

СИСТЕМА

ЦИФРОВОЙ ДВОЙНИК

Объектное мышление для руководителя производства

🕒 24 мин чтения • Полный текст

[← К ИНТЕРАКТИВНОЙ МОДЕЛИ](#)

Часть 1. О чём молчат станки: объектное мышление для тех, кто льёт пластик

Как перестать гасить пожары и начать видеть систему — без программирования, цифровых двойников и лишних затрат

Введение: «Ваш завод — это конструктор Lego, из которого сбиты все инструкции»

Вы когда-нибудь пытались объяснить мастеру, почему деталь «серебрит», а технолог говорит про влажность, а наладчик — про давление? Каждый прав. Каждый видит свой кусок. А вместе — каша.

На производстве пластмасс десятки типов станков, сотни форм, тысячи режимов. И все они разные. Но если присмотреться — у них есть общее. ТПА №5 и ТПА №7 отличаются усилием, но одинаково смыкают форму. Экструдер для трубы и для пленки разные, но оба плавят полимер.

Эта книга — не про компьютеры. Она про **способ мышления**, который позволяет видеть не только уникальность каждого станка, но и то, что их объединяет. В IT это называется объектно-ориентированное программирование (ООП). Но ООП — это не про программирование. Это про умение выделять **классы, объекты, наследование, полиморфизм** — и применять это в цехе.

Вы уже так думаете, когда говорите: «Это типовой ТПА 250 тонн», «Это типичная проблема сушки ПА6». Просто не дали названия.

Задача книги — дать эти названия и показать, как системное мышление помогает:

Быстрее находить причину брака.

Согласовывать действия разных отделов.

Переносить настройки с одного станка на другой.

Обучать новичков без «уникального опыта» старых мастеров.

Никакого кода. Никаких обещаний «цифрового двойника за выходные». Только ваш цех, ваши станки и новый взгляд на них.

Глава 1. О чём молчат станки: проблема уникальности

История Петра, технолога

Пётр работает на заводе «ПластПром» 15 лет. Он знает: ТПА №3 «капризничает» при смене формы, экструдер №2 любит более горячую зону 3, а сушилка для АБС-пластика требует на 10°C больше, чем по паспорту. Это знание — его сила. И его проклятие.

Когда Пётр в отпуске, сменный мастер звонит ему в 11 вечера: «Почему деталь хрупкая?». Пётр спрашивает: «Какой станок? Какая форма? Полимер?». Мастер не знает, как связать эти три вещи.

Проблема не в том, что станки уникальны. Проблема — в **неявных связях**. Каждый сотрудник держит в голове свою карту связей. Технолог — одну, мастер — другую, ремонтник — третью. Когда эти карты не совпадают — начинается хаос.

Упражнение «Пять объектов»

Возьмите лист бумаги. Выпишите 5 объектов из вашего цеха:

Станок

Оснастка (форма, калибратор)

Материал

Деталь

Оператор

Теперь соедините их стрелками: станок использует оснастку, материал перерабатывается в деталь, оператор управляет станком. У вас получился **схема связей**. Это первый шаг к тому, чтобы сделать неявное явным.

Вывод из главы: уникальность не исчезает, но становится управляемой, когда вы начинаете видеть не только каждый объект, но и типы объектов, и их отношения.

Глава 2. Классы и объекты: как ваш мозг уже это делает

Класс = чертёж, объект = конкретный станок

Скажите честно: когда вы смотрите на ТПА №5, вы видите не просто «железяку», а «ТПА с усилием 250 т». В вашей голове есть обобщённый образ ТПА — это и есть **класс**.

Конкретный ТПА №5 — **объект**.

Класс — это описание: «у всех ТПА есть усилие смыкания, объём впрыска, зоны нагрева».

Объект — это значения: «у ТПА №5 усилие 250 т, объём 400 см³, зона 1 — 200°C».

Зачем это вам

Когда вы переносите режим с ТПА №5 на ТПА №7, вы делаете это потому, что оба принадлежат одному классу. Если бы классы были разными, вы бы не смогли. Понимая классы, вы:

Создаёте единый язык (все знают, что значит «ТПА»).

Быстрее находите аналогии.

Легче обучаете: «Это обычный ТПА, только с горячеканальной формой».

