



ИССЛЕДОВАТЕЛЬСКАЯ СТАТЬЯ

Редакционная дата: 22 октября 2026 года

От отчёта к исполнимому контракту: концептуальное и вычислительное исследование экспортируемого знания о программных отказах

From Incident Reports to Executable Contracts: A Conceptual and Computational Study of Exportable Software-Failure Knowledge

Научная классификация: концептуальная исследовательская работа, дополненная ограниченным обзором первичных свидетельств и программным демонстратором, исполненным на синтетических случаях и средах. **Выполнено:** извлечение указанных источников, построение модели, написание и запуск кода, проверка файлов результатов. **Не выполнено:** надёжное автоматическое извлечение из реальных отчётов языковой моделью, испытания продуктов клиентов, фактическое экономическое исследование саудовской компании, независимый слепой тест или научное рецензирование.

Аннотация на русском языке

Работа исследует, можно ли превратить документированный опыт происшествий в программных и распределённых системах в исполнимый контракт отказа (Executable Failure Contract) — технический актив, который допускает перенос и продажу за пределами страны разработки. Региональная нагрузка на государство сама по себе не

считается источником экспортной отрасли. Вместо этого разделяются четыре уровня: описание происшествий; ограниченно обобщаемый механизм отказа; исполнимый тест с проверяемым правилом решения; доказательство того, что иностранный покупатель получает ценность, превышающую стоимость установки и обслуживания. Обзор связывает три документированных случая: отказ Cloudflare 18 ноября 2025 года из-за удвоения файла свойств после изменения прав доступа; сбой Google Cloud 12 июня 2025 года из-за данных о квотах с пустыми полями и перегрузки при восстановлении; отключение AT&T Mobility 22 февраля 2024 года после сетевой ошибки конфигурации с широкими последствиями. Затем выполняется синтетический опыт с 16 тестовыми контрактами и четырьмя специально созданными реализациями — всего 64 результата. Число совпадений с эталонными исходами составляет соответственно 7, 12, 11 и 16. Эти результаты предопределены конструкцией программы и контрактов; они не измеряют эффективность в производственной среде или вероятность предотвращения отказа. Коммерческая гипотеза, требующая независимой проверки, состоит не в применении искусственного интеллекта (Artificial Intelligence; AI) для инженерии хаоса (Chaos Engineering) — такая практика уже существует, — а в измерении смыслового переноса (Semantic Transfer) и тестового оракула (Test Oracle) между разными средами, стоимости такого переноса и повторного спроса покупателей.

Ключевые слова: исполнимые контракты отказа (Executable Failure Contracts); смысловой перенос (Semantic Transfer); тестирование на основе свойств (Property-Based Testing); ограниченный эксперимент (Bounded Experiment); программное обеспечение вывода (Inference Software); инженерия хаоса (Chaos Engineering); экономика единицы продукта (Product Unit Economics); экспорт программных услуг (Software Services Exports).

English abstract

This conceptual and computational working paper examines whether documented software failures can be transformed into portable, independently inspectable test artifacts with exportable commercial value. It distinguishes incident evidence, failure-mechanism abstraction, executable-oracle construction, cross-system semantic transfer, and demonstrated foreign willingness to pay. Three primary records anchor the inquiry: Cloudflare's November 2025 feature-file propagation outage, Google Cloud's June 2025 quota-policy failure and recovery congestion, and the February 2024 US AT&T mobile-network outage investigated by the FCC. A reproducible, fully synthetic Python demonstration manually specifies 16 toy test contracts spanning input schemas, bounded feature files, authenticated versioning and retry limits. Four intentionally constructed toy adapters yield 64 executions and respectively match 7, 12, 11 and 16 declared oracle outputs. The exercise demonstrates contract encoding and repeatability, not predictive incident prevention or product-market fit; all outcomes are

consequences of artificial design choices. The paper proposes prospective tests for cross-platform semantic transfer, oracle accuracy, false positives, incremental value over established chaos-testing products, integration cost and actual foreign renewals. The central claim is a falsifiable opportunity hypothesis, not evidence that a Saudi deep-tech resilience-export industry already exists.

Keywords: executable incident contracts; semantic transfer; chaos engineering; test oracle; fault injection; Saudi technology exports; reproducibility; product-market fit.

1. Научный вопрос и последствия ответа

Литература о надёжности систем призывает понимать отказы и готовиться к ним, но не доказывает существование самостоятельного рынка для всякого нового инструмента, обещающего устойчивость. «Экспортная отрасль» здесь означает нечто экономически более определённое, чем способность местной инженерной команды решать собственные задачи. Проверяется следующая гипотеза:

Некоторые виды знания о программных отказах можно извлечь из документированных происшествий или разрешённых сред, представить как сравнительно независимые от поставщика тестовые контракты, переносить между системами разных клиентов с приемлемой предельной стоимостью и многократно продавать покупателям за пределами Саудовской Аравии, которые платят за доказанную инженерную пользу.

