

Почему эта заявка третий день не двигается?

Почему по такой же заявке полгода назад решили иначе?

На оба вопроса у вас отвечает человек, а не система.

Георгий Бах

`george.bv@verdikt-app.ru`

Коротко, если дальше читать не будете

Что происходит.	Ошибок в ваших процессах не стало больше. Дороже стало каждое объяснение. И дорожает оно от размера процесса, а не от количества ошибок.
Почему.	Ваши системы отвечают на вопрос «что произошло». На вопрос «почему выбрали это, а не другое» они дают правдоподобный неполный ответ, и полный приходится собирать вручную.
Что делаем мы.	Процесс описывается один раз: условия, шаги, сроки, обратные действия. Это описание проверяется до выпуска и исполняется как есть. Каждая операция оставляет запись, из которой видно, что выбрали, что отклонили и по какому правилу.
Что это меняет.	Разбор перестаёт быть расследованием: ответ записан в момент решения. Что изменится от правки правила, видно до выпуска. Обязательство не теряется по дороге. Исход отката назван в записи. И в такой процесс становится возможно пустить программу, не отдав ей полномочия.
Что нужно от вас.	Шесть вопросов к собственному процессу — на странице 27; отвечать на них лучше без нас.

Ошибок не стало больше. Дороже стало каждое объяснение

Процессы у вас выстроены. Записи ведутся. Ответственные назначены. Доля сбоев за последние годы, скорее всего, не выросла.

Выросло другое. Разбор одного спорного случая раньше занимал час, теперь занимает дни. Те же люди, те же данные.

Этот расход никто не считает, потому что он и не выглядит расходом. Он выглядит как «Иван Петрович посмотрит».

И растёт он не от количества ошибок, а от количества шагов, условий и людей в процессе.

Три вопроса, которые задают везде, где ошибка стоит денег

Дошло ли до конца.

Письмо клиенту не ушло. Возврат не сделан. Проверка не запущена. Обычно об этом узнают от клиента, а не от системы.

Что изменится после правки.

Порог двигают через заявку и релиз, а на что повлияет правка, узнают уже на действующих операциях.

Почему решили именно так.

Через полгода спрашивают, и ответ собирают по переписке и журналам.

Ни отрасль, ни размер здесь ничего не меняют. Меняется только цена ответа: чем больше шагов, систем и людей, тем она выше.

Пример будет не из вашей отрасли

Дальше будет пример про кофейный автомат. Четыре шага, одна кнопка, ничего общего с вашей работой.

Автомат нужен как линейка. Примерьте её к своему процессу и посмотрите, что совпадёт. Неважно, четыре у вас шага или четыреста: чем больше, тем хуже сходится.

Второй вопрос, который возникает сразу: а как это вообще будет работать у нас.

Коротко: заменять ваши системы не нужно — ВЕРДИКТ ставится рядом и отдаёт им поручения по адресам, которые задали вы. Подробно на странице 22.

Если после чтения этот вопрос всё-таки останется, значит документ сработал. Ответ на него не в файле, а в разговоре про ваш процесс.

Жалоба

«Оплатила латте 170 рублей, автомат налил чёрный кофе без молока. Я его не пью. Прошу вернуть деньги и объяснить, почему так вышло. Это уже не первый раз.»

Последнее предложение и есть задача. Вернуть 170 рублей просто. Объяснить, почему так вышло, и сделать так, чтобы не повторилось, это другое.

Смотрим, что записал автомат по этой операции. Записано всё.

14:29:14	молоко: остаток ниже порога
14:29:14	режим меню: авто-замена
14:32:01	оплата принята, 170 ₽, карта ****4417
14:32:04	напиток: американо
14:32:06	стакан выдан
14:32:31	налито, 200 мл
14:32:33	операция завершена успешно

Почему автомат налил американо?

Ответьте, прежде чем перевернуть страницу.

Вы нашли ответ. Он правдоподобный и неполный

Скорее всего вы прочитали первые две строки и собрали ответ: молоко кончилось, включилась автозамена, поэтому американо. Так и ответит поддержка. Клиент получит объяснение и деньги, обращение закроют.

ЧЕГО ЗДЕСЬ НЕТ

О том, что клиент нажимал латте, здесь ни слова. Есть только результат.

О том, что латте и капучино убрали из меню, тоже ни слова.

А кто и когда решил, что при отсутствии молока надо наливать замену, а не отказывать и возвращать деньги, здесь нет вообще.

ЧТО ЗДЕСЬ ЕСТЬ

14:29:14	молоко: остаток ниже порога
14:29:14	режим меню: авто-замена
14:32:01	оплата принята
14:32:04	напиток: американо
14:32:06	стакан выдан
14:32:31	налито
14:32:33	операция завершена

Ничего не потеряно. Ничего не сломано. Запись событий полная и точная.

Она отвечает на тот вопрос, ради которого её вели: что произошло. А спросили про решение.

Обращение закрыто. Правило продолжает работать

Клиент получил объяснение и деньги. Обращение закрыто. Никто ничего не нарушил.

Правило, из-за которого всё произошло, никуда не делось. Завтра снова кончится молоко, и снова кто-то получит не тот напиток. И так на каждом автомате.

Разбор остановился не потому, что данных не было. Он остановился потому, что нашёлся ответ, похожий на правильный.

Ваши системы редко отвечают «не знаю».

Они отвечают правдоподобно и неполно. Кто-то принимает такой ответ, закрывает вопрос, и причина продолжает работать.

«Ну так добавим туда строчку»

Разумное предложение. Пишем, какую кнопку нажал клиент, и вопрос закрыт. Три возражения, и все три не про технику.

Строчку надо было придумать заранее.

Чтобы она была там сегодня, кто-то полгода назад должен был угадать, какой вопрос зададут сейчас.

Каждая строчка это изменение в программе.

Заявка, очередь, релиз. Закрывает каждое из них ровно один вопрос.

И главное.

Запись можно прочитать. Спросить её нельзя.