Пример из экструзии

Класс «Экструдер»: шнек, цилиндр, зоны нагрева, фильера. Объект «Экструдер для трубы РЕ80 №3»: шнек диаметром 60 мм, зоны 180-190-200°C.

Когда вы меняете фильеру с трубы 50 мм на 63 мм, вы меняете не весь экструдер, а только один атрибут. Класс остаётся тем же. Это экономит время на настройку.

Задание «Сделайте завтра»

Выберите три станка одного типа (например, три ТПА). Напишите, что у них общего (класс) и чем они отличаются (уникальные атрибуты объектов). Вы удивитесь, как много общего.

Глава 3. Наследование: не изобретайте велосипед

Что такое наследование простыми словами

Если у вас есть «Экструдер» и вы хотите описать «Экструдер для трубы», вы не пишете все свойства заново. Вы говорите: «Экструдер для трубы — это экструдер, плюс у него есть калибратор и ванна охлаждения».

В ООП это называется **наследование**. Дочерний класс берёт всё от родительского и добавляет своё.

На заводе

Класс «Оборудование» (ID, статус, локация).

От него наследуют «ТПА» и «Экструдер» (добавляют усилие или диаметр шнека).

От «ТПА» наследует «ТПА с роботом» (добавляет метод «захват детали»).

Почему это важно

Наследование предотвращает дублирование. Когда вы меняете общее свойство (например, «рабочее давление» для всех ТПА), оно меняется везде. Не нужно править каждый станок вручную.

Реальный кейс

На заводе поменяли стандарт безопасности: теперь на всех ТПА аварийный стоп должен звучать иначе. Вместо того чтобы программировать каждый ТПА, изменили в классе «ТПА» метод «Аварийный стоп». Все дочерние объекты подхватили изменение. Экономия времени — 3 дня против 5 минут.

Задание

Нарисуйте дерево наследования для вашего цеха: «Оборудование» → «Термическое оборудование» (сушилки, печи) и «Механическое» (ТПА, экструдеры). Добавьте один уникальный подкласс.

Глава 4. Полиморфизм: одна команда — разный смысл

Что это такое

Скажите «Пуск». Для ТПА это значит: смыкание, впрыск, выдержка, раскрытие. Для экструдера — нагрев, запуск шнека, стабилизация. Для сушилki — включение вентилятора. Одна и та же команда вызывает **разное поведение** у разных объектов. Это полиморфизм.

На заводе

Полиморфизм работает, когда вы даёте указания людям: «Исправь режим». Технолог лезет в PID, мастер проверяет охлаждение, оператор смотрит давление. Каждый реагирует по-своему.

Зачем вам это осознать

Когда вы строите систему (даже бумажную), полиморфизм позволяет давать общие команды, не вникая в детали каждого станка. Это ускоряет управление в разы.

Пример с наладкой

Мастер смены пишет в журнале: «Провести плановую чистку». Для ТПА это — чистка сопла и цилиндра. Для экструдера — чистка шнека и фильеры. Команда одна, работы разные. Если бы мастер описывал каждое действие отдельно, он бы не успел.

Задание

Составьте список из 5 общих команд (Пуск, Стоп, Настройка, Диагностика, Чистка). Для каждого типа оборудования опишите, что эта команда означает. Вы создадите полиморфную модель.

Глава 5. Инкапсуляция: каждому — своё

Что прячем и зачем

Оператору не нужно знать, как работает ПИД-регулятор. Ему достаточно кнопки «Пуск» и «Стоп». Наладчику нужен доступ к параметрам. Ремонтнику — к диагностике.

Инкапсуляция — это скрывание внутренностей и предоставление только нужного интерфейса.

В цехе

Вы уже используете инкапсуляцию: у станка есть кнопки, но нет доступа к контроллеру без пароля. У форм есть паспорт, но не вся информация нужна оператору.

Ошибка, которую мы часто делаем

Когда вы даёте оператору доступ к настройкам PID «на всякий случай», он случайно меняет режим. Или когда вы не даёте наладчику смотреть логи аварий — он гадает.

Инкапсуляция — это не секретность, а **разделение ответственности**. Каждый видит ровно столько, сколько нужно для его работы.

