roadmap.pdf · версия 2.0 · июль 2026 · методический roadmap

# AI-Native QA

Ручное тестирование с нуля и ИИ как рабочий инструмент. Системная программа для абсолютного новичка: от первых проверок до end-to-end проекта и защищаемого портфолио.

16 недель | 10–12 часов в неделю | 10 модулей | сквозной проект | старт — 1 сентября

// Главный принцип: ИИ не изучается отдельно от профессии. Каждая компетенция проходит цикл: самостоятельно → с ИИ → проверка → доказательство → решение человека.

Раздел 01 / Назначение программы

## Для кого этот курс — и для кого нет

Курс рассчитан на человека без опыта в IT, тестировании и программировании. За 16 недель участник получает базу ручного QA и учится применять ИИ внутри реальных тестовых задач, не подменяя им понимание продукта.

**Итог:** начинающий ручной QA-инженер, который умеет анализировать требования, проектировать проверки, тестировать web, API и данные, оформлять доказательства и безопасно использовать ИИ для ускорения работы.

для кого

- Для абсолютных новичков, которым нужен последовательный вход в профессию.
- Для людей без опыта программирования: код — только в простых прикладных QA-задачах.
- Для тех, кто готов заниматься 10–12 часов в неделю и объяснять собственные решения.

что программа не пытается сделать

- Не превращает новичка в автоматизатора, performance-инженера или специалиста по безопасности.
- Не учит копировать ответы модели: любой значимый AI-результат проверяется и защищается человеком.
- Не гарантирует трудоустройство — создаёт навыки и артефакты, которые можно подтвердить.

Раздел 02 / Структура · почему 16 недель

## Четыре части программы

Основная часть курса занимает около трёх месяцев. Дополнительное время — не «второй курс про ИИ», а безопасная практика, сборка повторяемого workflow и защита сквозного проекта.

| | |
|---|---|
| недели 1–7 | **Вход и база QA.** Профессия, процессы, требования, тест-дизайн, документация и дефекты |
| недели 8–13 | **Техническая база.** Web, DevTools, HTTP, API, SQL, интеграции, логи и поставка изменений |
| недели 14–15 | **Расширение и AI-native практика.** Mobile, NFR и один повторяемый workflow ручного тестировщика |
| неделя 16 | **Финальная защита.** End-to-end прогон, quality report, практическая задача и защита проекта |

Раздел 03 / Измеримые результаты

## Что участник умеет к завершению

- Объяснить место QA в жизненном цикле продукта и связать проверки с рисками.
- Проанализировать требования и макеты, найти пробелы и сформулировать вопросы.
- Применить основные техники тест-дизайна и провести исследовательскую сессию.
- Оформить чек-листы, тест-кейсы, матрицу покрытия, баг-репорты и quality report.
- Проверить web-интерфейс и подтвердить вывод данными DevTools.
- Собрать и запустить Postman-коллекцию по API-контракту.
- Проверить данные SQL-запросами и провести базовый анализ интеграции и логов.
- Адаптировать проверки под mobile и назвать основные нефункциональные риски.
- Использовать ИИ безопасно, фиксировать источники и проверять каждый значимый вывод.
- Защитить сквозной проект и показать подтверждаемые учебные артефакты.

Раздел 04 / Учебная модель

## Из чего состоит каждая неделя

- Короткая теория: понятия, демонстрация и источник истины — 20%
- Самостоятельная практика без ИИ — фиксация ручного baseline — 35%
- AI-assisted практика: расширение и проверка того же навыка — 25%
- Review и защита: разбор ошибок, доказательств и решений — 20%

```
обязательный цикл → источник истины → ручной анализ → AI-черновик или review → запуск или проверка инструментом → фиксация ошибок → решение человека
```

# Один продукт на весь курс

Все модули работают на одном учебном продукте (интернет-магазин, сервис записи, доставка или личный кабинет) с web-интерфейсом, API, ролями, базой данных и внешней интеграцией. Что накапливается:

- Описание продукта, пользователей, ролей, сред и основных рисков.
- Review требований, тестовая модель и набор тестовых данных.
- Чек-листы, тест-кейсы, TMS-прогоны, дефекты и матрица покрытия.
- Web, API и SQL-результаты с техническими доказательствами.
- Карта интеграций, log-разбор и risk-based regression pack.
- AI-журнал, проверенные prompts и один воспроизводимый AI-workflow.
- Итоговый quality report и портфолио-кейс.

