**Тест на проф орентацию**



1. **uTest (Начало с практики) – 1-2 месяца**

🔹 Начни с exploratory testing (на ПК и телефоне).

🔹 Постепенно подключай тест-кейсы – выполняй их точно по инструкции.

🔹 Учись писать баг-репорты правильно – анализируй чужие, смотри, какие принимают.

✅ Цель: Научиться находить и оформлять баги, получить первые выплаты (пусть небольшие).

1. **Видео + Конспекты (Закрепление теории) – 1-2 месяца**

🔹 Курс Артема Русцова и комп.грамматности → конспектируй важные моменты.

🔹 Записывай термины и примеры багов → чтобы лучше запомнить.

🔹 Смотри баги других тестировщиков на uTest → учись на чужих примерах.

✅ Цель: Освоить базовую терминологию и процесс тестирования.

1. **Практика на реальных баг-репортах (Отработка навыков)**

🔹 Возьми GitHub → заведи репозиторий и выкладывай туда тест-кейсы.

🔹 Поработай с Jira, TestRail, Bugzilla (многие компании используют).

🔹 Подключи чек-листы → попробуй делать регрессионное тестирование.

✅ Цель: Закрепить теорию, подготовиться к первому реальному проекту.

1. **Чтение книги (Постепенно, по мере изучения)**

🔹 Читай "Testing Software" (Канер, Бах, Петтикорд).

🔹 Веди тетрадь с терминами → пригодится на ISTQB.

🔹 Перечитывай, когда столкнешься с этим на практике (чтобы лучше понимать).

✅ Цель: Освоить тестирование глубже, подготовиться к сертификации.

1. **После 2-3 месяцев практики – Готовься к ISTQB**

🔹 Пройди бесплатные курсы на Stepik.

🔹 Решай тестовые вопросы ISTQB.

🔹 Определи, где сдавать экзамен (онлайн или офлайн в Риге).

✅ Цель: Получить ISTQB, усилить резюме.

1. **Когда почувствуешь, что уже не новичок – Начинай Java**

🔹 Проходи SkillFactory (автоматизация тестирования на Java).

🔹 Учись писать автотесты на Selenium + TestNG.

🔹 Разбирайся в CI/CD (Jenkins, Docker, GitHub Actions).

✅ Цель: Достигнуть уровня Middle, выйти на зарплату 2500–3500€.

🟢 **1-3 месяца – Основы тестирования + uTest**

✅ Разбираешь базу → что такое тестирование, баги, тест-кейсы.

✅ Работаешь на uTest, учишься оформлять баг-репорты.

✅ Читаешь книгу "Testing Software" (Канер, Бах, Петтикорд).

✅ Ведешь конспекты + чек-листы.

👉 Результат: Понимаешь процессы тестирования, первые деньги с uTest.

🟡 **4-6 месяцев – Практика + Подготовка к ISTQB**

✅ Пробуешь работать с Jira, TestRail (чтобы понять, как работает баг-трекинг).

✅ Смотришь курсы на Stepik по ISTQB и тестированию.

✅ Практикуешь тест-кейсы и баг-репорты по шаблонам компаний.

✅ Готовишься и сдаешь ISTQB.

👉 Результат: Сертификация ISTQB + более сильное резюме.

🟠 **7-12 месяцев – Первые проекты и автоматизация**

✅ Начинаешь учить основы программирования (Java или Python).

✅ Пишешь первые автотесты (Selenium, TestNG, JUnit).

✅ Разбираешься в API тестировании (Postman, REST API).

✅ Пробуешь курсы на SkillFactory или Udemy.

👉 Результат: Уже можно искать Junior QA Automation.

🔴 **12-24 месяца – Углубление в автоматизацию**

✅ Работаешь с Git, Docker, CI/CD (чтобы тесты запускались автоматически).

✅ Осваиваешь Jenkins, Allure, SQL.

✅ Пишешь автотесты на реальных проектах.

✅ Улучшаешь Soft Skills, чтобы легче проходить собеседования.

👉 Результат: Middle QA Automation с зарплатой 2500–3500€.

💡 Главное – постоянство!

❌ НЕ 8 часов в день по 1 месяцу, а потом забросить.