Как наладить

Нарисуйте матрицу: роли (оператор, наладчик, мастер, технолог, ремонтник) и для каждой роли — какие атрибуты/методы станка открыты. Например:

Оператор: Пуск, Стоп, Сброс аварии.
Наладчик: + изменение температур, давлений.
Ремонтник: + логи, тестовые режимы.
Это снизит ошибки и ускорит работу.

Глава 6. События и состояния: вместо «бегаем с бумажками»

Событие — то, что случилось

Авария, конец цикла, смена формы, достижение температуры — всё это **события**. На заводе события происходят постоянно, но их редко фиксируют явно. Чаще просто кричат или пишут в тетрадь.

Состояние — где объект сейчас

ТПА может быть: Выключен → Прогрев → Ждёт материал → Работает → Авария → Ремонт. Состояний не много, и переходы между ними обычно строгие.

Конечный автомат — правила переходов

Конечный автомат — это просто табличка: «если ТПА в состоянии "Работает" и случилась авария, перейти в "Авария", запустить сигнал». Такие правила исключают нелогичность: нельзя из «Авария» перейти в «Работает» без «Ремонт».

Пример из литья

Смена формы. Состояния:

ТПА работает (с формой А)

Поступила команда «Смена формы» → переход в «Останов»

Остывание → «Готова к смене»

Демонтаж → «Без формы»

Установка новой формы → «Установлена»

Прогрев → «Готов к работе»

Если пропустить «остывание» — можно повредить форму. Явные состояния дисциплинируют.

Задание

Возьмите любой процесс (замена фильеры, смена партии, чистка). Нарисуйте состояния и переходы. Это займёт 10 минут, но сэкономит часы объяснений.

Глава 7. Как увидеть систему на своём заводе: практикум без IT

Шаг 1. Обойдите цех с блокнотом

Запишите всё, что вы видите: станки, оснастку, людей, документы, сырьё, детали. Это ваши объекты.

Шаг 2. Сгруппируйте по типам (классам)

Например: ТПА (5 штук), экструдеры (3), сушилки (2), формы (15), полимеры (типы). Это ваши классы.

Шаг 3. Выделите общие атрибуты для каждого класса

Для ТПА: ID, усилие, объём, дата последнего ТО. Для формы: ID, количество гнёзд, масса детали. Для полимера: тип, влажность, MFR.

Шаг 4. Нарисуйте связи

Какая форма на каком ТПА? Какой материал используется? Какие документы связаны с заказом?

Шаг 5. Сделайте карту событий

Какие события происходят каждый день (завершение цикла, авария, смена оснастки)? Кто на них реагирует?

Итог — у вас готова **онтология завода** без единой строчки кода. Просто на бумаге. Вы теперь видите не кучу уникальных железяк, а систему взаимосвязанных типов и объектов.

Что это даёт

Вы можете показать карту новичку — он быстрее поймёт.

Вы можете найти «белые пятна» — где связи потеряны.

Вы можете спросить: «А что случится, если мы добавим новый класс?» — и обсудить, не ломая оборудование.

Глава 8. Возражения и ответы (честные)

1. «Это всё теория, у меня каждый станок уникален»

Да, уникален. Но у каждого есть тип. Тип — это общее. Вы же не переучиваетесь заново на каждый новый ТПА? Потому что вы знаете класс. Уникальность — в 20% деталей. Эта книга про то, чтобы не терять 80% общего.

2. «Я не программист, зачем мне ваши классы»

Вы уже используете классы, просто не даёте им имя. «ТПА» — класс. «Полиамид 6» — класс. Название помогает договориться. Программирование здесь ни при чём.

3. «Нам нужны не классы, а снижение брака»

Понимание классов и связей помогает быстрее находить причину брака. Когда вы связываете станок, форму и материал в одну схему, вы видите, где стык не работает. Без схемы — гадаете.

4. «Мы пробовали цифровизацию, не взлетело»

Эта книга не про цифровизацию. Она про мышление. Даже если вы никогда не купите софт, объектное мышление поможет вам на бумаге навести порядок. Цифровизация проваливается, когда нет ясной модели. Модель вы можете создать сами.

5. «Где взять время на это?»

Вы и так тратите время на разбор полётов, на согласование, на обучение новичков.

