# **Дневник проекта**

Я, Сидорова Полина Олеговна, выполняю в проекте «Сайт для планирования путешествий» роль тестировщика. Веду этот дневник, чтобы:

* **Отследить свой профессиональный рост**: фиксировать, какие навыки я развиваю.
* **Понять, как теория превращается в практику**: записывать ключевые выводы из курса, статей, учебной литературы, которые я изучаю, и показывать, как применяю их в проекте.
* **Фиксировать успехи и неудачи**: честно отражать и победы, и моменты, когда что-то пошло не так, чтобы извлекать уроки и не повторять ошибок.
* **Создать руководство для себя и коллег**: по итогу получатся не просто даты и задачи, а цепочка размышлений и решений — как зеркало процесса разработки и тестирования.

«Тестировщик — это голос пользователя в команде разработки».  
Я воспринимаю эту фразу как девиз: хочу делать продукт понятным и надёжным, предвидеть проблемы ещё до того, как они проявятся, и делиться своими инсайтами.

Причина выбора книги Святослава Куликова «Тестирование программного обеспечения. Базовый курс»: это бесплатное, регулярно обновляемое издание опытного преподавателя, в котором доходчивым языком изложены базовая терминология, классификация видов тестирования, правила составления тест-кейсов и чек-листов и приёмы фиксации дефектов . Я решила, что именно её принципы станут надёжным каркасом моей практики в проекте.

## **Первая неделя (03.02.2025 – 09.02.2025)**

Девиз недели: «Только вперёд»

Я выделила основные цели на эту неделю:

* Сформировать команду и выбрать канал коммуникации.
* Провести мозговой штурм по идеям проекта.
* Изучить ключевые источники и выбрать инструмент управления.

**05.02.2025**

Что сделано:

* Создана команда и общий чат в Telegram для быстрой связи.
* Мы обсуждали три концепции — платформу обмена учебными материалами, платформу совместной работы с материалами и сервис курсов программирования с отзывами и карьерной поддержкой.

Я прочитала статью «ТОП бизнес-идей, которых ещё нет в России 2024» (Е. Звягин), оказалось очень полезным углубиться обзор свежих трендов на рынке стартапов и вдохновиться на поиск уникальной ниши.

**Что изучено**: Описание перспективных бизнес-идей, критерии оценки новизны и ценности для пользователя.

**Вывод:** Хорошая идея – это проект-агрегатор, объединяющий несколько сервисов, имеет больше шансов предложить уникальный опыт и привлечь аудиторию.

**Вопрос:** Какие комбинации сервисов создают истинную ценность для пользователя?

**Ответ:** Интеграция маршрутов, билетов и «досуга» в едином интерфейсе сокращает время на планирование и минимизирует переключение между сайтами — это то, что мы будем брать за основу.

**07.02.2025**

* Провели первый командный созвон и утвердили тему: «Сайт для планирования путешествий».
* Проанализировала аналогичные сервисы: Aviasales, Яндекс.Путешествия, Островок — отметила сильные и слабые стороны их UX/UI.

**Возник вопрос:** Как организовать процесс разработки и тестирования так, чтобы все понимали свой вклад?

**Ответ:** На основании статьи Полины Воловик «Что такое Trello и как им пользоваться» выбрали Trello: карточки в списках отражают статусы задач (Backlog → In Progress → Review → Done), что упрощает планирование и отслеживание тест-кейсов

Как итог, мы выбрали Trello для трекинга задач и тестовых сценариев.

Подметила на лекции: фформулировать открытые вопросы и сразу находить на них ответы позволяет двигаться быстро и скоординировано.

## **Вторая неделя (10.02.2025 – 16.02.2025)**

\*подметила, что с целями на неделю лучше отслеживать результат, так что решила их прописывать на каждую неделю.

**Цели на неделю:**

* Провести очную встречу команды и детализировать идею проекта.
* Чётко распределить роли и подготовить устав проекта.
* Исследовать конкурентов, набросать первые артефакты (эскизы, требования, вопросы для интервью).