Раздел 06 / Карта программы · 10 модулей · 16 недель

## Карта программы

| | |
|---|---|
| M0 · нед 1 | Старт, профессия и безопасная работа с ИИ — рабочая среда и первый проверенный AI-assisted результат |
| M1 · нед 2–3 | Основы тестирования и жизненный цикл продукта — роль QA, риски, процессы, релизный цикл |
| M2 · нед 4–5 | Требования, макеты и проектирование проверок — тестовая модель и техники тест-дизайна |
| M3 · нед 6–7 | Тестовая документация, дефекты и TMS — рабочие артефакты, прогоны, доказательные баг-репорты |
| M4 · нед 8–9 | Клиент-сервер, Web и DevTools — проверка интерфейса и локализация проблемы по данным браузера |
| M5 · нед 10–11 | HTTP, API, Postman и авторизация — самостоятельное тестирование API по контракту |
| M6 · нед 12–13 | SQL, интеграции, логи и поставка изменений — проверка данных и локализация межсервисных ошибок |
| M7 · нед 14 | Mobile и нефункциональные проверки — платформенные и нефункциональные риски junior QA |
| M8 · нед 15 | AI-native workflow ручного тестировщика — повторяемый и измеримый процесс работы QA с ИИ |
| M9 · нед 16 | Финальный проект, защита и портфолио — защищённый end-to-end кейс с артефактами |

Раздел 07 / Содержание модулей

## Модули подробно

M0 · Старт, профессия и безопасная работа с ИИ — 1 неделя · 8–10 часов

**Результат:** участник понимает маршрут курса, настраивает рабочую среду и отличает собственное решение от AI-черновика.

темы

Роль ручного тестировщика · объект тестирования, качество, дефект, риск, доказательство · цифровая грамотность и рабочая среда · источники истины · как работает LLM: контекст, вероятностный ответ, ограничения · галлюцинации и выдуманные ссылки · персональные данные, NDA, токены, обезличивание.

практика и артефакты

Настройка среды, трекера и AI-журнала · поиск ошибок в подготовленном AI-ответе · описание формы регистрации без ИИ и с ИИ. Артефакты: чек-лист среды, личный протокол безопасной работы с ИИ, baseline-задача + AI-версия, первая запись AI-журнала.

```
hard gate → ✓ показывает свой вклад, называет источник истины, объясняет проверку AI-ответа, не передаёт модели чувствительные данные
```

M1 · Основы тестирования и жизненный цикл продукта — 2 недели · 20–24 часа

**Результат:** участник понимает место тестирования в разработке, различает основные проверки и принимает решения на основе риска.

темы

Тестирование, QA и QC · ценность для пользователя и стоимость дефекта · семь принципов тестирования · верификация и валидация · уровни тестирования · smoke, sanity, retest, regression · позитивные, негативные и злоупотребляющие сценарии · SDLC/STLC, Agile, Scrum, Kanban · роли в команде и церемонии · среды, сборка, релиз, rollback.

роль ИИ и практика

ИИ — наставник и генератор гипотез: риски и негативные сценарии с фильтрацией шума, суммаризация refinement с проверкой. Практика: классификация проверок, карта рисков для оплаты/регистрации, путь задачи от идеи до релиза. Артефакты: карта понятий QA, карта жизненного цикла, реестр рисков, smoke и post-release чек-листы.

```
hard gate → ✓ объясняет связь проверки с риском, не путает retest с regression, не считает QA единственным ответственным за качество
```

M2 · Требования, макеты и проектирование проверок — 2 недели · 20–24 часа

**Результат:** участник находит пробелы в требованиях и строит достаточное покрытие, связанное с рисками и источниками.

темы

Виды требований · user story, use case, acceptance criteria, спецификация, Figma, API-контракт · качество требований: полнота, однозначность, тестируемость · серые зоны · декомпозиция фичи и тестовая модель · классы эквивалентности и границы · таблицы решений · состояния и переходы · pairwise · исследовательское тестирование и charter.

