Геннадий Никитин

QA-инженер (Senior) · 8+ лет · мануальное тестирование + автоматизация + API

Асунсьон, Парагвай · удалённо (UTC-3), рабочий день пересекается с CET и EST · релокация обсуждаема · доступен сразу
Оформление: B2B, EOR или прямой контракт — обсуждаемо; для российских юрлиц — как самозанятый.
Telegram: @ni_genn · Email: g.nikitin14@gmail.com · WhatsApp: +595-993-353-602 (PY) · +7-931-288-7553 (RU)
LinkedIn: https://www.linkedin.com/in/gennadiy-nikitin-21337022b · GitHub: https://github.com/ge-a-nikitin

О СЕБЕ

QA-инженер, 8+ лет на клиент-серверных продуктах для десктопа, веба и мобильных устройств — высоконагруженные потребительские промо и геймификация: призовые игры, сезонные кампании, механики лояльности, включая призовую акцию с аудиторией до 3M+ уникальных пользователей в месяц. Отвечаю за полный цикл (SDLC) на своих проектах: требования и тестируемость до разработки, тест-дизайн, функциональное, регрессионное, API- и интеграционное тестирование, автоматизация там, где она окупается, — TypeScript на внутреннем фреймворке компании на базе Playwright, нагрузочное тестирование на k6 на двух проектах. После себя оставляю документацию и процессы, которые команда подхватывает.

КЛЮЧЕВЫЕ ДОСТИЖЕНИЯ

- Сократил регресс с **2 рабочих дней (2 тестировщика) до ~2 часов (1 тестировщик)**: автоматизировал ключевые десктопные сценарии на Python + PyAutoGUI с поиском элементов по изображению.
- **Взял на себя большую часть роли менеджера проекта на «Пушкин 225» без самой должности**, пока команда была недоукомплектована; акция **вышла в срок**, а я оставался на ней единственным QA — готовность к запуску, ежедневные статусы, приоритизация бэклога, требования вместе с продуктом-стейкхолдером, ресёрч новых интеграций (API и Kafka), менторство новых коллег.
- ИИ-чатбот, отвечавший на обращения первой линии (L1) по подготовленной документации, **закрывал 30% обращений, не доводя их до команды** — с разбивкой по каналам и проектам на дашборде.
- Вместе с коллегами спроектировал шаблон автоматизации департамента на фреймворке компании eva.io; **забрал шаблон в пять своих следующих проектов, а коллеги из смежных команд — ещё примерно в десять**.
- Перевёл нагрузочное тестирование с ручных прогонов на проде на **отдельное окружение под ответственностью QA**: сценарии benchmark, stability и maxperf на k6 в двух проектах, метрики в Grafana и Cosmos; утверждённая в департаменте методика с шаблоном и quickstart теперь переиспользуется на новых проектах.
- Автотесты были условием допуска к релизу во всех проектах, где мы писали e2e: красный пайплайн держал релиз до выяснения причины.
- Визуальная регрессия по скриншотам — на **ежедневных прогонах в CI** с отчётами в корпоративный мессенджер.

КЛЮЧЕВЫЕ НАВЫКИ

Тестирование (обеспечение качества): функциональное, регрессионное, интеграционное, системное, UI, API, кросс-браузерное, нагрузочное и производительное, визуальная регрессия (скриншотные тесты).

Тест-дизайн и анализ: анализ граничных значений, классы эквивалентности, диаграммы состояний и переходов, анализ требований, three-amigo-сессии (PM, разработчик и QA вместе), ревью тестируемости.

Автоматизация тестирования: TypeScript на eva.io (Selenium → Playwright: основная работа пришлась на период, когда сосуществовали оба), Playwright (модули eva.io на работе; публичный e2e-набор ниже), Python (PyAutoGUI, распознавание изображений, Tillypad), параметризованные тесты, визуальная регрессия, отчётность в Allure TestOps, переиспользуемая структура модулей и страниц.

API и данные: Postman, Insomnia, Swagger/OpenAPI, Charles, Fiddler, OWASP, SQL (SELECT, JOIN, GROUP BY, подзапросы, INSERT/UPDATE), проверка репликации MSSQL.

Производительность: k6 (benchmark, stability, maxperf), профили нагрузки по логам прода, Grafana (и внутренний сервис метрик Cosmos).

Платформы и инфраструктура: Docker, BrowserStack, реальные устройства iOS/Android, Android Studio, Chrome и Firefox DevTools.

ИИ в тестировании: чек-листы и тест-сценарии с помощью LLM, итерации промптов, Nestor AI (внутренний Cline-подобный агент компании) для тест-кода внутри IDE; делился практикой с командой.

Процесс QA: тест-стратегия, тест-планы и тест-кейсы, shift-left (требования и ревью тестируемости до разработки), критерии допуска к релизу (quality gates), жизненный цикл дефекта и триаж, тест-документация, метрики качества; Agile — Scrumban.

