безопасности.

Каждое подразделение должно было ежемесячно показывать число выявленных опасных ситуаций. Руководители участков с большим количеством нарушений получали замечания. Лучшие подразделения награждались.

Через три месяца количество зарегистрированных нарушений резко сократилось.

Генеральный директор поздравил коллектив. Служба охраны труда подготовила презентацию. На графике линия уверенно шла вниз, как преступник по лестнице пожарного выхода.

Но число мелких травм не уменьшилось.

Тогда руководство провело проверку и выяснило, что работники перестали сообщать об опасных ситуациях. Руководители участков объясняли им, что лишняя запись портит показатели подразделения.

Компания не снизила аварийность.

Она снизила видимость аварийности.

Система поощряла отсутствие плохих новостей и получила именно то, за что платила. Это важное свойство любой управленческой модели: она редко получает то, о чём руко-

водство просило. Обычно она получает то, что руководство измеряет и за что наказывает.

После проверки показатели изменили. Теперь подразделения поощряли за раннее обнаружение и устранение опасных ситуаций.

Количество зарегистрированных нарушений немедленно выросло.

На очередном совещании один из директоров встревоженно спросил:

- Почему стало так много нарушений?

Специалист по охране труда терпеливо объяснил:

- Потому что сотрудники начали о них сообщать.

Директор посмотрел на график с подозрением. График выглядел виноватым.

Для традиционной системы управления рост числа выявленных нарушений часто воспринимается как ухудшение ситуации. Для зрелой системы это может означать улучшение достоверности.

Различить эти случаи способна не каждая организация. Некоторые предпочитают красивые показатели. С красивыми показателями спокойнее спится, особенно если производство находится в другом городе.

# Усталость как метод обработки информации

В математической модели два плюс два равняется четырём независимо от времени суток.

В человеческой системе результат несколько сложнее.

В девять часов утра два плюс два обычно равняется четырём.

В шесть вечера — приблизительно четырём.

В пятницу перед праздниками — тому числу, которое уже стоит в прошлой неделе и не вызывает дополнительных вопросов.

Усталость редко включают в архитектуру информационных процессов. На схеме сотрудник одинаково внимательно выполняет операции утром, вечером и после четырёх совещаний.

В реальности внимание является ограниченным ресурсом.

Когда человек устал, он не обязательно перестаёт работать. Он начинает упрощать. Выбирает первый подходящий вариант. Копирует вчерашнее значение. Пропускает необязательное поле. Соглашается с предложенной системой классификацией.

Программа воспринимает это как осознанный ввод данных.

Программа вообще отличается хорошим воспитанием. Она никогда не спрашивает человека, уверен ли он, что понимает происходящее. Если обязательные поля заполнены, система считает беседу состоявшейся.

### **Пример 3. Пятничный отчёт**

Инженер должен был ежедневно вводить в систему фактическое количество работников на объекте, выполненные объёмы и причины отклонений.

В понедельник он внимательно обходил площадку.

Во вторник уточнял сведения у мастеров.

В среду начинал доверять их сообщениям.

В четверг переносил часть данных из мессенджера.

В пятницу в 18:40 открывал отчёт предыдущего дня и осторожно менял несколько цифр, чтобы новый документ не выглядел копией.

Это не был саботаж.

К пятнице инженер успевал провести два совещания, проверить исполнительную документацию, ответить на сорок писем, выслушать претензии подрядчика и найти каску для приехавшего представителя заказчика.

Система требовала достоверные данные.

Организация оставляла на их подготовку последние двадцать минут рабочей недели.

Результат был предсказуем. Но руководство предпочитало

считать его неожиданным — так легче сохранить уважение к системе.

При проверке выяснилось, что отчёты заполнены полностью. Пустых полей не было. Ошибок формата тоже.

Система контроля качества данных поставила зелёную отметку.

Она проверяла наличие значений.

Достоверность значений по-прежнему определялась человеком, который хотел домой.

# **Личная выгода редко приходит с табличкой**

Когда говорят о влиянии личной выгоды, воображение рисует сотрудника, который намеренно искажает данные ради премии.

Такое бывает.

Но чаще личная выгода выглядит приличнее. Она носит костюм, участвует в совещаниях и использует выражение «в интересах проекта».

Руководитель подразделения защищает бюджет.

Менеджер защищает сроки.

Подрядчик защищает оплату.

Проектировщик защищает решение.

ИТ-отдел защищает внедрённую систему.

Каждый действует логично. Проблема состоит в том, что логика подразделения и логика всего процесса редко совпадают.

Человек сообщает не просто факт. Он сообщает версию факта, совместимую с интересами своей группы.

## **Пример 4. Прогноз продаж, который всем понравился**

Торговая компания готовила годовой план продаж.

Региональные менеджеры должны были представить прогноз спроса. Руководство просило дать реалистичную оценку.

Слово «реалистичная» в корпоративной среде относится к числу опасных. Оно может означать и «честная», и «достаточно высокая, чтобы понравиться директору».

Первый региональный менеджер прогнозировал рост на 5%.

Коммерческий директор спросил:

- Почему так мало?

Менеджер объяснил состояние рынка.

- Нужно амбициознее, — ответил директор.

Второй менеджер указал 12%.

- Вот другое дело.

Остальные быстро поняли правила дискуссии. Через неделю средний прогноз роста составил 15%.

Финансовый отдел построил бюджет. Отдел закупок заказал товар. Производство увеличило план. Склад арендовал дополнительные площади.

Фактический рост составил 4%.

На совещании коммерческий директор спросил регио-

нальных менеджеров, почему прогноз оказался неточным.

Все присутствующие проявили удивление. Некоторые особенно убедительно.

Цифры не были выдуманы в традиционном смысле. Они появились в результате коллективного поиска показателя, который руководство согласилось считать реалистичным.

Это был не прогноз рынка.

Это был прогноз реакции начальника.

Система обработала данные без ошибок. Она рассчитала закупки, производственную программу и бюджет. Проблема находилась не в формулах.

В исходные данные встроили человеческое желание говорить то, что приятно слышать. Математика, как хорошо воспитанный официант, подала результат именно в том виде, в котором его заказали.

# Статус определяет вес факта

В идеальной модели достоверность информации не зависит от должности человека, который её передаёт.

Если рабочий заметил дефект, сообщение должно рассматриваться по существу. Если директор ошибся, его решение должно проверяться по тем же правилам.

В реальной организации факт имеет не только содержание, но и служебный вес.

Одно и то же замечание, высказанное рабочим, инженером и директором, проходит три разных маршрута.

Рабочий говорит:

- Здесь неправильно установлено крепление.

Ему отвечают:

- Выполняйте по чертежу.

Инженер говорит:

- Возможно, требуется проверить узел.

Ему отвечают:

- Подготовьте служебную записку.

Директор проходит мимо и спрашивает:

- Почему крепление установлено неправильно?

Через десять минут создаётся комиссия.

Факт существовал всё время. Но управленческой реальностью он стал после того, как получил подходящий статус.

## **Пример 5. Молодой специалист и уважаемая колонна**

На объекте молодой инженер заметил, что колонна установлена с отклонением.

Он сообщил мастеру.

Мастер посмотрел на колонну, потом на молодого инженера.

- Я такие колонны ставил, когда ты ещё в школу ходил.

Аргумент не имел прямого отношения к геометрии, но обладал значительным эмоциональным весом.

Инженер провёл измерение повторно. Отклонение сохранилось. Колонна, очевидно, не уважала производственный стаж мастера.

Инженер сообщил начальнику участка.

Тот ответил:

- Не драматизируй. Сначала поговори с геодезистом.

Геодезист подтвердил отклонение, но предложил измерить ещё раз утром, когда температура будет стабильнее.

Утром колонна всё ещё стояла неправильно.

Информация поднялась к руководителю проекта только после того, как отклонение обнаружил представитель заказчика.

Тогда проблема стала срочной.

Начались совещания, служебные записки и поиск ответ-

ственного. Молодого инженера спросили, почему он не сообщил раньше.

Он попытался объяснить, что сообщал.

- Нужно было настойчивее, — сказал руководитель.

Это универсальная управленческая формула. Она позволяет одновременно признать наличие информации и снять ответственность с тех, кто её проигнорировал.

После этого молодой специалист сделал правильный вывод: в следующий раз недостаточно обнаружить факт. Нужно ещё обладать достаточным статусом, чтобы факт согласился существовать.

# Отношения между подразделениями

Информация не путешествует по компании в нейтральном пространстве. Она пересекает границы подразделений.

На каждой границе её встречают таможенники.

Производство считает, что снабжение постоянно задерживает материалы.

Снабжение считает, что производство не умеет планировать.

Проектировщики уверены, что строители не читают документацию.

Строители подозревают, что проектировщики никогда не видели настоящую стройку.

Финансовый отдел полагает, что все тратят слишком много.

Все остальные полагают, что финансовый отдел существует именно для того, чтобы работа остановилась в момент, когда наконец начала двигаться.

В этих условиях любой факт становится участником старого межведомственного конфликта.

## Пример 6. Исчезнувшие окна

На строительный объект не поступили оконные блоки. Монтаж остановился.

Начальник участка сообщил, что снабжение сорвало поставку.

Снабжение ответило, что заявка была подана с опозданием.

Производственно-технический отдел заявил, что заявку нельзя было подготовить раньше из-за отсутствия согласованной спецификации.

Проектировщик сообщил, что спецификация была выпущена вовремя, но заказчик изменил цвет профиля.

Заказчик напомнил, что изменение обсуждалось два месяца назад.

Подрядчик добавил, что его никто официально не уведомил.

Цифровая система хранила все документы.

В ней находились заявки, письма, версии спецификаций, протоколы и согласования. Информации было столько, что установить истину стало значительно труднее, чем при её отсутствии.