роль ИИ и практика

ИИ извлекает утверждения из требований с указанием источника, ищет неоднозначности (факт/допущение/вопрос), расширяет ручной набор тестов — участник удаляет неподтверждённое. Практика: review пакета требований, проектирование проверок вручную + с ИИ, exploratory-сессия. Артефакты: список вопросов и противоречий, реестр рисков + матрица покрытия, тестовая модель, charter и отчёт сессии.

```
hard gate → ✓ каждая существенная проверка связана с требованием или риском; участник объясняет выбор техники и где заканчивается покрытие
```

M3 · Тестовая документация, дефекты и TMS — 2 недели · 20–24 часа

**Результат:** участник создаёт понятные тестовые артефакты, проводит прогон и оформляет воспроизводимые дефекты без выдуманной причины.

темы

Чек-лист vs тест-кейс vs charter · предусловия, шаги, данные, ожидаемый результат · test suite, test run, статусы, регрессия · матрица трассируемости · тест-план и лёгкая стратегия · структура баг-репорта · severity и priority, жизненный цикл дефекта · скриншот, видео, HAR, Console, Network как доказательство · TMS и отчётность · quality report.

роль ИИ и практика

ИИ — генератор и редактор черновиков: поиск непроверяемых шагов и выдуманных деталей, улучшение языка баг-репорта без добавления root cause, преобразование в формат TMS, поиск дублей в матрице. Практика: фича чек-листом и кейсами, прогон с 3+ дефектами, итоговый отчёт. Артефакты: набор кейсов и чек-листов, матрица трассируемости, 3 доказательных баг-репорта, TMS-прогон + quality report.

```
hard gate → ✓ другой человек выполняет проверки без устных пояснений; дефекты воспроизводимы; в документации нет придуманных причин
```

M4 · Клиент-сервер, Web и DevTools — 2 недели · 20–24 часа

**Результат:** участник понимает путь данных в web-системе, проверяет интерфейс и использует DevTools для локализации проблемы.

темы

Клиент, сервер, база, внешние сервисы · монолит и микросервисы · DNS, TCP, TLS, HTTPS прикладно · путь запроса от ввода до ответа · HTML, CSS, JS, DOM для тестировщика · Elements, Console, Network, Application, Performance · headers, payload, timing, waterfall · cookies и storage · валидация форм · адаптивность, совместимость, базовая доступность.

роль ИИ и практика

ИИ строит черновую схему потока данных, анализирует сохранённые Network-запросы и HAR только по фактическим данным, формирует баг-репорт из доказательств. Практика: путь действия от интерфейса до сервера, локализация дефектов через Console/Network, проверка формы в разных состояниях и браузерах. Артефакты: схема клиент-серверного взаимодействия, web/DevTools чек-лист, HAR-разбор, баг-репорты с техдоказательствами.

```
hard gate → ✓ показывает факт, на котором основан вывод, и честно говорит, когда данных недостаточно для назначения frontend/backend
```

M5 · HTTP, API, Postman и авторизация — 2 недели · 22–26 часов

**Результат:** участник тестирует API независимо от интерфейса: контракт, методы, данные, ошибки и права доступа.

темы

Запрос и ответ: method, URL, headers, body, status · GET/POST/PUT/PATCH/DELETE, идемпотентность · коды 2xx–5xx, отдельно 400/401/403/404/409/422/500 · JSON и проверка схемы · REST, обзорно SOAP/GraphQL/WebSocket · Swagger/OpenAPI как контракт · Postman: коллекции, окружения, переменные, скрипты · Basic, Bearer, JWT, OAuth 2.0 · негативные проверки, права, повторные запросы · кеширование и ETag.

роль ИИ и практика

ИИ генерирует запросы и Postman-тесты строго по OpenAPI; участник ищет выдуманные endpoints, поля и коды в AI-черновике; всё проверяется реальным запуском. Практика: CRUD-коллекция с позитивом и негативом, проверка ролей и чужих данных, локализация изменения поведения API. Артефакты: Postman-коллекция + окружение, API-чек-лист, матрица ролей и авторизации, отчёт о дефектах.

```
hard gate → ✓ коллекция запускается повторно, тесты связаны с контрактом, участник различает ошибку запроса, прав и сервера
```

M6 · SQL, интеграции, логи и поставка изменений — 2 недели · 22–26 часов

**Результат:** участник проверяет данные безопасными запросами и разбирает интеграции, логи, очереди и CI/CD без лишней специализации.

