## ****1. Анализ требований и планирование (Неделя 1)****

### ****Действия:****

* Провести анализ требований (если документация отсутствует — составить чек-лист на основе функционала).
* Выявление основных функций продукта:
  + Просмотр каталога товаров (поиск, сортировка);
  + Добавление товаров в корзину (добавление/удаление);
  + Оформление заказа (заполнение форм);
  + Выбор способа доставки (заполнение форм);
  + Оплата через сторонний сервис (интеграция API).
* Определить ключевые риски:
  + Нестабильность платежного сервиса;
  + Проблемы с отображением товаров в разных браузерах;
  + Ошибки при оформлении заказа (потеря данных);
  + Безопасность данных пользователей (никому не надо, чтобы их имена и/или адреса утекли в сеть)
* Составить стратегию тестирования (подходы, уровни тестирования, инструменты).
* Оценить трудоемкость:
  + 30% - функциональное тестирование;
  + 25% - тестирование производительности:
    - 5% - нагрузочное тестирование (тестирование при заданной нагрузке);
    - 10% - стресс тестирование (что произойдет при незапланированной нагрузке, важна планируемая деградация);
    - 10% - тестирование производительности (как быстро работают функции);
  + 10% - Конфигурационное тестирование (проверка совместимости на различном железе/ОС/Браузерах)
  + 10% - Юзабилити тестирование (для анализа удобства использования сайта)
  + 10% - Тестирование локализации (вероятно наш сайт будет работать не только на одном языке)
  + 15% Тестирование безопасности.
* Создать возможные тестовые кейсы и проанализировать их трудоъемкость.

**Результат:** Документ со стратегией тестирования, чек-лист ключевых сценариев, тест-кейсы.

## ****2. Тестирование альфа-версии (Недели 2–3)****

**Цель:** Проверить базовую функциональность, формирование баг-репортов.

### ****Действия:****

### ****Модульное тестирование:****

* + Просмотр каталога (поиск, сортировка);
  + Работа корзины (добавление/удаление товаров, обновление суммы);
  + Оформление заказа (валидация полей, переход к оплате).
* **Интеграционное тестирование:**
  + Проверка API (например, между фронтендом и бэкендом корзины) при помощи Postman;
  + Тестирование платежного шлюза;
  + Кросс-браузерность (Chrome, Firefox, Safari) через BrowserStack;
  + Адаптивность (мобильные устройства);
  + Тестирование локализации;
  + Юзабилити тестирование.
* **Регрессия:** Быстрая проверка критичных багов после фиксов.

**Результат:** Проверена базовая функциональность сайта и собраны баг-репорты.

## ****3. Углубленное тестирование (Недели 4–8)****

**Цель:** Полноценное покрытие функционала перед бета-тестированием.

### ****Действия:****

### Составление тест-кейсов (пример: "Оплата при недостаточном балансе карты", "Попытка заказа отсутствующего сейчас товара").

* Применение техник: классы эквивалентности, граничные значения (например, цена товара = 0).
* **Автоматизация:**
  + Пилотные автотесты на API (например, запросы к корзине);
  + Фаззинговое тестирование, если применимо (случано сгенерированные данные например в поисковую строку).
* **Тестирование производительности:**
  + Проверка реакции сервера например 100+ одновременных заказов, важно гарантировать запас перед реальным сценарием, но и не перестараться;
  + Проверка скорости работы функций (проверка и грубо говоря установление эталона);
  + Проверка стабильности работы программы в течении продолжительного времени (учечки памяти, накопление ошибок округления и т.п.)
* **Тестирование безопасности:**
  + Проверка на SQL-инъекции в форме заказа.
  + Уязвимости сессий .

**Результат:** Полный набор тест-кейсов, отчет о произоводительности.

## ****4. Релизная подготовка (Недели 9–12)****

**Цель:** Обеспечить готовность к релизу.

### ****Действия:****

* **Регрессионное тестирование:** Прогон всех ключевых сценариев.
* **Запланированное тестирование:** например каждый день частичное тестирование основного функционала, а раз в неделю ночью полное тестирование функционала.
* **Тестирование документации:** Соответствие README и пользовательских подсказок.
* **Мониторинг после релиза:**
  + Логи ошибок;
  + Метрики: % успешных оплат, среднее время загрузки страницы;

## ****5. Поддержка (Месяц после релиза)****

**Цель:** Быстрое реагирование на инциденты.

### ****Действия:****

* Приоритизация багов от пользователей;
* Анализ причин сбоев (например, падение платежного API);
* Обновление тест-кейсов под реальные сценарии;

**Метрики:**

* Количество критичных багов.
* Время реакции на инциденты.