ЗАПИСЬ СОБЫТИЙ — ТЕКСТ

```
14:32:04 напиток=американо; замена=да
14:33:10 напиток=американо; замена=да
14:34:02 напиток=латте;      замена=нет
14:35:21 напиток=американо; замена=да
14:36:08 напиток=американо; замена=да
... ещё десятки тысяч таких строк за март
```

ЗАПИСЬ РЕШЕНИЯ — СТРУКТУРА СО СВЯЗЯМИ

14:32:04
выбор напитка

нажато: латте

отклонено: латте, капучино — нет молока

выбрано: американо

правило «замена вместо отказа», вер. 12.03

Сколько раз за март клиент выбирал латте и получал замену?

Ответ где-то там есть, но текстом, среди десятков тысяч строк. Достать его можно, но это человек, выгрузка и вечер.

Дело не в том, что не хватает одной строки. Дело в том, что записаны события, а спрашивают всегда про решение. И записаны они текстом, поэтому любой вопрос к ним превращается в чтение.

Чем аккуратнее система, тем убедительнее неполный ответ

Скорее всего у вас не строчки в файле. У вас карточка со статусами, история изменений, таблица действий пользователей, а то и отдельный контур учёта. Всё аккуратно, всё на местах.

И всё это отвечает на тот же вопрос: что произошло. Просто гораздо красивее.

Красивый ответ хуже некрасивого. Он выглядит завершённым, поэтому разбор на нём останавливается раньше.

ЭТО ПРОВЕРЯЕТСЯ НА ОДНОМ СЛУЧАЕ

Возьмите случай, когда система что-то не пропустила или отклонила, и попробуйте ответить на три вопроса:

- 01 Видно ли, какое именно условие сработало и какие варианты рассматривались, чтобы объяснить эту блокировку человеку со стороны?
- 02 Видно ли значение, при котором условие не выполнилось?
- 03 Видно ли, какая версия правил действовала в тот день?

Если ответы находятся в системе, а не у людей, процесс уже управляем и пилот вам не нужен.

**Объяснение нельзя
восстановить задним числом.
Его можно только записать в
момент решения.**

■ АВТОМАТ

В 14:32 автомат знал
ответ целиком.

**В 14:33 его уже не знал
никто.**

Процесс описывается один раз и исполняется как описан

- 01 Условия, шаги, сроки и обратные действия описаны в одном месте, читаемым текстом, отдельно от программы
- 02 Это описание проверяется до выпуска: что противоречит, что недостижимо, что не покрыто
- 03 Исполняется ровно то, что описано. В момент решения записывается, что выбрали, что отклонили и по какой причине
- 04 Запись это структура со связями, а не строка текста: что было допустимо, что выбрано, почему не подошло остальное и по правилам какой версии

Ничего не собирается потом. Всё пишется во время работы.

■ АВТОМАТ

Так выглядит та же запись:

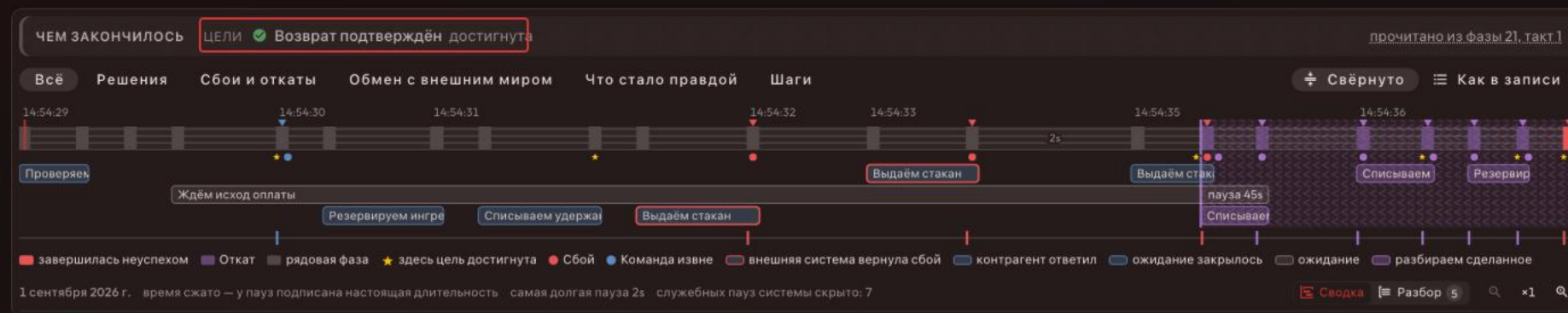
```
14:32:04  выбор напитка
          нажато: латте
          латте      отклонён: нет молока с 14:29
          капучино   отклонён: нет молока с 14:29
          американо  выбран по правилу «замена
                               вместо отказа», версия от 12.03
```

Дальше экраны продукта.

Запись операции. Тот же процесс, что и в примере.

Исход, причина и вся жизнь операции на одном экране

ЗАПИСЬ ОДНОЙ ОПЕРАЦИИ ЦЕЛИКОМ



Разбор операции без разработчика: исход, причина и вся жизнь операции на одном экране. Здесь сбой на выдаче, откат и возврат денег.