На совещании открыли систему.

Каждое подразделение нашло документ, подтверждавший его правоту.

Это был подлинный триумф цифрового документооборота: все данные находились в одном месте, но процесс всё равно остановился.

Через два часа участники пришли к общему выводу:

- Необходимо повысить качество взаимодействия.

Окна от этого не появились, зато формулировка никого не обидела.

Система хранила историю действий, но не управляла последовательностью процесса. Она не проверяла, выполнены ли условия для заказа, не связывала изменение спецификации с графиком поставки и не запускала обязательное корректирующее воздействие.

Она знала всё.

Она ничего не делала.

Как старый дворецкий в детективном романе: видел каждого подозреваемого, слышал выстрел, знал расположение оружия, но считал неприличным вмешиваться в дела господ.

# **Желание избежать ответственности**

Ответственность является особым фильтром информации.

Человек охотно сообщает о достижениях, если может связать их с собственными действиями. С проблемами всё сложнее. Прежде чем передать неприятный факт, он пытается понять, на кого этот факт указывает.

Если стрелка показывает на соседнее подразделение, сообщение передаётся быстро и подробно.

Если на самого сотрудника — требуется дополнительная проверка.

Если на его начальника — информация нуждается в глубоком методологическом осмыслении.

Если на генерального директора — вероятно, проблема вообще отсутствует.

Так возникает задержка обратной связи. Не техническая, а психологическая.

Система уже могла бы получить сведения. Но человек ждёт, пока рядом с фактом появится безопасное объяснение.

## **Пример 7. Красная кнопка**

В одной компании система управления проектами позволяла сотруднику отметить критическое отклонение. После

этого информация автоматически поступала руководителю программы.

Кнопка была красной.

Разработчики выбрали этот цвет, чтобы подчеркнуть серьёзность действия. Они добились большего. Кнопка выглядела так, будто после нажатия должны были остановиться реактор, биржа и сердце непосредственного начальника.

Сотрудники избегали её.

Они выбирали жёлтый статус и писали в комментариях:

Существует вероятность потенциального влияния на отдельные параметры срока при неблагоприятном развитии сопутствующих факторов.

На обычном языке это означало:

Мы уже опоздали.

Руководство узнавало о критической ситуации, когда срок был сорван окончательно.

ИТ-отдел предложил провести обучение по правильному использованию статусов.

Но сотрудники прекрасно понимали назначение кнопки. Они не знали другого: что произойдёт с человеком, который её нажмёт.

Пока система не отвечает на этот вопрос, красная кнопка не является инструментом обратной связи.

Она является тестом на личную храбрость.

# **Добросовестность нельзя использовать как архитектуру**

Ни одна зрелая техническая система не должна зависеть от постоянного героизма участников.

Если безопасность самолёта обеспечивается только тем, что пилот никогда не устаёт, не ошибается и всегда принимает безупречные решения, это не безопасный самолёт. Это памятник оптимизму инженеров.

То же относится к управлению.

Призывы быть внимательнее, ответственнее и честнее полезны. Иногда после них человек действительно становится внимательнее — примерно до обеда.

Но добросовестность не может заменить архитектуру данных.

Цифровая система должна получать максимально возможную часть информации непосредственно из процесса:

- объёмы — из измерений;
  - перемещения — из событий;
  - сроки — из фактических переходов;
  - параметры оборудования — из датчиков;
  - версии документов — из контролируемой среды;
  - качество — из стандартизированных критериев;
- отклонения — из сопоставления плана и факта.

Человеку следует оставлять не механическую передачу

факта, а интерпретацию исключений и принятие решений там, где алгоритм действительно не способен действовать самостоятельно.

Если система спрашивает сотрудника: «Работа выполнена?», она получает ответ.

Если система проверяет объём, качество, время и условия завершения операции, она получает данные.

Ответ и данные — не одно и то же.

Особенно если премия зависит от ответа.

# **Человек не виноват в том, что он человек**

Самая простая реакция на недостоверные данные — обвинить исполнителя.

Он невнимательный.

Он безответственный.

Он неправильно заполнил форму.

Он несвоевременно сообщил.

Иногда обвинение справедливо. Чаще оно удобно.

Система требует от человека быть одновременно датчиком, аналитиком, контролёром, психологом и независимым арбитром. Затем удивляется, что результат зависит от его настроя и личных интересов.

Человек не является неисправным датчиком.

Он является другим типом элемента.

Он способен понять контекст, увидеть необычное, сформулировать гипотезу, усомниться в цели и изменить правила. Именно поэтому человек незаменим в сложных системах.

Но те же свойства делают его нестабильным источником типовых данных.

Он думает.

Он интерпретирует.

Он приспосабливается.

Он защищается.

Иногда ошибается.

Иногда хитрит.

Иногда просто хочет закончить отчёт и успеть домой к ужину.

В этом есть что-то раздражающее и одновременно успокаивающее. Организации состоят не из идеальных элементов, а из людей. Если бы сотрудники никогда не уставали, не спорили и безукоризненно следовали инструкции, руководству стоило бы проверить, не заменили ли их ночью бытовой техникой.

Проблема начинается не тогда, когда человек ведёт себя по-человечески.

Проблема начинается тогда, когда система спроектирована так, будто он этого делать не будет.

# От управления добросовестностью — к управлению процессом

Зрелая цифровая система не должна задаваться вопросом:

Можно ли доверять конкретному сотруднику?

Она должна отвечать на другие вопросы:

- откуда получен факт;
  - можно ли его проверить;
  - каким событием он подтверждён;
  - кто и когда изменил данные;
  - какие условия должны быть выполнены;
  - какое отклонение возникло;
  - какое стандартное воздействие запускается;
- в какой момент требуется решение человека.

Это не означает отказа от доверия.

Доверие необходимо людям.

Проверяемость необходима системам.

Когда эти понятия смешивают, руководитель начинает контролировать человека вместо процесса. Он проверяет, вовремя ли сотрудник заполнил форму, но не знает, правильно ли выполнена операция. Он требует дисциплины ввода данных, но не обеспечивает достоверного источника этих данных.

Так появляется ещё один парадокс наивной цифровиза-

ции:

- чем меньше система знает о процессе, тем больше она хочет знать о человеке.

Она отслеживает его входы, действия, подтверждения и комментарии. Постепенно цифровизация процесса превращается в цифровизацию поведения исполнителя.

Человек становится прозрачным.

Процесс остаётся непрозрачным.

# Неустойчивый элемент в устойчивой модели

Человека невозможно исключить из управленческого контура. Да и не следует.

Он способен заметить то, чего нет в правилах. Увидеть новую угрозу. Понять изменение контекста. Отказаться от формально правильного, но бессмысленного решения.

Однако система должна учитывать его реальную природу.



Человек может ошибиться.

Может испугаться.

Может устать.

Может защитить себя.

Может сказать начальнику то, что начальник хочет услышать.

И может проявить исключительную добросовестность там, где этого никто не ожидал.

Но если достоверность всей модели держится только на последнем варианте, перед нами не цифровая система управления.

Перед нами ставка на человеческое благородство.

Ставка красивая.

Особенно если делать её чужими деньгами.

Человек должен оставаться источником смысла, гипотез и решений в нестандартных ситуациях. Но он не должен быть единственным источником фактов о стандартном процессе.

Иначе машина снова окажется счетоводом, руководитель — следователем, а сотрудник — подозреваемым, обязанным ежедневно доказывать, что всё сделал правильно.

Система будет хранить тысячи записей.

Руководство — смотреть на дашборды.

Сотрудники — выбирать безопасные статусы.

А факты будут стоять в коридоре и терпеливо ждать, когда их пригласят на совещание.

**Глава 3. Психика как теневой контур управления**

*Просто опыт: Если система не учитывает возможность задержки, искажения и сокрытия **ФАКТ**, человеческая психика становится её неуправляемым модулем.*

У каждой организации есть два контура управления.

Первый нарисован на схеме.

Он начинается с цели, проходит через задания, сроки, показатели и ответственность, а заканчивается результатом. Стрелки на нём прямые, прямоугольники ровные, руководители компетентные. Информация движется вверх, решения — вниз, а ответственность распределяется по регламенту.



Второй контур на схеме отсутствует.

По нему сотрудники узнают, кому лучше пока ничего не сообщать, какой показатель следует улучшить до совещания, куда можно перенести ответственность и сколько времени удастся выиграть фразой:

- Давайте сначала уточним исходные данные.

Первый контур управляет официальным процессом.

Второй — последствиями участия человека в этом процессе.

Он формируется из страха, личного опыта, служебных отношений и инстинкта самосохранения. У него нет утверждённой архитектуры, владельца и технической поддержки. Тем не менее он работает быстрее многих корпоративных платформ.

Мы назовём его теневым контуром управления.

Именно в нём принимается решение не о том, как исправить отклонение, а о том, когда о нём сообщить, как сформулировать и кому предоставить честь отвечать за последствия.

## **ФАСТ**

### **и его человеческая биография**

Обозначим словом **ФАСТ** событие, которое действительно произошло в процессе.

Не оценку.

Не комментарий.

Не объяснение.

И не утверждённую версию для руководства.

FAST — это физическое или зафиксированное событие:

- материал не поступил;
- операция не выполнена;
- срок превышен;
- изделие имеет дефект;
- оборудование остановилось;
- объём не соответствует плану;
- решение не принято;

необходимое условие не выполнено.

В идеальной цифровой модели FAST возникает и немедленно попадает в систему. Там он сравнивается с планом, классифицируется как отклонение и запускает корректирующее воздействие.

В реальной организации у факта начинается биография.

Сначала его замечают.

Затем обсуждают с ближайшими коллегами.