Это составная гипотеза, и подтверждение одного звена не доказывает её целиком. Тест может быть технически исполним, но не переносить смысл между платформами. Перенос может требовать дорогой настройки в каждом проекте. Команда может продавать экспертные услуги за рубеж, не создавая повторяемого продуктового бизнеса. Ответ влияет на решение о специализации и инвестициях: финансировать ли лабораторию адаптеров и повторно используемых спецификаций, вложиться ли в специализированный консалтинг или ограничиться существующими инструментами.

Научная единица анализа: конкретный тестовый контракт, исполненный на определённой системе при заданных условиях и сопоставленный с объявленным оракулом. **Коммерческая единица анализа:** контракт независимого иностранного клиента на продукт или его продление в пределах определённого цикла продажи и обслуживания. Эти единицы не равнозначны.

2. Что уже существует до предлагаемого проекта?

2.1 Опыты с отказами уже существуют и не являются предполагаемым саудовским изобретением

«Принципы инженерии хаоса» (Principles of Chaos Engineering) описывают проверку гипотезы об устойчивом поведении, опору на наблюдение и уменьшение радиуса воздействия (Blast Radius). Они не утверждают, что опыт можно построить по публичному отчёту без знания системы. AWS Fault Injection Service (AWS FIS) с 2023 года предлагает библиотеку сценариев, описанную компанией в руководстве 2024 года [S06](#). Сервис поддерживает условия остановки, связанные с конкретными сигналами мониторинга, и не сводится к выключению машин [S08](#). В 2025 году Amazon опубликовала инженерный пример, где большие языковые модели (Large Language Models; LLMs) создают шаблоны опытов из естественно-языковых описаний внутри её среды [S07](#). Поэтому утверждение о новизне простого «перевода текста в шаблон теста» или отсутствии конкуренции противоречит имеющимся свидетельствам.

Проект Jepsen с 2013 года публикует проверки согласованности распределённых баз данных, очередей и систем сообщений. Он использует реальные реализации, генерирует операции и отказы — в том числе разделение сети и сдвиг времени, — а затем проверяет историю результатов по объявленным моделям согласованности [S09](#). Возможное дополнение следует определить точнее: единое представление библиотеки случаев, обновляемой по новым происшествиям; смысловая проверка между поставщиками; либо воспроизводимое доказательство для покупателя. У каждого варианта иные конкуренты, а рыночный пробел нельзя установить без анализа закупок и сравнительных тестов.

2.2 Три семейства механизмов отказа в исходных свидетельствах

Рамка опирается на следующие записи; они не образуют репрезентативной выборки частоты технологических отказов:

Происшествие	Что документировал источник	Предлагаемый абстрактный механизм	Что остаётся неизвестным
18 ноября 2025 года, Cloudflare	После изменения прав доступа результаты запроса различались, файл свойств удвоился, превысил предел считывателя, а сервис работал с перебоями S01	Неожиданный рост промежуточного распределённого результата при скрытом предположении считывателя о размере	Достаточно ли удалить повторы в другой системе? Нет: смысл и порядок элементов могут различаться
12 июня 2025 года, Google Cloud	Неполные данные квот вызвали программное исключение в глобально распределённом пути; повторные попытки при восстановлении создали нагрузку на хранилище S02	Активация непроверенного пути неполными данными, затем перегрузка восстановления	У нас нет исполнимой копии внутренней системы для повторения исходного отказа
22 февраля 2024 года, AT&T Mobility	Ошибка конфигурации сетевого элемента нарушила связь по всей стране; по отчёту регулятора были заблокированы 92 миллиона вызовов и не прошли более 25 тысяч звонков в службы спасения S03	Распространение эффекта изменения управляющего элемента и массовая повторная регистрация	Эти данные не определяют полный безопасный опыт для другого оператора

Первые два источника — самоотчёты операторов. Издатель располагает прямым техническим знанием, но заинтересован в собственной версии происшествия; сопоставления с закрытыми сырыми журналами не проводилось. Третий источник — сравнительно независимое расследование регулятора, часть чисел в котором сообщена самим оператором. Эти границы сохранены в реестре утверждений.

2.3 Контекст подтверждает путь к экспорту, но не рынок этого продукта

Saudi Press Agency (SPA) сообщило об участии **28 национальных технологических и телекоммуникационных компаний** в павильоне Made in Saudi на GITEX AFRICA в апреле 2025 года [S04](#), а также о подписании меморандума между Elm и Huawei во время LEAP East в июле 2026 года [S05](#). Эти документы подтверждают только институциональное присутствие и международную деятельность. Они не содержат независимого измерения оплаченного экспорта в предлагаемом секторе тестирования, регулярных подписок, зарубежной клиентской базы или маржи таких продуктов. Они указывают на возможные каналы продаж, но не служат зависимой переменной, доказывающей тезис.