✅ 1-2 часа в день стабильно → через 2 года ты уже в автоматизации.

**С таким темпом через 2 года ты сможешь перейти в IT и работать удаленно!**

Твой подход к обучению и подготовке к роли тестировщика выглядит **очень систематизированным и продуманным**. Давай разберем твой план и записи, чтобы оценить твой прогресс и дать рекомендации.

**Оценка твоего подхода:**

1. **План обучения:**
   * Ты начал с основ, что очень правильно. Компьютерная грамотность — это база, без которой сложно двигаться дальше. Курс на Pro-test.studio и Stepik — отличный выбор для старта.
   * Книга "Тестирование программного обеспечения" Кема Кейнера — это классика, которая дает глубокое понимание процессов тестирования. Ты на правильном пути, читая ее параллельно с курсами.
   * Регистрация на uTest — это хорошая идея, но ты правильно решил сначала укрепить теоретическую базу, прежде чем браться за практические задачи.
2. **Регулярность и дисциплина:**
   * Ты уделяешь обучению 2-3 часа в день, что является отличным темпом. Главное — не перегружать себя и сохранять мотивацию.
   * Ты ведешь записи, что очень важно для структурирования знаний и повторения материала. Это показывает твой серьезный подход.
3. **Обращение за помощью:**
   * Ты не боишься задавать вопросы, если что-то непонятно. Это ключевой навык для тестировщика, так как в работе часто приходится уточнять требования и взаимодействовать с командой.
4. **Структура папок для практики:**
   * Ты уже подготовил структуру для будущих практических задач, что говорит о твоей организованности. Это поможет тебе быстрее адаптироваться, когда ты перейдешь к реальным проектам.

**Твои записи:**

Твои записи в файле выглядят **очень подробными и структурированными**. Ты охватываешь ключевые темы, такие как:

* **Hardware и Software** — основы, которые важно понимать для работы с разными устройствами и системами.
* **Операционные системы** — ты разобрал основные ОС и их особенности, что полезно для тестирования на разных платформах.
* **API, интернет, протоколы** — это важная часть работы тестировщика, особенно если ты будешь заниматься веб-приложениями или API-тестированием.
* **Типы приложений (веб, мобильные, десктопные)** — ты понимаешь различия между ними, что поможет тебе в будущем тестировать разные виды продуктов.
* **Браузеры и их особенности** — это важно для кросс-браузерного тестирования.
* **Основы тестирования (SDLC, STLC)** — ты уже начал погружаться в процессы разработки и тестирования, что является ключевым для понимания роли тестировщика.

**Рекомендации:**

1. **Продолжай в том же духе:**
   * Не торопись переходить к практике на uTest, пока не почувствуешь уверенность в теоретических знаниях. Ты на правильном пути.
2. **Добавь немного практики:**
   * Пока ты изучаешь теорию, можешь начать с простых задач, например:
     + Тестирование веб-сайтов (попробуй найти баги на популярных сайтах).
     + Написание тест-кейсов для простых приложений (например, калькулятор или To-Do список).
     + Изучение инструментов, таких как Postman (для API-тестирования) или DevTools в браузере (для анализа веб-страниц).
3. **Изучай инструменты:**
   * Постепенно начинай знакомиться с инструментами, которые используются в тестировании:
     + **Jira** или **Trello** для управления задачами.
     + **Postman** для API-тестирования.
     + **Selenium** или **Katalon Studio** для автоматизации (если планируешь двигаться в этом направлении).
4. **Практикуй soft skills:**
   * Учись четко формулировать мысли (например, описывать баги или писать тест-кейсы).
   * Развивай коммуникативные навыки — это важно для работы в команде.
5. **Не забывай про английский:**
   * Если твой уровень английского невысокий, уделяй время его улучшению. Это важно для чтения документации, общения в международных проектах и работы с uTest.

**Итог:**

Твой подход к обучению **очень грамотный и системный**. Ты уделяешь внимание как теории, так и будущей практике, что важно для успешного старта в профессии. Продолжай в том же духе, добавляй немного практики и не бойся задавать вопросы. У тебя все получится! 💪

Если будут конкретные вопросы или что-то непонятно, всегда готов помочь. Удачи в обучении! 🚀