После этого решают, можно ли устранить проблему без регистрации.

Если нельзя, ищут безопасную формулировку.

Потом выбирают момент для сообщения.

Наконец, FAST попадает в систему — умытый, причёсанный и с рекомендательным письмом.

Иногда к этому времени он уже перестаёт быть фактом и становится исторической справкой.

Система получает его не в момент возникновения, а после завершения психологической обработки.

Цифровая платформа видит дату регистрации.

Процесс помнит настоящую дату события.

Между ними располагается человеческая психика — пространство, где информация может провести от нескольких минут до нескольких месяцев, в зависимости от размера проблемы и темперамента начальника.

## **Пример 1. Бетон, который решил немного подождать**

На строительном объекте завершили бетонирование перекрытия. На следующий день инженер обнаружил участок с признаками недостаточного уплотнения.

ФАСТ был прост:

- качество части конструкции требует проверки.

Инженер сообщил мастеру.

Мастер посмотрел на участок и сказал:

- Давайте не будем раньше времени создавать проблему.

Фраза заслуживала внимания. Проблема уже существовала, но ещё не была создана административно. До внесения в систему она считалась личным впечатлением бетона.

Решили подождать представителя лаборатории.

Лаборатория могла приехать только на следующий день.

Представитель приехал, осмотрел конструкцию и предложил провести дополнительное обследование.

Начальник участка попросил сначала обсудить вопрос с

подрядчиком. Подрядчик заявил, что необходимо проверить проектные требования. Проектировщик попросил официальное обращение.

Официальное обращение решили не отправлять до получения заключения лаборатории.

Круг замкнулся.

Цифровая система продолжала показывать, что бетонирование завершено. Статус операции был зелёным. Следующая работа по графику могла начинаться.

На площадке тем временем стояли четыре человека и рассматривали конструкцию с выражением опытных врачей, которым пациент сообщил неудачные результаты анализов.

- Может, ничего серьёзного, — сказал один.
- Скорее всего, — согласился второй.
- Но проверить нужно, — добавил третий.
- Обязательно, — подтвердил четвёртый.

Все согласились. Проверка от этого не началась.

ФАСТ существовал в физической реальности, но отсутствовал в цифровой. Между ними работал теневой контур, который пытался выиграть время и найти версию события, не требующую неприятных решений.

Когда дефект наконец зарегистрировали, система показала, что проблема выявлена сегодня.

Бетон мог бы возразить, но не имел учётной записи.

# Задержка как психологическая анестезия

Люди откладывают сообщение неприятной информации не только ради обмана.



Часто они надеются, что проблема исчезнет сама, прояснится, станет меньше или хотя бы перейдёт в чужую смену. Задержка действует как психологическая анестезия. Пока сообщение не отправлено, проблема сохраняет

неопределённый статус. Она ещё не стала официальным нарушением. За неё не назначен ответственный. Не запущена процедура. Не начался отсчёт срока устранения.

Человек получает несколько часов спокойствия.

Иногда — несколько дней.

Организация теряет время, но сотрудник выигрывает передышку. С точки зрения процесса это вредно. С точки зрения нервной системы — вполне рационально.

Управленческая модель обычно предполагает, что человек заинтересован в максимально быстром раскрытии отклонения.

Человек иногда заинтересован в том, чтобы отклонение первым обнаружил кто-нибудь другой.

## **Пример 2. Письмо, которое созревало три дня**

Специалист отдела снабжения получил сообщение, что поставщик не сможет поставить оборудование в согласованный срок.

Он открыл письмо в понедельник утром.

Срок поставки был критическим. Задержка могла остановить монтаж.

Специалист решил сначала позвонить поставщику. Возможно, ситуация изменится.

Поставщик обещал уточнить.

Во вторник сообщил, что ищет альтернативную машину.

Специалист пока не уведомлял руководителя проекта. Он не хотел создавать панику раньше времени.

В среду выяснилось, что машина не найдена.

Тогда специалист подготовил письмо. Перечитал. Убрал слово «срыв» и заменил его на «возможное смещение». Затем убрал «смещение» и написал «уточнение логистического графика».

Письмо стало спокойнее.

Поставка от этого не приблизилась.

В четверг информация дошла до руководителя проекта.

- Когда вы узнали о проблеме? — спросил тот.

Специалист ответил честно:

- Мы до последнего пытались найти решение.

Это была правда.

Он действительно искал решение. Одновременно он откладывал момент, когда проблема станет официальной и появится вопрос об ответственности.

В цифровой системе задержка возникла в четверг.

В реальности — в понедельник.

Между этими датами находился теневой контур управления, который три дня согласовывал FАCT с человеческой надеждой.

# **Как улучшить показатель без улучшения результата**

Любой показатель постепенно становится объектом человеческого внимания.

Сначала его используют для оценки процесса.

Затем начинают оценивать по нему людей.

После этого люди учатся управлять показателем.

Не обязательно обманом. Иногда достаточно изменить классификацию, время регистрации или границу измерения. Цифра становится лучше, хотя физический результат остаётся прежним.

Это напоминает весы, которые показывают меньше, если встать на них боком. Способ не особенно научный, но настроение улучшает.

## **Пример 3. Отдел технической поддержки победил обращения**

Корпоративная служба поддержки измеряла среднее время закрытия заявок.

Руководство поставило цель: сократить его с трёх дней до одного.

Сотрудники получили чёткий показатель и быстро нашли

способ его улучшить.

Если проблема требовала длительного решения, заявку закрывали с комментарием:

Пользователю предоставлена консультация. При повторном возникновении необходимо создать новое обращение.

Пользователь создавал новое обращение.

Первое считалось успешно закрытым.

Среднее время обработки сократилось. Количество обращений выросло. Пользователи стали раздражительнее. Служба поддержки получила премию за повышение эффективности.

Руководитель отдела демонстрировал график:

- Как видите, мы значительно ускорили работу.

График это подтверждал. Графики вообще редко спорят с человеком, который выбирает для них данные.

Проблемы не решались быстрее.

Они просто получали новые номера.

Система измеряла время жизни заявки, а не время устранения неисправности. Сотрудники не нарушали установленного правила. Они научились достигать показателя способом, не предусмотренным авторами модели.

Так теневой контур выполнил корректирующее воздействие. Только корректировал он не процесс, а его цифровое отражение.

# **Показатель становится мишенью**

До тех пор, пока показатель используется для понимания процесса, он может быть полезен.

Когда от него начинают зависеть премия, статус и безопасность сотрудника, показатель превращается в мишень.

Люди стреляют в неё всеми доступными способами.

Если оценивают количество закрытых задач, задачи начинают дробить.

Если оценивают отсутствие дефектов, дефекты перестают регистрировать.

Если оценивают соблюдение сроков, дату завершения переносят ближе к фактической.

Если оценивают производительность, незавершённую работу объявляют почти завершённой.

Если оценивают точность планирования, план корректируют до тех пор, пока он не совпадёт с фактом.

Последний способ особенно эффективен. Будущее не оправдало прогноз — значит, необходимо исправить прошлую версию будущего.

## **Пример 4. Проект, который никогда не опаздывал**

Компания управляла проектами через календарные гра-

фики. Каждый месяц руководители должны были показывать отклонение от базового срока.

Красная зона начиналась при задержке более десяти дней.

Один проект приближался к красной границе. Руководитель знал, что восстановить срок невозможно. Для этого требовалось увеличить ресурсы, изменить последовательность работ и договориться с подрядчиками.

Это было сложно.

Скорректировать график оказалось проще.

Часть работ перенесли на следующий этап. Несколько задач объединили. Для одной операции изменили длительность. Затем утвердили «актуализированный план».

После актуализации отклонение сократилось с восемнадцати дней до трёх.

Проект снова стал жёлтым.

Физически не изменилось ничего. Рабочих не прибавилось. Материалы не поступили. Отставание не исчезло.

Изменился документ, относительно которого оно измерялось.

На совещании руководитель доложил:

- После актуализации графика ситуация стабилизирована.

Это была изящная фраза. Она создавала впечатление, что кто-то стабилизировал ситуацию, хотя на самом деле стабилизировали способ её описания.

Через три месяца срок снова оказался под угрозой.

График актуализировали ещё раз.

Если бы проект продолжался достаточно долго, он мог бы достичь абсолютной точности: каждая фактическая дата немедленно становилась бы плановой.

Компания получила бы идеально управляемый проект, который завершился на два года позже, но ни разу официально не опоздал.

# **Перенос ответственности как корпоративный спорт**

Теневой контур работает не только с информацией. Он перемещает ответственность.

В официальной модели каждое отклонение должно иметь причину, владельца и корректирующее действие.

В неофициальной модели участники стараются сделать так, чтобы причина находилась за пределами их полномочий, владелец сидел в другом подразделении, а корректирующее действие требовало решения руководства.

Лучший результат — доказать, что проблема возникла из-за отсутствия информации, которую должен был предоставить кто-то другой.

Тогда сотрудник превращается из участника происшествия в его первую жертву.

## **Пример 5. Совещание о недостающем кабеле**

Монтажная бригада не могла приступить к работе: на объекте отсутствовал кабель необходимого типа.

Начальник участка спросил снабжение:

- Где кабель?

Снабжение ответило:

- Не было подтверждённой заявки.

Производственно-технический отдел сообщил:

- Мы отправили заявку две недели назад.

Снабжение уточнило:

- Вы отправили предварительную потребность, а не подтверждённую заявку.

ПТО возразило:

- Для подтверждения требовалась рабочая документация.

Проектировщик сказал:

- Документация была выдана.

Начальник участка открыл папку:

- Здесь старая версия.

Проектировщик проверил:

- Новая версия направлена письмом.