Инструменты и CI/CD: Git, GitLab CI с ежедневными прогонами по расписанию, Jira, Confluence, Allure TestOps, Redmine, интеграции через топики Kafka.

ОПЫТ РАБОТЫ

Текущий статус: ушёл из T-Bank по личным обстоятельствам.

T-Bank (ex Tinkoff) — Quality Engineer — апрель 2023 – сентябрь 2026

Веб- и webview-части мобильных приложений Android/iOS: промо-механики, которые вырастали в продукты, — призовая игра в стиле «Монополии» с аудиторией до 3M+ уникальных пользователей в месяц, сервисы с долгим жизненным циклом, сторонние интеграции. Первые два года — единственный QA, затем один из двух равноправных QA-инженеров — после пересборки команд вокруг крупных продуктов; отвечал за тест-стратегию своих проектов. Работа шла в Scrumban — гибрид Scrum и Kanban для проектной работы. Релизный цикл: 3–7 месяцев разработки на стенде, один день релиза, дальше — релизы ежедневно или еженедельно по критичности. Дежурства по обращениям и алертам: триаж делает QA и передаёт PM, разработчику или тимлиду; инциденты разбирали с бэкендом и тимлидом.

- **Автоматизация на уровне департамента.** Вместе с коллегами спроектировал шаблон тестирования департамента на eva.io (TypeScript, Selenium → Playwright): авторизация через внутренний тестовый контур QA Utils (готовые тестовые пользователи), запуск тестов и отчётность в Allure TestOps, структура репозитория, гайд по развёртыванию. Делал ревью кода и вносил изменения в репозиторий шаблона; шаблон приняли свои и смежные команды.
- **Game Center (единая точка входа для спецпроектов).** Построил процесс тестирования с нуля: тест-планы, отчёты Allure, FAQ и база знаний; вёл системное, UI-, API- и интеграционное тестирование; добавил визуальную регрессию по скриншотам на модуле Playwright в eva.io с прогонами в CI/CD (красный примерно раз в месяц — когда менялись сезонные ассеты или список проектов). Передал команде с документацией и процессами.
- **Тестовая пирамида и TestOps.** Последние полтора года — вместе со вторым QA: прогоны unit- и интеграционных тестов из репозиториев бэкенда и фронтенда, e2e- и скриншотные наборы, вся отчётность в TestOps — статистика, тест-кейсы, уровни намеренно разделены. Сквозная проверка контрактов и валидации нашла сломанный контракт API и невалидируемое поле.

- **Shift-left по умолчанию, а не как инициатива.** На этапе требований, до старта разработки: писал функциональные требования вместе с командой в three-amigo-сессиях, работал по RFC департамента о раннем подключении тестирования, вёл страницу открытых вопросов и указывал на слабые места, пока менять планы было ещё дёшево.
- **Снижение нагрузки на поддержку.** Настроил автоответы ИИ-чатбота в корпоративном мессенджере по подготовленной документации; дашборд по каналам и проектам показывал, что **30% обращений закрываются в чатботе и не доходят до команды**; продолжал вести поддержку и трекинг дефектов.
- **Нагрузочное тестирование.** Вместе с SRE поднял отдельное окружение для нагрузочных тестов; сценарии benchmark, stability и maxperf на k6; метрики в Grafana и Cosmos. Прогоны показали насыщение базы — медленные запросы и конкуренцию за блокировки под нагрузкой — и это исправили до запуска. Написал утверждённую в департаменте методику с шаблоном и quickstart, которую команда QA переиспользовала на новых проектах.
- **«Пушкин 225»: единственный QA и большая часть роли PM.** При нехватке людей забрал операционную работу менеджера — ежедневные статусы и приоритеты, готовность к запуску и сроки вместе с ведущим PM, наполнение приложения контентом, требования от продукта-стейкхолдера — а параллельно ввёл в проект двух новых членов команды: разработчика бэкенда на испытательном сроке и стажёра-PM (онбординг, 1:1, консультации).
- **Помощь соседним командам.** Дважды на 1–2 месяца закрывал отсутствие QA в соседней команде: релизное тестирование и контекст стратегии, релизы шли без штатного QA, сдавал обратно с документацией.

Tillypad — Quality Engineer — май 2018 – апрель 2023

Прошёл от Test Specialist через Test Engineer до Quality Engineer в команде из 3 тестировщиков и 6 разработчиков. Клиент-серверная система ресторанов: десктопные приложения для официантов, барменов, поваров, менеджеров, позже — мобильная (Android/iOS) и веб-версии; MSSQL с репликацией между базой ресторана и головным офисом или облаком.