3. Определение экспортируемого объекта

Предлагаемый исполнимый контракт отказа (Executable Failure Contract) определяется как запись с пятью обязательными элементами:

1. **Происхождение (Provenance).** Ссылка на запись, из которой извлечена гипотеза, место абзаца, степень независимости источника и различие между заявленной причинностью и интерпретацией.
2. **Предусловия (Preconditions).** Свойства приложения, хранилища и протокола, при которых опыт имеет смысл, а также ограничения против вводящего в заблуждение переноса.
3. **Экспериментальный стимул (Stimulus).** Синтетический ввод, разрешённое изменение в изолированной среде или искусственная последовательность операций с заданными областью, длительностью и условиями остановки.
4. **Тестовый оракул (Test Oracle).** Наблюдаемое свойство или функциональный инвариант, отделяющий допустимые результаты от недопустимых, с указанием способа наблюдения и неопределённости.
5. **Пакет свидетельств (Evidence Package).** Версии программ, параметры запуска, разрешения, журнал событий и теста, результаты, пробелы и объяснение недопустимых выводов.

Единица должна переноситься через **промежуточное представление (Intermediate Representation; IR)**, но перенос не сводится к механическому равенству имён полей. Если контракт предполагает, что удаление двух повторов не меняет смысл файла, его применение к сервису, где повтор значим, разрушает контракт. Требуется функциональное описание и смысловые предусловия, а не одно преобразование формата (Serialization).

Предлагается иллюстративная форма, не имеющая статуса стандарта:

```
incident_spec_version: proposed-v0
provenance:
  source_id: S01
  claim_boundary: documented mechanism; not original payload
applicability:
  data_semantics: order-insensitive feature set
  environment: isolated local simulator
stimulus:
  type: repeated_features
  records: [a, b, a, b, a, b, a]
oracle:
  acceptable: ACCEPT after semantic deduplication
  forbidden: uncaught parser exception
safe_execution:
  production_connection: false
  stop_after_seconds: 10
  rollback: discard isolated fixture
limits:
  does_not_reproduce: Cloudflare production code or actual outage
```

Имя Cloudflare в поле происхождения не устраняет технические различия между компанией и любой другой системой. Если пользователь не доказал, что целевой файл представляет множество, а не список со смысловым порядком, контракт должен быть отклонён: воздержание от преобразования (Abstention) лучше уверенного, но бессмысленного результата.

4. Главная трудность: переносится СМЫСЛ или ТОЛЬКО команды?

Переносимость (Portability) имеет три разных уровня:

- **Синтаксическая переносимость (Syntactic Portability):** файл можно прочитать и преобразовать в корректный шаблон другого движка.
- **Операционная переносимость (Operational Portability):** тест можно исполнить в ясных пределах полномочий, записав требуемые данные и не подвергая живую систему опасности.
- **Смысловая переносимость (Semantic Portability):** новая операция проверяет то же свойство и обращается с допустимыми и недопустимыми случаями согласно назначению нового приложения.

Первые два уровня могут пройти, а третий — нет. Имитация отказа облачного региона способна отключить тестовую сеть, но не воспроизводит деградацию общей политики версий, расходящейся по нескольким регионам. Богатая временными данными трасса (Trace) сама по себе не выявляет потерю или повтор транзакции без логического отношения между параллельными операциями.

Пусть имеются две экспериментальные системы A и B , трассы выполнения τ_A и τ_B , абстрактные преобразования наблюдений α_A и α_B и общее свойство φ . Утверждение о смысловом переносе приблизительно требует сравнить значение свойства на эквивалентных представлениях трасс при соблюдении предусловий, а не просто убедиться, что оба опыта завершились без исключения. Требование можно записать так:

$$\text{MeaningMatch}(A, B, \varphi) := \text{Comparable}(\alpha_A, \alpha_B, \varphi) \wedge \text{OracleSound}(A, \varphi) \wedge \text{OracleSound}(B, \varphi).$$

Это концептуальная формула, а не доказательство эквивалентности программ и не автоматическое правило решения. Некоторые свойства нельзя сохранить вне конкретной среды, например семантику финансовой транзакции или юридические полномочия, отличающиеся в другой системе.

5. Устройство реально выполненного эксперимента

5.1 Ограниченная цель

Мы не создавали языковой компилятор происшествий, не испытывали Cloudflare, Google или AT&T и не импортировали образцы их систем. Был построен демонстратор доказательства концепции (Proof-of-Concept Demonstrator) для одного вопроса: можно ли записать небольшой набор инвариантов поведения так, чтобы в локальной детерминированной среде получить сопоставимые результаты независимых реализаций?

Примеры написаны вручную. Шестнадцать контрактов состоят из четырёх случаев в каждом из четырёх семейств. Каждый случай содержит `case_id`, `family`, `description`, `stimulus`, `expected` и `observed_property`. Затем написаны четыре тестовые реализации с четырьмя разными сочетаниями защит. Программа создана на Python и использует только стандартную библиотеку; к сети она не подключается. Цифровая запись содержит хеш SHA-256 конфигурации случаев, чтобы обнаружить необъявленную подмену входов, не претендуя на внешнюю проверку безопасности.

5.2 Семейства контрактов

Входные данные и политики (Policy Schema). Корректная числовая квота, отсутствующая квота, отрицательная квота и значение, не преобразуемое в число. Предлагаемое условие: если данные невозможно истолковать, ввод намеренно

отклоняется, а необработанное исключение не останавливает весь путь. Это не означает, что любой производственный сервис должен закрываться при каждом пропущенном поле.