темы

Реляционная база: таблицы, ключи, связи · SELECT, WHERE, ORDER BY, DISTINCT, NULL · JOIN, GROUP BY, HAVING, агрегаты · проверка дат, дублей, статусов · INSERT/UPDATE/DELETE и транзакции в безопасной среде · связь UI–API–база · интеграции, контракт, зависимость · синхронное и асинхронное взаимодействие, webhook · mock и stub · логи и correlation ID · Kafka и RabbitMQ концептуально · Git, CI/CD, deploy и регрессия по изменению.

роль ИИ и практика

ИИ пишет SQL только по реальной схеме, объясняет запросы пошагово, суммаризирует логи со ссылками на строки, генерирует сценарии отказов с фильтрацией по реальной архитектуре. Практика: запросы от SELECT до JOIN/GROUP BY, проверка данных после действия в UI/API, разбор интеграционного сбоя, минимальный regression pack по diff. Артефакты: набор SQL-запросов, чек-лист качества данных, карта интеграций + матрица отказов, отчёт log-расследования, risk-based regression pack.

```
hard gate → ✓ объясняет запросы и выводы, соблюдает безопасный режим работы с данными, не выдаёт гипотезу за установленную причину
```

M7 · Mobile и нефункциональные проверки — 1 неделя · 10–12 часов

**Результат:** участник понимает особенности mobile и основные нефункциональные риски — в границах роли junior QA.

темы

Native, hybrid, cross-platform, mobile web · установка, обновление, миграция · разрешения, уведомления, deeplink · foreground/background, прерывания · сеть, offline, переключение Wi-Fi/mobile · device matrix · эмулятор, ADB, logcat, proxy на уровне знакомства · производительность, безопасность, доступность, usability · границы junior QA.

роль ИИ и практика

ИИ строит device matrix по аудитории, генерирует сценарии прерываний с удалением нереалистичных, превращает расплывчатое NFR-требование в проверяемый вопрос. Практика: установка/обновление/разрешения/сеть, воспроизведение проблемы на эмуляторе, риск-чек-лист NFR. Артефакты: mobile-чек-лист + device matrix, сценарии прерываний, мобильный баг-репорт с логами, карта нефункциональных рисков.

```
hard gate → ✓ адаптирует проверки под mobile, указывает устройство и состояние приложения, ограничивает выводы фактическими данными
```

M8 · AI-native workflow ручного тестировщика — 1 неделя · 10–12 часов

**Результат:** участник собирает повторяемый процесс работы QA с ИИ, измеряет пользу и сохраняет человеческий контроль.

темы

Context engineering: инструкции, структура, источники, ограничения · структурированный вывод: таблицы, JSON, Markdown · библиотека проверенных запросов для требований, тест-дизайна, API, SQL, логов · AI-review по рубрике · простые QA-утилиты: генератор данных, parser логов, diff требований · agentic workflow: план, действие, наблюдение, отчёт · read-only режим, разрешения, журнал действий · измерение эффекта · честное описание AI-вклада в портфолио.

роль ИИ и практика

Практика: автоматизация одного повторяющегося QA-процесса с ручным подтверждением, сборка утилиты или шаблона, воспроизводимого другим человеком, замер «до и после». Артефакты: личный AI-native playbook, библиотека prompts и рубрик, утилита или workflow с инструкцией, замер эффекта + карта ограничений.

```
hard gate → ✓ workflow воспроизводится, внешние изменения требуют подтверждения, заявленная польза подтверждена замером
```

M9 · Финальный проект, защита и портфолио | 1 неделя · 12–16 часов

**Результат:** участник завершает сквозной проект, защищает решения и оформляет подтверждаемый учебный кейс.

темы

Финальная приоритизация рисков · end-to-end прогон web, API, данных, интеграций · smoke, regression, retest, post-release · итоговый quality report и остаточные риски · аудит AI-журнала · портфолио-кейс: задача, контекст, действия, доказательства, результат · самопрезентация · практическая задача с неизвестной фичей.

