

TestOps Copilot

AI-ассистент для автоматизации работы
QA-инженеров на базе Cloud.ru Evolution
Foundation Model

Автоматизирует генерацию тест-кейсов,
оптимизацию и проверку стандартов.
Поддерживает UI и API-тестирование для
продуктов Cloud.ru.

 AI QA Assistant

8 Команда и роли

Мы — команда из трех специалистов, мотивированных оптимизировать QA-процессы в IT. Мы сталкиваемся с рутинной тестирования ежедневно и создали это решение для реальной бизнес-пользы.

Каждый член команды внес вклад в ключевые аспекты проекта, от дизайна до интеграций.

Наша цель — сделать QA быстрее и эффективнее для всех.



Краснокутский Алексей

DevOps + Frontend



Фомин Иван

Backend + Архитектура



Усачёва Ульяна

Design + Presentation

+ Проблема и бизнес-контекст

Представьте: QA-инженер тратит часы на ручную генерацию тест-кейсов, анализ покрытия и проверку стандартов. Это приводит к ошибкам, задержкам и перегрузке команды.

Пользователи: QA-инженеры, тест-лиды в IT-компаниях. Текущие решения: ручной труд, фрагментированные инструменты. Метрики "до":

2-3 часа

На генерацию тестов в неделю

20%

Ошибок в тестах

2 дня

Задержки релизов

Наше решение

AI-агент на базе Cloud.ru Evolution Foundation Model автоматизирует генерацию ручных и автоматизированных тест-кейсов (UI/API), оптимизацию и проверку стандартов. Совместим с Evolution AI Agents.

Решение включает полный цикл: от парсинга требований до коммита в GitLab.

Обеспечивает 80%+ покрытия тестов с минимальными правками.



Генерация в формате Allure TestOps as Code



Автотесты: Playwright (UI), Pytest (API)



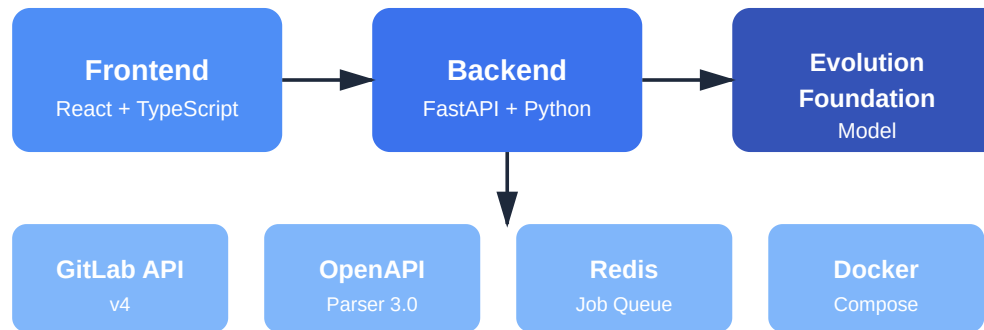
Оптимизация: анализ покрытия, дубликаты



Интеграции: GitLab, OpenAPI, A2A

🗄 Архитектура решения

Система построена на взаимодействии AI-агента и интеграционного слоя. Агент использует LLM через Cloud.ru Evolution Foundation Models для интерпретации запросов и генерации тестов.



Core Technologies

Python 3.10+

AI & Testing

LangChain

Pytest

Infrastructure

Docker

Redis

Integrations

GitLab API v4

FastAPI

React 19

TypeScript

Playwright

Allure

GitLab CI

OpenAPI 3.0

A2A Protocol

Демонстрация: UI-тестирование

Кейс 1: Генерация тест-кейсов для калькулятора цен Cloud.ru [\[https://cloud.ru/calculator\]](https://cloud.ru/calculator)

Входные данные

Описание требований к UI калькулятора: начальная страница, каталог продуктов, конфигурация Compute, управление сервисами, мобильная адаптация

Процесс агента



LLM парсит требования и выделяет 5 блоков тестирования



Генерация 25-35 ручных тест-кейсов в Allure формате с автотестами на Playwright (улучшенная структура, AAA паттерн)



Оптимизация: анализ покрытия (65% для UI), удаление дублей

Выходные данные

Python-код тест-кейсов, автотесты, отчет о покрытии. Полностью соответствует ТЗ с учетом последних доработок.

👤 Демонстрация: API-тестирование

Кейс 2: Генерация тест-кейсов для Evolution Compute Public API v3 (<https://compute.api.cloud.ru>)

Входные данные

OpenAPI спецификация: VMs, Disks, Flavors. Аутентификация Bearer.

Процесс агента



Парсинг OpenAPI через OpenAPIParser



Генерация 25-35 ручных тест-кейсов для VMs (6 оп.), Disks (7 оп.), Flavors (2 оп.)



Автотесты на Pytest с httpx, учетом аутентификации и последних правок (улучшенный парсинг)



Оптимизация: полное покрытие (100% для API), выявление устаревших тестов

Выходные данные

Код тест-кейсов, автотесты, GitLab commit. Соответствует ТЗ с GitLab.

🏠 Эффект и измеримый результат

После внедрения QA-команды экономят время и снижают ошибки. Сравнение метрик:

Показатель	До	После
Время на генерацию тестов	180 мин/нед	5-10 мин
Ошибки в тестах	~20%	<2%
Скорость релизов	1-2 дня	30 секунд
Экономия времени	-	140 часов/год на специалиста

Улучшена производительность: генерация теста <5 сек, пакета <30 сек.

Снижение нагрузки на QA позволяет фокусироваться на сложных задачах.

⚠ Масштабирование и итоги

Решение подходит для 500+ компаний в ритейле, финансах, госсекторе. Легко адаптируется под новые API.

Будущие планы: интеграция с дефектами, unit-тесты, Swagger-документация.

Масштабирование: добавление multi-agent, RAG для политик, батч-обработка.

Multi-agent

RAG

UI Integration

Batch Processing

Edge Cases

Unit Tests

✓ **Итоги работы**

Гордимся: надежностью, бизнес-ценностью, соответствием ТЗ. Улучшения: тест-планы, исторические дефекты.