**10.02.2025**

Очная встреча команды, расширение идеи проекта, распределение тем для изучения. «Чёткое начало позволяет выстроить эффективный процесс и избежать дублирования усилий»

**11.02.2025**

Первый семинар с преподавателем. Получены ключевые рекомендации:

* Явное распределение ролей.
* Активное использование Trello и ведение проектного дневника.
* Проведение интервью с целевой аудиторией (студенты туристических направлений). Зачем? Интервью позволит выявить реальные pain-points и сформировать проверенные требования.

**12.02.2025**

Начала работу над уставом проекта, в этом мне помогла изученная статья «Как написать устав проекта» (Юлия Мартинс)

**Вывод:** Устав — базовый документ, фиксирующий рамки проекта и необходимые ресурсы. Без него сложно выстроить тест-план и согласовать приоритеты

**Вопрос:** Почему тестировщику важно участвовать в создании устава?  
**Ответ:** Участие в уставе помогает тестировщику рано увидеть scope (объём) проекта, определить риски и заложить критерии качества, которые лягут в основу тест-плана.

**13–15.02.2025**

Основной уклон был на анализ конкурентов и составление сравнительной таблицы.

**Инсайты:**

* Роль тестировщика выходит за рамки поиска багов: нужно участвовать в формировании требований и артефактов.
* Глубокий анализ конкурентов помогает понимать «точки боли» и сразу закладывать валидные сценарии тестирования.
* Первичные эскизы и вопросы к ЦА формируют основу для будущего CJM и Use Case.

**Вопрос:** Какие данные из интервью критичны для написания тест-кейсов?  
**Ответ:** Понимание типовых шагов пользователей (например, подбор билета → добавление жилья → сохранение маршрута) позволит строить полные end-to-end сценарии и

## **Третья неделя (17.02.2025 – 23.02.2025)**

**Цели недели:**

* Проработка ключевых артефактов
* Изучение моделей жизненного цикла разработки и выбор оптимальной.
* Защита первого чекпоинта

**18.02.2025**

Участие в первом чекпоинте, но, к сожалению, не смогли защитить. Выделили ключевые моменты, которые нужно улучшить: налаживать регулярную работу с целевой аудиторией, вести подробный дневник проекта, доработать артефакты Use Case и CJM.

**Решение:** мы фокусируемся на наиболее важных этапах и используем лучшие практики работы с требованиями и пользовательскими сценариями.

**Вывод от нашего первого провала:**  полученные рекомендации четко задали вектор: приоритезировать обратную связь от реальных пользователей и формализовать все шаги из их пути в системе.

**20.02.2025**

**Изучены методические статьи**:

* «Customer journey map» (Загребина И., Зуйкова А.): CJM позволяет визуализировать все точки взаимодействия пользователя с системой и выявить «узкие места» на его пути.
* «Как написать полезный use case» (Нижельская М., Патрушева А.): Use Case помогает структурировать сценарии и описать все альтернативные потоки взаимодействия.

Рассмотрены Waterfall, V-Model, RUP и итерационно-инкрементальная модель. Как итог приняли итерационно-инкрементальную модель как наиболее гибкую и удобную для приоритизации требований и быстрых релизов.

## **Четвёртая неделя (24.02.2025 – 02.03.2025)**

Как всегда ставлю цели на неделю:

* Защитить ранее созданные артефакты (Use Case, CJM, устав).
* Доработать документы по замечаниям преподавателя.
* Оптимизировать процесс подготовки и согласования артефактов.

**25.02.2025**

Всё ещё пытаемся защитить артефакты проекта: Use Case и CJM. – точка, в которой мы очень затормозили.

**Уроки и выводы:**

* Необходимо заранее готовить подробную презентацию артефактов и проводить мини-ревью внутри команды.
* Следует вводить промежуточные встречи с преподавателем для ускорения правок.

## **Пятая неделя (03.03.2025 – 09.03.2025)**

Начинаю эту неделю с целей:

* ЗАЩИТИТЬ ЧЕКПОИНТ 1 и согласовать логику артефактов.
* Начать углублённое изучение методик тестирования веб-приложений по рекомендованному курсу.

**04–05.03.2025**

**УРА-УРА!!** Защитили первый чекпоинт, внесли правки в командную работу: утвердить хронологию разработки и единую версию документов.

