

ИЗМЕНЕНИЯ

УПРАВЛЕНИЕ КАК ЭКСПЕРИМЕНТ

PDCA, гипотезы и как перестать гадать

🕒 23 мин чтения • Полный текст

[← К ИНТЕРАКТИВНОЙ МОДЕЛИ](#)

Введение. Почему ваши «исправления» не работают?

Практическое руководство по циклу PDCA для директоров и руководителей

Вы — руководитель. Каждый день вы слышите: «Надо провести планёрку», «Утвердить регламент», «Внедрить KPI». Вы это делаете. Но через два месяца проблема возвращается. И вы снова тушите тот же пожар.

Знакомо?

Я прошёл через это на нескольких предприятиях — от небольших цехов до крупных производств. Схема всегда одна: проблема → экстренное решение → временное затишье → рецидив. И каждый раз мы удивляемся: «Ну почему они опять не делают то, о чём мы договаривались?»

Ответ прост: вы меняли действия, а не поведение. Вы планировали мероприятия, а не систему.

Но есть и второй ответ, более личный. Когда я начинал, я понятия не имел о цикле Деминга. Я просто методом проб и ошибок выстроил на своих заводах процесс: придумать изменение → проверить → посмотреть на результат → либо закрепить, либо выбросить. А потом узнал, что это уже называется PDCA, и обрадовался: значит, логика верная.

А потом я пошёл учиться «правильному» PDCA у бизнес-школ и консультантов — и ужаснулся. Они превратили эксперимент в бюрократию. План, отчёт, наказание за отклонение. Сухой шаблон без живого смысла. Именно эта профанация дискредитировала великий инструмент.

(Историческая справка для любознательных: идея цикла восходит к Уолтеру Шухарту (конец 1930-х), а четырёхшаговую формулу развил и популяризировал в Японии Уильям Эдвардс Деминг уже после войны. Деминг предпочитал название PDSA — «Study», исследование, а не «Check», инспекция. В этой книге я оставляю привычную аббревиатуру PDCA, но по духу следую PDSA.)

Эта книга — возвращение к первоначальному смыслу. Который я заново открыл через производственный опыт. Здесь не будет «правильных» шаблонов ради шаблонов. Будут техники, которые я проверил на живых коллективах.

Главный инструмент — цикл PDCA, но не тот, сухой из учебников, а живой, поведенческий, экспериментальный. Я пересобрал его под реальные задачи руководителя.

Что вас ждёт:

Глава 1. PDCA как научный метод — зачем он нужен.

Глава 2. Как мы чиним классический цикл (и возвращаем ему смысл).

Глава 3. Гипотеза — ваш новый план. Забудьте про «списки дел».

Глава 4. Поведение, а не задачи: четыре рычага изменений.

Глава 5. Роль руководителя — не царь и не надсмотрщик, а экспериментатор.

Глава 6. Карта гипотез — пошаговый метод от цели до проверки.

Глава 7. Цикл в действии — три живых примера (и один провал — тоже полезный).

Глава 8. Валидация: работает ли это в реальности? И как PDCA связан с Agile, Lean и Scrum.

Заключение. Ваш первый эксперимент — план на неделю.

Приложение. Шпаргалка, глоссарий и шаблоны для быстрого старта.

Без воды. Только практика.

Глава 1. PDCA как научный метод

Цикл Деминга — четыре шага: Plan, Do, Check, Act. Казалось бы, что тут нового? Новое — в том, как вы их проживаете.

Для большинства руководителей PDCA выглядит так:

Plan: составить список работ.

Do: заставить людей сделать.

Check: сравнить с планом, обнаружить отклонения.

Act: наказать.

Это путь в никуда. Так работают только бюрократические системы, которые имитируют бурную деятельность.

Правильный PDCA — это научный метод, перенесённый в управление. Каждый ваш цикл должен отвечать на вопрос: «Что мы узнали нового о том, как заставить систему работать лучше?»