Вложение 10 часов в создание объектной карты окупится через месяц за счёт меньшего числа нестыковок.

6. «Мне проще сделать, чем описать»

Для простой детали — да. Для сложного заказа из 500 операций — нет. Описание вынуждает вас проверить логику. «Сделать» без описания приводит к ошибкам, которые дороже.

7. «Это подходит только для больших заводов»

На малом заводе каждый сбой болезненнее, потому что нет резерва. Упорядочивание классов и связей на 5 станках сэкономит вам пару часов в день — для малого завода это огромно.

Заключение: Пришло время выключить пожарную сирену

Мы прошли путь от хаоса уникальности до ясной системы классов, объектов, связей, событий и состояний. Мы не написали ни строчки кода. Мы не купили ни одного датчика. Мы просто **изменили способ думать** о заводе.

Главные выводы:

- Ваш завод — не уникальный снежинка. В нём есть повторяющиеся типы (классы). Найдите их.
- Каждый объект (станок, форма, материал) имеет атрибуты и поведение. Отделите общее от частного.
- Наследование и полиморфизм экономят время на перенастройку и обучение.
- Инкапсуляция (разные права) предотвращает ошибки.
- События и состояния делают процессы прозрачными.

Что делать завтра утром:

- Возьмите лист А1 и фломастеры.
- Напишите в центре «Наш завод».
- Добавьте 5 главных классов: Оборудование, Оснастка, Материал, Продукция, Персонал.
- Соедините их стрелками.
- Повесьте на стену в цехе.
- Через неделю добавьте события. Через две — состояния. Через месяц вы увидите, как изменилось качество разговоров: «А у нас в модели класс "сушилка" не связан с классом "влажность", поэтому технолог сушилки спорит с технологом ТПА». Это уже не конфликт, а задача на проектирование.

Объектное мышление не заменит опыт и интуицию. Но оно превратит ваш опыт из набора личных трюков в систему, которой можно управлять, передавать и улучшать.

Вы не обязаны становиться программистом. Но вы можете стать **системным технологом** — человеком, который видит не только деталь, но и весь завод как связанное целое. Это сложно. Это не даст быстрых побед. Но это единственный способ перестать тушить пожары и начать проектировать производство, которое не горит.

Приложение: Глоссарий для производственника

ТЕРМИН	ПЕРЕВОД
Класс	Тип, общее описание (например, «термопластавтомат»)
Объект	Конкретный станок, форма, партия

ТЕРМИН	ПЕРЕВОД
Атрибут	Характеристика (усилие, температура)
Метод	Действие, которое объект может сделать (нагреть, впрыскнуть)
Наследование	Дочерний тип берёт всё от родительского и добавляет своё
Полиморфизм	Одна команда вызывает разные действия у разных объектов
Инкапсуляция	Разные права доступа для разных ролей
Событие	То, что случилось (авария, конец цикла)
Состояние	В каком режиме объект (работает, ремонт)
Конечный автомат	Правила переходов между состояниями

Конец книги.

Если вы применили хотя бы одно упражнение и почувствовали, что стали лучше понимать свой цех — автор доволен. Если нет — напишите, что не так. Без вашей критики следующая книга не станет лучше.

Часть 2. Управление как конструктор: объектный подход для руководителей производства пластмасс

Как превратить хаос заказов, станков и людей в систему, которой можно управлять, не увеличивая штат

Для кого: директоров, начальников цехов, менеджеров по производству, мастеров, плановиков. **Не требуется:** знание программирования, IT-бюджет, цифровые двойники. **Требуется:** желание увидеть завод не как кучу проблем, а как набор управляемых объектов.

Введение: почему управление сложнее, чем станок

Вы умеете управлять одним ТПА. Выставили температуру, давление, время цикла — он работает. А попробуйте управлять десятью ТПА, тремя экструдерами, двумя складами,

пятью сменами и сотней заказов. Появляются нестыковки: заказ ждёт материал, станок простаивает из-за формы, мастер не знает, что приоритет изменился.

Проблема: вы пытаетесь управлять процессами так же, как станком — линейно, командно. Но производство — это сеть взаимодействующих объектов. Нужен другой способ мышления.