ИТ-специалист посмотрел журнал:

- Письмо было заблокировано из-за размера вложения.

На несколько секунд в комнате воцарилась тишина. Все посмотрели на ИТ-специалиста с благодарностью охотников, неожиданно обнаруживших следы подходящего животного.

ИТ-специалист почувствовал опасность и добавил:

- Пользователю было направлено автоматическое уведомление.

Теперь внимание переключилось на пользователя.

Пользователь сказал, что уведомление попало в папку нежелательной почты.

На совещании присутствовали семь человек, и каждый внёс посильный вклад в отсутствие кабеля. Это создавало редкое чувство коллективного единства.

После часа обсуждений подготовили протокол:

Причиной задержки стало несвоевременное прохождение информации между заинтересованными подразделениями.

Формулировка обладала большим достоинством: в ней не было ни одного живого человека.

Кабель заказали.

Но главное достижение состояло в другом — ответственность удалось растворить в процессе.

Она теперь принадлежала «взаимодействию».

Взаимодействие не получало премию, не писало объяснительных и не присутствовало на совещании. Идеальный виновный.

# **Выиграть время — значит изменить реальность**

Иногда задержка информации имеет практическую цель. Человек рассчитывает устранить проблему до того, как о ней узнает руководство.

Логика понятна:

- возникло отклонение;
  - если сразу сообщить, начнётся контроль;
  - контроль потребует отчётов и совещаний;
  - отчёты и совещания отнимут время;
  - лучше потратить это время на устранение отклонения;
- если всё получится, сообщать вообще не придётся.

На первый взгляд это разумно.

Иногда такой подход действительно спасает процесс от лишней бюрократии.

Проблема появляется, когда устранить отклонение не удаётся. Тогда организация узнаёт о нём позже, располагает меньшим временем и получает уже не одну проблему, а две: само отклонение и потерянный период реакции.

Человек хотел защитить систему от преждевременной тревоги.

В итоге защитил тревогу от своевременной системы.

## Пример 6. Поставщик, который почти успел

Поставщик оборудования обнаружил, что изготовление шкафа управления задерживается на десять дней.

Менеджер не сообщил заказчику. Он рассчитывал ускорить сборку и сократить задержку.

Через несколько дней удалось вернуть три дня.

Менеджер ободрился и продолжил молчать.

Затем обнаружился дефект комплектующего. Задержка выросла до двух недель.

К этому моменту заказчик уже подготовил площадку, вызвал монтажников и заказал подъёмную технику.

Когда информация наконец появилась, изменить график без потерь было невозможно.

Заказчик спросил:

- Почему вы не сообщили раньше?

Менеджер ответил:

- Мы надеялись решить вопрос собственными силами.

Это был честный и даже благородный ответ. Менеджер пытался спасти ситуацию. Он не хотел перекладывать проблему на заказчика.

Но вместе с проблемой он удерживал и право заказчика управлять последствиями.

Теневой контур часто возникает из хороших намерений.

Это делает его особенно опасным. Злой умысел можно запретить. Добрые намерения обычно награждают.

# **Неофициальный маршрут быстрее официального**

В организации неприятная информация редко движется по утверждённому маршруту.

Сначала её обсуждают неофициально.

Рабочий говорит мастеру.

Мастер — начальнику участка.

Начальник участка звонит знакомому в производственно-техническом отделе.

Тот советуется с проектировщиком.

Проектировщик просит пока ничего не регистрировать, потому что хочет проверить решение.

Параллельно информация распространяется через мессенджеры, телефонные разговоры и курилку. Через час о проблеме знают все, кроме цифровой системы и руководителя проекта.

У цифровой модели нет статуса «все уже знают, но официально никто не сообщил».

Между тем это один из самых распространённых статусов корпоративной жизни.

## **Пример 7. Лифт, о котором знали все**

В новом здании периодически останавливался лифт.

Технический персонал знал о проблеме. Охрана знала. Сотрудники, работавшие на верхних этажах, знали особенно хорошо.

В системе технического обслуживания зарегистрировали только два обращения.

Большинство сотрудников сообщало охране устно. Охрана перезапускала лифт и считала проблему временно решённой.

Когда лифт останавливался снова, процедура повторялась.

Через месяц директор оказался внутри.

Через пятнадцать минут проблема получила высший приоритет, официальный номер, ответственного и срок устранения.

На совещании спросили:

- Почему раньше не было информации?

Техническая служба открыла систему:

- Официально было только два обращения.

Сотрудники возмутились:

- Мы постоянно говорили охране!

Охрана объяснила:

- Мы каждый раз перезапускали лифт.

Каждая сторона была права.

Сотрудники сообщали.

Охрана реагировала.

Система почти ничего не знала.

Официальный контур считал проблему редкой. Теневой контур обслуживал её ежедневно.

Лифт оказался единственным участником, который последовательно придерживался своей позиции.

# **Психика не ломает систему — она к ней приспосабливается**

Удобно считать, что теневой контур возникает из-за плохих сотрудников.

Это позволяет заменить нескольких людей, провести обучение и объявить проблему решённой.

Но теневой контур создаётся не характерами, а условиями.

Если сообщение о проблеме приводит к наказанию, люди задерживают проблемы.

Если выполнение показателя приносит премию, люди учатся управлять показателем.

Если ответственность распределяется через поиск виновного, участники заранее готовят защиту.

Если официальная процедура слишком медленна, появляется неофициальная.

Если система не помогает устранить отклонение, сотрудники используют её только для отчётности.

Люди не обязательно саботируют управление. Они приспосабливаются к нему.

Человек вообще обладает удивительной способностью приспосабливаться. Он может привыкнуть к шуму, пробкам, неудобному креслу и еженедельному совещанию без повестки. Поэтому не стоит удивляться, что он приспосабливается

и к цифровой системе.

Если для закрытия операции нужно нажать кнопку, он нажмёт кнопку.

Если требуется прикрепить фотографию, он найдёт фотографию.

Если нельзя завершить задачу с открытым замечанием, замечание будет закрыто.

Вопрос о физическом результате может при этом остаться без ответа.

Человек не нарушает систему.

Он выполняет её требования с такой точностью, что их бессмысленность становится особенно заметной.

# Почему цифровая модель не видит теневой контур

Большинство корпоративных платформ регистрирует только формальные действия:

- документ создан;
- статус изменён;
- задача назначена;
- комментарий добавлен;
- согласование завершено;

срок перенесён.

Но система не видит:

- когда человек впервые узнал о проблеме;
- почему не сообщил сразу;
- с кем советовался неофициально;
- какого решения опасался;
- какой показатель пытался защитить;
- чью реакцию хотел смягчить;

сколько времени потратил на поиск безопасной формулировки.

Именно поэтому цифровая история процесса может быть формально точной и фактически неполной.

В ней отражено то, что люди решили зарегистрировать.

Не обязательно то, что происходило.

Система видит следы официального движения. Теневой

контур ходит по ковру и старается не оставлять следов.

# Как сделать теневой контур видимым

Психику невозможно полностью формализовать. Но можно спроектировать систему так, чтобы ей было сложнее управлять фактами.

Для этого необходимо разделить регистрацию FАСТ от оценки виновности.

Сначала фиксируется событие.

Затем определяется его причина.

Только после этого рассматривается ответственность.

Если все три действия соединены в одной форме, человек понимает: сообщая о факте, он одновременно участвует в собственном допросе. Разумеется, он захочет сначала посоветоваться с адвокатом, даже если адвокатом выступает более опытный коллега из соседнего кабинета.

Регистрация отклонения должна помогать процессу, а не автоматически запускать наказание.

Кроме того, цифровая система должна:

- получать факты непосредственно из операций и измерений;
- фиксировать время возникновения события, а не только время регистрации;
- хранить историю изменений;
- сопоставлять статусы с физическими результатами;
- обнаруживать массовое закрытие и повторное открытие

задач;

- отслеживать систематический перенос сроков;
- выявлять расхождение между показателями и конечным

результатом;

- автоматически запускать стандартные корректирующие действия;

передавать человеку только исключения, действительно требующие решения.

Цель состоит не в том, чтобы поймать сотрудника на искажении данных.

Цель — сделать так, чтобы искажение перестало быть удобным способом управления ситуацией.

# Два контура должны поменяться местами

Сегодня во многих организациях официальный цифровой контур занимается регистрацией документов, а теневой человеческий — реальным управлением.



Люди звонят друг другу, договариваются, ускоряют, задерживают, скрывают, уточняют и исправляют. После этого результаты неофициального управления аккуратно заносят-

ся в программу.

Получается странная конструкция.

Реальный процесс управляется в телефонных разговорах.

Цифровая система хранит его посмертную документацию.

Чтобы перейти к зрелому цифровому управлению, эти контуры должны поменяться местами.

Стандартные события, отклонения и корректирующие воздействия должны обрабатываться цифровой моделью.

Человеческий контур должен работать там, где требуется понимание контекста, изменение правил или решение нестандартной проблемы.

Человек остаётся в системе.

Но он перестаёт быть тайным диспетчером, который вручную решает, какой факт заслуживает существования.

# Тень, которая всегда следует за моделью

Психологический контур невозможно уничтожить.

Пока в организации работают люди, они будут обсуждать решения, бояться последствий, защищать интересы и пытаться выиграть время.

Это нормально.

Ненормально строить систему так, будто ничего этого не существует.

Если цифровая модель не учитывает задержку информации, задержка станет невидимой.

Если не учитывает возможность искажения показателя, показатель станет удобной декорацией.

Если не отделяет факт от наказания, люди будут прятать факты.

Если не связывает данные с физическим результатом, цифровая картина постепенно начнёт жить собственной жизнью.

Она станет аккуратнее реальности, спокойнее реальности и значительно дешевле в обслуживании — потому что в ней не потребуется ничего исправлять.