Моя трактовка четырёх фаз:

ФАЗА ЧТО ЭТО НА САМОМ ДЕЛЕ

Plan	Не список дел, а проверяемая гипотеза о причинно-следственной связи.
Do	Не массовое внедрение, а контролируемый эксперимент (часто пилот).
Check	Не отчёт, а анализ данных и извлечение уроков.
Act	Не наказание, а решение: стандартизировать успех или начать новый цикл с другой гипотезой.

Я помню свой первый осознанный цикл. На одном заводе брак на сварке достиг 7%. Мы не стали проводить внеплановый инструктаж (как делали всегда). Вместо этого мы повесили на верстак образцы правильных и дефектных швов. Через неделю брак упал до 4%. Мы проверили гипотезу «если дать наглядный эталон, то сварщик будет сверяться». Сработало. Но главное — мы поняли, что дело не в лени людей, а в отсутствии быстрой подсказки. Это изменило наше мышление.

Глава 2. Как мы чиним классический цикл

Когда я начал применять PDCA так, как его учат на курсах MBA, я наткнулся на две проблемы.

Первая: план не имеет встроенной логики. Формальный план не объясняет, *почему* это действие должно сработать. А без «почему» проверка превращается в гадание: «Ну, не получилось — значит, плохо сделали».

Вторая, самая опасная: цикл предполагает, что вы проверяете только результат в конце. А если вы ошиблись в начале? Две недели работы коту под хвост.

Это не ошибка Деминга. Это ошибка интерпретации. Оригинальный PDCA был задуман как спираль с короткими обратными связями. Но бизнес-школы упростили его до линейной стрелы.

Мои исправления (возвращающие нас к замыслу Шухарта)

Я добавил в цикл два элемента:

Гипотезу как обязательное условие плана (об этом — следующая глава).

Микро-циклы проверки внутри фазы «Do» — но с важной оговоркой: это не разрешение менять гипотезу на ходу, а проверка допущений. Вы смотрите: идём ли мы по плану? Что

мешает? Может, нужно скорректировать условия (инструменты, обучение), а не саму идею? Если вы поняли, что исходная гипотеза неверна — вы не меняете её на лету. Вы завершаете цикл досрочно и запускаете новый. Это честная экспериментальная дисциплина.

Как определить момент досрочного завершения?

Правило простое: если после двух-трёх чек-поинтов (4–6 дней) нет ни малейшего намёка на желаемое изменение поведения или метрики — гипотеза, скорее всего, ошибочна. Что значит «ни малейшего намёка»? Например:

За три дня ни один сотрудник ни разу не выполнил целевое действие (не взял чек-лист, не сверился с эталоном, не задал вопрос по шаблону).

Метрика не сдвинулась даже на 1% в нужную сторону.

Вы слышите от команды: «Мы пробовали, но это невозможно, потому что...» (и барьер не устранить быстрым фиксом).

Не ждите две недели. Останавливайтесь, фиксируйте урок и стартуйте новый цикл.

Исключение — процессы с естественным длительным циклом (например, выращивание культуры безопасности на вахте). Но в большинстве случаев трёх дней достаточно для первого сигнала.

Где PDCA не работает (честно о границах)

Никакой инструмент не универсален. Вот три случая, когда цикл Деминга не подходит (или нужна его серьёзная адаптация):

Кризис «горит» — нет времени на цикл. Сначала тушим пожар любым способом, потом применяем PDCA, чтобы понять причину.

Жёстко регламентированные среды (атомная энергетика, военные операции) — отклонение от инструкции недопустимо даже в эксперименте. Но цикл можно применять для улучшения самих инструкций (на уровне штаба).

Проекты с необратимым результатом (строительство моста, запуск ракеты) — нельзя «проверить и откатить». Здесь нужна модель «водопад» с тотальным контролем на каждом шаге.

Если ваш контекст подпадает под эти ограничения — не форсируйте PDCA. Используйте его только для тех процессов, где допустима проба и ошибка.

Теперь ваш PDCA выглядит как **спираль**, а не стрела. И это ключевое отличие моего подхода от профанации, которую продают под видом «внедрения PDCA».