Пакет свойств (Feature Bundle). Небольшое множество; допустимый повтор; повтор, при котором сырой размер превышает предел, но после удаления дубликатов укладывается в него; уникальное множество сверх предела. Только в синтетическом примере задано, что порядок и повтор элементов не меняют смысл, поэтому удаление дубликатов допустимо. Без доказательства этого свойства в реальной системе перенос запрещён.

Версия и подтверждение обновления (Update Version). Новое подтверждённое обновление; повторная отправка такой же подтверждённой версии; старая подтверждённая версия; новая версия с недействительной подписью. Предлагаемое условие не разрешает откатывающее или неподтверждённое обновление, но допускает идентичный повтор там, где его последствия определены. Реальная криптографическая подпись и управление ключами не моделируются: `signature_ok` — готовое синтетическое значение.

Ограниченные повторные попытки (Bounded Retry). Немедленный успех; три отказа перед успехом на четвёртой попытке; продолжительный отказ в течение двенадцати попыток; шесть последовательных отказов. В примере предел равен четырём попыткам в сумме. Оракул различает допустимый успех в пределах лимита и остановку по его исчерпанию. Код не использует настоящую сеть или случайную задержку и не измеряет перегрузку хранилища.

5.3 Четыре синтетические реализации

1. **Без защит (Naive).** Напрямую преобразует поля, применяет к сырому списку предел с исключением, принимает версии обновлений и повторяет попытки чрезмерное число раз.
2. **Только проверка входа (Input Guard Only).** Проверяет тип и значение квоты, версию и подпись обновления; обработка файла и повторных попыток остаётся уязвимой.
3. **Только границы ресурсов (Capacity Guard Only).** Обрабатывает дубликаты файла и ограничивает повторы; проверка политики и версий остаётся уязвимой.
4. **Объединённые защиты (Combined Guards).** Совмещает проверки, явно заданные контрактами примера.

Каждая реализация намеренно показывает ограниченное различие. Первая не изображает плохую реальную компанию, а четвёртая не является доказанно безопасной конструкцией. Поскольку тесты и защиты создавались в одном проекте,

успех четвёртой реализации не служит статистическим доказательством превосходства какой-либо методологии разработки. Можно придумать новые случаи, в которых откажут все четыре.

5.4 Запуск и записанные результаты

Скрипт `analysis/reproduce.py` СОЗДАЛ `contract_cases.json`, `contract_test_results.csv`, `contract_summary.csv` И `execution_meta.json`. Всего получено **16 контрактов × 4 реализации = 64 запуска**. Каждая строка хранит ожидаемый и наблюдаемый результат и признак совпадения, а не «успех предотвращения происшествия».

Синтетическая реализация	Политика входа (из 4)	Пакет свойств (из 4)	Версии (из 4)	Повторы (из 4)	Всего (из 16)
Без защит	1	2	2	2	7
Только проверка входа	4	2	4	2	12
Только границы ресурсов	1	4	2	4	11
Объединённые защиты	4	4	4	4	16

Каждый синтетический контракт на четырёх тестовых адаптерах

Двоичное значение показывает совпадение фактического результата с объявленным тестовым оракулом

	Данные политики				Файл свойств				Доверие версии				Лимит повторов				Итого
	Q1	Q2	Q3	Q4	F1	F2	F3	F4	V1	V2	V3	V4	R1	R2	R3	R4	
Без защит	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	7/16
Проверка входа	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	12/16
Границы ресурсов	1	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	11/16
Все защиты	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16/16

1 = совпадение 0 = расхождение

16 случаев, составленных вручную, × 4 синтетических адаптера = 64 запуска.

Результат означает согласие с оракулом, а не измеренное предотвращение производственных сбоев.

Матрица результатов 64 запусков на синтетических данных

Рисунок 1. Результаты вычислены из data/contract_summary.csv. Оценка показывает совпадение вручную спроектированного вывода с созданным вместе с ним оракулом; она не измеряет рыночную эффективность или вероятность реальных происшествий.

Каждый частичный путь проверки прошёл семейства, для которых был создан, и остался неполным в других. Но результаты **почти предопределены**: здесь нет ни статистической проверки гипотезы, ни способности обобщать за пределами программы. Польза примера учебная и архитектурная: единый язык сценария, отделение правила оценки от механизма исполнения и сохранение результата каждого контракта в сравнимой и воспроизводимой форме. Результат 16/16 нельзя истолковывать как систему, которая не отказывает.

6. Контроль, проверка и чувствительность: что сделано и что не сделано

Сделано: применены детерминированные правила без случайности, экспортированы все 64 случая, программно проверены число контрактов и итог каждой реализации, записан хеш содержимого случаев. Пользователь может запустить код в Python 3.10+ без внешних зависимостей и сравнить повторно полученные файлы. Это вычислительная воспроизводимость (Computational Reproducibility) примера, а не повтор естественного события.

Не сделано: контракты не переносились на производственные системы; оракул не проходил независимую проверку; время работы внешней команды не измерялось; клиентское принятие, проникновение, реальная языковая модель, множество синтетических зёрен и изменения самоадаптирующихся сред не исследовались. Для примера нет доверительных интервалов или вероятностных значений, поскольку набор не является статистической выборкой. Шестьдесят четыре запуска не создают совокупность из 64 независимых происшествий.

