Мутируйте все!

От обычного кода до запросов в базы данных

Евгений Блинов для Data Fest 2026

Обо мне

- Разработчик-питонист
- Энтузиаст опенсорса
- Разрабатываю собственный движок мутационного тестирования
- Амбассадор мутационного тестирования

Контакты

Гитхаб: github.com/pomponchik

Линкедин: linkedin.com/in/pomponchik

Личный сайт: pomponchik.org


Hildebrands Deutsche Schokolade
Будущее!
Deutscher Kutter.

3 факта о будущем

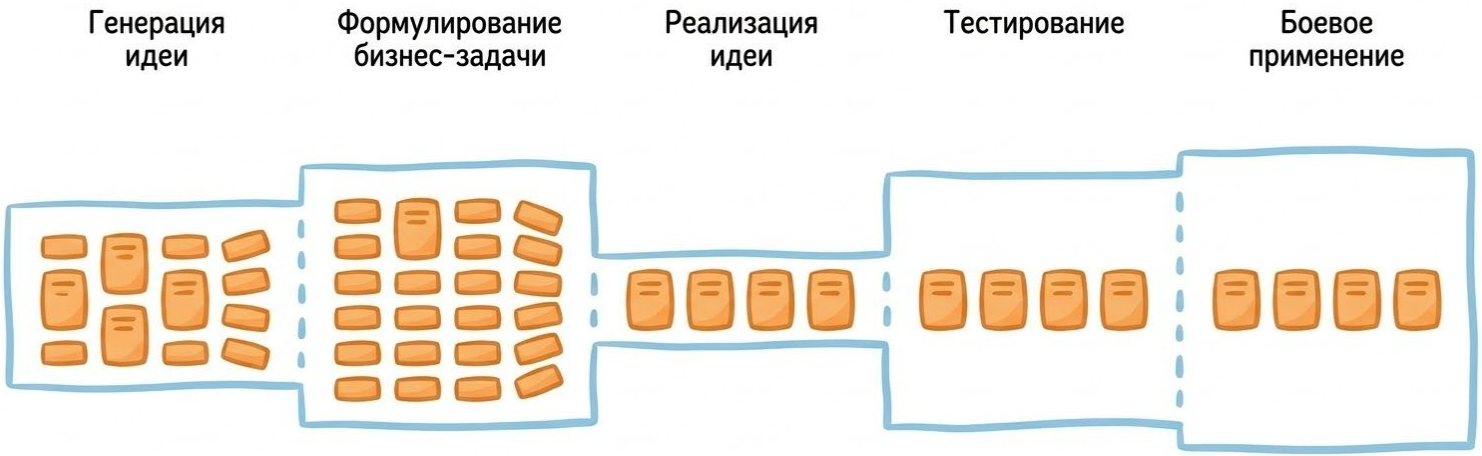
1. Парадокс Джевонса

- Чем дешевле код, тем больше его нужно
- IT будет продолжать расти в капитализации
- Бенефициары перераспределятся
- Какую долю среди них займут “операторы LLM” – вопрос пока открытый

2. Бутылочные горлышки переместятся

- По сути все IT – это фабрики ифов
- Код пишется, чтобы проверять чьи-то гипотезы
- Код – промежуточный артефакт
- Генерация кода – только один из конвееров
- Расширение одного конвеера перемещает ботлнеки в другие

2.1. Нагрузка на тестирование резко возрастает



Тесты – бутылочное горлышко агентского кодинга

- Все используют LLM для генерации кода
- Тесты кажутся легкой целью
- Но программисты по факту все равно перечитывают весь код
- Тесты и их корректность – по-прежнему сложнейшая задача в программировании

> LLM пишет тесты

Вот каким будет будущее

Каким, по твоему, будет будущее?

я не Каким, меня Сашей зовут. да по моему будет будущее

Как расширить бутылочное горлышко?

?

Нужно автоматизировать тесты

Нужно автоматизировать тесты

- Но ведь LLM может писать тесты!

Нужно автоматизировать тесты

- Но ведь LLM может писать тесты!
- Да, она ошибается, но может мы попросим LLM провести ревью тестов?

Нужно автоматизировать тесты

- Но ведь LLM может писать тесты!
- Да, она ошибается, но может мы попросим LLM провести ревью тестов?
- Как насчет ревью ревью тестов?

Решение есть:
мутационное тестирование

Плюсы мутационного тестирования

- Детерминированный алгоритм
- Много исследований о реальной корреляции дефектов ПО и мутационного скора

Быстрое введение:
что такое MT?

