Uber выжег годовой AI-бюджет за четыре месяца: тормоза проектируют до запуска

2026-06-07

Uber выжег годовой AI-бюджет за четыре месяца: тормоза проектируют до запуска

В мае 2026 года технический директор Uber публично признал, что компания израсходовала весь свой годовой бюджет на AI-кодинг за четыре месяца. По разбору Forbes, средний инженер тратил 150–250 долларов в месяц, а у тех, кого внутри называли power-users, счёт доходил до 500–2000 долларов в месяц на человека — и при масштабе инженерной организации Uber это сложилось в перерасход, после которого компания, по словам её же CTO, вернулась «к чертёжной доске» (Forbes). История разлетелась по деловым лентам как анекдот про дорогие токены. Это неверное прочтение.

Цена токена у Uber была ровно такой же, как у любой другой компании с тем же контрактом. Разъехалась не цена — разъехалась архитектура. Инженерам выдали мощный агент без квот, без маршрутизации задач по моделям разного тиража, без обязательного кэширования и без внутреннего учёта, кто сколько жжёт. В такой конфигурации расход — это не строка в смете, а открытый кран, и единственный сигнал о том, что кран открыт, приходит в виде месячного счёта. Проблема Uber не финансовая, а проектная: контроль расходов на AI (далее — FinOps) у них появился после прода, а должен был появиться до первой строки кода.

Почему счёт приходит в виде шока

Чтобы увидеть, что здесь сломано, полезно отделить цену модели от стоимости контура. Это разные величины, и путаница между ними и порождает шоковые счета.

Цена модели — это публичный прайс-лист: сколько стоит миллион входных и выходных токенов. Она известна заранее и одинакова для всех. Стоимость контура — это сколько денег в год съест работающий агент с учётом объёма запросов, длины контекста, числа повторов, доли кэш-попаданий и количества людей, которые им пользуются без ограничений. Эта величина не лежит в прайс-листе. Она вытекает из того, как контур спроектирован, и именно поэтому её почти никто не считает до запуска.

Порядок этой величины уже измерен на стороне. По оценке SearchUnify, совокупная стоимость владения одним корпоративным AI-агентом в первый год составляет 108–306 тысяч долларов: капитальные затраты на разработку 70–150 тысяч, операционные расходы 3,2–13 тысяч долларов в месяц, и сверх этого ежегодное обслуживание, которое съедает 15–25% от стоимости разра-

ботки. Последняя цифра — самая недооценённая: агент не строится один раз, он требует постоянной подгонки к меняющимся API, схемам данных и регламентам, и эта подгонка стоит как четверть исходной разработки каждый год. Команда, которая заложила в бизнес-план только цену токенов, не заложила три четверти реальной стоимости.

Дальше работает простая арифметика без тормозов. Современный агент в задаче кодинга прогоняет через модель не один запрос, а десятки итераций: читает файлы, держит в контексте историю, перечитывает её на каждом шаге. Без кэширования каждая итерация оплачивает весь контекст заново по полной цене входного токена. Один инженер, гоняющий агент по крупной кодовой базе восемь часов в день, легко выходит на те самые 2000 долларов в месяц — не потому что токен дорогой, а потому что один и тот же контекст оплачивается сотни раз. Умножьте на масштаб организации, в которой, по данным того же разбора, заметная доля бэкенд-кода писалась агентами фактически без человека в цикле, — и годовой бюджет действительно сгорает к маю.

Четыре тормоза, которые ставят на чертеже, а не после счёта

FinOps как дисциплина — это не «следить за облачным счётом постфактум». FinOps Foundation определяет её как операционную модель, в которой видимость затрат, квотирование и unit-экономика встроены в принятие инженерных решений, а не приклеены сверху (FinOps Foundation). Для AI-нагрузок та же логика переносится почти дословно: фонд выделил отдельную категорию FinOps for AI именно потому, что у токенов сложность затрат, скорость роста и непредсказуемость выше, чем у классической облачной инфраструктуры. Четыре механизма из этой дисциплины проектируются вместе с агентом, а не докручиваются после.