Человек же останется рядом.

Нелепый, осторожный, самоуверенный и по-своему симпатичный.

Он будет нажимать правильные кнопки, выбирать безопасные статусы и искренне надеяться, что до следующего совещания проблема как-нибудь передумает быть проблемой.

Официальный контур покажет, что всё находится под контролем.

Теневой контур будет знать правду.

А FАСТ, как терпеливый свидетель в коридоре суда, продолжит ждать, когда его наконец пригласят внутрь.

#### **Глава 4. Глаз и совесть начальника**

*Просто опыт: Глаз начальника полезен для профессиональной оценки, но опасен как постоянная архитектура информационной системы.*

Когда в организации отсутствуют стандарты, статусы, критерии готовности и алгоритмы реакции, она всё равно должна как-то работать.

Тогда появляется он.

Опытный начальник.

Он входит на объект, смотрит по сторонам и через несколько минут понимает то, чего информационная система не смогла установить за неделю.

Он замечает, что подрядчик отстаёт, хотя график остаётся зелёным. Понимает, что материал ещё не заказан, хотя заявка имеет статус «в работе». По выражению лица мастера определяет, что техника снова сломалась. По необычной вежливости поставщика догадывается, что поставка не со-

стоит.

Он знает, что руководитель одного подразделения никогда не сообщает плохих новостей до обеда, а другого — после. Помнит, какой подрядчик обещает срок с запасом, а какой воспринимает календарь как произведение художественной литературы.



Программа видит статусы.

Начальник видит людей.

Программа читает данные.

Начальник читает паузы между словами.

Именно поэтому организация постепенно делает его уни-

версальным регулятором. Там, где нет правила, он принимает решение. Где нет достоверных данных — оценивает ситуацию. Где не определена ответственность — назначает ответственного. Где процесс остановился — звонит нужному человеку.

Система держится на его опыте, памяти, авторитете и совести.

Это кажется разумным.

Особенно пока начальник находится на рабочем месте.

# Универсальный регулятор

Опытный начальник обычно не считает себя частью информационной архитектуры.

Он просто работает.

Утром отвечает на звонки, днём проводит совещания, вечером проверяет отчёты. Между этими занятиями он устраняет противоречия, которые система не смогла обнаружить, и принимает решения, для которых никто не написал алгоритма.

Если проектировщик и строитель по-разному понимают чертёж, начальник определяет, кто понял его менее неправильно.

Если два подразделения спорят об ответственности, он находит человека, способного выполнить работу независимо от распределения ответственности.

Если график показывает одно, а объект — другое, он верит объекту. Объект, в отличие от графика, редко успевает подготовиться к совещанию.

Начальник становится одновременно:

- датчиком;
- аналитиком;
- интегратором;
- диспетчером;
- арбитром;

- регулятором;
  - аварийным каналом связи;
- хранителем неформальных правил.

Вечером он уходит домой с чувством, что весь день решал мелкие вопросы.

На самом деле он вручную исполнял функции отсутствующей системы управления.

Организация называет это опытом.

Иногда — лидерством.

Если начальник просит увеличить зарплату, это называют сложной экономической ситуацией.

## **Пример 1. Начальник, который услышал отставание**

На строительном объекте проходило еженедельное совещание.

Представитель подрядчика докладывал:

- Работы идут в соответствии с актуализированным графиком. Ресурсы мобилизованы. Критических ограничений нет.

На экране действительно не было критических ограничений. Несколько задач горели жёлтым, но жёлтый цвет в корпоративных системах обычно означает, что ситуация уже плохая, однако участники пока не готовы использовать красный.

Руководитель проекта посмотрел на подрядчика.

- Сколько человек сегодня работает на участке?
- Ресурс соответствует текущей потребности.
- Я спросил, сколько человек.
- С учётом инженерно-технического персонала?
- Без учёта слов. Сколько людей?

После короткой паузы выяснилось, что вместо сорока рабочих на площадке находились двадцать три.

Руководитель продолжил:

- Материал на следующую неделю заказан?
- Заявка находится в работе.
- Это значит, что заказан?
- Она передана ответственному специалисту.
- Материал заказан?

После второй паузы выяснилось, что заказ ещё не размещён.

Цифровая система не обнаружила проблему. Формально ресурсы были «мобилизованы», а заявка — «в работе».

Начальник обнаружил её по двум паузам и одному уклончивому наречию.

У него был профессиональный слух. Он слышал отставание раньше, чем оно появлялось в отчёте.

После совещания молодой аналитик спросил:

- Как вы поняли, что они не успевают?

Руководитель пожал плечами:

— Когда подрядчик говорит, что всё идёт по графику,

нужно смотреть график. Когда он говорит это особенно уверенно, нужно идти на площадку.

Так работала система раннего предупреждения.

Она носила пиджак, пила крепкий чай и не имела резервной копии.

# Глаз начальника

Выражение «глаз начальника» часто звучит иронично. Но профессиональный взгляд действительно существует.

Опытный руководитель замечает слабые сигналы, которые трудно сразу формализовать:

- непривычное снижение темпа;
  - изменение поведения команды;
  - несоответствие между докладом и физическим результатом;
  - опасную последовательность мелких отклонений;
  - попытку подразделения заранее подготовить оправдание;
  - проблему, ещё не достигшую установленного порога;
- решение, которое формально правильно, но практически не работает.

Такой взгляд формируется годами.

В нём соединяются знания, память о прошлых случаях и способность сопоставлять признаки, которые по отдельности ничего не доказывают.

Программа видит, что срок поставки ещё не нарушен.

Начальник знает, что поставщик третий день отвечает необычно вежливо.

Программа считает, что документ находится на согласовании.

Начальник замечает, что согласующий перестал отвечать на звонки.

Программа показывает выполнение на 80%.

Начальник видит, что оставшиеся 20% потребуют половину общего времени.

Его глаз полезен там, где необходимо понять контекст. Проблема возникает, когда без этого глаза невозможно выполнить ни одну стандартную операцию.

## **Пример 2. Склад, на котором всё было**

На производственном предприятии система показывала, что необходимый подшипник находится на складе.

Количество — две штуки.

Статус — доступен.

Место хранения — секция В-14.

Механик пришёл в секцию В-14 и обнаружил пустую полку.

Кладовщик проверил систему:

- Должно быть две штуки.

Механик посмотрел на полку:

- Здесь ни одной.
- Но в системе две.

Возник классический конфликт цифровой и физической реальности. Обе стороны держались уверенно. Система имела базу данных. Полка обладала преимуществом непосред-

ственного присутствия.

Позвали начальника склада.

Он выслушал участников и спросил:

- Кто последний брал подшипники?

Выяснили, что неделю назад один передали ремонтной бригаде без оформления, а второй переложили в другой шкаф, чтобы не мешал инвентаризации.

Начальник пошёл к нужному шкафу и достал подшипник.

- Откуда вы знали? — спросил механик.

- Я знаю своих людей.

Эта фраза прозвучала гордо.

Но в ней находился диагноз системы.

Цифровой учёт не знал, где лежит деталь. Кладовщики не соблюдали процедуру. Движение материальных ценностей восстанавливалось по памяти начальника.

Предприятие имело складскую информационную систему.

Складом управляла биографическая память одного человека.

Когда начальник уходил в отпуск, предприятие временно переходило к археологии.

# Совесть начальника

Глаза позволяют увидеть отклонение.

Совесть решает, что с ним делать.

При слабой системе руководитель постоянно оказывается перед выбором:

- остановить работу или разрешить продолжить;
- зарегистрировать дефект или устранить без огласки;
- сообщить наверх или сначала попробовать решить самостоятельно;
- наказать сотрудника или признать ошибку процесса;
- защитить срок или качество;

выполнить формальное требование или сохранить практический результат.

Регламент часто не даёт однозначного ответа. Данные неполны. Времени мало. Тогда решение зависит от личной профессиональной этики руководителя.

Хороший начальник принимает ответственность на себя. Он не прячет проблему, защищает команду от бессмысленного давления и не позволяет формальному показателю уничтожить результат.

Организация быстро привыкает к такому человеку.

Она перестаёт улучшать процесс, потому что начальник компенсирует его слабости.

Зачем создавать алгоритм реакции, если можно позво-

нить Ивану Николаевичу?

Зачем определять критерии готовности, если он сам посмотрит?

Зачем связывать данные, если он всё помнит?

Чем лучше руководитель справляется с хаосом, тем меньше у организации стимулов этот хаос устранять.

В результате профессиональная совесть превращается из нравственного преимущества в элемент производственной инфраструктуры.

Инфраструктура надёжная.

Но иногда хочет в отпуск.

### **Пример 3. Партия изделий и один честный человек**

Завод готовил крупную партию изделий к отправке заказчику. Срок был критическим. За задержку предусматривался штраф.

Во время финального контроля обнаружилось отклонение одного параметра. Оно не выглядело катастрофическим, но выходило за установленный допуск.

Инженер по качеству предложил задержать отгрузку и провести проверку.

Коммерческий директор возразил:

- У нас нет подтверждения, что отклонение влияет на эксплуатацию.

Инженер ответил:

- У нас также нет подтверждения, что не влияет.

Финансовый директор сообщил размер возможного штрафа. Цифра произвела сильное впечатление. После неё отклонение стало казаться меньше.

В документации не было ясного алгоритма для такой ситуации. Решение должен был принять директор производства.

Он осмотрел изделие, изучил результаты измерений и сказал:

- Не отправляем. Проверяем всю партию.

Отгрузку задержали. Несколько изделий потребовали доработки. Компания понесла дополнительные расходы, но не передала заказчику сомнительную продукцию.

Все согласились, что директор поступил ответственно.