Глава 3. Гипотеза вместо плана

Плохой план: «Назначить ответственного за контроль качества».

Хорошая гипотеза: «Если мы введём ежечасную самопроверку по чек-листу, то количество дефектов снизится на 30%, потому что оператор будет видеть ошибку сразу, а не через смену, тогда брак упадёт».

Чувствуете разницу?

Формула моей гипотезы (адаптирована из карты гипотез Александра Бындю под цеха и офиса)

«Если *\$воздействие\$*, то *\$изменение поведения\$*, потому что *\$мотив или устранение барьера\$*, тогда *\$влияние на метрику\$*»

Определение поведения (важное уточнение)

Поведение — это не «стал ответственным» и не «понял важность». Это **конкретное, наблюдаемое действие**, которое можно зафиксировать: взял чек-лист, нажал кнопку, задал вопрос на планёрке, сверил с эталоном, заполнил шаблон. Если вы не можете сказать «да, я видел, что он это сделал» — значит, вы планируете не поведение, а намерение.

Примеры плохих формулировок поведения: «осознал важность», «стал более ответственным», «понял ценность». Хороших: «заполнил форму в CRM», «позвонил поставщику по шаблону», «сверил деталь с эталоном на верстаке».

Пример из реального производства

Проблема: брак на сварке (7% партии).

Обычное решение: провести внеплановый инструктаж (не работает).

Гипотеза: «Если мы разместим прямо на верстаке образцы правильных и дефектных швов с подписями, то сварщик будет перед каждым швом сверяться с эталоном, потому что ему не надо лезть в телефон или звать мастера (устранение барьера), тогда процент брака снизится с 7% до 2% за две недели».

Второй кейс — из офиса (логистика)

Проблема: менеджеры по закупкам затягивают согласование спецификаций с поставщиками, сроки поставок срываются.

Гипотеза: «Если мы введём стандартный шаблон запроса на согласование с тремя обязательными пунктами (цена, срок, качество) и будем требовать его заполнения до отправки, то менеджеры станут структурировать вопросы, потому что шаблон снижает их страх пропустить важное, тогда среднее время согласования сократится с 5 до 2 дней».

Эта гипотеза — ваш план. В «До» вы просто внедряете шаблон. В «Check» замеряете время. В «Act» решаете: распространять на всех поставщиков или уточнять гипотезу.

Важно: мотив в формуле может быть как «устранение барьера» (как в примере), так и прямой выгоды («потому что это влияет на его премию»). Главное — чтобы он был правдивым и работал для конкретного человека.

Для более глубокого учёта интересов участников рекомендую использовать «карту стимулов» — она подробно описана в моей книге «Управлять хаосом на производстве» (глава 15).

Глава 4. Поведение — главный актив.

Четыре рычага

Вы когда-нибудь внедряли «обязательную» процедуру, а через месяц она умирала? Где ваши чек-листы, где эти регламенты? В мусорке.

Потому что вы забыли: устойчиво меняется только поведение, встроенное в контекст.

Мне помогла простая таблица — четыре системных рычага. Прежде чем запустить цикл PDCA, я прохожу по каждому:

РЫЧАГ	ВОПРОС СЕБЕ
Процессы и инструкции	Как новое поведение вписывается в ежедневную рутину? Оно там прописано?
Мотивация и KPI	Человеку выгодно делать по-новому? Или он теряет время и деньги?
Ресурсы и инструменты	Всё ли у него есть для нового поведения? Шаблон, доступ, кнопка, освещение?
Обучение и навыки	Он вообще умеет это делать? Или надо показать три раза?

Возьмём пример с самопроверкой сварщика

Процесс: добавить пункт «сверка с эталоном» в маршрутную карту.

Мотивация: ежемесячная премия зависит от процента брака после самопроверки.

Инструменты: эталонные образцы на верстаке, лупа, подсветка.

Обучение: 15 минут показать, как сравнивать — и всё.

Важное правило

Если вы провалили два и более рычага — не запускайте цикл. Сначала почините их.