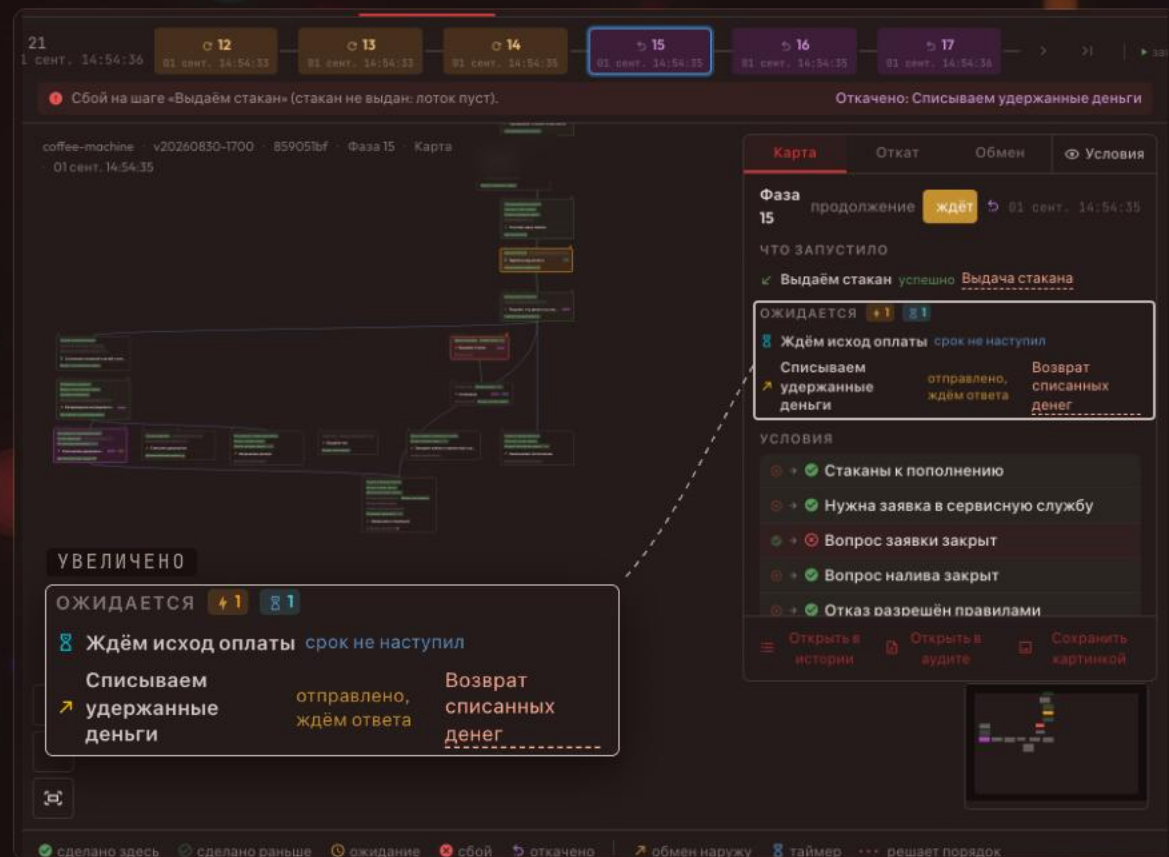
Тот самый вопрос из жалобы. Ответ виден целиком и не требует ни выгрузки, ни человека, который помнит.

«Завершена» и «достигла цели» здесь два разных ответа. Операция может дойти до конца и не сделать того, ради чего затевалась.

Процесс на выбранном шаге.

Почему операция не двигается, видно за один взгляд

ПРОЦЕСС НА ВЫБРАННОМ ШАГЕ



Это первый вопрос с обложки. Его задают каждую неделю, и обычно на него отвечает человек.

Увеличенный фрагмент показывает, чего процесс ждёт прямо сейчас: исход оплаты, срок по нему ещё не наступил, а списание удержанных денег отправлено и ждёт ответа.

Чего ждём, какие условия уже выполнены, сколько целей закрыто.

Обязательство не может потеряться по дороге

Срок это поле процесса, а не строка в регламенте.

Когда он подходит, процесс сам решает, что делать: напомнить, эскалировать, отменить. Само действие выполняет ваша система, как и любое другое.

У каждого шага есть предел числа попыток, у каждой цели свой срок.

«Зависло» перестаёт быть ощущением и становится исчерпанным пределом, у которого есть имя. Это видно в записи, а не в чьей-то голове.

Повтор идёт с растущей паузой.

Внешнюю систему, которая отвечает через раз, процесс не добивает, а ждёт дольше и пробует снова.

Пока процесс ждёт, он слушает.

«Клиент доплатил», «партия отгружена» приходят как объявленные в процессе команды и сразу меняют состояние. Ждущий процесс не глухой.

Ни один из четырёх пунктов не живёт в голове сотрудника или в напоминаниях. Всё это части описания процесса, и проверяются они там же, где всё остальное.

СРОКИ И ПРЕДЕЛЫ ВНУТРИ ПРОЦЕССА

▼

ЗАВЕРШЕНИЕ

все цели достигнуты

ЦЕЛИ

Операция завершена достигнута

Денежный вопрос закрыт достигнута

ПРОДВИЖЕНИЕ

идёт вперёд 1

БЮДЖЕТЫ

Ждём исход оплаты 1 из 10

Согласуем спорный случай с оператором 0 из 12

Списываем удержанные деньги 6 из 7

Проверяем готовность автомата 2 из 8

Завершаем операцию 1 из 4

Решаем, что делать по операции 1 из 4

Выдаём стакан 0 из 12

Заводим заявку в сервисную службу 4 из 8

Выдаём чек 0 из 12

Наливаем 0 из 8

Считаем цену заказа 1 из 4

Снимаем удержание 2 из 8

Заказываем пополнение 0 из 8

Резервируем ингредиенты 2 из 8

Промываем группу 0 из 8

Закрываем резерв 3 из 8

Срок и предел попыток это поля процесса, а не строки в регламенте. Видно, чего процесс ждёт и сколько попыток у него осталось.

Условие меняют без разработчика. Идущие операции это не ломает

ТАБЛИЦА УСЛОВИЙ: ЧТО МОЖНО НАЛИТЬ

#	ЕСЛИ	ТО	
1	Остаток молока не меньше 220 мл Остаток молока не меньше {Порог остатка молока} Остаток воды не меньше {Воды на порцию} Остаток зерна не меньше {Зерна на порцию} Остаток стаканов не меньше 1 шт	Полное меню Разрешено Почему набор такой	Латте, Капучино, Американо, Эспрессо «всё меню доступно»
2	Остаток молока не меньше {Порог остатка молока} Остаток воды не меньше {Воды на порцию} Остаток зерна не меньше {Зерна на порцию} Остаток стаканов не меньше 1 шт	Молока впритык: капучино снят Разрешено Почему набор такой	Латте, Американо, Эспрессо «молока хватает только на латте»
3	Остаток воды не меньше {Воды на порцию} Остаток зерна не меньше {Зерна на порцию} Остаток стаканов не меньше 1 шт	Без молока Разрешено Почему набор такой	Американо, Эспрессо «молока ниже порога — остаётся чёрный кофе»
4	Остаток зерна не меньше {Зерна на порцию} Остаток стаканов не меньше 1 шт	Без воды Разрешено Почему набор такой	Эспрессо «воды ниже порога — остаётся эспрессо»

Правило как таблица: условие, действие и основание решения — в одной строке. Содержимое меняют без разработчика.

Возвращаемся к вопросу, который остался без ответа. Кто решил, что при отсутствии молока надо наливать замену.