Затем совещание закончилось, и никто не изменил процедуру контроля.

Критерии реакции остались прежними. Алгоритм блокировки партии не появился. Порог автоматического решения не установили. Ответственность не формализовали.

Организация не устранила системный пробел.

Она убедилась, что у директора хорошая совесть.

Так моральное качество одного человека было принято в эксплуатацию вместо процедуры.

Следующая партия снова зависела от того, кто окажется в кабинете и насколько убедительно финансовый директор

назовёт сумму штрафа.

# Герой как симптом слабой системы

Организации любят героев.

Герой остаётся после работы. Лично обзванивает подрядчиков. Находит дефицитный материал. Уговаривает проектировщика выпустить решение ночью. Вручную сводит противоречивые данные и утром приносит руководству правильный отчёт.

Его благодарят.

Награждают.

Иногда даже дают помощника, который должен научиться работать так же героически.

Но регулярный героизм — это не признак сильного управления. Это показатель того, что стандартный процесс не способен получить результат без чрезвычайного человеческого вмешательства.

Если каждую неделю требуется подвиг, подвиг следует включить в технологическую карту.

## Пример 4. Алексей Викторович спасает понедельник

Каждую пятницу строительный объект должен был получать план работ на следующую неделю.

Для его подготовки требовалось собрать сведения о го-

товности фронтов, наличии материалов, рабочих, техники и документации.

Формально каждое подразделение вносило данные в общую систему.

Практически в пятницу вечером Алексей Викторович звонил всем участникам и задавал короткие вопросы:

- Люди будут?
- Материал придёт?
- Чертёж окончательный?
- Кран работает?
- Вы точно уверены?

Последний вопрос был наиболее важным и отсутствовал в цифровой форме.

После звонков Алексей Викторович корректировал план. Одну работу переносил, другую добавлял, третью переставлял местами. Иногда звонил подрядчику второй раз — уже менее вежливо, но более результативно.

В понедельник объект начинал работать.

Руководство считало, что система недельного планирования действует успешно.

Однажды Алексей Викторович уехал в отпуск.

В следующий понедельник две бригады пришли на неготовый фронт, для третьей не оказалось материала, а кран был занят на соседнем участке.

На совещании руководитель спросил:

- Почему не сработал недельный план?

Участники посмотрели друг на друга.

План был заполнен. Все поля содержали данные. Документ был утверждён вовремя.

В нём отсутствовало только одно обязательное условие — Алексей Викторович.

После его возвращения работа восстановилась. Руководство вздохнуло с облегчением и решило назначить ему заместителя.

Так компания попыталась масштабировать человека вместо процесса.

# Начальник как интеграционная платформа

Современные организации используют множество информационных систем.

Одна хранит документы.

Вторая — графики.

Третья — бюджет.

Четвёртая — заявки.

Пятая — дефекты.

Шестая помогает сотрудникам не забыть пароли от первых пяти.

Между системами часто отсутствует полноценная связь. Тогда интеграцию выполняет руководитель.

Он открывает календарный план и видит, что работа должна начаться завтра. Затем проверяет документооборот и обнаруживает несогласованный чертёж. Звонит в снабжение и узнаёт, что материал не заказан. После этого переносит задачу и сообщает подрядчику, чтобы люди не выходили.

Ни одна система отдельно не увидела проблему.

Начальник сопоставил данные вручную.

Он стал интеграционной платформой с голосовым интерфейсом.

## Пример 5. Четыре программы и один Николай

В девелоперской компании использовали четыре основные системы:

- систему календарного планирования;
  - электронный документооборот;
  - программу бюджетного контроля;
- приложение для фиксации работ на площадке.

В календарном плане монтаж фасада должен был начаться 15 апреля.

В документообороте рабочая документация находилась на согласовании.

В бюджете договор с подрядчиком числился неподписанным.

В приложении площадки фронт работ был отмечен как готовый.

Каждая система показывала свою часть правды.

Николай Сергеевич, руководитель проекта, открыл четыре окна на двух мониторах, сделал несколько звонков и установил следующее:

- фронт работ готов частично;
- чертежи требуют корректировки;
- договор согласован, но не подписан;
- подрядчик готов вывести людей, если получит гарантий-

ное письмо;

материал прибудет на неделю позже.

После этого он вручную изменил график, подготовил поручения и назначил совещание.

Цифровая экосистема работала.

Экосистемой был Николай Сергеевич.

ИТ-директор сообщил, что в следующем году планирует-ся интеграция систем.

Николай Сергеевич спросил:

- А до следующего года что делать?
- Пока продолжать в текущем режиме.

Текущим режимом называлась жизнь Николая Сергеевича.

# Когда глаз становится бутылочным горлышком

Чем опытнее начальник, тем больше решений начинают ждать его оценки.

Сотрудники привыкают:

- Пусть посмотрит руководитель.
- Нужно согласовать с директором.
- Без него лучше не решать.
- Он вернётся и скажет.

Первоначально начальник включается только в сложные ситуации. Затем — в важные. Потом — в спорные. Наконец, без него невозможно заказать канцелярские скрепки, потому что прошлый раз приобрели не тот размер.

Универсальный регулятор постепенно превращается в бутылочное горлышко.

Он перегружен, но организация считает это доказательством его незаменимости. Сам начальник тоже может испытывать определённое удовлетворение. Человек редко возражает против собственной незаменимости, пока не пытается уйти в отпуск.

## **Пример 6. Директор и двести согласований**

Руководитель компании настаивал на личном согласовании всех значимых расходов.

Он хорошо знал бизнес и часто замечал ошибки, которые пропускали сотрудники. Благодаря ему компания избегала необоснованных закупок.

Постепенно список согласований расширялся.

Сначала директор проверял крупные договоры.

Затем — все договоры.

Потом — дополнительные соглашения, командировки и закупки оборудования.

Наконец, у него накопилось более двухсот документов.

Подразделения ждали решений. Сроки сдвигались. Поставщики нервничали. Сотрудники начали заранее закладывать в график две недели на ожидание подписи.

Директор работал вечерами и возмущался:

- Почему никто не способен самостоятельно принять решение?

Сотрудники могли бы ответить, что самостоятельные решения всё равно требовали его согласования. Но сочли это решение недостаточно самостоятельным.

Организация создала замкнутый контур:

- директор не доверяет правилам;

- поэтому проверяет решения лично;
- сотрудники перестают принимать решения;
- качество самостоятельных решений снижается;

директор получает подтверждение, что доверять нельзя.

Система была устойчивой.

Как болото.

# **Зависимость от личности**

Если процесс существует только в голове руководителя, организация не может его надёжно воспроизводить.

Она не знает:

- какие признаки начальник учитывает;
- как он оценивает риск;
- почему выбирает определённое воздействие;
- какие исключения считает допустимыми;
- кому доверяет;

при каких условиях меняет решение.

Уход такого человека воспринимается как природное бедствие.

Проводятся встречи по передаче дел. Создаются папки. Пишутся инструкции. Преемнику говорят:

- Тут всё просто. Постепенно разберёшься.

Фраза обычно означает, что формальная модель отсутствует, а настоящее обучение начнётся после первой серьёзной ошибки.

## **Пример 7. Когда Павел Андреевич ушёл**

Павел Андреевич управлял производственным участком восемнадцать лет.

Он знал оборудование, людей, поставщиков и особенно-

сти каждой смены. Помнил, какой станок нельзя перегружать летом, какой оператор слишком оптимистично оценивает срок и как быстро получить редкую деталь, не начиная официальную процедуру закупки.

Когда Павел Андреевич решил уйти, руководство попросило его передать знания преемнику.

Он подготовил сорок страниц инструкций.

Через месяц после его ухода участок начал регулярно сры-  
вать план.

Инструкции были подробными. Но в них не было записано, что:

Ивану лучше давать задания утром;

- поставщику необходимо звонить дважды;
- станок № 8 вибрирует перед поломкой особым образом;
- ночная смена занижает остаток материала, чтобы оста-

вить резерв;

главный механик быстрее реагирует на личную просьбу, чем на официальную заявку.

Павел Андреевич не скрывал знания.

Он просто не считал их знаниями.

Для него это была жизнь.

Компания потеряла не руководителя, а неописанную модель управления, восемнадцать лет работавшую в человеческой голове.

ИТ-система продолжила функционировать без ошибок.

Она и раньше не знала, как на самом деле работает уча-

СТОК.

# Опасность доброго начальника

Особенно опасен для слабой системы хороший начальник.

Плохой руководитель быстро обнаруживает недостатки процесса. При нём всё останавливается, конфликты выходят наружу, данные расходятся с реальностью. Организация хотя бы понимает, что у неё проблема.



Хороший начальник незаметно закрывает пробелы собой. Он исправляет неточные задания.

Договаривается между подразделениями.

Проверяет сомнительные данные.

Компенсирует отсутствие ресурсов.

Защищает людей от противоречивых решений.

В результате система выглядит устойчивой.

Руководство не видит причин что-либо менять. Оно видит результат.

Только результат создаёт не система.

Его создаёт человек, который каждый день удерживает конструкцию руками и старается не показывать, насколько она тяжёлая.

Когда он уходит, организация удивляется:

- Почему всё развалилось? Ведь процесс работал.

Работал не процесс.

Работал начальник.

# Цифровизация глаза начальника

Наивная цифровизация не устраняет зависимость от руководителя. Она предоставляет ему больше экранов.

Раньше начальник обходил площадку, разговаривал с людьми и читал отчёты.

Теперь он открывает дашборд, просматривает уведомления, проверяет фотографии, читает комментарии и всё равно звонит тем же людям.

Количество информации возрастает. Способ принятия решения остаётся прежним.