Исключение: вы сознательно идёте на эксперимент, чтобы проверить, перекроют ли два сильных рычага один слабый. Но в этом случае вы должны чётко зафиксировать это в гипотезе. Для обычного внедрения — лучше не рисковать.

Пример провала из-за игнорирования рычагов: на одном заводе мы решили ввести ежедневные пятиминутки по качеству. Мотивацию и процесс прописали, но забыли про инструменты (не было проектора для показа дефектов) и обучение (мастер не умел проводить короткие разборы). Через неделю пятиминутки превратились в пустые посиделки. Пришлось остановить цикл, дооснастить кабинет и провести тренинг для мастера. Только после этого гипотеза заработала.

Глава 5. Что на самом деле делает руководитель в цикле?

Вот таблица, которая перевернула отношение моих директоров к PDCA.

ФАЗА	ДЕЛАЕТ СОТРУДНИК	ДЕЛАЕТ РУКОВОДИТЕЛЬ (ЭТО ВАЖНО!)
Plan	Предлагает идеи, участвует в анализе	Формулирует гипотезу, обеспечивает методологию (5 Why, диаграмма Исикавы)
Do	Выполняет запланированные действия	Создаёт условия, снимает блокеры, организует чек-поинты
Check	Предоставляет данные, факты	Анализирует вместе с командой, сравнивает с гипотезой, извлекает уроки
Act	Корректирует своё поведение (если стандартизировано)	Принимает решение: стандартизировать, скорректировать, отбросить

Чек-поинты: как они выглядят на практике

В фазе «Do» руководитель проводит короткие встречи (20 минут, раз в 2–3 дня), на которых команда отвечает на три вопроса:

Что идёт по плану?

Что пошло не так?

Что мы меняем в рамках текущей гипотезы (а не меняем саму гипотезу)?

Реальный диалог чек-поинта — позитивный сценарий

Внедрение шаблона ТЗ в IT-отделе.

Руководитель: «Итак, среда, 10:00. Мы внедрили шаблон ТЗ для менеджеров проектов. Прошло три дня. Что идёт по плану?»

Менеджер Ольга: «Я использовала шаблон для двух новых проектов. Клиенты дали обратную связь — им стало понятнее, они быстрее подписывают».

Руководитель: «Отлично. Что пошло не так?»

Менеджер Иван: «Я потратил лишние 30 минут на заполнение раздела «риски» — там много полей, и я не знаю, что писать».

Руководитель: «Это важный сигнал. Мы не меняем гипотезу, но корректируем условия — упростим раздел рисков до трёх пунктов и проведём короткий тренинг по их заполнению завтра в 16:00. Через три дня проверим, стало ли легче».

В этом примере гипотеза (шаблон ускоряет согласование) не меняется, но мы улучшаем инструмент и обучение. Это и есть микро-циклы внутри Do.

Шаблон повестки чек-поинта (раздаточный материал для команды)

Дата/время:

Участники:

Вопрос 1: Что из запланированного сделано? (факты, цифры)

Вопрос 2: С какими трудностями столкнулись? (конкретные препятствия)

Вопрос 3: Какие корректировки условий (инструменты, обучение, доступ) мы вносим?

Решение: Продолжаем эксперимент / останавливаем (если гипотеза не работает).

Следующий чек-поинт: (дата)

О том, как ваш управленческий стиль влияет на проведение чек-поинтов, читайте в главе «Два типа управления» (энергия Run и Change).

Глава 6. Карта гипотез: инструкция для этапа «План»

Я долго мучился, как научить руководителей строить хорошие гипотезы. Пока не нашёл (и переработал под свои задачи) **карту гипотез** Александра Бындю. Это простой четырёхшаговый шаблон.

Шаг 1. Цель

Формулируем **SMART**. Не «улучшить качество», а «снизить количество рекламаций с 4 до 1 в месяц за квартал». Обязательно добавляем **балансирующий показатель**: «без увеличения времени на контроль более чем на 10%».