Вот оно. Поменять «предложить замену» на «отказать и вернуть деньги сразу» это одна строка в этой таблице. Не заявка в разработку.

Прикладной код при этом не выкатывается: выходит новая версия процесса, и работать она начинает, когда её включают.

Операции, которые шли в этот момент, доработали по тем правилам, которые действовали на их старте. Любые две редакции можно сравнить по делу, а не строками кода.

ВЕРСИИ ПРАВИЛА: КТО И КОГДА

Свойство: **v20260820-2046** выпущена 20 авг. 23:46:25 · 3 дня назад

После нее выпущено: 152. Ни одна из них ничего не обслуживает. [Найти в списке](#)

Показаны 20 из 152

1 / 18

Версия	Выпущена	
v20260824-0130-2	04:30:52	Свойство: v20260824-0130-2
v20260824-0130	04:30:14	Свойство: v20260824-0130
v20260824-0127-4	04:27:33	Свойство: v20260824-0127-4
v20260824-0127-3	04:27:30	Свойство: v20260824-0127-3
v20260824-0127-2	04:27:11	Свойство: v20260824-0127-2
v20260824-0127	04:27:08	Свойство: v20260824-0127

Спор про мартовский случай разбирается по мартовской версии, потому что она никуда не делась.

Содержимое условий меняют без разработчика. Новый шаг процесса или новое поле это по-прежнему работа разработчика.

Прогон случаев до выпуска.

Прежде чем менять правило, видно, что изменится

Поддержка предлагает заменить «предложить замену» на «отказать и вернуть деньги». Раньше это был спор мнений на совещании.

ПРОГОН СЛУЧАЕВ ДО ВЫПУСКА

случаи

ПРОГОН

БЫЛО / СТАНЕТ

draft

ПРОГНАТЬ СЛУЧАИ

случаев: 23

✖ есть расхождения

23
случаев

21
КАК ОЖИДАЛИ

2
РАЗОШЛОСЬ

Показать все случаи

✖ Полный автомат: латте наливается как заказан

расхождений: 4

что ответили

drink_decision

Решение о напитке

Налить drink Латте вместо Американо

Что делаем outcome Налить как заказано вместо Налить замену

Это замена substituted нет вместо да

order_total

Денежный итог

Итого amount_rub 170 Р вместо 220 Р

✖ Молоко ниже порога: латте заменён американо

расхождений: 1

что ответили

substitute_ranking

Кандидат на замену

Чем заменить drink Американо вместо Эспрессо

Набор случаев прогоняется сразу на прежней и новой версии правила — до того, как новая заработает.

Видно не «сколько процентов сломалось», а какие именно случаи разошлись и в каком поле.

Теперь это не спор, а решение по числам. И принимает его владелец процесса, не разработчик.

Цена изменения известна до выпуска — с точностью до поля.

Прогон случаев это только часть проверки. Остальное не требует случаев вообще

Часть ошибок видна прямо в описании процесса, ещё до того, как оно хоть раз исполнилось.

ОТЧЁТ ПРОВЕРКИ ОПИСАНИЯ

234 из 237 условий разобрано (99 %) НЕ РАЗОБРАНО: 3 ВХОДЫ ДЛЯ ПРАВИЛ

Разобрано: таблиц 59, правил 239, процессов 2, шагов 34, функций 60. Правил, прочитанных приблизительно: 3

0 ошибок предупреждений: 14 замечаний: 24 после черты: 37 было раньше: 0 так задумано: 1 ПРОВЕСТИ ЧЕРТУ

Раскождений с движком: 0 Не показано без примера: 0 Таблиц не удалось проверить прогоном: — Вопросы без ответа: 3

НОВОЕ (37) БЫЛО РАНЬШЕ (0) ТАК ЗАДУМАНО (1)

Предупреждение: Откат не снимает обещание шага Шаги не разобраны Остаток зеркала не больше шага 6 Таблицу не читает никто 5 Есть непокрытые входы 3 vending 12

vending_life 2 СНЯТЬ СУЖЕНИЕ

Откат не снимает обещание шага ОТКРЫТЬ
ТАК ЗАДУМАНО

Код компенсатора не пишет ни одного поля, от которого зависит обещанный признак

vending ↑

шаг: Списываем удержанные деньги `capture_payment`

Денежный вопрос закрыт `payment_accepted` Примечание: таймер

Что с этим делать

Признак-Денежный вопрос закрыт-вычисляется по полям: Платёжная система отказала `state_declined`, Срок ожидания истёк `state_expired`, Деньги удержаны `state_held`, Удержание снято `state_hold_released`, Деньги списаны `state_payment_captured`. Значит, снять обещание шага откат сможет, только если код компенсатора запишет одно из этих полей.

Есть непокрытые входы ANY Подтверждено исполнением ОТКРЫТЬ
ТАК ЗАДУМАНО

Есть вход, на который не срабатывает ни одно правило

↑ таблица: Автомату нужен сервис `service_required`

Что с этим делать

Правила молчат на входах, где

Все условия выполнены:

- Положить зерно ниже не больше (Остаток зерна)
- Положить стаканы ниже не больше (Остаток стаканов)
- Положить молоко ниже не больше (Остаток молока)
- Положить воду ниже не больше (Остаток воды)
- Режим обслуживания — **нет**

Добавить правило на этот случай или расширьте одно из существующих. Пока такого правила нет, таблица на таком входе вернёт пустое значение: «нет» у признака, «не указано» у

УВЕЛИЧЕНО

НОВОЕ (37) БЫЛО РАНЬШЕ (0) ТАК ЗАДУМАНО (1)

ПРИМЕР ВХОДА, НА КОТОРОМ ЭТО ПРОИСХОДИТ

Отчёт называет свою границу первой строкой, до всякого перечня находок.

- Правила, которые противоречат друг другу.
- Входы, на которых не срабатывает ни одно правило.
- Шаги, которые ничего не меняют.
- Цели, которых нельзя достичь.
- Сроки, которые никто не обрабатывает.
- Откаты, которые ничего не отменяют.

Каждое из этого обычно находят уже на живых операциях, по инциденту, и разбирают потом.

И одно про сам отчёт. Проверка первой строкой называет свой охват — на какие вопросы она ответила.

Чистый отчёт легко прочитать как «ошибок нет». Первая строка не даёт прочитать молчание шире, чем оно значит.

