

САД

Контур живого решения

Открытый метод проверки решений до и после исполнения

Не система, которая решает за человека. Система, которая делает последствия видимыми до и после решения.

Человек задаёт цель и принимает решение



сервис расширяет поле зрения



решение исполняется в пределах



последствия измеряются



ошибка возвращается в пересмотр

Версия 1.0 · август 2026

1. Зачем нужен Контур

Управление часто ломается не потому, что все участники хотят вреда. Решение может приниматься в условиях страха, неполных данных, давления сроков, ведомственных стимулов, привычки к запрету и плохой обратной связи. Контур нужен не для поиска виноватого, а для того, чтобы уменьшить слепые зоны до запуска и вернуть последствия в обучение после запуска.

Он не гарантирует правильность. Он делает слепоту менее полной.

Что Контур пытается не допустить

- решение без ясно названной человеческой проблемы;
- скрытую цену, которую платят люди, отсутствовавшие при обсуждении;
- принуждение там, где есть менее ограничивающая альтернатива;
- решение без пилота при высокой цене ошибки;
- систему, где плохая новость опаснее плохого результата;
- ошибку, которую невозможно признать и пересмотреть;
- ИИ, который постепенно становится хозяином решения вместо человека.

Место в Саде

Сад задаёт вопрос «ради чего существует система?». Карта 101 показывает места, где жизнь сталкивается с лишним трением. Контур живого решения — это рабочий цикл: как принимать и пересматривать конкретное решение, не отдавая руль ни идеологии, ни автоматике.

2. Кто за что отвечает

Человек	Сервис	Затронутые люди
задаёт цель	разделяет факт и гипотезу	получают объяснение
принимает решение	ищет отсутствующих	могут сообщить о вреде
определяет допустимую цену	показывает возражение	имеют канал обжалования
несёт ответственность	ищет цену и альтернативы	не обязаны понимать устройство системы
назначает пересмотр	сравнивает прогноз и факт	не становятся материалом для скрытой оптимизации

Пять запретов

1. Сервис не принимает политическое или управленческое решение сам.
2. Сервис не скрывает от пользователя значимое возражение и не подталкивает тайно к заранее выбранной идеологии.
3. Сервис не подменяет отсутствие данных уверенной психологической, экономической или политической теорией.
4. Сервис не превращает помощь в власть: рекомендация должна оставлять возможность отказа, проверки и пересмотра.

5. Сервис не оценивает успех только отчётом системы: итоговая проверка возвращается к последствиям для людей.

3. Цикл живого решения: 12 шагов

Контур не выносит вердикт «хорошо/плохо». Он заставляет решение пройти через двенадцать вопросов, а затем возвращается к нему после появления реальных последствий.

1

Назвать решение

Вопрос: Что именно мы собираемся изменить, разрешить, запретить, профинансировать или прекратить?

На выходе: Короткая формулировка решения без лозунга.

2

Назвать человеческую трудность

Вопрос: Какую конкретную трудность человека мы уменьшаем?

На выходе: Описание одной наблюдаемой проблемы.

3

Разделить известное и неизвестное

Вопрос: Что подтверждено данными? Что является интерпретацией? Что мы пока не знаем?

На выходе: Три колонки: факт / гипотеза / неизвестное.

4

Увидеть стимулы и власть

Вопрос: Кто принимает решение, кто исполняет, кто получает выгоду и кто может скрыть плохую новость?

На выходе: Карта ответственности и стимулов.

5

Найти отсутствующих

Вопрос: Кто будет жить с последствиями, но не присутствует при обсуждении?

На выходе: Список затронутых сторон, включая слабых и зависимых.

6

Посчитать цену

Вопрос: Кто заплатит деньгами, временем, свободой, здоровьем, риском или потерей выбора?

На выходе: Карта выгод и издержек.

7

Дать сильнейшее честное возражение

Вопрос: Что сказал бы разумный человек, который не согласен с нами?

На выходе: Steelman-возражение, которое нельзя

высмеять.

8

Проверить альтернативы

Вопрос: Есть ли менее принудительный, менее дорогой или обратимый способ решить ту же проблему?

На выходе: Не менее двух альтернатив, если они существуют.

9

Ограничить риск

Вопрос: Какой малый пилот, предел, стоп-условие или независимая проверка нужны до масштаба?

На выходе: Пилот + стоп-условия + зона ответственности.

10

Зафиксировать ожидания до запуска

Вопрос: Что мы ожидаем увидеть через 30/90/365 дней?

На выходе: Предварительно записанные метрики и прогноз.

11

Запустить с человеческим выходом

Вопрос: Как человек может понять решение, пожаловаться, отказаться там, где это допустимо, и добиться пересмотра?

На выходе: Понятное объяснение + канал обжалования + ответственный.