Начальник по-прежнему должен:

- определить достоверность данных;
- понять причину отклонения;
- оценить критичность;
- выбрать воздействие;
- назначить ответственного;

проверить результат.

Цифровая система ускоряет доставку материалов для его размышлений, но не замыкает контур.

Это может повысить эффективность сильного руководителя.

Но одновременно увеличивает зависимость организации от его внимания.

Теперь начальник способен контролировать не один объ-

ект, а пять. Значит, вместо одного процесса, зависящего от его глаза, компания получает пять.

Прогресс налицо.

Особенно у начальника.

# Как превратить опыт в систему

Задача цифровизации состоит не в том, чтобы убрать опытного руководителя.

Попытка заменить его несколькими индикаторами обычно заканчивается тем, что индикаторы продолжают гореть зелёным возле уже остановившегося процесса.

Нужно сделать другое: превратить повторяющиеся профессиональные решения в элементы системы.

Если начальник всегда проверяет готовность фронта по пяти признакам, эти признаки должны стать критериями готовности.

Если он определяет риск задержки по сочетанию нескольких событий, система должна отслеживать это сочетание.

Если при определённом отклонении он всегда запускает одну и ту же последовательность действий, она должна стать стандартным алгоритмом реакции.

Если руководитель сравнивает сведения из четырёх источников, данные следует связать.

Если он регулярно исправляет одинаковую ошибку, проблема находится уже не в людях, а в архитектуре процесса.

Опыт начальника необходимо разложить на:

- наблюдаемые признаки;
- правила оценки;
- допустимые границы;

- типовые причины;
- алгоритмы воздействия;
- условия эскалации;

исключения, требующие профессионального решения.

После этого глаз начальника не исчезает.

Он поднимается на другой уровень.

Вместо ежедневной проверки стандартных ситуаций руководитель анализирует исключения, меняет правила и улучшает модель.

Он перестаёт быть частью механизма.

Он становится его конструктором.

# Глаз, совесть и алгоритм

Управление без человеческого опыта опасно. Алгоритм способен последовательно выполнять правило, даже если правило давно потеряло смысл.

Управление только через человеческий опыт не менее опасно. Оно зависит от памяти, состояния и присутствия конкретного человека.

Зрелая система соединяет три элемента:

- алгоритм управляет повторяющимися ситуациями;
- глаз начальника распознаёт новое и нестандартное;
- совесть начальника не позволяет формально правильному решению уничтожить реальный результат.

У каждого элемента своя роль.

Проблема наивной цифровизации в том, что алгоритм занимается хранением документов, а глаз и совесть начальника продолжают управлять всем остальным.

Начальник становится последней инстанцией истины.

Он смотрит на объект, слушает объяснения, проверяет цифры и выносит решение. В этот момент он похож на частного детектива из старого романа: вокруг все что-то скрывают, документы противоречат друг другу, а единственный свидетель утверждает, что ничего не видел.

Разница лишь в том, что детектив получает гонорар за одно расследование.

Начальник проводит их ежедневно.

# Не уничтожить начальника системой

У опытного руководителя есть достоинство, которого долго не будет у любой цифровой модели: он понимает смысл происходящего.

Он способен увидеть, что люди формально выполнили процедуру, но не достигли результата. Может нарушить правило ради безопасности. Может остановить процесс, который на экране выглядит идеальным.

Поэтому цель цифровизации — не заменить начальника.

Цель — перестать использовать его внимание для выполнения операций, которые можно стандартизировать.

Начальник не должен ежедневно проверять, заказан ли материал, согласована ли документация и готов ли фронт работ. Эти условия должна контролировать система.

Он должен решать, что делать, когда стандартный порядок больше не работает.

Его глаз нужен для исключений.

Его совесть — для решений, затрагивающих людей, риск и ответственность.

Его опыт — для изменения самой модели.

Всё остальное должно перестать жить в его голове.

Иначе однажды начальник уйдёт в отпуск, заболеет или просто выключит телефон.

Цифровая система продолжит показывать статусы.

Сотрудники продолжают заполнять формы.  
Дашборды будут светиться спокойным зелёным цветом.  
А организация впервые увидит себя без глаз.

## **Часть II. Советский эксперимент**

Советская система допустила автоматизацию расчётов и рабочих мест, но не создала устойчивого сквозного управления экономическими процессами.

Она охотно разрешила машине считать.

Гораздо труднее оказалось разрешить ей сравнивать обещание с результатом, видеть отклонение и задавать неприятный вопрос:

- Кто принял это решение и почему оно не сработало?

**Глава 5. Кибернетику разрешили — управление не отдали**

*Просто опыт. Власть легко принимает автоматизацию вычислений и значительно труднее принимает автоматизацию контроля над собственными решениями.*

Распространённая история о советской кибернетике выглядит просто.

Сначала её объявили буржуазной лженаукой. Затем поняли ошибку, реабилитировали и начали развивать. Учёные создали вычислительные центры, экономико-математические модели и проекты автоматизированного управления. Но консервативная бюрократия испугалась машин, поэтому великий цифровой проект не состоялся.

История удобная.

В ней есть учёные, чиновники, запрет, прозрение и неосуществлённое будущее. Не хватает только сыщика в плаще и папки с надписью «Совершенно секретно».

В действительности всё было сложнее и потому интереснее.

В начале 1950-х годов кибернетика действительно подвергалась идеологическим нападкам и получала ярлыки «буржуазной» и «псевдонаучной». Но развитие самой вычислительной техники, особенно связанной с обороной, атомным проектом и сложными расчётами, не прекратилось. Машины создавались, математические методы развивались, а государственные структуры прекрасно понимали практическую ценность быстрых вычислений. Во второй половине 1950-х кибернетика получила признание и превратилась в широкую научную рамку — от теории управления до экономических моделей и исследований мышления. Поэтому утверждение «кибернетику в СССР просто запретили» слишком грубо: идеологическая атака была реальной, но вычислительная техника продолжала развиваться, а затем сама кибернетика стала институционально допустимой и престижной. Исследования истории советской вычислительной техники и истории советской кибернетики подтверждают именно эту более сложную картину.

Кибернетику не просто включили обратно.

Ей выдали кабинет, научный журнал и поручили помогать

строительству будущего.

Но между разрешением заниматься наукой и разрешением перестроить управление находилась та же разница, что между приглашением инженера на совещание и передачей ему ключей от директорского кабинета.

Первое считалось полезным.

Второе вызывало вопросы.

Машина только считала — и этим нравилась

Советское государство охотно использовало вычислительную технику там, где задача имела чёткие границы.



Рассчитать траекторию.

Обработать результаты испытаний.

Оптимизировать отдельную технологическую операцию.  
Рассчитать конструкцию.

Подготовить статистическую таблицу.

Выполнить большой объём арифметических действий.

Машина не спорила с целью. Она получала исходные данные, выполняла вычисления и выдавала результат.

Это был идеальный работник.

Не опаздывал, не требовал жилплощадь и не задавал вопроса, почему исходные данные не совпадают с действительностью. Последнее особенно ценилось.

Проблемы начинались, когда вычислительная машина переставала быть усовершенствованным арифмометром и приближалась к управленческому контуру.

Для сквозного управления было недостаточно быстрее считать. Нужно было собирать достоверные данные с предприятий, сопоставлять их между отраслями, обнаруживать отклонения, перераспределять ресурсы и связывать решение с ответственностью.

Тогда машина начинала интересоваться не числами, а властью.

Кто определяет план?

Кто формирует данные?

Кто распоряжается ресурсами?

Кто имеет право изменить решение?

Кто отвечает за расхождение плана и факта?

На такие вопросы арифмометр не претендовал.

Управленческая система претендовала.

Именно поэтому автоматизация вычислений могла восприниматься как техническое развитие, а сквозная автоматизация управления — как попытка перестроить отношения между ведомствами, предприятиями и центром.

Машине разрешили считать государственный план.

Но право понимать, почему он не выполнен, оставили людям.

## **Пример 1. Счётная машина в кабинете директора**

Представим советское промышленное предприятие.

В плановом отделе появляется вычислительная машина. Она ускоряет обработку данных, рассчитывает потребность в материалах и помогает формировать производственную программу.

Начальник отдела доволен.

Раньше сотрудники сводили таблицы десять дней. Теперь справляются за два.

Директор доволен.

Министерство получает отчёт быстрее.

Все выиграли.

Теперь представим следующий шаг. Система должна автоматически сопоставить производственный план с фактическим наличием материалов, состоянием оборудования, тру-

довыми ресурсами и заказами смежных предприятий. Она обнаруживает, что утверждённый план невозможно выполнить при имеющихся ограничениях.



Вычислительная машина сообщает:

Для достижения установленного результата необходимо изменить распределение ресурсов или пересмотреть плановое задание.

В комнате становится тише.

Пока машина помогала выполнять решение, она была полезным инструментом.

Когда она показала, что само решение противоречиво, она приобрела неприятную привычку вмешиваться в дела руко-

водства.

Директор смотрит на расчёт.

Начальник планового отдела смотрит на директора.

Все понимают, что математически машина права.

Но план уже утверждён.

Следовательно, исправлять придётся не план, а объяснение его невыполнения.

Так вычислительная техника снова возвращается к безопасной роли: помогает быстрее подготовить отчёт.

Научная судьба и управленческая судьба

Историю кибернетики в СССР необходимо разделить на две линии.

Первая — научная.

Она включает развитие вычислительной техники, теории автоматов, математической экономики, оптимизации, распознавания образов, управления сложными системами и многих других направлений. Здесь работали сильные научные школы, создавались оригинальные методы и амбициозные проекты.

Вторая линия — управленческая.

Она связана с вопросом, удалось ли превратить математические методы и вычислительные машины в единый контур управления экономикой.

