

# Портфолио

Разработка · интеграции · CRM · приём платежей · автоматизация

**Алексей Никонович — full-stack разработчик**

Ниже — реальные проекты последнего времени: связка сайтов и CRM, приём платежей, работа с внешними API и оборудованием, аудит и восстановление данных. Названия компаний, суммы и контакты убраны, цифры настоящие — из журналов рабочих систем.

## Как я работаю

- Работа на копии, а не на боевом. Правки живут в режиме предпросмотра: их видно по моей ссылке, посетители сайта — нет. Рабочая версия всё это время не меняется.
- Резервная копия перед каждым шагом. Настройки, структура и изменяемые файлы сохраняются до правки. Откат — одна команда.
- Проверка боевым прогоном. Тестовая регистрация, платёж в тестовом режиме, отправка формы с контролем, что письмо и заявка дошли. «Готово» говорю после проверки, а не после написания кода.
- Аудит до запроса материалов. Сначала смотрю, что уже есть в системе. На одном проекте так нашлись 13 услуг с фото, 10 логотипов партнёров, 7 сертификатов и реквизиты — всё лежало в базе и не показывалось на сайте.

## Заявки с сайта в CRM: попутно нашли, что письма не доходили год

*Металлоторговая компания, Московская область.*

Сайт на коробочном Битриксе, CRM — в облачном Битрикс24, две разные системы. Заявки с форм уходили только письмом. Сделал обработчик на событие формы, отправку через входящий вебхук портала методом `crm.lead.add`, отсечение дублей по телефону и почте через `crm.duplicate.findbycomm`, нормализацию номера к `+7XXXXXXXXXX` и запись в лог при сбое, чтобы заявка не терялась.

По ходу вскрылось главное: за последний год с сайта не ушло ни одного письма. Тринадцать заявок висели в очереди неотправленными, самая старая — полтора года; восемь живые, с телефонами и конкретными запросами на трубы и профнастил. Причина — встали фоновые задачи Битрикса, которые разбирают очередь. Заявки выгрузил в файл, заказчик обзвонил.

```
форма на сайте
-> событие onAfterResultAdd
-> входящий вебхук портала (crm.lead.add)
-> дедуп по телефону/почте (crm.duplicate.findbycomm)
-> лид в Битрикс24
```

**Результат. Заявки с сайта попадают в CRM без дублей; найдено и возвращено 8 живых потерянных обращений, о которых бизнес не знал.**

## Приём платежей через Робокасса

*Сервис по подписке для авторов.*

Сделал формирование ссылки оплаты с подписью и фискальным чеком, обработчик Result URL с проверкой подписи и идемпотентную обработку повторных уведомлений: Робокасса шлёт их

регулярно, и без защиты подписка продлевалась бы дважды, а промокод сгорал повторно.

Дефект нашёл до запуска: в тестовом режиме Робокасса проверяет подпись отдельной парой паролей, а код использовал одну пару на оба режима. Платёж падал бы с «неверной подписью» на стороне банка, а в логах сервиса было бы чисто — такое ищут днями. Развёл тестовые и боевые пароли.

```
кнопка оплаты
-> Робокасса (оплата)
-> Result URL: проверка подписи
-> идемпотентное продление подписки
```

**Результат. Рабочий приём платежей с чеком и защитой от двойного продления; критичная ошибка режима поймана до первого реального платежа.**

## 1С Альфа-Авто: почему не работал планировщик загрузки постов

*Сеть автосервисов.*

При аудите выяснилось, что планировщик не работал не из-за ошибки, а потому что под ним не было данных: постов не заведено, графиков смен нет, цех демонстрационный из коробки, расчёт шёл по общему графику компании без учёта табеля. Завёл графики смен «Слесари 4/2» и «Приёмщики 4/2», привязал сотрудников к цехам и подразделениям.

Из 44 человек привязал 33: шестеро складских не участвуют в планировании постов, ещё шести не нашлось в базе — вскрылось расхождение табеля и учётной системы по фамилиям, отдал заказчику списком.

**Результат. Планировщик считает загрузку постов по реальным сменам; попутно найдено расхождение кадрового табеля и базы 1С.**

## Автопропуск по номеру машины: камера → распознавание → шлагбаум

*Закрытая территория, около 600 машин в списке, работа без оператора.*

Шлагбаум открывается сам: камера распознаёт номер, программа сверяет со списком жильцов и подаёт импульс на привод. Событийная подписка на камеру по HTTP Digest (событие ловит не только пересечение линии, но и машину, уже вставшую перед стрелой), сверка с локальной базой, импульс на USB-реле. Список машин ведётся на сервере и раз в минуту забирается на объект через обратный SSH-туннель (белого IP на объекте нет). Управление, заявки на гостей, ежедневный отчёт и ежечасная автопроверка — через Telegram-бота.

| День                  | 1     | 2     | 3     | 4    | 5     | 6    |
|-----------------------|-------|-------|-------|------|-------|------|
| Проездов без задержки | 89.8% | 98.0% | 96.7% | 100% | 98.6% | 100% |

Рост — доработка разбора номеров, камеру и железо не меняли. Отдельно: система однажды «тихо отказала» — процесс жив, снимки обновляются, а въезд не обрабатывался 19 часов. После этого поставил ежечасную проверку, которая контролирует не признаки жизни, а результат: приходят ли номера и открывается ли шлагбаум. Простоев с тех пор нет.

```
op = urllib.request.build_opener(
    urllib.request.HTTPDigestAuthHandler(mgr))
url = base + "/eventManager.cgi?action=attach&codes=[...]&heartbeat=20"
r = op.open(url, timeout=CAM_READ_TIMEOUT)    # timeout обязателен:
...                                           # без него подписка
plate = d["Object"]["Text"].strip().upper()  # «умирает молча»
```

**Результат. Проезд по списку без оператора; после доработки распознавания — до 100% машин без задержки, с ежечасным контролем реального результата.**

## Восстановление клиентской базы из переписок

*Металлоторговая компания.*

Заказчик считал 10 419 учёток мусором и собирался удалить. Проверил — это клиенты с Авито за три года, контактов в анкетах не было ни у одного. Прошёл 218 903 сообщения открытых линий, вытащил телефоны и почты из текста переписок, нормализовал номера, собрал таблицу с именем, контактами и датами обращений.

**Результат. 2113 клиентов с контактами и 2132 уникальных телефона спасены от удаления. Сначала посчитал, потом предложил решение.**

## Диагностика: регистрацию не мог завершить никто

*Веб-сервис по подписке.*

Ссылка подтверждения почты из письма вела на домен без такой страницы — 404. После правки домена всплыло второе: письмо звало на /confirm/{токен}, а обработчик жил по /api/confirm/{токен}; общий перехватчик отдавал главную страницу, и человек думал, что почту подтвердил. На деле подтверждения не было, а без него в сервисе закрыто сохранение работ. Добавил маршрут с редиректом, проверил живой регистрацией.

**Результат. Регистрация заработала для всех; проблема найдена боевым прогоном, а не по отчёту «должно работать».**