Шаг 2. Субъекты

Чьё поведение мы хотим изменить? Не «отдел контроля», а «контролёр ОТК Анна Петровна». Опишите её **боли и желания** (Болит голова от брака на предыдущей операции? Боится пропустить дефект из-за большой партии? Хочет, чтобы хвалили?)

Шаг 3. Гипотезы

По нашей формуле: **Если... то... потому что... тогда...** Пишите три-пять альтернативных гипотез. Не одну.

Шаг 4. Задачи и рычаги

Какие конкретные действия проверим? И сразу по рычагам: что меняем в процессе, мотивации, инструментах, обучении?

Теперь у вас есть не план работ, а **дизайн эксперимента**. И когда что-то не сработает, вы не скажете «мы провалились». Вы скажете: «Гипотеза №2 не подтвердилась. Идём дальше». Это меняет культуру управления с наказания на обучение.

Пример заполненной карты гипотез (для офиса — снижение времени согласования)

Цель: Сократить среднее время согласования счетов с 5 до 2 рабочих дней без увеличения числа ошибок.

Субъект: Менеджер по закупкам Сергей. Его боль: боится пропустить сроки, перепроверяет каждый документ по 3 раза. Хочет чёткого алгоритма.

Гипотезы:

«Если мы создадим чек-лист из 5 обязательных пунктов перед отправкой счета, то Сергей перестанет перепроверять вручную, потому что чек-лист даст ему чувство завершённости, тогда время сократится до 3 дней».

«Если мы внедрим автоматическое уведомление о статусе согласования, то Сергей не будет тратить время на дозвон, потому что он будет видеть прогресс в системе, тогда время сократится до 2 дней».

Рычаги: процесс (чек-лист), инструменты (уведомления), обучение (как пользоваться уведомлениями).

Важное предупреждение

Если в вашей компании принято наказывать за любую ошибку, метод не взлетит. Культура страха убивает эксперименты. Без изменения этой культуры (а это отдельная работа с топ-менеджментом) лучше даже не начинать. Или начинайте с малого — с проверок, которые не грозят увольнением, и доказывайте пользу на мелких успехах.

Как поговорить с «директором-карателем»? Придите с коротким докладом: «У нас есть проблема X. Я хочу провести один маленький эксперимент за две недели, который не потребует бюджета и не повлияет на план. Если не сработает — мы просто вернёмся к старому способу. Но если сработает — снизим потери на Y%. Разрешите?» Чаще всего разрешают, когда нет риска. После 2–3 таких успехов можно расширять.

Глава 7. Как это выглядит в жизни: три примера и один провал

Пример 1. Успех: отдел разработки перестал срывать сроки

Возьмём болезненную тему почти любого офиса — срыв сроков. У нас был отдел, который постоянно сдавал проекты с опозданием.

Шаг 1. Наблюдение. Провели «5 почему». Выяснили: причина не в лени, а в том, что требования меняются поздно. Почему меняются? Потому что менеджеры проектов не прорабатывают их глубоко перед стартом. Почему не прорабатывают? Нет обязательного шаблона технического задания. Всё — коренная причина найдена.

Шаг 2. Карта гипотез.

Цель: снизить количество изменений требований после старта на 70% за два квартала.

Субъект: менеджер проектов. Его боль: хочет быстрее подписать договор, боится загружать заказчика.

Гипотеза: *«Если внедрить обязательный шаблон ТЗ и провести тренинг «Как собирать требования», то менеджеры станут тщательнее прорабатывать детали, потому что это снизит их личные риски переделок в будущем, тогда количество изменений упадёт».*

Рычаги: процесс (шаблон в чек-листе старта), мотивация (качество анализа в KPI), обучение (тренинг), инструмент (библиотека шаблонов в Confluence).

Шаг 3. Эксперимент (Do). Моя роль — не контролировать каждый ТЗ, а обеспечить время на тренинг, доступ к Confluence, и каждую неделю на планёрке спрашивать: «С какими сложностями вы столкнулись при заполнении шаблона?» — и сразу править шаблон (микро-циклы корректировки условий).