Берем функцию…

```
def function(a, b):
    return a + b
```

И тестируем!

```
def test_function():
    function(2, 2)
```

Вы восхитительны!

Благодарим Вас за использование Google Chrome!

Покрытие 100%

Ладно, вторая попытка:

```
def test_function_2():
    assert function(2, 2) == 4
```

Вы восхитительны!

Благодарим Вас за использование Google Chrome!

Все же круто, да? Да?

- Ладно, тут есть ассерт
- Это должно работать

Но что, если…

```
def function(a, b):
    return a * b
```

Наш тест, напоминаю:

```
def test_function_2():
    assert function(2, 2) == 4
```

Вы восхитительны!

Благодарим Вас за использование Google Chrome!

Добавим лучшие практики!

```
def test_function_3():
    assert function(2, 2) == 4
    assert function(0, 0) == 0
```

Вы восхитительны!

Благодарим Вас за использование Google Chrome!

Мы только что освоили MT

ИИ *может* писать качественные тесты!

- ИИ-агент может запускать MT на своих тестах и убеждаться в их качестве
- Можно строить агентские системы, которые покрывают качественными тестами ваш код
- Систему MT можно использовать как источник обратной связи при обучении LLM с подкреплением

Ограничения

- Эта штука не выявляет баги, несоответствие спеке
- Она проверяет только то, что *что-то* проверяют
- Не заменяет другие виды тестирования
- Дорого по компьюту
- Должно применяться после всех других видов тестирования
- Эквивалентные мутанты

Этим вообще кто-то пользуется?

Google

- На 2021 год: активно используется более чем 24 000 инженеров, проверяются все PRы
- Система встроена в продакшен
- Все крутится вокруг дешевых инкрементальных проверок

research.google/pubs/practical-mutation-testing-at-scale-a-view-from-google/

Meta (запрещено в РФ)

- Активно мутируют
- Интегрируют с LLM: нейронка как ломает код, так и пишет тесты

engineering.fb.com/2025/09/30/security/llms-are-the-key-to-mutation-testing-and-better-compliance/

Окей, как затащить это к себе?

Спроси своего чат-бота)))

Лидеры рынка в целом

Java: лидер по MT

- Платформа с самой сильной MT-историей
- Куча научных работ и куча тулинга
- Наиболее мейнстримный инструмент: PIT

C/C++ (вкл. LLVM): тоже неплохо

- Развитый тулинг
- Популярен инструмент Mull

Python

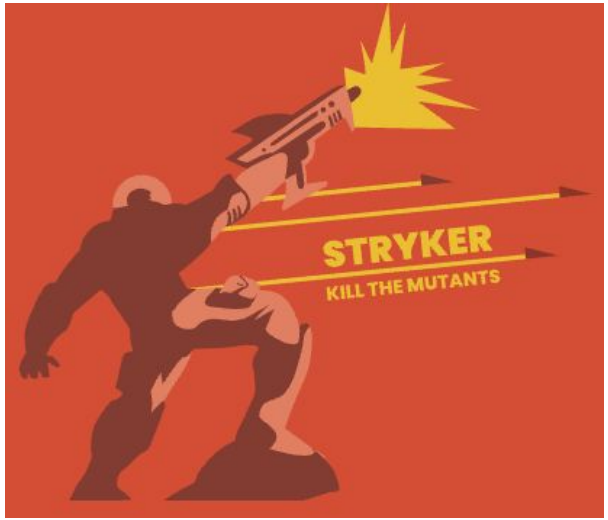
mutmut

- Тулинг разрозненный и местами незрелый
- Популярные инструменты: mutmut, Cosmic Ray

Но мы над этим работаем!

JS, PHP, C#, Scala

- Высококлассный стандартизированный тулинг от Stryker
- Конкуренты где-то позади



Как насчет SQL?

- Базы данных – самая уязвимая деталь любого бекенда
- Да, SQL тоже можно мутировать
- Есть вариант загуглить по “<имя вашей ORM> + mutation testing tool”
- Примеры инструментов:
  - QACover / SQLMutation – лучший старт для сервисов: мутанты WHERE/JOIN/NULL/границ → тесты должны упасть.
  - EvoSQL – помогает закрывать дыры: пустые выборки, NULL, даты, deleted-записи, edge cases.

За МТ будущее!

Извините пожалуйста и спасибо за понимание, всего вам доброго

Ссылки:

- Гитхаб: github.com/mutating
- Сайт: mutating.tech

За красивую тему для презентации отдельная благодарность компании Evrone