Найти все ошибки проверка не обещает, и молчание у неё не значит «всё хорошо».

Самое неприятное лежит не в правилах и не в шагах, а между ними

Откат, который ничего не отменяет.

Обратное действие описано, названо и выглядит правильным. А по его коду видно, что оно не трогает ни одного поля, от которого зависит то, что надо отменить. Узнать об этом можно только по аварии.

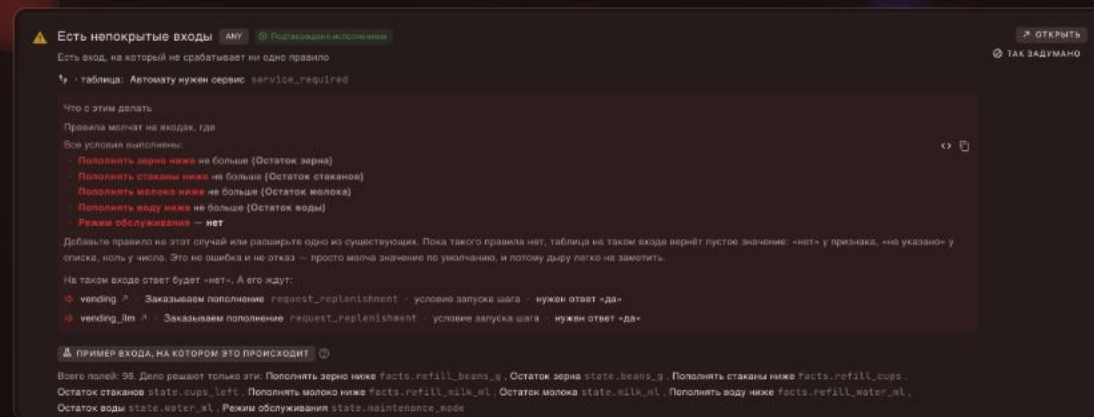
Взведён срок, у которого нет читателя.

Процесс поставил себе напоминание на третий день, а обработчика для него никто не написал. Правила в порядке. Шаги в порядке. Срок наступит, и ничего не произойдёт.

Шаг обещает результат, которого не производит.

По описанию шаг закрывает условие «оплата подтверждена», а его обработчик не записывает ни одного поля, от которого это условие зависит. Процесс будет ходить по кругу и упрётся в предел попыток.

одна находка со свидетелем



Находка, которую не видит ни один слой по отдельности.



Такие вещи не видит ни один инструмент, который смотрит на правила отдельно, на процесс отдельно и на код отдельно. Каждый из трёх скажет, что у него всё хорошо, и все трое будут правы.

Мы смотрим на это как на один предмет, потому что это и есть один предмет. Процесс описан одним артефактом, и проверяется он целиком.

Запись отмены.

Когда шаг не прошёл, отмена уже сделанного тоже записывается

ЗАПИСЬ ОТМЕНЫ: ОТКАТ ЦЕЛИКОМ

Фаза 15	14:54:35	2x от фазы 13	идёт откат	Показать в карте	Показать в истории
НАЧАТА ТЕМ, ЧТО	пришёл ответ	Выдаём стакан	запрос из фазы 14		
ОТКАТЫВАЕТСЯ ПОТОМУ, ЧТО	Выдаём стакан	этот шаг не удался			
ДАННЫЕ ИЗМЕНИЛИСЬ	+5 -3	показать			
СВОЙ	Выдаём стакан	стакан не выдан: лоток пуст			
ОТКАТ	Возвращаем списанное	выполнен			
Фаза 16	14:54:35		идёт откат	с фазы 15	Показать в карте
НАЧАТА ТЕМ, ЧТО	пришёл ответ	Списываем удержанные деньги	запрос из фазы 15		
ДАННЫЕ ИЗМЕНИЛИСЬ	+3	показать			
ОТКАТ	Возвращаем списанное	продолжен после ответа			
Фаза 17	14:54:36		идёт откат	с фазы 15	Показать в карте
НАЧАТА ТЕМ, ЧТО	подошёл повтор компенсации				
ДАННЫЕ ИЗМЕНИЛИСЬ	-1	показать			
ОТКАТ	Возвращаем списанное	выполнен			
Фаза 18	14:54:36		идёт откат	с фазы 15	Показать в карте
НАЧАТА ТЕМ, ЧТО	пришёл ответ	Списываем удержанные деньги	запрос из фазы 17		
ДАННЫЕ ИЗМЕНИЛИСЬ	+2 -1	показать			
ОТКАТ	Возвращаем списанное	продолжен после ответа			
Фаза 19	14:54:36		идёт откат	с фазы 15	Показать в карте
НАЧАТА ТЕМ, ЧТО	внутренний такт движка	начали ждать в фазе 18			
ОТКАТ	Освобождаем резерв	выполнен			
Фаза 20	14:54:36		идёт откат	с фазы 15	Показать в карте
НАЧАТА ТЕМ, ЧТО	пришёл ответ	Резервируем ингредиенты	запрос из фазы 19		
ДАННЫЕ ИЗМЕНИЛИСЬ	+2	показать			
ОТКАТ	Освобождаем резерв	продолжен после ответа			

Откат это часть процесса, со своими пределами и повторами. В записи назван его исход — включая тот, при котором он не доехал.

У каждого обратного действия своё деловое имя. Когда машина возвращает деньги, а в записи подряд идёт «Фиксируем оплату», аудитору остаётся догадываться. Когда у каждого действия своё имя, он просто читает.

Удержать, списать частично, снять удержание, вернуть это четыре разных обязательства перед клиентом, и в записи они названы четырьмя разными словами, а не одним «отменить».

Результат каждого обратного действия проверяется правилами, а не берётся на веру.

Повторная доставка приходит с тем же ключом — по нему ваша система не повторяет действие.

«А сейчас она ещё так делает?»

Спорный случай можно запустить заново на тех же данных — из Конструкторской.