**Новая активность:**

* Пообщалась с тестировщиками из других команд — порекомендовали курс **«Тестирование ПО — основы профессии тестирования веб-приложений»** на Stepik.[[1]](#footnote-1)
* Начала прохождение 1 модулей, очень понравился баланс между качественной и понятной теорией и практическими заданиями, которые помогают быстрее запоминать изученный материал.

**Вопрос:** Как интегрировать знания из курса в текущий тест-план?

**Ответ:** Запланировала расширить раздел «Нефункциональные требования» в тест-плане и добавить чек-листы по безопасности и производительности на основании модулей курса.

**07.03.2025**

Произошёл ккомандный созвон. Приняты решения:

1. Согласовать хронологию разработки.
2. Доработать функциональные требования.
3. Переработать Use Case с примерами бронирования билетов и составления маршрутов.
4. Подготовить индивидуальные версии видения проекта.

### **Неделя 6 (31.03.2025 – 07.04.2025)**

**Цели недели**

* Старт курса Stepik «Тестирование ПО - основы профессии тестирования веб-приложений» и закрепление теории из книжки Куликова.
* Оформление документации: чек-листы, баг-репорты, Test Plan.
* Начало автоматизации тестов на pytest.

**01–03.04.2025**

Начинаю подготовку ко 2 чекпоинту. Решили углубиться больше в теоретическую часть для составления грамотного плана разработки на ближайшие 45 дней и для плана тестирования.

Изучила до конца **модули 1.1–1.5** курса (понятия, STLC, роли тестировщика, виды тестирования, методологии).

**Вывод:** Как и в Куликове, поняла, что «тестировщик — это не просто «ловец дефектов», а участник всего цикла разработки» (ср. «Тест-кейс… если у тест-кейса не указаны входные данные… — это плохой тест-кейс» ).

**04–05.04.2025**

Выполнила **модули 2.1–2.4** курса. Разобралась в различиях между функциональным и нефункциональным тестированием, статическом/динамическом тестировании и принципах тестирования.

**Вывод:** невозможно протестировать всё, поэтому важно приоритизировать критичные флоу и регулярно обновлять тест-кейсы (принцип «пестицида»).

**Книга Куликова** подчёркивает роль чек-листов:

«Чек-лист… это просто набор идей: идей по тестированию, идей по разработке… Главное, чтобы они помогали в работе» .

Итог: составила первый чек-лист для сценариев регистрации: многоуровневый список идей по позитивным и негативным случаям.

**06–07.04.2025**

Перешла к изучению модуля 2.7: Пирамида тестирования. Для нашего проекта применила таким образом:

* Юнит-тесты (pytest) для логики API — база пирамиды.
* Интеграционные тесты для проверки стыков модулей.
* E2E-тесты на UI — минимальное покрытие для ключевых сценариев.

**Вывод:** грамотное сочетание уровней ускоряет CI-цикл и снижает стоимость поддержки тестов.

### **Неделя 7 (08.04.2025 – 13.04.2025)**

**Цели недели**

* Запуск backend на FastAPI и проверка базовых эндпоинтов.
* Писать юнит- и интеграционные тесты с учётом рекомендаций Куликова и курса.
* Начать формализованный анализ рисков.

Продолжается подготовка ко 2 чекпоинту. Провела анализ рисков проекта по методике Бриткина А.И.:[[2]](#footnote-2)

* Автор рассматривает метрики и показатели, позволяющие адекватно оценить и контролировать рискованные участки проекта: процент оставшейся работы, связанной с рисками; возраст нерешённых проблем; распределение рисков по времени, типу, сложности и срочности; распределение проблем по командам и процессам. Автор приводит визуальные примеры (графики), как такие метрики могут указывать на настоящую «зону проблем» в проекте и делает акцент на том, что вовремя выявленные риски и проблемы — залог устойчивости проекта и более точного планирования.
* Добавила эти метрики в дашборд Trello — теперь «горячие» зоны проекта видны сразу.

**Вывод:** регулярный мониторинг рисков помогает вовремя перераспределять ресурсы и корректировать приоритеты тестирования.