Шаг 4. Check. Через два квартала считаем: изменения требований упали на 65% — почти цель достигнута. Поведение: 90% проектов идут через шаблон.

Шаг 5. Act. Внедряем шаблон как обязательный стандарт компании. Обновляем регламент. Запускаем новый цикл (теперь проблема задержки данных из CRM).

Пример 2. Успех: снижение брака на литьевом участке

Проблема: брак на литьевой машине №3 — 12% за месяц. Потери — 2 млн руб.

Гипотеза: *«Если мы установим три дешёвых датчика температуры (9 тыс. руб.) и будем записывать температуру каждый цикл, то увидим корреляцию скачков с браком, потому что сейчас мы вообще не измеряем температуру».*

Эксперимент: датчики установили, попросили оператора фиксировать данные каждые 30 минут. Через 3 дня выяснили: температура падает на 15°C в конце каждого часа, когда чиллер переключается на другую форму. Решение: перенастроили расписание чиллера (бесплатно) и добавили буферную ёмкость (40 тыс. руб.). Брак снизился с 12% до 6%, потери — примерно с 2 млн до 1 млн руб. в месяц. Затраты на решение окупались за первые недели.

Урок: дешёвое измерение перед дорогими решениями — лучшее вложение.

Пример 3. Провал (и почему это полезно)

Другая проблема: высокая текучка на сборочной линии. Выдвинули гипотезу: *«Если мы введём ежемесячную премию за стаж (плюс 5% к зарплате за каждый отработанный год), то*

люди станут задерживаться дольше, потому что у них появится финансовая выгода от долгосрочной работы, тогда текучка снизится с 30% до 15% за полгода».

Запустили. Через три месяца чек-поинт: премию начисляем, люди её получают.

Поведение? Никакого изменения. Увольняются так же часто.

Что сделали? Остановили цикл. Не стали «доводить до конца». Провели быстрые интервью с увольняющимися. Выяснили: причина не в деньгах, а в том, что мастер-самодур создаёт невыносимую атмосферу.

Саморефлексия: ошибка была в том, что на этапе Plan мы не включили в субъектов мастера. Мы смотрели только на линейных сотрудников, хотя ключевое поведение (стиль управления) принадлежало мастеру. В следующем цикле гипотеза была про смену мастера или введение регулярной анонимной оценки руководителя.

Вывод для Act: гипотеза №3 (премия) не подтвердилась. Следующая гипотеза будет про смену мастера или введение анонимной оценки руководителя.

Этот провал — не поражение. Это экономия месяцев работы над не той проблемой. В старой парадигме мы бы продолжали повышать премию и удивляться: «Ну почему они всё равно уходят?»

Матрица решений фазы Check (ключевой инструмент)

Вот таблица, которую я использую при подведении итогов цикла. Она помогает принять правильное решение в Act.

ПОВЕДЕНИЕ ЗАКРЕПИЛОСЬ?	МЕТРИКА ДОСТИГНУТА?	РЕШЕНИЕ
✅ Да	✅ Да	Стандартизируем, распространяем. Гипотеза полностью подтверждена.
✅ Да	❌ Нет	Механизм не работает. Поведение есть, но оно не привело к метрике. Значит, наша причинно-следственная связь ошибочна. Запускаем новый цикл с другой гипотезой о том, как то же поведение (или новое) повлияет на метрику.
❌ Нет	✅ Да (случайно)	Скрытые факторы. Не стандартизируем , ищем реальную причину успеха. Скорее всего, сработало что-то другое.
❌ Нет	❌ Нет	Гипотеза не подтвердилась. Начинаем новый цикл с нуля с другой гипотезой.

Используйте её на каждом чек-поинте и в финале цикла.

Глава 8. Валидация и связь с Agile, Lean и Scrum

Реальные цифры за два года

Я не теоретик. Я проверил этот подход на нескольких предприятиях.