Важные проверки чувствительности на будущее: что произойдёт, если удаление повторов изменит смысл множества на упорядоченный список? Если старая версия разрешена в документированном откате? Если более четырёх повторов коммерчески необходимы и не вредят ресурсам? Если необработанное исключение предпочтительнее небезопасного разрешения? При изменении назначения системы часть четвёртой реализации способна перейти из «верной» в «нарушающую». Именно поэтому каждый перенос требует независимой смысловой калибровки.

7. Разрыв между работающим и полезным тестом

Реальное происшествие в начале сценария может придать результату незаслуженный авторитет. Отчёт не является полной спецификацией вызвавшей сбой программы. Оператор способен скрыть сведения для защиты системы; одинаково выглядящие поля в разных инструментах могут выполнять разные функции; расследование иногда основано на неполных данных. Поэтому в будущем опыте следует различать четыре исхода: точное представление исходного отказа; абстрактное представление семейства механизмов; учебный пример, повторяющий лишь словесную форму механизма; неприменимый опыт, который нельзя запускать.

Текущий пример в зависимости от контракта относится ко второй или третьей категории; доказанного совпадения с историческими системами нет. `pass` может быть верным для неверного контракта. При работе с искусственным интеллектом возникает дополнительная опасность: языковая модель создаёт синтаксически правильный опыт, а пробелы отчёта заполняет несуществующей связью — выдумывает долю, имя поля или значение повторения. Поэтому привязку к свидетельствам (Evidence Grounding), корректность компиляции контракта (Compilation Validity) и смысловую состоятельность оракула (Oracle Validity) необходимо проверять отдельно.

7.1 Хорошие проверки, на которых должен отказать сам инструмент

Продукт должен воздержаться от выпуска опыта, если не знает функцию. Предлагаются отрицательные калибровочные случаи:

- В отчёте скрыто необходимое определение поля — следует запросить дополнительное расследование.
- Издатель отозвал причинное объяснение — контракт следует отозвать или перевыпустить.
- Сервис работает с упорядоченными списками — нельзя удалять повторы лишь потому, что похожий случай прошёл в другом приложении.
- В среде нет достаточных условий остановки или отката — исполнимый опыт выпускать нельзя.
- Неполный журнал наблюдений не доказывает вывод оракула — тест помечается как «неопределённый», а не «успешный».

Подсчёт сгенерированных тестов способен вознаградить платформу, которая много говорит и мало проверяет. Более содержательная мера — число независимых контрактов, прошедших проверку источника, смысла, исполнения и результата и сохранивших состоятельность после обновления продукта клиента.

8. Начальная инженерная архитектура продукта

Независимую концепцию можно проверить в семи компонентах с ясными границами:

А. Версионизируемый корпус происшествий (Versioned Incident Corpus). Хранит ссылку, дату, вид свидетельства, версии и исправления. Не распространяет защищённый контент без разрешения, а сохраняет происхождение, допустимые фрагменты или указатели поиска.

Б. Извлекатель гипотез (Hypothesis Extractor). Локальная или законно доступная языковая модель предлагает механизмы, отделяет наблюдения от вывода и объявляет степень неполноты. Инструкции внутри импортированного отчёта не принимаются за команды системе.

В. Проверка смысловой применимости (Semantic Applicability Checker). Уточняет функции файла, права отката, версии, общее состояние и возможность подходящего оракула; может сформулировать вопросы инженеру клиента.

Г. Компилятор промежуточного представления (Intermediate Representation Compiler). Превращает утверждённый контракт в набор случаев и ограниченные программы внесения отказов. Он отделяет вид случая от инструмента исполнения и записывает невыполнимую информацию в `unsupported`, не отбрасывая её.

Д. Ограниченные адаптеры исполнения (Bounded Execution Adapters). Сначала работают в симуляторе или изолированной тестовой среде с узкими полномочиями, пределами длительности и нагрузки, остановкой и откатом. Эти гарантии не считаются автоматическими: даже рыночные инструменты нуждаются в настройке.

Е. Оракул и анализатор трасс (Oracle and Trace Analyzer). Сопоставляет вывод с объявленными свойствами, указывает на отсутствующую трассу или неопределённое поведение. При необходимости использует OpenTelemetry [S11](#), не сводя корректность транзакции к доле успешных запросов.

Ж. Пакет свидетельств и воспроизведения (Evidence Replay Bundle). Описывает версию системы, входы, решение, предостережения и ограничения. Независимая команда покупателя может повторить опыт в своей лаборатории или отвергнуть его при несовпадении предусловий.

Предлагаемая цепочка от происшествия к тесту

Пять преобразований знания; каждое требует свидетельств и проверки в конкретном приложении



Это концептуальная, не построенная платформа: нужно сохранить причинную неопределённость, проверить смысл и ограничить запуск. Независимый покупатель должен иметь возможность повторить результат либо отклонить перенос.

Путь от записи о происшествии к проверяемому тестовому продукту

Рисунок 2. Концептуальная производственная цепочка. Этапы не являются действующей саудовской системой или опубликованным стандартом; каждый интерфейс следует проверить и ограничить до применения в реальной среде.