Параллельно начинается реализация части backend. Изучаю статью «Возможности веб-феймворка fastapi для реализации серверной части веб-системы», Богачёв.[[3]](#footnote-3)

В данной публикации авторы рассматривают веб-фреймворк FastAPI как современное решение для разработки серверной части веб-систем. В статье проводится сравнительный анализ FastAPI с другими популярными фреймворками на Python — Django и Flask. Отдельное внимание в работе уделено практическим аспектам применения FastAPI, в частности, при построении веб-системы терминалов. Авторы демонстрируют, как использование встроенных инструментов FastAPI позволяет решать прикладные задачи (например, генерация API-документации и разграничение прав доступа) более эффективно и с меньшими издержками по сравнению с Django или Flask. Для нашего проекта полученные из статьи сведения о FastAPI оказались крайне полезными при выборе технологий для реализации серверной части платформы. Одним из ключевых аспектов, которые мы переняли из работы, стала асинхронная обработка запросов — критически важная возможность для нашего проекта, ведь при построении маршрутов необходимо параллельно обращаться к множеству внешних API (авиабилеты, отели, экскурсии и прочее). Кроме того, авторы подробно описали встроенную систему автоматической генерации документации Swagger и Redoc, что позволило нам быстро настроить понятную и наглядную документацию для REST API — это упростило внутреннее тестирование и улучшило взаимодействие между backend и frontend. Полученные из статьи знания легли в основу ключевых архитектурных решений и позволили создать надёжную и масштабируемую серверную часть.

Изучила статью Куликова по тест-кейсам:

«Test case — набор входных данных, условий выполнения и ожидаемых результатов… Если у тест-кейса не указаны эти элементы — это плохой тест-кейс» Software Testing - Base…Software Testing - Base….

### **Неделя 8 (14.04.2025 – 20.04.2025)**

**Цели недели**

* Применить знания из курса Stepik по модулям по документации и веб-тестированию.
* Разработать и уточнить тест-кейсы, чек-листы и шаблоны баг-репортов.

На этой неделе мы защищали первый раз 2 чекпоинт всей командой. Основная трудность, с которой мы сталкиваемся по мере реализации данного проекта – это военная кафедра по вторникам у Никиты. Из-за этого приходится искать компромиссы с преподавателей и договариваться на другие дни для защиты/доработки материалов. Для первого раза защита прошла хорошо, но нужно доработать план разработки и план тестирования (важно указывать распределение по ролям среди участников).

Для доработки плана разработки приходит на помощь курс и статьи! Изучаю статью «Функциональное тестирование веб-сайтов», Богданова В.С.[[4]](#footnote-4)

В статье рассмотрены основные принципы функционального тестирования веб-сайтов. Автор подробно описывает этапы данного процесса: от анализа функциональности до составления отчета о тестировании. Особое внимание уделено видам функционального тестирования (модульное, интеграционное, системное, регрессионное, sanity- и smoke-тестирование), а также инструментам, применяемым при его реализации, как ручным (TestIT, TestRail, Allure), так и автоматизированным (Selenium, Appium, Katalon Studio, Ranorex Studio). Автор делает акцент на важности соответствия программного продукта заявленным требованиям и обеспечении его надежности до передачи пользователю.

Для нашего проекта изучение данной статьи полезно для формирования практических навыков тестирования веб-приложений. Описание этапов тестирования и обзор инструментов позволяют грамотно выстраивать процессы тестирования в рамках командной разработки. Полученные знания могут быть непосредственно применены при работе над проектом, особенно в частях, касающихся проверки пользовательской логики, интерфейсов и устойчивости сайта.

**Что сделано:**

1. Изучила и применила **модули 3.1–3.4 курса**:
   * Создание Test Plan, Test Case, Bug Report, Check-lists.
   * Пересмотрела правила формулировки шагов, предусмотреть предусловия и постусловия каждого теста.
2. Применяла **модули 5.1–5.3** (основы веб-приложений, HTTP-запросы, архитектура клиент-сервер):
   * Проверяла тест-кейсы не только в UI, но и с помощью запросов через Postman, сверяя поведение API с документацией Swagger.
3. Разработала и согласовала с архитектором:
   * Шаблон чек-листа для ключевых сценариев (регистрация, поиск маршрутов, генерация чек-листа).
   * Единый формат баг-репорта с полями «Описание», «Шаги воспроизведения», «Ожидаемый результат», «Фактический результат», «Приоритет», «Скриншоты».