А если выбор отдать языковой модели

ВОПРОС МОДЕЛИ И ОТВЕТ НА НЕГО

Фаза 5 14:54:45

Показать в карте

Показать в истории

Выбор очередного шага job-bb28b68b1e8946f66d1351194697903b

ответ в фазе 6

Ключ повторного запроса: choose_next_step

Что можно сделать дальше StepChoiceRequest

> Разрешённые позиции allowedDrinks: Американо, Эспрессо

> Разрешённые шаги allowedSteps: Налить замену, Отказаться

Контекст context: молока ниже порога — остаётся чёрный кофе

Категория клиента customerTier: Гость

Удержано heldRub: 178 Р

Что нажал клиент requestedDrink: Латте

> СТАЛО 3

Сумма удержана да — терминал удержал сумму заказа

Исход оплаты известен да — сумма удержана

Денежный вопрос закрыт нет — деньги ещё в удержании

Фаза 6 14:54:45

Показать в карте

Показать в истории

НАЧАТА ТЕМ, ЧТО пришёл ответ

запрос из фазы 5

ДАННЫЕ ИЗМЕНИЛИСЬ +7 > показать

ОТВЕТ

Выбор очередного шага

ВЫБОР ШАГА

ПОЛУЧЕН

запрос из фазы 5 · 67ms

Выбор очередного шага

ПОЛУЧЕН job-bb28b68b1e8946f66d1351194697903b

запрос в фазе 5

Что предложил решатель StepChoiceResult

Выбранный шаг chosenStep: Завершить операцию

Обоснование rationale: предлагаю просто закрыть операцию

Полномочия задаёт не модель: что сейчас уместно, считают правила до обращения к модели. В записи остаются и перечень уместного, и ответ модели.

Пусть замену подбирает модель: знает предпочтения, время суток, остатки. Ночью она решает, что раз молока нет, а клиент постоянный, надо налить двойной эспрессо и денег не брать.

Правило «не брать денег» никто не заводил. Модель придумала его сама.

Предложить такое модель может, и помешать этому нельзя ни устройством системы, ни промптом. Но своих полномочий она не определяет: что сейчас уместно, считают правила до обращения к ней.

Шаг выполняется только при сошедшихся условиях, кто бы его ни выбрал. Ответ, для которого условия не сошлись, попадает в запись вместе с причиной отказа и не выполняется.

В записи каждого выбора видно, что выбрала стратегия и как она называется. Сама модель необязательна — встроенная стратегия обходится без неё.

Модель здесь не становится предсказуемой. Предсказуемой становится граница её полномочий.

Адреса задаёте вы. Обращаемся только к ним

ВСЯ ГРАНИЦА С ВНЕШНИМ МИРОМ

Фаза 13 14:54:10 Показать в карте Показать в истории

НАЧАТА ТЕМ, ЧТО внутренний такт движка начали ждать в фазе 12

ЗАПУЩЕНО ЗДЕСЬ Наливаем ответ в фазе 14

ЗАПРОСИЛИ Наливаем ответ в фазе 14 · 55ms

Фаза 14 14:54:10 Показать в карте Показать в истории

НАЧАТА ТЕМ, ЧТО пришёл ответ Наливаем запрос из фазы 13

ДАННЫЕ ИЗМЕНИЛИСЬ +2 -5 > показать

ОТВЕТ Наливаем ПОЛУЧЕН запрос из фазы 13 · 55ms

Фаза 15 14:54:10 Показать в карте Показать в истории

НАЧАТА ТЕМ, ЧТО внутренний такт движка начали ждать в фазе 14

ЗАПУЩЕНО ЗДЕСЬ Закрываем резерв ответ в фазе 17 Выдаём чек ответ в фазе 16

ЗАПРОСИЛИ Закрываем резерв ответ в фазе 17 · 691ms

Выдаём чек ответ в фазе 16 · 627ms

Фаза 16 14:54:11 Показать в карте Показать в истории

НАЧАТА ТЕМ, ЧТО пришёл ответ Выдаём чек запрос из фазы 15

ОСТАЛОСЬ НЕЗАКРЫТЫМ Закрываем резерв запрос из фазы 15

ДАННЫЕ ИЗМЕНИЛИСЬ +2 > показать

УВЕЛИЧЕНО

НАЧАТА ТЕМ, ЧТО пришёл ответ Выдаём чек запрос из фазы 15

ОСТАЛОСЬ НЕЗАКРЫТЫМ Закрываем резерв запрос из фазы 15

Ни одного адреса и доступа к вашим сервисам, и при этом вся граница видна: что поручили, что ответили, что пришло само.

Действия во внешнем мире выполняют ваши системы — своими доступами, по поручениям ВЕРДИКТА.

Адреса задаёте вы: один на всё или по виду поручения. Обращаемся только к ним.

Ваша система выполняет поручение и возвращает исход в заранее оговорённом виде.

БАЗОВЫЙ ИНТЕРФЕЙС ИНТЕГРАЦИИ — ПЯТЬ АДРЕСОВ

POST /api/v1/sessions	запустить процесс
GET /api/v1/sessions/{id}	посмотреть, что с процессом
POST /api/v1/sessions/{id}/callback	ответ на поручение
POST /api/v1/sessions/{id}/command	сообщение извне в процесс
POST /api/v1/sessions/{id}/abort	завершить процесс

Внешняя система не знает о нас ничего. Её вызывают её собственным интерфейсом, и отвечает она туда же, куда отвечала всегда.

Вы уже за это платите. Просто не счётом

ЗА ЧТО ПЛАТИТЕ СЕЙЧАС

- За часы людей, которые восстанавливают, почему полгода назад решили именно так.
- За очередь в разработку, когда надо подвинуть один порог.
- За разбор аварии, где половина сделана, а половина нет, и кто-то подбирает хвосты руками.
- За ошибку, которая нашлась на живых операциях, потому что до выпуска её искать было нечем.

У этого расхода нет строки в бюджете, поэтому его никто не считает. Он выглядит как занятые люди.

ЧТО МЕНЯЕТСЯ

- Разбор перестаёт быть работой: ответ уже записан в момент решения.
- Изменение правила больше не ждёт очереди в разработку.
- Противоречия и пробелы в правилах находятся до выпуска, а не по инциденту.
- Исход отката назван в записи — включая тот, при котором он не доехал.

ЗА ЧТО ВЫ НЕ ПЛАТИТЕ

- Не за замену систем: они остаются на месте.
- Не за консультантов внутри вашего бизнеса: предметную область знаете вы, и мы в неё не идём.