Конкуренция не требует строить все части с нуля. Адаптеры могут использовать существующие интерфейсы инструментов внесения отказов и открытых проектов наблюдения. Доказать необходимо отличительную ценность: снижение стоимости переноса смысла и ясность свидетельств теста для разных клиентов. Для систем за пределами облака могут понадобиться модули без сетевого доступа к клиенту или библиотеки генерации входов для контейнерных программ, но это инженерное решение для конкретной закупки, а не лозунг.

9. Экономика продукта: от первого опыта к третьему клиенту

Описательно различаются три конкурирующие модели бизнеса:

- 1. Индивидуальный консалтинг по надёжности (Bespoke Reliability Consulting).** Клиент покупает время экспертов, изучающих его систему и проектирующих особый опыт. Выручка по договору может быть высокой, но большая часть труда повторяется, а маржа зависит от инженерных часов.
- 2. Лицензируемый пакет тестов (Licensed Testpack).** Клиент приобретает библиотеку, которую можно повторно запускать и обслуживать с ограниченной настройкой.

Продавцу нужны адаптационные средства, правила версий и применимости, иначе растёт поток поддержки.

3. Подписка на исполнение и свидетельства (Subscription Execution Evidence). Клиент платит за связь опытов с постоянно меняющимися версиями. Подписка даёт регулярную выручку только при возобновляемой ценности и создаёт обязанности по эксплуатации, хранению, безопасности и конфиденциальности.

Чтобы не объявлять недоказанный рынок, для каждого иностранного пилотного клиента предлагаются следующие единицы измерения:

Обозначение	Переменная	Предлагаемое измерение
H_i	Часы первоначальной настройки	Ограниченные рабочие журналы, связывающие труд с клиентом и версией
H_r	Часы обновления продукта для новой версии	Труд по обновлению без несвязанной коммерческой работы
H_c	Часы особой поддержки клиента	Время вмешательства, которое нельзя повторно использовать у других покупателей
P	Оплаченная цена	Полученная договорная выручка, а не преysкурant или предварительная договорённость
V	Документированная польза	Дополнительные релевантные случаи сверх средств клиента с учётом стоимости их исполнения
R	Оплаченное продление	Реальное продление после рассмотрения результатов первой версии

Предварительно можно определить **долю повторно используемого инженерного труда (Reusable Engineering Share)**:

$$\mathcal{S} = 1 - \frac{H_c}{H_i + H_r + H_c}$$

Если все часы неотрицательны, а знаменатель положителен, доля лежит от нуля до единицы. Это не валовая маржа (Gross Margin) и не самостоятельная мера качества. Высокое значение может говорить как о хорошей платформе, так и о непредоставленной поддержке, в которой нуждались клиенты. Для сравнения нужны единые определения периода, системы и покупателя. В настоящей работе измеренных значений нет.

Покупатель также может оценить гипотетическую верхнюю границу ожидаемой ценности через вероятность отказа, величину ущерба и изменение вероятности предотвращения благодаря тесту, вычтя стоимость опыта, цену и побочные

последствия. Но причинный эффект непрямо: успешный тест может сопровождаться полной переработкой системы, а число сбоев — снизиться из-за обучения, меньшей нагрузки или смены архитектуры; сам отказ может не возникнуть в срок договора. Поэтому истории клиентов и число выполненных контрактов не позволяют вывести отдачу от инвестиций.

10. Как отличить продаваемый продукт от услуги под видом платформы?

Практическое различие требует опубликованной внутри проекта панели измерений, а не маркетинговых титулов. У масштабируемого продукта средний труд настройки должен уменьшаться при переходе к новому клиенту с похожей архитектурой; полезный результат должен сохраняться после обновлений; инженер покупателя должен уметь запускать его без постоянного присутствия разработчиков. Специализированная услуга может быть прибыльной и без этих свойств, но это другой бизнес с иной экономикой.

Рыночный опыт следует начать в гражданских секторах, где легко получить ясное разрешение: многопользовательские корпоративные программы, обработка нечувствительных заявок до выпуска и общие компоненты управления конфигурацией. После подтверждения успеха можно изучать более чувствительные классы при соответствующих мерах. Для выхода на рынок не требуется публиковать симуляции критических объектов или связывать продукт с закрытыми эксплуатационными данными.

Строгим начальным критерием входа — **предлагаемым будущим протоколом, а не результатом** — могут стать три независимых международных клиента на двух платформах. Хотя бы один должен повторно запустить пакет без вмешательства продавца; фиксируются стоимость настройки, замечания и случаи непереносимости. Это эвристический порог для опровержимой проверки, а не опубликованное научное правило классификации компаний.

11. Исследование, способное опровергнуть идею

11.1 Выборка и сравнения

Первичный реестр составляется из оригинальных опубликованных отчётов; источники заранее классифицируются по независимости и подробности. Организации для обучения и проверки разделяются. Если измеряется обобщение, в обучающий набор

нельзя включать происшествие того же семейства, что и отложенный случай. События следует разделять и по дате публикации, чтобы модель, обученная на поздней записи, не применяла знание будущего к контракту, якобы выпущенному раньше.