Эпизод 1. Завод металлоконструкций, 240 человек. Проблема: брак при сварке (7% от партии). Внедрили циклы с гипотезами по образцам на верстаке (как в Главе 3). Через 4 цикла (два из которых провальные) брак снизился до 2,3% за 5 месяцев. Экономический эффект — около 1,8 млн руб./год.

Эпизод 2. Отдел разработки ПО, 15 человек. Проблема: срыв спринтов. Использовали карту гипотез и чек-поинты. Из 7 гипотез подтвердились 2 (шаблон ТЗ и ежедневные 15-минутные синки). Время сдачи проектов сократилось на 22% без увеличения переработок.

Эпизод 3. Малое предприятие (мебельный цех, 12 человек). Внедрили только формулу гипотезы и рычаги, без формальных чек-поинтов. Заказчики жаловались на царапины при доставке. Гипотеза про прокладку из вспененного полиэтилена подтвердилась за 3 дня. Результат — снижение рекламаций на 80%. Руководитель сказал: «Это и так понятно, зачем книга?» Но без формулы он бы думал «надо строже наказывать грузчиков».

Где метод не сработал (честно)

Из 12 предприятий, где я внедрял лично, устойчивый результат получен на 9. Не сработало там, где:

директор наказывает за любую ошибку (даже за неподтверждённую гипотезу),
нет 2 часов в неделю на рефлексию,
проблема на самом деле кризисная (пожар, срыв поставок).

Поэтому валидация — честная: помогает многим, но не всем. И не требует веры — попробуйте один цикл на мелкой проблеме и решите сами.

PDCA — это ДНК современных методик

Многие спрашивают: «А как PDCA соотносится с Agile, Scrum, Lean? Они же сейчас в моде».

Отвечаю: **PDCA — не конкурент, а фундамент.** Вот краткая таблица:

МЕТОДИКА	КАК ИСПОЛЬЗУЕТ PDCA
Scrum	Спринт = миниатюрный цикл: планирование (Plan) → выполнение (Do) → обзор (Check) → ретроспектива (Act). Ретроспектива — это чистейший Act.
Lean (Кайдзен)	Непрерывное улучшение — это бесконечная спираль PDCA. Каждый маленький шаг улучшения — один цикл.
Теория ограничений (ТОС)	Шаги: найти ограничение → использовать → подчинить → расширить → повторить. Это тот же PDCA, только в другой упаковке.

Если ваша команда уже работает по Agile, вы всё равно делаете PDCA, просто не называете его так. Эта книга помогает делать это **осознанно** и **системно**, а не на интуиции.

Для углублённого изучения рекомендую мои другие материалы: «Неизбежное исполнение» (о стратегиях «учить, лечить, мочить») и «Управлять хаосом на производстве» (о Цикле принятия решений).

Заключение. Ваш первый эксперимент — план на неделю

Если вы прочитали книгу и закрыли её — ничего не изменится. Вот что я предлагаю сделать в ближайшие **семь рабочих дней** (не откладывайте):

День 1. Возьмите **одну проблему**, которая горит прямо сейчас (например, «опаздывают с отчётами» или «много брака на участке»).

День 2. Проведите **15 минут с командой** в формате «5 почему». Не надо решений, просто копайте.

День 3. Сформулируйте **одну гипотезу** по шаблону «Если... то... потому что... тогда...». Проверьте её по четырём рычагам.

День 4. Назначьте **чек-поинт через три дня** на 20 минут. Внедрите **минимальное изменение** — и смотрите.

Дни 5–7. На чек-поинте вы либо получите первые данные, либо поймёте, что гипотеза неверна. И то, и другое — прогресс.

Чек-лист первого цикла (держите перед глазами)

Проблема сформулирована одним предложением.

Проведено «5 почему» (или другой быстрый анализ).

Гипотеза записана по формуле.

Определены рычаги (все четыре проверены).

Запланирован пилот (один станок, одна смена, один менеджер).

Назначен чек-поинт через 3 дня.

Ответственный за эксперимент определён.

Красные флаги (критерии остановки) установлены.