Научные достижения не означали автоматического изменения управленческой архитектуры.

Можно построить вычислительный центр и оставить

прежний порядок формирования данных.

Можно автоматизировать расчёт плана, но не связать его с фактическим исполнением.

Можно установить ЭВМ в министерстве, не изменив отношения министерства с предприятиями.

Можно создать автоматизированное рабочее место, на котором сотрудник станет быстрее выполнять старую бюрократическую операцию.

Компьютеризация увеличивает скорость существующего процесса.

Но если сам процесс устроен неправильно, результатом становится более быстрая ошибка.

Машина не реформирует организацию только фактом своего присутствия. Это всё равно что поставить электрический двигатель на телегу и ожидать, что она самостоятельно превратится в железную дорогу.

Лошади, возможно, удивятся.

Система — вряд ли.

Машины коммунизма

К началу 1960-х отношение к кибернетике изменилось радикально. Её начали рассматривать как важный инструмент научно-технического и экономического развития. В партийных и научных публикациях обсуждалось широкое применение электронных вычислительных машин в производстве, строительстве, транспорте, проектировании, учёте и управлении.

Компьютеры получили почти героический образ.

Если раньше в них видели подозрительное западное учение о машинах и обществе, то теперь они становились помощниками плановой экономики.

Будущее выглядело убедительно.

Данные поступают с предприятий.

Вычислительные центры их обрабатывают.

Математические модели определяют оптимальное распределение ресурсов.

Руководство получает точную картину хозяйства.

Решения принимаются быстрее.

Потери сокращаются.

Экономика становится управляемой как единый организм.

Картина была настолько стройной, что оставалось решить мелкую техническую проблему: убедить все органы этого организма передавать друг другу достоверную информацию и отказаться от части собственной власти.

Органы отнеслись к предложению сдержанно.

Они были согласны стать единым организмом при условии, что каждый останется его головой.

Китов: первая попытка соединить машины и управление

Одним из ранних сторонников общегосударственной автоматизированной системы управления был Анатолий Китов.

Он предлагал использовать сеть вычислительных центров

не только для военных задач, но и для управления народным хозяйством. Идея была смелой: объединить вычислительные мощности, организовать обработку экономической информации и повысить качество управления на государственном уровне.

С инженерной точки зрения замысел выглядел логично.

Если вычислительные центры существуют, почему не использовать их мощности для хозяйственных расчётов? Если экономика плановая, почему не создать систему, которая будет быстро собирать информацию и помогать реализовывать план?

Но предложение затрагивало не только технику.

Оно затрагивало порядок доступа к данным, ведомственные границы и качество существующего управления. А это уже не вопрос мощности процессора.

Когда инженер предлагает автоматизировать расчёт, он просит оборудование.

Когда он предлагает автоматизировать управление, он просит организацию объяснить, как именно она принимает решения.

Организации не любят этот вопрос.



Особенно если ответ хранится в закрытом шкафу, а ключ находится у человека, который утверждает, что шкафа не существует.

## **Пример 2. Машина задаёт лишний вопрос**

Предположим, министерство требует от предприятия увеличить выпуск продукции на 12%.

Предприятие сообщает, что готово выполнить задание.

В обычном порядке вопрос считается решённым. План доведён. Ответственный назначен. Все проявили сознатель-

ность.

Теперь в контур включается система управления.

Она проверяет:

- производственную мощность;
- наличие сырья;
- график поставок;
- состояние оборудования;
- численность персонала;

загрузку смежных предприятий.

После этого система сообщает:

При существующих ограничениях максимальный рост составляет 4%.

Возникает неудобство.

Если принять расчёт, необходимо изменить план или перераспределить ресурсы.

Если оставить план без изменений, система заранее фиксирует его нереалистичность.

До появления сквозного контроля предприятие могло принять задание, а затем объяснить невыполнение объективными трудностями. Министерство могло утверждать, что поставило правильную цель, но предприятие не обеспечило исполнение.

Каждая сторона сохраняла собственную правоту.

Машина портила эту гармонию.

Она пыталась связать решение с ресурсами до начала работы.

Так техническая система становилась свидетелем. А свидетелей, особенно хорошо считающих, редко приглашают на семейные совещания.

Глушков и идея нервной системы экономики

Наиболее масштабное развитие идея получила в работах Виктора Глушкова.

Речь шла не просто о дополнительных вычислительных центрах. Предлагалась многоуровневая общегосударственная система сбора и обработки экономической информации, позднее известная как ОГАС.

В общих чертах предполагалась сеть, объединяющая предприятия, отраслевые и региональные центры и верхний уровень государственного управления. Данные должны были двигаться снизу вверх, а решения и управляющие воздействия — возвращаться в систему.

Для плановой экономики такая идея выглядела почти естественной.

Если государство уже стремилось управлять производством, распределением и ресурсами в масштабах страны, вычислительная сеть могла стать его информационной нервной системой.

Но нервная система отличается от канцелярии.

Канцелярия принимает документ, регистрирует и направляет по принадлежности.

Нервная система замечает боль и требует реакции.

Документ можно положить в папку.

Боль обычно проявляет настойчивость.

Исследования истории ОГАС (Общегосударственная автоматизированная система учёта и обработки информации.) показывают, что проект упирался не только в стоимость или технические ограничения. Он затрагивал существующее распределение полномочий: ведомства контролировали собственные данные, вычислительные центры и предприятия, а общая сеть требовала совместимости, межведомственного обмена и новых правил принятия решений. В 1966–1970 годах были созданы сотни отдельных автоматизированных систем, но без единой координирующей сети; позднее ведомственные решения продолжили развиваться как несовместимые острова. Исторический анализ проекта ОГАС показывает, как идея общенациональной сети постепенно превратилась в совокупность отдельных систем.

С технической точки зрения это выглядело как фрагментация.

С управленческой — как сохранение полномочий.

Каждое ведомство получало вычислительную технику.

Но не обязано было отдавать данные соседу.

Прекрасный компромисс: будущее наступило, а власть осталась на прежнем месте.

Автоматизация без сквозного процесса

Предприятия и министерства начали создавать автоматизированные системы управления.

Автоматизировались отдельные расчёты, функции и ра-

бочие места. Машины помогали обрабатывать документы, рассчитывать показатели, учитывать ресурсы и формировать отчётность.

Это давало реальный эффект.



Нет оснований высмеивать саму автоматизацию. Быстрый расчёт лучше медленного. Электронная обработка данных лучше ручного пересчёта огромных таблиц. Локальная система способна существенно повысить эффективность отдельного участка.

Проблема начиналась в точках перехода.

Данные одной системы не совпадали с данными другой.

Классификаторы различались.

Форматы были несовместимы.

Каждое ведомство создавало решение под собственные задачи.

В результате автоматизировались части административного аппарата, но не формировался единый контур управления экономическим процессом.

Возник своеобразный советский предок современного «зоопарка софта».

Только программ было меньше, компьютеры занимали больше места, а согласование интеграции могло длиться дольше срока службы оборудования.

### **Пример 3. Три правды об одном подшипнике**

Представим, что заводу необходимы подшипники.

Система предприятия показывает потребность в тысяче единиц.

Система отраслевого министерства оперирует укрупнённым планом и учитывает восемьсот.

Система снабжения располагает информацией о возможности поставить шестьсот.

Каждая система права в пределах собственной модели.

Предприятие знает производственную потребность.

Министерство — утверждённый лимит.

Орган снабжения — доступный ресурс.

Сквозная система должна обнаружить расхождение до остановки производства и предложить управляющее решение:

- изменить план;
- найти дополнительного поставщика;
- перераспределить запас;
- заменить конструкцию;

скорректировать производственную программу.

Локальные системы делают другое.

Каждая выпускает правильный документ.

Предприятие отправляет заявку на тысячу подшипников.

Министерство подтверждает восемьсот.

Снабжение выделяет шестьсот.

Через месяц завод объясняет невыполнение плана нехваткой четырёхсот подшипников.

Министерство указывает, что лимит был доведён.

Снабжение подтверждает, что поставило всё доступное.

Никто не ошибся.

Производство остановилось по всем правилам.

Данные как административная собственность

Сегодня принято говорить, что данные являются новым топливом или новой нефтью.

В управлении данные прежде всего являются властью.

Кто владеет информацией, тот способен:

- формировать картину положения;
- определять момент появления проблемы;

- выбирать показатели;
- объяснять отклонения;
- контролировать доступ;
- влиять на распределение ресурсов;

назначать виновных.

Сквозная информационная система меняет этот порядок.

Если данные предприятия доступны верхнему уровню напрямую, предприятие теряет часть возможности объяснять их самостоятельно.

Если показатели разных ведомств сопоставляются, становится видно, где возникают противоречия.

Если система хранит историю решений, труднее переписать причину задним числом.

Если план автоматически сравнивается с ресурсами, недостаточно просто утвердить красивую цифру.

Поэтому сопротивление сквозной автоматизации нельзя объяснять только страхом чиновников перед компьютерами.

Чиновники могли охотно поддерживать компьютеры, если те усиливали их ведомство.

Начальник редко выступает против хорошей вычислительной машины. Особенно если машина стоит в его здании, обрабатывает его данные и готовит отчёт для его начальника.

Трудности возникают, когда та же машина начинает передавать информацию поперёк ведомственных границ и сравнивать результат с чужими обещаниями.

Тогда она превращается из инструмента в участника ап-

паратной политики.

Компьютер не имел партийного билета.

Но проявлял опасный интерес к фактам.

Как каждое ведомство получило собственное будущее

Единая система требовала единых стандартов, классификаторов, правил обмена и ответственности.

Ведомственная система требовала только финансирования.

Поэтому второй путь был психологически и административно удобнее.

# 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.