Первое сравнение — с инженером, вручную пишущим тест в своём инструменте. Второе — с готовой библиотекой сценариев облачного поставщика. Третье — с генерацией шаблона из естественного языка без дополнительной смысловой проверки, как в некоторых рыночных практиках S06S07. Командам по возможности предоставляют одинаковое время и фиксируют квалификацию инженеров, иначе преимущество команды будет принято за преимущество инструмента.

11.2 Результаты и критерии приёмки

- **Точность происхождения механизма (Provenance Accuracy):** привязан ли механизм к конкретному месту первичного отчёта без добавленных деталей?
- **Согласие экспертов с оракулом (Oracle Agreement):** согласны ли независимые эксперты с ожидаемым результатом после изучения назначения системы? Разногласия записываются, а не удаляются.
- **Точность применимости (Applicability Precision):** какая доля признанных применимыми контрактов действительно оказывается применимой при проверке?
- **Выход переноса (Transfer Yield):** доля контрактов, сохранивших смысловую корректность после переноса в новую среду заданной выборки, с объявленным знаменателем и исключениями.
- **Дополнительное обнаружение отказов (Incremental Fault Discovery):** число дополнительных корректных дефектов сверх исходного инструмента после исключения повторов и ложных срабатываний.
- **Предельная стоимость интеграции (Marginal Integration Cost):** часы разработки, документирования и эксплуатации у продавца и клиента для каждой новой среды и версии.
- **Повторное оплаченное зарубежное удержание (Foreign Paid Retention):** число иностранных покупателей, продливших договор после настоящего цикла применения, и чистая оплаченная сумма, а не предложения или меморандумы.

Порог опровержения: гипотеза о самостоятельной экспортной программной отрасли ослабевает, если продукт в слепом тесте не находит больше функциональных дефектов, чем существующие средства за то же время и стоимость; если каждый проект требует неизменно большого объёма консалтинга; либо иностранные покупатели не продлевают договоры. Учебная или консультационная ценность при этом может сохраниться.

11.3 Причинный вывод

Даже снижение числа происшествий после покупки пакета нельзя приписать ему без контрольной группы или анализа, учитывающего версии, интенсивность использования и прочие исправления. С продуктом вначале наиболее непосредственно связан новый воспроизводимый дефект или сокращение времени до надёжного теста. Предотвращение будущего происшествия требует более долгого наблюдения и контрфактического объяснения. Тяжёлые отказы настолько редки, что статистически доказать пользу их сокращения трудно даже при большой экономической цене.

12. Полномочия и безопасность как свойства исполнения

Тест, обнаруживший дефект, не заслуживает научного успеха, если он вызвал реальный сбой за пределами разрешённой области. Все пакеты должны начинать в изолированной или предпроизводственной среде; для внесения отказа в работающую систему требуется явное разрешение границ. Необходимы пределы длительности, нагрузки и ресурсов и условия остановки, связанные с состоятельным показателем. AWS документирует условия остановки экспериментов на основе заданных сигналов мониторинга [S08](#), показывая, что безопасность входит в устройство опыта, а не в приписку к нему.

Продаваемые или публикуемые пакеты не должны хранить пароли, журналы доступа и подробные карты зависимостей чувствительных объектов. Названия публичных происшествий отделяются от данных эксплуатации клиента; при извлечении сведений из опубликованных отчётов соблюдаются права и лицензии. Глобальная классификация ошибок не требует знать секретное правило организации или раскрывать её внутреннюю учётную запись. Искусственный интеллект должен изолировать недоверенный исходный материал от операционных инструкций: извлечение текста не даёт права исполнения.

13. Что исследование добавляет к саудовским вопросам до 2035 года?

Работа различает четыре степени ценности:

1. **Местная экспертиза (Local Expertise):** саудовские команды умеют исправлять реальные проблемы; это важно, но недостаточно для экспорта.
2. **Кодифицированное знание (Codified Knowledge):** опыт превращён в проверяемые спецификации и свидетельства; это ещё не доказывает переносимость.

3. **Переносимый продукт (Transferable Product):** тест сохраняет близкий смысл у нового клиента при ограниченной стоимости; это не доказывает устойчивый спрос.
4. **Повторяемая экспортная отрасль (Repeatable Export Industry):** независимые покупатели платят и продлевают договоры, а продукт финансирует собственное обслуживание и улучшение; для такого экономического результата нужны иные данные, чем для инженерного доказательства.

Это предлагаемые описательные степени, а не официальный индекс. Нельзя прямо выводить четвёртую из первой или подменять сведения об устойчивой технологической выручке участием 28 компаний в международной выставке S04.

В рамках гипотезы наиболее полезной саудовской исследовательской инвестицией может быть ограниченная лаборатория, а не универсальная сервисная компания: разнообразный корпус документированных происшествий; протокол кодирования их условий; открытые адаптеры для изолированных опытов; слепые состязания между человеческими решениями и предложениями языковой модели после инженерной проверки; правила публикации, запрещающие неподтверждённые заявления. При независимом измеренном успехе можно переходить к специализированным продуктам и иностранным клиентам. При неудаче её следует опубликовать: сам отказ является исследовательским материалом, из которого предполагаемая отрасль хочет учиться.