Это и есть управление как эксперимент. Никакого героизма, только дисциплина циклов.

Особо для собственников и гендиректоров: если вы прочитали эту книгу и подумали «надо заставить моих руководителей так делать» — не надо заставлять. Сделайте первый цикл сами. Лично. На своей проблеме. Иначе культура экспериментов не запустится.

Я написал эту книгу не для того, чтобы вы восхитились моими схемами. А для того, чтобы вы **начали**. Даже с одного маленького PDCA. Потом будет второй, третий, и ваша компания превратится в самообучающуюся систему, где проблемы не копят, а решаются на корню. Надеюсь, у вас получится. Удачи.

Приложение. Шпаргалка, глоссарий и шаблоны

Формула гипотезы (напоминание)

Если *\$действие\$*, то *\$поведение\$*, потому что *\$мотив или устранение барьера\$*, тогда *\$метрика\$*.

Что такое поведение (коротко)

Наблюдаемое действие: взял, нажал, сверил, заполнил, спросил. Не «понял», не «осознал».

Четыре системных рычага

Процессы и инструкции

Мотивация и KPI

Ресурсы и инструменты

Обучение и навыки

Правило: если провалены два и более рычага — не запускайте цикл, сначала почините их.

Три вопроса на чек-поинте (20 минут, раз в 2–3 дня)

Что идёт по плану?

Что пошло не так?

Что меняем в рамках текущей гипотезы? (Или признаём гипотезу несостоятельной — и завершаем цикл?)

Типичные ловушки (чего не делать)

Начинать без гипотезы.

Пропускать чек-поинты.

Наказывать за неподтверждённую гипотезу (это убьёт культуру).

Игнорировать два и более рычага без осознанного эксперимента.

Пытаться внедрить метод в компании с культурой страха, не начав с изменения этой культуры.

Для малого бизнеса (до 20 человек)

Оставьте только формулу гипотезы и четыре рычага.

Чек-поинты замените ежедневным пятиминутным разговором.

Главное — не перегружать себя бюрократией.

Для компаний с токсичным топ-менеджментом

Не начинайте с больных тем.

Выберите проблему, где неудача не приведёт к оргвыводам.

Соберите 2–3 маленьких успеха, затем покажите результат и попросите разрешения расширить метод. Это путь снизу вверх.

Глоссарий (ссылки на смежные материалы)

Run и Change — два управленческих стиля: жёсткий (энергия границ) и мягкий (энергия комфорта). Подробнее в главе «Два типа управления».

Выученная беспомощность — состояние, когда сотрудники перестают проявлять инициативу из-за многолетнего наказания за ошибки. Как восстановить — см. материал «Выученная беспомощность на производстве».

Учить, лечить, мочить — три стратегии воздействия на сотрудников в зависимости от барьера (незнание, заблуждение, ригидная парадигма). Описаны в книге «Неизбежное исполнение».

Карта стимулов — анализ выгод и потерь участников изменений. Подробно — в книге «Управлять хаосом на производстве».

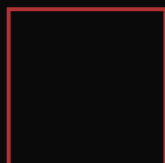
Цикл принятия решений (ЦПР) — расширенная версия PDCA для хаотичных ситуаций.

Короткая историческая справка (для любознательных)

Идея цикла восходит к **Уолтеру Шухарту** (конец 1930-х). **Уильям Эдвардс Деминг** развил её в четырёхшаговый цикл (Plan–Do–Study–Act) и популяризировал в Японии в 1950-е; название PDCA закрепилось позже. Деминг предпочитал **Study**, а не **Check**: «Check звучит как инспекция, а Study — как исследование». Современные модификации: **PDSA** (акцент на изучение) и **OPDCA** (Observe перед Plan). В этой книге мы негласно используем OPDSA, но оставили привычную аббревиатуру.

Статья целиком и рабочие артефакты — для скачивания и печати.

Печатные материалы к статье готовятся. Пока — сохраните страницу в PDF или придите за разбором в Штаб.



Изначальная идея:

Уолтер Шухарт (1891–1967) → PDCA