Первый — квоты. Лимит расхода на инженера, на команду, на проект в единицу времени. Это самый дешёвый тормоз и тот, отсутствие которого напрямую стоило Uber годового бюджета. Квота не обязана быть жёсткой отсечкой; достаточно мягкого потолка, после которого запрос уходит на согласование или на более дешёвую модель. Принципиально то, что квота — это архитектурное решение на входе: если её не заложили в шлюз доступа к модели с самого начала, добавить её после того, как все привыкли к безлимиту, — это уже не настройка, а отъём, на который команда реагирует так же болезненно, как на любое урезание привилегий. FinOps Foundation выносит политики и контроль в отдельную способность фреймворка не случайно: без них видимость затрат остаётся отчётом, который читают, но на который не реагируют (FinOps Foundation).

Второй — маршрутизация по тиражам моделей. Не каждая задача требует топовой модели. Классификация письма, извлечение полей из документа, простая правка — это работа для дешёвой быстрой модели, которая стоит в разы меньше. Разрыв в публичных прайс-листах огромен: между флагманской и компактной моделью одного и того же провайдера разница во входной цене

легко достигает порядка (OpenAI). Контур, который гонит всё через топовую модель, переплачивает за каждую тривиальную операцию. Маршрутизация — это слой, который смотрит на задачу и отправляет её на самую дешёвую модель, справляющуюся с ней; он либо встроен в архитектуру как явный компонент, либо его нет, и тогда «всё через флагман» становится дефолтом по умолчанию. Этот же сдвиг описан и для смешанных стеков: появление компактных моделей высокого качества и дешёвых альтернатив пограничного уровня сделало маршрутизацию архитектурным преимуществом, а не просто строкой экономии.

Третий — кэширование. Главный рычаг, доступный бесплатно и почти всегда недоиспользованный. Когда агент перечитывает один и тот же системный промпт, ту же документацию, ту же историю на каждом шаге, кэширование позволяет не оплачивать этот повторяющийся контекст по полной цене. В документации Anthropic кэш-чтение стоит существенно меньше обычного входного токена — на типовых конфигурациях речь идёт о порядковой экономии на повторяющейся части контекста (Anthropic). Для агента, который по своей природе крутит длинный стабильный контекст в цикле, это разница между жизнеспособной и сгорающей экономикой. Но кэширование требует, чтобы промпт был структурирован под кэш — стабильная часть отделена от изменчивой, — а это решение на этапе проектирования промпта, а не переключатель, который щёлкают после первого счёта.

Четвёртый — внутренний учёт. Биллинг по командам и проектам, который превращает безличный общий счёт в персональную ответственность. Пока расход анонимен, он растёт: никто не оптимизирует то, за что не отвечает. Как только каждая команда видит свою строку и отвечает за неё, поведение меняется само — без квот и запретов, просто потому что появилась видимость. Именно это FinOps Foundation называет основой дисциплины: не урезание, а перенос экономической ответственности туда, где принимается техническое решение. Контур без внутреннего учёта узнаёт свою экономику единственным способом — когда приходит общий счёт, и уже поздно спрашивать, кто его наполнил.

Что говорят данные о цене «всё через флагман»

Решение на этапе проектирования	Контур с FinOps на чертеже	Контур без тормозов
Квоты на инженера/команду	Заложены в шлюз доступа к модели	Появляются после шокового счёта как отъём привилегий
Маршрутизация задач	Тривиальное уходит на дешёвую модель	Всё идёт через флагман по умолчанию
Кэширование контекста	Промпт структурирован под кэш с первого дня	Повторяющийся контекст оплачивается заново каждый шаг
Внутренний учёт расходов	Биллинг по командам, персональная ответственность	Анонимный общий счёт, никто не оптимизирует
Когда команда узнаёт экономику	На этапе проектирования, в смете	В мае, когда сгорел годовой бюджет