И главное: объяснение перестаёт дорожать с каждым новым шагом — ответ записан в момент решения.

Сегодня каждый новый шаг, каждое новое условие и каждый новый человек делают его чуть дороже.

Где проходят границы

ЧЕГО МЫ НЕ ДЕЛАЕМ

- Мы не решаем за вас, каким должен быть процесс: плохо описанный будет исполнен ровно так, как описан, — и запись это покажет.
- Мы не заменяем ваши системы: действия во внешнем мире выполняют они, своими доступами.
- Однократность действия во внешней системе обеспечивает ваша система — по ключу, с которым приходит каждая доставка поручения.
- Проверка до выпуска не обещает найти все ошибки и называет свой охват первой строкой. Замысел проверяется прогоном случаев.
- Ручного перезапуска отката нет: откат, который не доехал, кончается названным исходом.
- Выпуск версии с находками проверки мы не запрещаем, но молча он не проходит: нужно явное решение, а «так задумано» — с причиной.
- Учётные записи людей и парольная политика живут у вашего поставщика удостоверений, а не у нас.
- Модель мы не удерживаем: предложить она может что угодно, а выполняется только шаг с сошедшимися условиями.
- Повторный запуск случая идёт на тех же данных и по тем же правилам. Если внешний мир с тех пор изменился, это увидит человек, а не мы.
- Граница проходит не между «кодом» и «без кода», а между описанием процесса и подключением ваших систем — оно остаётся работой на вашей стороне.

ПРО ОТЧЁТЫ

- Экраны разбора у нас есть, вы их только что видели.
- Отраслевых отчётов у нас нет: предметную область знаете вы.

ПРО ДАННЫЕ

- Решения принимаются по тем данным, которые вы передадите. Если в правилах нигде не участвует, скажем, почта клиента, передавать её не нужно.
- Смысл ваших данных остаётся на вашей стороне: мы в него не заходим, а запись отвечает на другой вопрос — как шло дело.
- Для этого есть отправка записей в вашу систему — она включается заданным адресом, и отчёты вы строите там.
- Всё, с чем работает процесс, объявлено в контракте данных: чего там нет, того для процесса не существует.
- На экране разбора видно то, что попало в запись. Кому открыт этот экран, определяете вы.
- Ставится у вас, из архива: без исходников и без инструментов сборки.

■ АВТОМАТ

Для решения автомату **не нужен номер карты**. Достаточно того, что оплата прошла.

Что меняется, в зависимости от того, за что вы отвечаете

ОПЕРАЦИОННЫЙ ДИРЕКТОР

Обязательство не теряется по дороге: срок и предел попыток живут внутри процесса. Про вставшую операцию вы узнаете по сроку, а не по звонку клиента.

ФИНАНСОВЫЙ ДИРЕКТОР

Каждое обратное действие названо своим именем; исход отмены виден в записи, даже когда она не доехала. Повторная доставка несёт тот же ключ — по нему ваша система не повторяет действие.

ВЛАДЕЛЕЦ ПРОЦЕССА, ГЛАВБУХ

Порог меняется без разработчика, и до выпуска видно, что изменится. Начатые операции правка не задевает, а новая находка проверки не пройдёт выпуск молча.

СОБСТВЕННИК

Разбор не останавливается на первом правдоподобном ответе: основание решения записано. И появляется процесс, в который можно пустить программу, не отдав ей полномочия.

■ АВТОМАТ

Пока автомат один, всё это держит в голове наладчик.

Вопрос в том, сколько у вас автоматов.

Четыре шага, одна кнопка

Вы смотрели на полную и точную запись.

Вы смотрели столько, сколько хотели.

Вы нашли ответ, и он оказался неполным ровно в том месте, где начиналась причина.

Это четыре шага и одна кнопка. Без всяких умных технологий.

Сколько шагов в процессе, за который вы отвечаете? И сколько таких процессов?

И кому вы собираетесь доверить решения в них, если объяснить эти решения вы пока не можете?

Шесть вопросов к вашему процессу. Отвечать лучше самостоятельно

Возьмите один процесс, где принимаются решения. Пройдите по вопросам. Неудобный ответ полезнее удобного.

- 01 На прошлой неделе что-то должно было произойти и не произошло. Вы узнали об этом из системы или от клиента?
- 02 Возьмите спорный случай прошлого квартала. Сколько людей и сколько часов понадобилось, чтобы объяснить, почему решили именно так?
- 03 Кто в компании может сказать, какая версия правил действовала в тот день?
- 04 Когда правило меняют, кто видит, что изменится, до того как оно заработает?
- 05 Сколько сейчас в ваших правилах условий, которые противоречат друг другу или до которых никогда не доходит очередь? Чем это проверить?
- 06 Если система что-то заблокировала или программа приняла решение вместо человека, чем ограничен набор разрешённых действий и где записано, почему выбрали именно это?

Если на четыре из шести ответ «зависит от конкретного человека», значит тот самый растущий расход у вас есть.

Один вопрос напоследок

**Вспомните последний раз, когда пришлось
объяснять уже принятое решение или искать
обязательство, которое не выполнилось.**

Сколько человек в этом участвовало и сколько времени заняло?

Назовите процесс и контакт — вернёмся с разбором и планом пилота.

А если отвечать не хочется, пройдите шесть вопросов с предыдущей страницы.
Они полезны и без нас.

Георгий Бах

george.bv@verdikt-app.ru



Георгий Бах

Основатель ВЕРДИКТА

20 лет в ИТ, 15 на руководящих позициях. Последние шесть техническим директором компании в кибербезопасности: два продукта для корпоративных клиентов, оба с нуля до продаж.

Проекты были разные по масштабу и отрасли, проблема везде одна: решение принято, причина нигде не осталась. Через полгода её собирают заново по переписке и логам. Из этого вырос ВЕРДИКТ.