роль ИИ и практика

ИИ — gap analysis по рубрике, редактура портфолио без выдуманных фактов, симуляция вопросов. Практика: финальный прогон и защита остаточных рисков, задача по незнакомой фиче, защита проекта + mock-собеседование. Артефакты: полный capstone-пакет, quality report, портфолио-кейс с обезличенными доказательствами, план дальнейшего развития.

```
hard gate → ✓ не менее 80% по финальной рубрике, самостоятельное объяснение решений, выполнение незнакомой практической задачи
```

Раздел 08 / Система оценки

## Как оценивается каждый модуль

Модуль считается пройденным после практики и короткой защиты. Хорошо оформленный AI-ответ без понимания не засчитывается.

| | |
|---|---|
| Практический артефакт: корректность, полнота, доказательства | 45% |
| Понимание и защита: объяснение решений, ответы на вопросы | 25% |
| AI-журнал: источник, вклад ИИ, способ проверки, ошибки | 15% |
| Проверка знаний: термины, связи, выбор действия | 15% |

| | |
|---|---|
| порог модуля | не менее 70% общей оценки |
| порог финала | не менее 80%; критические правила безопасности — на 100% |
| пересдача | AI-сгенерированный артефакт пересдаётся, если участник не может объяснить его содержание; после пересдачи — разбор причины первоначальной ошибки |
| definition of done | ручная часть или baseline → обязательный артефакт → AI-вклад в журнале → выводы подтверждены источником или запуском → замечания review исправлены → короткая защита → результат в сквозном проекте |

Раздел 09 / Минимальный AI-журнал

## Что фиксируется по каждой значимой задаче

| | |
|---|---|
| задача | какой рабочий результат требовался |
| источник истины | требование, контракт, данные, лог, документация или наблюдаемое поведение |
| инструкция | ключевой запрос, контекст, ограничения и формат ответа |
| результат ИИ | ссылка или сохранённая версия |
| проверка | источник, инструмент или запуск, которым проверялся ответ |
| ошибки | что модель пропустила, придумала или исказила |
| решение человека | что принято, отклонено или исправлено |
| эффект | время, полнота и число итераций |

Раздел 10 / Минимальный стек

## Инструменты — по функции, а не по бренду

| | |
|---|---|
| работа с ИИ | доступная LLM-платформа и отдельная проверка источников |
| управление | Jira, Confluence или аналоги |
| документация | TMS и табличные форматы |
| web | Chrome или Firefox, DevTools, Figma |
| api | OpenAPI, Postman или аналоги |
| данные | PostgreSQL, DBeaver |
| интеграции | mock server, Charles, Kibana или аналоги |
| mobile | Android Studio, ADB |

Раздел 11 / После базового курса

## Дополнительные треки развития

Эти темы сознательно вынесены из обязательного ядра, чтобы не перегружать вход в профессию. Они осваиваются отдельными треками после базы:

| | |
|---|---|
| Лёгкая автоматизация QA — API/UI автотесты и работа с кодом | 4–6 недель |
| Тестирование AI-функций и evals — LLM, RAG, недетерминированные результаты | 3–4 недели |
| Нагрузочное тестирование — k6/JMeter, метрики, анализ bottleneck | 3–4 недели |
| Безопасность приложений — угрозы, OWASP, безопасная практика | 3–4 недели |
| Интеграции, Kafka и CI/CD — очереди и поставка изменений как зона QA | 2–3 недели |

условия первого потока

✓ набор открыт

**Старт — 1 сентября · 10 мест · 40 000 ₽** (цена первого потока; следующие — 50 000 ₽). Авторы проверяют каждую работу лично.

Авторы курса: **Павел Виноградов** (Team Lead QA Engineer, 7 лет в тестировании) и **Эд Зияев** (QA-инженер, основатель EDVERSE).

// Мы не обещаем трудоустройство — мы даём навыки и артефакты, которые можно подтвердить.
// Версия программы 2.0, июль 2026. Программа обновляется после каждого потока.