**Зачем изучено:**

* **Модули 3** дали чёткую структуру тест-документации, благодаря чему тест-кейсы стали однозначны для всех участников.
* **Модули 5** позволили глубже понять, как фронтенд взаимодействует с бекендом, и строить тест-кейсы, проверяющие не только кликабельность кнопок, но и корректность HTTP-статусов и тел запросов.

**Выводы:**

* Формализованная документация снижает риск недопонимания между тестировщиком и разработчиком и ускоряет воспроизведение бага.
* Совмещение UI-тестов и API-проверок повышает надёжность: критичные ошибки на уровне HTTP-слоя можно обнаружить раньше, чем они проявятся в интерфейсе.

### **Неделя 9 (21.04.2025 – 27.04.2025)**

**Цели недели:**

* ЗАЩИТА 2 ЧЕКПОИНТА

Хорошая новость: удалось закрыть 2 чекпоинт!!

Происходит подготовка к интеграционному тестированию, решила изучить статью:

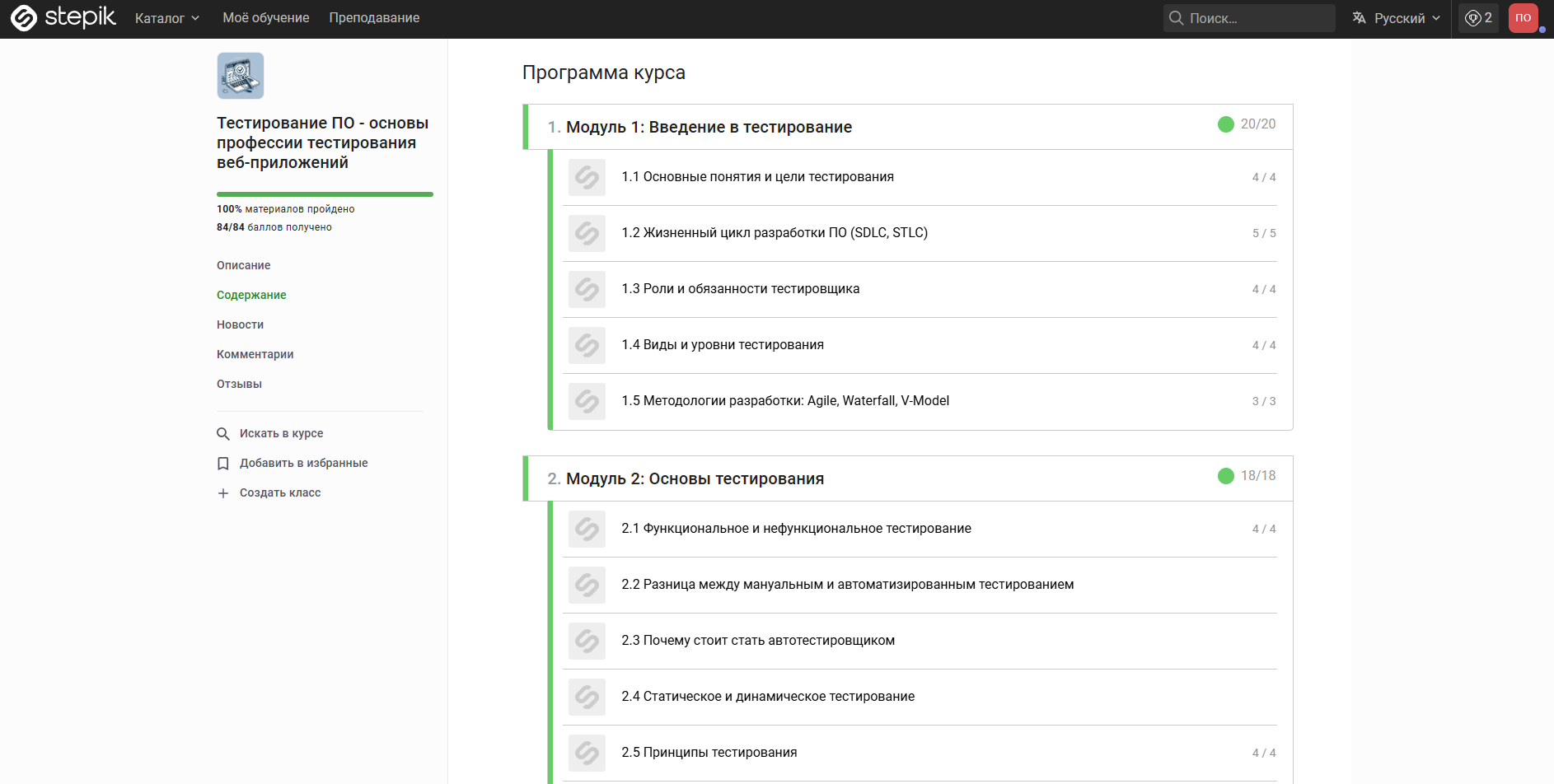
«Взаимодействие с БД при интеграционном тестировании приложений», Дронова М.В..[[5]](#footnote-5) Для нашего проекта данная статья оказалась особенно полезной при построении инфраструктуры тестирования. Мы столкнулись с необходимостью тестировать модули, работающие с различными источниками данных: расписания авиарейсов, отели, экскурсии и т.д. Основываясь на методике из статьи, мы вместе с Никитой настроила отдельный экземпляр MySQL для автоматических и ручных тестов:

* + Спроектировали структуру таблиц для каталога маршрутов и хранения логов запросов.
  + Автоматически инициализируем БД SQL-скриптом перед каждым запуском тестов.

**Вывод:** Раннее подключение тестовой среды важно для выявления архитектурных проблем до появления большого числа зависимостей.

**Конец прохождения моего курса:**

Пройдено 100 % материалов, набрано 84 из 84 баллов.



**Выводы по прохождению данного курса:**

1. Умение выбирать правильный уровень (юнит, интеграция, E2E) ускоряет обратную связь и снижает стоимость поддержки тестов.
2. Стандартизированные шаблоны тест-планов и баг-репортов сделали процесс прозрачным и позволили сократить время на уточнение деталей при оформлении дефектов.
3. Применение техник эквивалентного разбиения и граничных значений помогло выстроить компактный, но исчерпывающий набор тестов для фильтров и форм.
4. Теперь я чётко вижу стадии дефекта и умею управлять ими — от «Новый» до «Закрыт», что повысило скорость реакции разработчиков.
5. Тестирование превратилось из «штамповки» багов в проактивный анализ рисков и улучшений.
6. Этот курс помог мне связать теорию и практику, выстроить системный подход к QA и гладко вписаться в командный процесс разработки.

**Неделя 10-11 (28.04.2025 – 11.05.2025)**

**Цели недели:**.

1. Выполнить нефункциональные проверки совместно с архитектором.
2. Начать подготовку финального отчёта по результатам тестирования.

Обсуждали с командой результаты тестирования, занимаемся доработкой. Работа «плечом к плечу» с Никитой (архитектором) и Захаром (тим-лидом) ускорила принятие решений по оптимизации API-слоя и обеспечила глубокий охват нефункциональных требований.

**10.05.2025**

На общем созвоне обсудили пройденный путь, насколько процентов уже реализован и выполнен в срок план разработки. Подвели итог и выделили каждый по 10 задач. Помогала это всё документировать и оформлять в общий анализ. Очень понравился момент, когда каждый из нас по очереди давал свои комментарии по выполнению поставленных задач других участников. Так мы смогли выделить сильные и слабые стороны каждого из команды и подметить зоны роста.

Активно готовимся к защите 3 чекпоинта. Оформляю документацию по тестированию. Очень помогает практическая часть из курса (модуль 3-4) правильно оформлять отчёты.

Очередной раз убедилась в правильности выбора курса!

**Отчёт по книге «Тестирование программного обеспечения. Базовый курс» — Святослав Куликов.**

Особенно ценными для меня оказались следующие разделы:

• Раздел 2.1.2 «Жизненный цикл тестирования» — демонстрирует итеративную модель STLC, где тестирование планируется, проектируется, выполняется и анализируется в замкнутом цикле, что помогает своевременно возвращаться к критическим этапам.