В первой книге мы говорили об объектном подходе для оборудования. Теперь применим его к **управлению**. Вы увидите:

Производственный заказ как объект со своей жизнью.

Ресурсы (станки, люди, оснастка) как объекты с состояниями.

События (запуск, остановка, брак) как триггеры для управленческих решений.

Метрики как методы, которые объекты сами умеют вычислять.

Обещание: вы перестанете бегать с бумажками и начнёте проектировать управленческие правила. Не за месяц, но за квартал — точно.

Глава 1. Управленческая онтология: из чего состоит управление

Что вы управляете каждый день?

Возьмите лист. Выпишите всё, чем вы управляете:

Заказы (их статусы, сроки)

Станки (доступность, загрузка)

Оснастка (где, чья, когда вернётся)

Люди (смены, квалификация)

Материалы (остатки, поставки)

Документы (наряды, акты, отчёты)

Это **объекты управления**. Они не висят в воздухе, а связаны. Заказ привязан к техпроцессу, техпроцесс — к станкам, станки — к людям.

Онтология управления — реестр типов и связей

Класс «Производственный заказ». Атрибуты: номер, номенклатура, количество, дата отгрузки, приоритет, статус. Класс «Ресурс». Атрибуты: ID, тип (станок/человек/оснастка), текущее состояние, календарь. Класс «Операция». Атрибуты: номер, описание, норма времени, требуемый ресурс.

Связи: Заказ состоит из Операций. Операция требует Ресурс. Ресурс может быть занят или свободен.

Зачем это руководителю: когда вы явно описали классы и связи, вы можете ответить на вопрос «Что случится, если я добавлю новый заказ?» — не гадать, а пройти по связям.

Упражнение «Управленческая карта»

Нарисуйте три объекта: Заказ (с номером), Станок (с именем), Мастер (с именем).

Соедините стрелками: Заказ → Станок (закреплён за), Мастер → Станок (отвечает за).

Теперь добавьте событие «поломка». Какие связи затронуты? Это и есть онтология управления.

Глава 2. Объекты управления: заказы, сменные задания, наряды

Заказ как объект

У заказа есть **жизненный цикл**:

Черновик (согласование состава)

Подтверждён (передан в производство)

В работе (распределён по операциям)

Частично выполнен

Выполнен

Закрыт

Каждому статусу соответствуют **методы**: «отправить в цех», «приостановить», «изменить приоритет», «закрыть».

Сменное задание — объект-посредник

Вы не можете дать заказ напрямую станку. Нужно разбить на сменные задания. Класс

«Сменное задание»: атрибуты — ID заказа, ID станка, плановое время начала/конца, фактические время, статус (назначено/выполняется/завершено/срыв).

Когда мастер принимает смену, он видит список объектов «Сменное задание». Каждое задание знает, какой заказ, какой станок, какие материалы. Это заменяет гору бумаг.

Наряд на работу — объект для людей

Для оператора или наладчика нужен наряд. Он наследует от сменного задания, но добавляет ФИО, часы, выработку. Метод «Выполнить» у наряда автоматически отмечает время окончания, списывает материал, генерирует событие для заказа.

Пример из литья

Заказ №789: деталь «Крышка», 5000 шт., срок 20.05. Система создаёт сменные задания:

ТПА №5 — 2000 шт., ТПА №7 — 3000 шт., по сменам. Каждое сменное задание порождает наряды для операторов. При завершении наряда заказ автоматически уменьшает остаток. Когда остаток 0 — заказ переходит в «Выполнен».

Что получает руководитель: прозрачность, кто, на чём и сколько сделал. Никаких «а я думал, что ты сделал».

Глава 3. Наследование в управлении: типовые маршруты и семейства деталей

Зачем нужны шаблоны

Вы выпускаете 100 видов деталей. Каждый раз писать техпроцесс заново — безумие. В объектном управлении вы создаёте **класс-родитель** «Техпроцесс литья» и наследуете от него конкретные процессы.

Пример

Класс «Базовый техпроцесс»: операции — сушка, загрузка, впрыск, охлаждение, извлечение. Для детали «Крышка простая» наследуем ничего не меняя. Для детали «Крышка с армированием» добавляем операцию «укладка вставок». Для детали «Корпус толстостенный» переопределяем время охлаждения (удлиняем).