ВЕРДИКТ — от описания до объяснения

Порядок подключения

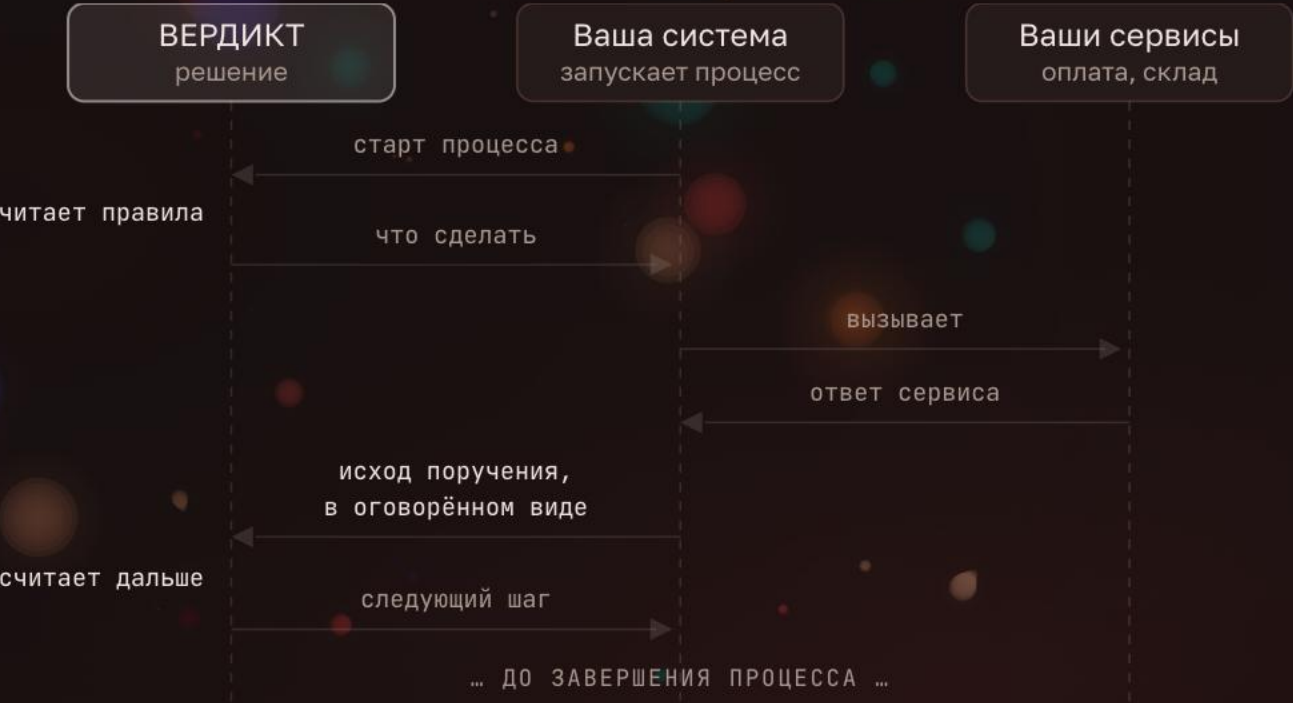
Для технического человека, которому перешлют файл. Главное здесь одно: адреса задаёте вы, и обращаемся мы только к ним.

БАЗОВЫЙ ИНТЕРФЕЙС ИНТЕГРАЦИИ

POST /api/v1/sessions	запуск процесса
GET /api/v1/sessions/{id}	состояние процесса
POST /api/v1/sessions/{id}/callback	ответ на поручение
POST /api/v1/sessions/{id}/command	сообщение извне
POST /api/v1/sessions/{id}/abort	завершить процесс

/callback и **/command** это прикладные входы: у них есть смысл в предметной области, «поручение выполнено», «терминал ответил».

ПОРЯДОК ОБМЕНА



Решатель, отправка записей и «мы напишем это сами»

ПРО ВНЕШНИЙ РЕШАТЕЛЬ

- Выбор действия можно передать наружу, включая языковую модель. Решатель получает список действий, допустимых по правилам на этот момент, и возвращает свой выбор. Список — не рекомендация, а входные данные; исполнение шага всё равно требует сошедшихся условий.
- Ответ решателя приходит событием и записывается — вместе с допустимым набором и сделанным выбором.
- Молчание решателя ограничено объявленным сроком и кончается названным исходом, а не ожиданием без конца.

ПРО ОТПРАВКУ ЗАПИСЕЙ

- Записи можно отправлять во внешнюю систему — это включается заданным адресом. Отчёты и отраслевая аналитика строятся там, вашими средствами.

ЧТО ОСТАЁТСЯ НА ВАШЕЙ СТОРОНЕ

- Вызов ваших сервисов. Относительные сроки отсчитываем мы; календарных моментов вида «в 9:00» у нас нет как примитива.

Если возникает мысль «мы напишем это сами»

Скорее всего уже написали. В каждом сервисе по-своему, разными людьми, в разные годы. Именно поэтому объяснить решение, прошедшее через четыре сервиса, сегодня и нельзя.

Дорого здесь не написать. Дорого договориться об одной форме записи между командами, а потом удержать её, когда пойдут сроки. Обычно эта договорённость не переживает первый горящий релиз.

Из вашей карточки — в причину

ВАШ ЭКРАН И ЗАПИСЬ ЗА НИМ

ВАША СИСТЕМА

ORD-14-32-04

заказ

Нажали

Латте 170 Р

Налито

Американо 200 мл

Списано

120 Р

Чек

ФД-2233

Статус

ГОТОВО

Запись процесса

837ee582...1ea23 >

Почему налили не то, что нажали?

Кто разрешил замену?

Почему списали 120 Р, а удержали 170 Р?

Карточка не отвечает ни на один. Нижняя строка — всё, что добавляет интеграция.

837ee582

→

ключ — идентификатор записи

ВЕРДИКТ

Запись 837ee582...1ea23

ваш номер ORD-14-32-04

26 фаз · 2 цели закрыты · 2 команды извне

Что на них отвечает запись

Молока меньше порога пополнения

остаток 60 мл

дано на входе

Нажатой позиции нет; заменяем: чёрный кофе того же объёма

решили правила · фаза 6

Уровень внимания требует человека

уровень 3

фаза 6

Оператор согласовал

Ковалёв: «замена уместна, наливаем»

команда извне · фаза 9

Удержание превращено в списание

120 Р вместо удержанных 170 Р

фаза 13

★ Налита замена: Американо

цель закрыта · фаза 25

В той же записи и попутное: заявка на пополнение RPL-3204 и закрытый резерв RSV-3204.

Ни одной строки этой записи вы не программировали.

Интеграция добавляет в вашу карточку одно поле — идентификатор записи.