Контр-аргумент в защиту моноархитектуры звучит разумно: на старте проще пустить всё через одну топовую модель, не строить маршрутизатор, не возиться с кэшем — и быстрее выйти на работающий прототип. Это правда, и на горизонте первого прототипа это даже рационально. Проблема в том, что эта простота берётся в долг. Прототип без тормозов незаметно переходит в прод без тормозов, а на масштабе долг гасится с процентами — в виде того самого счёта, после которого приходится возвращаться «к чертёжной доске» и встраивать FinOps в уже работающую систему, где каждая правка дороже, чем она была бы на чертеже. Эта же арифметика себестоимости определяет, какая модель оплаты контура вообще выживает на дистанции (Подписка против проекта: три класса экономики B2B-агентов).

Этот сюжет рифмуется с более широкой картиной корпоративного AI. Исследование MIT NANDA зафиксировало, что 95% корпоративных AI-пилотов не дали измеримого возврата через полгода, несмотря на 30–40 млрд долларов совокупных вложений (MIT NANDA). Среди причин — не слабость моделей, а то, что экономика контура ломается на архитектуре раньше, чем на модели. Кейс Uber — это та же болезнь в зеркальном отражении: там агент как раз работал и давал отдачу, но архитектура без экономических ограничений превратила рабочий инструмент в неуправляемую статью расходов. В обоих случаях граница между успехом и провалом проходит не по качеству модели, а по тому, что построено вокруг неё.

Дополнительный сигнал того же порядка — движения крупных игроков по управлению доступом к агентным инструментам. Сообщалось, что Microsoft пересматривает внутренние лицензии на сторонние агенты для кодинга в

пользу собственного стека к середине 2026 года. Независимо от деталей, направление читается одинаково: после первой волны безлимитного внедрения корпорации возвращают контроль над тем, кто, чем и в каком объёме пользуется. Это и есть запоздалый FinOps — дисциплина, которую дешевле было заложить в проект, чем вводить под давлением счёта.

Что проверить инженеру и что — руководителю

Из всего сказанного следуют два разных набора проверок — для того, кто строит контур, и для того, кто за него платит.

Инженеру, проектирующему агента. Первый тест — на кэш: посмотрите долю кэш-попаданий в реальной нагрузке. Если стабильная часть контекста (системный промпт, документация, инструкции) не вынесена в кэшируемый префикс и доля попаданий низкая — вы оплачиваете один и тот же контекст десятки раз, и это первое, что чинится без потери качества. Второй тест — на маршрутизацию: пройдитесь по типам запросов и честно ответьте, какая доля из них реально требует флагманской модели; если простая классификация и извлечение полей идут через топовую модель, контур переплачивает на ровном месте. Третий — на квоты: существует ли в шлюзе доступа к модели хоть какой-то потолок на инженера или проект, и что произойдёт, если один пользователь за ночь упрётся в десятикратный обычный расход. Если ответ «ничего не произойдёт, кроме счёта» — тормоза не спроектированы.

Руководителю, который санкционирует бюджет. Первый тест — потребовать в смете не цену токена, а годовую стоимость владения контуром: разработка плюс операционные расходы плюс 15–25% ежегодного обслуживания. Если в плане есть только цена API — план занижен в разы, и шоковый счёт уже заложен, просто ещё не пришёл. Второй тест — на видимость: может ли каждая команда увидеть свою строку расхода на AI отдельно от общей; если расход анонимен, оптимизировать его никто не будет, и рост счёта — вопрос времени. Третий — на обратимость решения «всё через флагман»: спросите, что будет стоить ввести маршрутизацию и квоты через полгода работы без них; если ответ «придётся переписывать ядро контура и ломать привычки команды» — значит, дешевизну взяли в долг, и пора отдавать до того, как набегут проценты.