Наследование позволяет:

Не дублировать общее.

Менять общее в одном месте (например, добавить контроль качества после сушки — и все процессы получают).

Быстро создавать новый техпроцесс под новую деталь.

Управленческий выигрыш

Когда приходит срочный заказ на деталь, похожую на уже выпускавшуюся, вы берёте класс-родитель, наследуете, меняете 2 параметра. Время на планирование — 10 минут вместо 2 часов.

Задание для плановика

Составьте список семейств деталей. Для каждого семейства опишите базовый маршрут. Укажите, что обычно меняется (время, материал, оснастка). Получится библиотека наследуемых техпроцессов.

Глава 4. Полиморфизм управления: одна команда — разная реакция

Задача диспетчера

Диспетчер говорит: «Останови станок!» Причины могут быть разными: авария, смена оснастки, отсутствие материала, конец смены. Но команда одна. **Полиморфизм** означает, что каждый объект (станок) сам знает, что делать при команде «Стоп» в зависимости от контекста.

Расширим на управление заказами

Команда «Приоритет повышен». Для заказа в статусе «Черновик» — ничего не менять. Для заказа «В работе» — пересчитать расписание, возможно вытеснить другие заказы. Для заказа «Частично выполнен» — догрузить оставшееся в более ранние слоты. Один метод «повысить_приоритет» ведёт себя по-разному для разных состояний заказа. Это полиморфизм.

Управленческий пример

Мастер цеха даёт команду «Ускорить выполнение заказа №456». Система (даже мысленная или бумажная) проверяет:
Заказ на ТПА №3 — там осталось 2 часа, ускорить нельзя, но можно перебросить на ТПА №4.
Если бы заказ был на экструдере, ускорение означало бы повышение линейной скорости. Разные объекты — разная реакция. Мастеру не нужно думать за каждый станок.

Как внедрить без IT

Создайте таблицу: «Команда» → «Тип объекта» → «Действие». Например:

КОМАНДА	ОБЪЕКТ	ДЕЙСТВИЕ
Приоритет 1	Заказ в очереди	Переместить в начало
Приоритет 1	Заказ в работе	Не прерывать, но после него — следующий
Приоритет 1	Сменное задание	Добавить сверхурочную смену

Это явное описание полиморфизма. Раздайте мастерам.

Глава 5. Инкапсуляция: кто что видит

Управленческая информация тоже требует защиты

Мастер должен видеть загрузку станков и сменные задания, но не видеть себестоимость заказа (коммерческая тайна).
Плановик — видеть остатки материалов, но не зарплаты.
Директор — видеть всё, но не вдаваться в детали каждой операции.
Инкапсуляция в управлении — это **ролевой доступ к объектам и их методам**.

Матрица доступа

роль	заказы (статус)	ресурсы (загрузка)	финансы	документы
Оператор	только свой текущий наряд	только свой станок	нет	наряд
Мастер	сменные задания своего участка	все станки участка	нет	сменные планы
Плановик	все заказы, все статусы	все ресурсы	нормы времени	техпроцессы
Директор	все, плюс отчёты	все	себестоимость, прибыль	стратегические

Зачем это внедрять сейчас (на бумаге)

Когда у вас нет системы, информацию вы даёте всем одинаково — либо скрываете от всех. Мастер не знает загрузки соседнего участка, плановик не знает реальных простоев. Чёткая инкапсуляция наоборот **улучшает информационный обмен**, потому что каждый знает, что он должен видеть и что требовать.

Задание

Напишите список ролей в вашем управлении. Для каждой укажите 3 класса объектов, которые ей нужны, и 3 — которые должны быть скрыты. Согласуйте на совещании.

Глава 6. События в управлении: не дёр-гайте, а реагируйте

Управление через события

Вместо того чтобы каждые 5 минут бегать и спрашивать «как идёт?», вы настраиваете **события**. Событие — это сообщение от объекта: «Я изменил состояние».

Примеры управленческих событий:

«Сменное задание началось»

«Станок в аварии»

«Поступил новый заказ с приоритетом»

«Закончился материал на складе»

«Оператор закончил наряд»

На каждое событие есть **подписчик** — роль или правило, которое должно среагировать.