14. Ограничения исследования и предел задания

Единственный завершённый вычислительный результат — сопоставление 64 выходов с заранее заданными синтетическими контрактами. Внешний обзор ограничен выбранными первичными источниками; это не всесторонний систематический обзор (Systematic Review), поиск по индексируемым публикациям с заранее зарегистрированными критериями исключения не проводился. Размер рынка, цена договора, вероятность происшествия и реальная отдача не вычислялись. Абсолютная научная новизна модели, её патентоспособность и независимость от существующих инструментов не подтверждены.

Допустимый вывод уже, но содержателен: ограниченные семейства отказов ввода, обновления и нагрузки технически можно описать исполнимыми контрактами и снабдить проверяемыми свидетельствами — это демонстрирует небольшой пример. На рынке уже есть зрелые средства для близких форм опытов с отказами и их генерации из естественного языка. Поэтому предполагаемую добавленную ценность следует проверять по **корректности смыслового переноса, снижению стоимости настройки и повторной международной оплате**. До таких испытаний предлагаемая отрасль остаётся исследовательской возможностью, а не сообщением о новой саудовской экономике.

15. Доступность и воспроизведение

Данные в `data/` полностью синтетические и созданы для этой работы; внутри пакета они предоставлены как учебные примеры, а не данные третьей стороны. Код находится в `analysis/reproduce.py`, использует только стандартную библиотеку Python (Python Standard Library) и запускается из корневого каталога исследования, выдавая те же файлы и хеш случаев. Полученные графики представлены в векторном (SVG) и растровом (PNG) форматах, а таблица Excel позволяет визуально проверить объявленные вычисления. Облачные службы, вход в учётную запись и скрытые данные не нужны.

В пакете нет исходных эксплуатационных данных Саудовской Аравии, коммерческих секретов, кода атак на объекты или непроверенных сообщений о конфликтах. Исходные отчёты пересказываются по смыслу без длинных цитат и чувствительных сведений исполнения.

Литература

1. Cloudflare, Matthew Prince, *Cloudflare outage on November 18, 2025*, original incident postmortem, 18 November 2025. <https://blog.cloudflare.com/18-november-2025-outage/>
2. Google Cloud, *Incident details: June 12, 2025 service disruption*, incident status and root-cause note, 2025. <https://status.cloud.google.com/incidents/ow5i3PPK96RduMcb1SsW>
3. Federal Communications Commission (FCC), *February 22, 2024 AT&T Mobility Network Outage*, Public Safety and Homeland Security Bureau Report, 22 July 2024. <https://docs.fcc.gov/public/attachments/DOC-404150A1.pdf>
4. Saudi Press Agency (SPA), *'Made in Saudi' Pavilion at GITEX AFRICA Showcases 28 Leading Tech Companies*, 14 April 2025. <https://www.spa.gov.sa/en/N2297562>
5. Saudi Press Agency (SPA), *Elm Signs MoU with Huawei to Advance Cooperation in AI and Digital Infrastructure*, 9 July 2026. <https://www.spa.gov.sa/en/N2630981>
6. Amazon Web Services (AWS), *Bootstrap your chaos engineering journey with AWS Fault Injection Service Scenarios Library*, 1 October 2024. <https://aws.amazon.com/blogs/mt/bootstrap-your-chaos-engineering-journey-with-aws-fault-injection-service-scenarios-library/>
7. Amazon Web Services (AWS), *Chaos engineering made clear: Generate AWS FIS experiments using natural language through Amazon Bedrock*, 13 May 2025. <https://aws.amazon.com/blogs/publicsector/chaos-engineering-made-clear-generate-aws-fis-experiments-using-natural-language-through-amazon-bedrock/>
8. Amazon Web Services (AWS), *Stop conditions for AWS FIS*, online technical documentation. <https://docs.aws.amazon.com/fis/latest/userguide/stop-conditions.html>

9. Jepsen (Kyle Kingsbury / Aphyr), *Analyses and Research Ethics*, technical analyses and methods. <https://jepsen.io/analyses> and <https://jepsen.io/analyses/ethics>
10. Principles of Chaos Engineering, *Principles of Chaos Engineering*, foundational practitioner document. <https://principlesofchaos.org/>
11. Cloud Native Computing Foundation (CNCF), *OpenTelemetry has graduated... Now what?*, 24 July 2026. <https://www.cncf.io/blog/2026/07/24/opentelemetry-has-graduated-now-what/>
12. Amazon Web Services (AWS), *REL12-BP04 Test resiliency using chaos engineering*, Well-Architected Reliability Pillar. https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/rel_testing_resiliency_failure_injection_resiliency.html
13. Author-supplied materials, *Executable Incident Contract Demonstration*, version 1.0. Python code and CSV/JSON outputs are included in this package; explicitly synthetic, not independently peer reviewed.

Статус: концептуальная и вычислительная рабочая статья, не прошедшая независимого научного рецензирования. Независимая проверка носителем языка не проводилась.