За чем смотреть в 2026 году

Первый сигнал — появление в корпоративных AI-бюджетах отдельной защищённой строки на FinOps: квоты, маршрутизацию, учёт. Пока её нет, история Uber будет повторяться у каждого, кто проходит фазу безлимитного внедрения. Когда такая строка станет стандартом сметы, рынок усвоит урок про тормоза на чертеже.

Второй сигнал — смещение разговора у поставщиков с «насколько умна наша модель» на «как контролировать расход на нашей модели»: появление встроенных квот, бюджетных лимитов и пер-командного биллинга в управляемых средах. Когда контроль расходов станет частью продукта, а не самодельной

обвязкой клиента, это будет означать, что отрасль признала: экономика AI-контура решается архитектурой, а не прайс-листом — и решается до запуска, а не после счёта.

Главное

- Uber израсходовал годовой AI-бюджет за четыре месяца не из-за дорогих токенов, а из-за архитектуры без квот, маршрутизации, кэширования и внутреннего учёта; цена токена была у всех одинаковой.
- Годовая стоимость владения одним агентом оценивается в 108–306 тысяч долларов, а обслуживание съедает 15–25% стоимости разработки ежегодно — три четверти этой суммы не видны тому, кто считает только цену API.
- FinOps для AI — не пост-продакшн-оптимизация, а проектное ограничение: квоты, маршрутизация по тиражам моделей, кэширование и пер-командный биллинг встраиваются в архитектуру с первого дня или превращаются в шоковый счёт.
- Дешевизна схемы «всё через топовую модель» берётся в долг: на прототипе она рациональна, на масштабе гасится с процентами в виде перерасхода и дорогой переделки уже работающего контура.

FAQ

Что такое FinOps для AI и чем он отличается от обычного контроля облачных расходов? FinOps — операционная модель, в которой видимость затрат и экономическая ответственность встроены в инженерные решения, а не приклеены постфактум. Для AI выделена отдельная категория, потому что у токенов выше сложность учёта, скорость роста расходов и непредсказуемость: один и тот же контекст может оплачиваться сотни раз, а безлимитный доступ масштабирует счёт быстрее любой классической инфраструктуры.

Почему кэширование называют главным рычагом экономики? Агент по своей природе крутит в цикле длинный стабильный контекст — системный промпт, документацию, историю. Без кэша эта повторяющаяся часть оплачивается заново на каждом шаге по полной цене входного токена. Кэш-чтение стоит существенно дешевле обычного входа, поэтому при правильно структурированном промпте экономика на повторяющейся части сжимается в разы — без потери качества ответа.

Когда маршрутизация по моделям не нужна? Если весь поток задач контура действительно требует максимального качества рассуждения — например, узкий контур только на сложном синтезе, — выигрыш от маршрутизации мал. Но в большинстве реальных нагрузок заметная доля запросов тривиальна (классификация, извлечение полей, простые правки), и для них флагманская модель — переплата. Чем разнороднее поток, тем сильнее окупается маршрутизация.

Сколько на самом деле стоит один агент в первый год? По оценке SearchUnify — 108–306 тысяч долларов совокупной стоимости владения: 70–

150 тысяч на разработку, 3,2–13 тысяч долларов в месяц операционных расходов и сверх этого 15–25% стоимости разработки ежегодно на обслуживание. Точная цифра зависит от объёма нагрузки и сложности контура, но порядок показывает: цена API — меньшая часть счёта.

Как измерить, что контур спроектирован экономно? Три метрики: доля кэш-попаданий на повторяющемся контексте (чем выше, тем лучше), доля запросов, уходящих на модель ниже флагмана (показывает, работает ли маршрутизация), и наличие пер-командного биллинга с квотами в шлюзе доступа. Если все три на нуле — контур не имеет тормозов, и шоковый счёт лишь вопрос масштаба и времени.