Событие «Авария ТПА №5»

Подписчики:

Мастер смены → должен переназначить сменные задания с ТПА №5 на другой станок.

Ремонтная служба → получить заявку.

Диспетчер → уведомить плановика о сдвиге сроков.

Директор (опционально) → если время простоя > 2 часов.

События превращают хаос в управляемую цепочку реакций. Вы перестаёте гадать, кто должен что сделать.

Как начать без IT

Возьмите ватман. Напишите основные события дня (начало смены, поломка, выполнение заказа, приход материала). Напротив каждого события напишите, кто и что должен сделать. Повесьте в диспетчерской. Через месяц вы будете делать это автоматически.

Пример для экструзии

Событие «Давление расплава превысило порог». Подписчик — мастер-технолог. Его реакция: проверить фильеру, при необходимости остановить линию, вызвать ремонт.

Событие «Готовность калибратора» — подписчик оператор, который включает тянущее устройство.

Когда события прописаны, новичок знает, что делать при любом сигнале. Не нужно бежать к мастеру.

Глава 7. Конечные автоматы для управленческих процессов

Процесс как конечный автомат

Управление производством — это цепочки состояний. Конечный автомат задаёт разрешённые переходы. Это дисциплинирует.

Пример: согласование заказа

Состояния:

Черновик

На согласовании у технолога

На согласовании у плановика

На согласовании у экономиста

Согласован

Отклонён

Переходы:

Из Черновика → На согласовании у технолога (только когда заполнены все поля).

От технолога → к плановику (только если технолог поставил визу).

От плановика → к экономисту (если ресурсы есть).

Из любого статуса → Отклонён (с комментарием).

Из Отклонён → Черновик (с правками).

Без явного автомата заказ может «зависнуть» у технолога на неделю, а плановик даже не знает.

Автомат для сменного задания

Состояния: Назначено → Подтверждено оператором → Выполняется → Завершено →

Принято мастером. Запрещено: из Выполняется перейти в Назначено (нельзя отменить в процессе) или из Завершено в Выполняется.

Как нарисовать автомат для любого процесса

Перечислите все возможные состояния.

Укажите стартовое и конечное состояния.

Проведите стрелки переходов и подпишите событие, вызывающее переход.

Запретите все остальные стрелки.

Повесьте автомат на стену. Когда кто-то говорит «а давайте сразу из черновика в работу», вы показываете на схему и говорите «нарушаем процесс».

Глава 8. Управленческие метрики как методы объектов

Метрики не висят в воздухе

Каждый объект управления сам умеет вычислять свои KPI. Это его **методы**.

У объекта «Станок»: $OEE = (\text{доступность} \times \text{производительность} \times \text{качество})$

У объекта «Сменное задание»: $\text{отклонение от плана} = \text{факт_время} - \text{план_время}$

У объекта «Заказ»: $\text{выполнение_процента} = \text{сделано_шт} / \text{всего_шт} \times 100\%$

У объекта «Оператор»: $\text{выработка_за_смену} = \text{сумма_нарядов}$

Почему это важно

Когда метрика привязана к объекту, она всегда актуальна и контекстна. Вы не ищете «общую производительность цеха» в абстрактной таблице — вы спрашиваете у каждого станка его OEE и агрегируете.

Пример из литья

Директор хочет понять, почему цех не выполняет план. Вместо того чтобы собирать отчёты, он смотрит на список объектов «Станок». У ТПА №5 OEE = 62% из-за низкой производительности (медленный цикл). У ТПА №7 OEE = 88% — норма. Причина: на ТПА №5 устаревшая форма. Метод «рассчитать_OEE» у формы тоже есть — и он показывает, что форма даёт больше брака. Управленческое решение: заменить форму.

Как внедрить без софта

В Excel создайте лист «Объекты-метрики». Колонки: Объект (ТПА №5), Параметры (рабочее время, плановое время, количество деталей, брак). Формулы OEE. Обновляйте раз в смену. Это уже даст 80% пользы.

Глава 9. Как внедрить объектное управление без IT: пошагово

Шаг 0. Осознание: провести воркшоп «Классы нашего управления»

Соберите руководителей на 4 часа. На флипчарте нарисуйте:

Классы: Заказ, Ресурс, Операция, Событие, Метрика.