12

Вернуть последствия

Вопрос: Что произошло в реальности? Где прогноз был неверен? Кому стало хуже? Что нужно изменить?

На выходе: Сравнение ожиданий и результата + решение о продолжении, изменении или остановке.

4. Карточка решения

Каждое решение должно уместиться в одну читаемую карточку. Если смысл невозможно объяснить без внутренней ведомственной схемы, значит человек пока не может нормально проверить решение.

Решение

Человеческая трудность

Факты

Гипотезы

Неизвестное

Кто выигрывает

Кто платит цену

Кого нет в комнате

Сильнейшее возражение	Альтернативы
Пилот	Стоп-условия
Метрики	Срок пересмотра
Ответственный человек	Канал обжалования
Результат после запуска	Что изменили после обратной связи

5. Саморастущая часть: память последствий

Система накапливает не 'правильные ответы', а проверенные уроки о последствиях.

Если новое требование сократило риск, но добавило семь посещений ведомств, следующий похожий проект автоматически получает вопрос об административной нагрузке.

Что память хранит

- исходную цель
- предварительный прогноз
- метрики
- побочные эффекты
- жалобы/сигналы
- решение о пересмотре
- причину изменения

Что она не должна хранить по умолчанию

- лишние персональные данные
- психологические ярлыки
- идеологическую оценку личности

Как появляется новый вопрос

Урок не становится догмой. Если в похожем процессе ранее возникла неожиданная цена, сервис при новом решении говорит: «В похожем случае возник такой побочный эффект. Проверьте, применимо ли это здесь». Он не говорит: «Мы уже знаем правильный ответ».

Пример

До запуска	Чтобы снизить мошенничество, вводят дополнительную справку.
Ожидание	Число ошибочных выплат снизится на 20%, срок услуги почти не изменится.
После запуска	Ошибочные выплаты снизились, но людям стало нужно три дополнительных визита, а срок вырос с 7 до 31 дня.
Урок	В следующих похожих решениях автоматически проверять административную нагрузку и возможность межведомственной передачи данных.

6. Пересмотр после действия

Контур заканчивается не публикацией решения, а возвращением к нему. В момент запуска заранее назначается дата пересмотра и человек, который отвечает за сравнение прогноза с реальностью.

- Что произошло по факту?
- Какие ожидания подтвердились?
- Где прогноз оказался неверным?
- Кому стало хуже, хотя мы этого не ожидали?
- Какие жалобы повторяются?
- Нужно продолжить, изменить или остановить решение?
- Что должно измениться в следующем похожем решении?

Ошибка не должна автоматически превращаться в стыд. Она должна превращаться в ответственность, исправление и новый урок.

7. Уровни применения

Личный. Чиновник, врач, руководитель или гражданин проверяет собственное решение.

Командный. Несколько участников видят карту фактов, возражений, цены и ответственности.

Организационный. Решения и последствия формируют внутреннюю память процессов и типовых рисков.

Публичный. Публикуются обезличенные карточки решений, метрики и причины пересмотра там, где это безопасно и законно.

8. MVP для сервиса

Вход: Текст решения/регламента/проекта + цель + контекст.

Выход: Карточка проверки из 12 шагов без вердикта 'хорошо/плохо'.

Человеческий шлюз: Перед исполнением пользователь явно подтверждает решение, пределы и ответственность.

После действия: Через заданный срок сервис запрашивает фактические результаты и создаёт карточку пересмотра.

Как выглядит интерфейс без образа оракула

- Не красно-зелёный рейтинг решения.
- Статусы: «известно», «нужно проверить», «есть возражение», «есть цена», «назначен пересмотр».
- Черновик ИИ визуально отделён от подтверждённого человеком текста.
- Неизвестное можно оставить неизвестным.
- Нет кнопки «принять решение за меня».
- История изменений сохраняет, кто и когда изменил формулировку.

9. Проверка самого Контура

Контур нельзя считать полезным только потому, что он выглядит разумно. Его тоже надо проверять.

- нашёл ли он значимую слепую зону до запуска;
- уменьшил ли число необратимых решений без пилота;

- сделал ли видимой цену для отсутствующей стороны;
- привёл ли к пересмотру после новых данных;
- не увеличил ли зависимость человека от ИИ;
- не превратился ли в ритуальную форму, которую заполняют ради отчёта.

10. Ядро

Не дать злу лёгкой дороги.
Не дать добру стать властью.
Сделать последствия видимыми.
Оставить человеку выбор.
Возвращать ошибку в обучение, а не в стыд.

Контур живого решения не обещает идеального управления. Его задача скромнее: сделать так, чтобы страх, давление, ведомственная привычка или уверенный ИИ не получали автоматического права вести человека и систему по самому лёгкому пути. Руль остаётся у людей. Реальность получает право возражать.

Открытый материал. Можно критиковать, переписывать, проверять и применять без культа автора.