- **Полный цикл ручного тестирования и поддержка.** Создавал и поддерживал тест-планы и тест-кейсы; проводил модульное, интеграционное, системное и регрессионное тестирование; проверки API в Postman и Swagger, анализ трафика в Charles и Fiddler; вёл дефекты в Redmine; прорабатывал требования, чтобы находить слабые места в начале цикла. Разбирал проблемы пользователей вместе с поддержкой; вводил в продукт и процессы новых членов команды.
- **Автоматизация UI.** Автоматизировал ключевые десктопные сценарии на Python + PyAutoGUI с распознаванием изображений кнопок и элементов интерфейса для определения координат и сравнения с эталоном. Регресс ушёл с 2 рабочих дней (2 тестировщика) до ~2 часов (1 тестировщик).
- **Тестирование репликации MSSQL.** Участвовал в формировании подхода; писал SQL-скрипты, которые вставляют, обновляют и удаляют данные в источнике, после чего проверял, что в целевой базе есть всё нужное и нет лишнего. Запускал в начале каждого релизного цикла: **около десяти дефектов за три года, все высокого приоритета и критичности**, в логике, без которой продукт не работает.

ИЗБРАННЫЕ ПРОЕКТЫ

- **hecho-demo** — https://github.com/ge-a-nikitin/hecho-demo — HECHO, клиент-серверная доска заказов, сделанная, чтобы показать полный цикл QA на практике: SPA на React 19 + TypeScript, API на NestJS со Swagger/OpenAPI, PostgreSQL + Prisma, SeaweedFS, Docker Compose. Unit- и интеграционные API-тесты на Vitest, двуязычная продуктовая и QA-документация, шаблон баг-репорта в GitHub Issues, деплой из CI.
- **hecho-e2e** — https://github.com/ge-a-nikitin/hecho-e2e — набор e2e-тестов на Playwright, который гоняет настоящий стек (веб, API, PostgreSQL, SeaweedFS): сценарии заказов клиента в проектах chromium и Pixel 7, теги, синхронизация каталога тест-кейсов с Qase TestOps, SUT, зафиксированный по коммиту, отчёты.
- **TShop** — магазин мерча T-Bank внутри мобильного приложения; интеграции фулфилмента заказов, единственный QA.
- **Birthday 4.0** — сервис поздравлений с розыгрышем призов, выросший из разовой акции в продукт; тестирование с нуля, автоматизация на фреймворке, затем два года — держатель знаний, точка входа по вопросам тестирования: обращения, разборы, триаж, доработки к релизу.
- **New Year 2024** — ежегодная геймифицированная акция; интеграции через топики Kafka и проверка диплинков между версиями приложения. Собрал из логов версии, которыми реально пользовались люди, и построил графики на вики департамента — по этим числам решали, какие версии тестировать.
- **«Пушкин 225»** — задания, призы и лояльность к 225-летию поэта; проект с нуля, разработка апрель – июль 2024, жизнь июль – август 2024. Единственный QA на фронтенде и бэкенде: требования по shift-left, интеграции через топики Kafka для новых типов заданий, API-скрипты для работы с проектом, тестовые данные с помощью ИИ. От автоматизации отказались сознательно: сроки, кадры и глубина функционального тестирования не оставляли места набору, который бы себя окупил.
- **Stickermania with Vkusno i Tochka** — совместная акция с сетью ресторанов; параметризованные автотесты, модуль авторизации, добавленный в общий шаблон, чек-листы с помощью ИИ. R&D по автоматизации тестирования вёрстки на параметризации показал высокий процент ложных срабатываний — от внедрения отказались и сэкономили ресурсы команды; саму параметризацию перенёс на функциональные автотесты и сократил дублирование.
- **UserAgent Redirect** — исследование user-agent браузеров для страницы-фолбэка; результаты — в ADR и гайде по использованию, автотесты на больших списках значений, которые ручной проверкой не покрыть, доклад для команд департамента.
- **Prizopoly 2.0** — призовая акция с аудиторией до 3M+ уникальных пользователей в месяц; нагрузочное тестирование на k6 по всем критичным эндпоинтам, три запланированных сценария сведены в один универсальный, нагрузочное тестирование перешло от бэкенда к QA.

ОБРАЗОВАНИЕ

- **Санкт-Петербургский технологический колледж** (sptcol.ru) — программирование, среднее профессиональное образование — 2023 – 2027 (ожидается).
- **Петербургский государственный университет путей сообщения Императора Александра I** (ПГУПС) — механический факультет, подвижной состав железных дорог и вагонное хозяйство — 2007 – 2011, окончено три курса (незаконченное высшее).

Обучение. Внутренние курсы работодателя перед каждым новым участком: сквозная автоматизация на eva.io и нагрузочное тестирование на k6. Внешних сертификатов нет: SQL, теория тестирования, программирование и английский — колледж, вуз и самообразование.

ЯЗЫКИ

Русский — родной (C2). Английский — B2, рабочий уровень чтения, письма и речи: документация, задачи, отчёты и интервью.