Атрибуты и методы.

Связи.

Шаг 1. Создать реестр объектов в Excel

Три листа:

«Заказы» (ID, статус, срок, приоритет)

«Ресурсы» (ID, тип, состояние, ответственный)

«Операции» (ID, описание, норма времени, требуемый ресурс)

Шаг 2. Прописать конечные автоматы для главных процессов

Процесс исполнения заказа

Процесс ремонта станка

Процесс смены оснастки

Шаг 3. Определить события и подписчиков

Создать доску «Если случилось X, то делаем Y». Примеры повесить в цехе.

Шаг 4. Внедрить метрики как методы

Научить мастеров рассчитывать ОЕЕ, выполнение плана, отклонения по формулам в Excel.

Потребуется 2 часа обучения.

Шаг 5. Еженедельный обход объектов

Каждый понедельник руководитель обходит не станки, а **объекты управления** в системе: открытые заказы, простаивающие ресурсы, незакрытые события. Управление становится проактивным.

Шаг 6. Масштабировать

Когда привыкли на одном участке, распространите на весь завод. Добавьте классы «Поставщик», «Транспорт», «Склад». Свяжите с финансами.

Бюджет: 0 рублей на софт, только время людей. **ROI:** снижение простоев из-за чёткой реакции на события, ускорение согласования заказов, прозрачность для принятия решений.

Заключение: от тушения пожаров к проектированию системы

Управление производством без объектного мышления похоже на попытку собрать Лего без инструкции — вы видите много деталей, но не знаете, как они соединяются. Объектный подход даёт вам **инструкцию**.

Выводы:

Производство состоит из объектов: заказы, ресурсы, операции, события. Назовите их.

У каждого объекта есть атрибуты (данные) и методы (что он может). Опишите это.

Наследование и полиморфизм позволяют не плодить сущности, а переиспользовать общее.

Инкапсуляция разграничивает доступ — каждый видит нужное.

События и конечные автоматы делают процессы явными и контролируемыми.

Метрики — это методы объектов, не внешние отчёты.

Что сделать завтра:

Соберите управленческую команду.

Нарисуйте на доске три класса: Заказ, Станок, Мастер.
Добавьте событие «Заказ просрочен».
Пропишите реакцию каждого объекта на это событие.
Сфотографируйте и выложите в чат завода с комментарием: «С сегодня работаем по-новому».
Вы не купили софт, не наняли программистов, не потратили миллионы. Но вы изменили способ думать. Это и есть цифровая трансформация без цифры.

Приложение 1. Шаблоны для управленческих воркшопов

Шаблон 1. Карта классов управления

КЛАСС	АТРИБУТЫ	МЕТОДЫ	СВЯЗИ
Производственный заказ	№, статус, срок, приоритет	изменить_приоритет, приостановить	состоит из операций
Ресурс (станок)	ID, тип, состояние	запустить, остановить, рассчитать_OEE	обслуживается мастером
Событие	тип, источник, время	обработать, уведомить	инициирует переход

Шаблон 2. Матрица инкапсуляции (роли и объекты)

РОЛЬ	ЗАКАЗЫ (СТАТУС)	РЕСУРСЫ	МЕТРИКИ	СОБЫТИЯ
Директор	все	агрегированные	все	критические
Плановик	все, кроме финансов	свободные, занятые	загрузка	создание/отмена заказа
Мастер	сменные задания	своего участка	выполнение смены	авария, конец наряда
Оператор	только свой наряд	свой станок	выработка	начало/конец работы

Шаблон 3. Конечный автомат для заказа

Состояния: Черновик → Согласование → Передан в цех → В работе → Контроль качества → Готов → Отгружен. Под каждым переходом — событие: «заполнены поля», «виза технолога», «назначены ресурсы» и т.д.

Конец книги.

Если вы хотя бы на один день перестали бегать и начали проектировать — автор гордится вами. Управление — это не скорость, это правильные правила.

МАТЕРИАЛЫ ПО ТЕМЕ

Статья целиком и рабочие артефакты — для скачивания и печати.



Статья «Цифровой двойник»

PDF • полный текст

СКАЧАТЬ