• Раздел 2.4.1 «Чек-листы» — объясняет, что чек-лист — это не формальный документ, а набор идей (тест-кейсов) в виде структурированного списка, позволяющий фиксировать любые мысли и быстро возвращаться к ним при тестировании.

• Подраздел «Жизненный цикл тест-кейса» (рисунок 2.4.a) — описывает ключевые состояния тест-кейса (создан, запланирован, выполняется, пройден, провален и др.), подчёркивая важность контроля статусов на каждом шаге Software Testing - Base….  
• Раздел 2.5 «Отчёты о дефектах» — подчёркивает два золотых правила: один отчёт = один дефект и для каждого нового дефекта создаётся отдельный отчёт, что предотвращает путаницу и улучшает прослеживаемость Software Testing - Base…Software Testing - Base….

• Рисунок 2.3.g и комментарии к нему — показывает разницу между дымовым (smoke), критическим и регрессионным тестированием, а также отличие sanity-тестов как быстрых проверок стабильности билда Software Testing - Base….

К концу мая я прочитала около 85 % этого руководства. Главные открытия:

* Понимание STLC позволяет вовремя включаться в проект, корректировать планы и не пропускать важные проверки.
* Чек-лист как «хранилище идей» освободил меня от беспорядочного вороха мыслей: теперь могу моментально зафиксировать идею теста и впоследствии развернуть её в полноценный тест-кейс.
* Управление состояниями тест-кейсов дало прозрачность выполнения: я всегда знаю, какие проверки готовы к запуску, а какие требуют доработки.
* Жёсткие правила дефект-репортов гарантировали, что ни один баг не потерялся и каждая правка фиксировалась новым отчётом.
* Разграничение smoke, sanity и regression помогает балансировать скорость проверок и глубину тестирования, сохраняя качество при частых релизах.

**Ключевая мысль из книги:**  
Настоящий тестировщик не ограничивается поиском ошибок — он проникает в логику продукта с позиций конечного пользователя и выявляет потенциальные слабые места ещё на этапе проектирования. Такой проактивный подход отделяет эксперта от рядового «кликера».

**Что дала мне эта книга:**  
Руководство Куликова помогло синхронизировать теорию и практику: я научилась системно оформлять свои наблюдения, видеть продукт в целом и говорить с разработчиками на одном профессиональном языке.

Работа над проектом стала для меня цельным путешествием — от первых идей до финального релиза — в котором я полностью реализовала роль тестировщика как голоса пользователя и гаранта качества.

**Спасибо за внимание!**

1. https://stepik.org/course/233978/syllabus [↑](#footnote-ref-1)
2. БРИТКИН А.И. АНАЛИЗ И КОНТРОЛЬ РИСКОВ В ПРОЕКТАХ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ - 2008 - № 1 (8) - С. 18–22 <https://www.elibrary.ru/item.asp?id=18113008> [↑](#footnote-ref-2)
3. БОГАЧЁВ Р.Е., ЗАРИКОВСКАЯ Н.В., ВОЗМОЖНОСТИ ВЕБ-ФРЕЙМВОРКА FASTAPI ДЛЯ РЕАЛИЗАЦИИ СЕРВЕРНОЙ ЧАСТИ ВЕБ-СИСТЕМ – 2021 – Всероссийская научная конференция «Достижения науки и технологий-ДНиТ-2021» – № 2 – С. 289–292 <https://www.elibrary.ru/item.asp?id=47383366> [↑](#footnote-ref-3)
4. БОГДАНОВА В.С. ФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ ВЕБ-САЙТОВ // ТЕХНОЛОГИЧЕСКИЕ ИННОВАЦИИ И НАУЧНЫЕ ОТКРЫТИЯ: Сборник научных статей по материалам XVI Международной научно-практической конференции. - 2024. – С. 341–344. <https://www.elibrary.ru/item.asp?id=75096556> [↑](#footnote-ref-4)
5. ДРОНОВА М.В., ВЗАИМОДЕЙСТВИЕ С БАЗОЙ ДАННЫХ ПРИ ИНТЕГРАЦИОННОМ ТЕСТИРОВАНИИ ПРИЛОЖЕНИЙ // Синергия наук. – 2021. – № 62. – С. 45–50. <https://www.elibrary.ru/item.asp?id=38565315> [↑](#footnote-ref-5)