

Лекция 6.

**Техники эффективного
тестирования**

Что делает тест-кейс «хорошим»?

✓ Хороший тест-кейс содержит:

- Чёткую цель
- Ясные входные данные
- Условия выполнения
- Однозначные ожидаемые результаты

✗ Плохой тест-кейс — это:

- Непонятная цель
- Отсутствие данных или ожиданий
- Формальное выполнение без понимания сути

Цель — во главе угла. Без неё невозможно создать осмысленную проверку.

Качество = ценность для пользователя

- Качество — это соответствие реальным потребностям и ожиданиям пользователя.
- Пользователь платит **временем, усилиями, доверием** — даже если не платит деньгами.
- Тестирование должно выявлять **нарушения ценности**, а не просто «ошибки по документу».

! Продукт может быть «идеален по метрикам», но бесполезен на практике.

Фокус на важном

- В любом ПО есть:
 - Критически важные функции/свойства
 - Важные, но второстепенные
 - Незначительные детали

 **Задача тестировщика:**

Найти максимум важных ошибок за ограниченное время.

Важные ошибки = те, что ломают то, что пользователь считает ценным
(функциональность, безопасность, производительность, удобство...)

Алгоритм создания эффективных проверок

Перед написанием чек-листа или тест-кейса задайте себе:

1. **Что передо мной?** → Понимание объекта тестирования
2. **Кому и зачем это нужно?** → Определение ценности
3. **Как обычно используется?** → Позитивные сценарии
4. **Как может сломаться?** → Негативные сценарии

Ответы на эти вопросы — основа осмысленного тестирования.

Практические рекомендации

- 📝 Пишите чек-листы — память ненадёжна
- 🔍 Начинайте рано — ещё на этапе требований
- 🌱 Декомпозируйте сложные системы на модули
- ❓ Фиксируйте вопросы прямо в документах
- 📄 Используйте форматирование — но умеренно
- 👁️ Проверяйте документы коллегами (техника беглого просмотра)
- ⌚ Планируйте время на доработку тест-кейсов

Как улучшить тест-кейсы

- 🎯 Цель должна формулироваться за 10 секунд
- 🚫 Избегайте дублирования — используйте техники:
 - Классы эквивалентности
 - Граничные значения
 - Доменное тестирование
- 💡 Повышайте показательность без усложнения
- ⚙️ Описывайте подготовку (предусловия!)
- + Объединяйте позитивные проверки, не объединяйте негативные
- ♻️ Оптимизируйте — снижайте трудозатраты на выполнение
- ✅ Перечитывайте финальную версию — найдёте ошибки!

Техника классов эквивалентности (Equivalence Partitioning)

Суть:

Разделение входных данных на группы (классы), где поведение системы **одинаково**.
Достаточно протестировать **по одному значению из каждого класса**.

Преимущество:

Снижает количество тест-кейсов без потери покрытия.

Примеры:

1. Поле «Возраст» (18–65 лет):

- Валидный класс: 25, 40, 60
- Неважные классы: <18 (напр., 15), >65 (напр., 70)

2. Логин (5–20 символов):

- Валидный класс: "user123" (8 симв.)
- Неважные: "ab" (2 симв.), "verylongusername12345" (22 симв.)

3. Выбор категории товара («Электроника», «Книги», «Одежда»):

- Тестируем по одному значению из каждой категории — поведение системы должно быть одинаковым внутри класса.

Техника граничных значений (Boundary Value Analysis)

Суть:

Ошибки чаще всего возникают **на границах** допустимых диапазонов. Тестируем значения **на границе и рядом с ней**.

Типичные значения:

$\text{min}-1$, min , $\text{min}+1$, $\text{max}-1$, max , $\text{max}+1$

Примеры:

1. Пароль длиной 8–16 символов:

- Тестируем: 7 (ошибка), 8 (успех), 9 (успех), 15 (успех), 16 (успех), 17 (ошибка)

2. Заказ товара (1–100 шт.):

- Проверяем: 0 (ошибка), 1 (успех), 2 (успех), 99 (успех), 100 (успех), 101 (ошибка)

3. Оценка по 5-балльной шкале (1–5):

- Граничные тесты: 0 → ошибка, 1 → ОК, 5 → ОК, 6 → ошибка



Часто комбинируется с классами эквивалентности!

Доменное тестирование (Domain Testing)

Определение:

Доменное тестирование — это проектирование тестов на основе **области допустимых значений (домена)** входных данных с учётом:

- бизнес-логики,
- пользовательских сценариев,
- контекста использования,
- ограничений предметной области.

Чем отличается от классов эквивалентности и граничных значений?

Аспект	Классы/Границы	Доменное тестирование 
Фокус	Формальные диапазоны	Смысловые зоны и контекст
Подход	Математический / структурный	Прагматический / поведенческий
Цель	Проверить корректность обработки значений	Проверить, работает ли система так, как нужно пользователю в реальности

Главная идея:

Не все значения внутри формального диапазона одинаково важны.

Нужно тестировать те, что реально используются — и учитывать их смысл.

Примеры доменного тестирования

Пример 1: Онлайн-кинотеатр

- Домены времени:
 - Утро (6–12) → семейный контент
 - Вечер (18–24) → премьеры, блокбастеры
- Тест: При выборе фильма в 20:00 система должна предлагать новинки, а не детские мультфильмы.

Пример 2: Банковское приложение — переводы

- Домены сумм:
 - До 10 000 ₺ → без подтверждения
 - 10 001–100 000 ₺ → SMS-код
 - Свыше 100 000 ₺ → звонок в банк
- Тест: Перевод 10 000 ₺ — проходит мгновенно; 10 001 ₺ — требует код.

Пример 3: Электронная медицинская карта

- **Домены пациентов:**
 - Дети (0–17 лет) → нельзя назначать некоторые лекарства
 - Взрослые (18–65) → стандартные протоколы
 - Пожилые (>65) → учитываются хронические болезни
- **Тест:** При попытке назначить «Лекарство X» ребёнку система выдаёт предупреждение.


Пример 4: Доставка еды

- **Географические домены:**
 - Центр города → доставка за 30 мин
 - Окраина → 60 мин
 - За МКАД → недоступна
- **Тест:** При вводе адреса «улица Ленина, 100» (за МКАД) — кнопка «Заказать» неактивна.

Вывод:

Доменное тестирование помогает находить ошибки, которые не видны при формальном подходе, но критичны для пользователя.

Сравнение ключевых техник проектирования тестов

Техника	Цель	Когда применяется	Пример 
Классы эквивалентности	Сократить количество тестов, группируя входные данные с одинаковым поведением	При наличии чётких диапазонов или категорий	Возраст: [18–65] → 1 тест из диапазона; <18 и >65 → по одному невалидному
Граничные значения	Выявить ошибки на границах допустимых значений	Всегда вместе с классами эквивалентности	Пароль длиной 8–16 симв.: тестируем 7, 8, 16, 17
Доменное тестирование	Проверить поведение системы в реальных, смысловых контекстах	Когда данные связаны с бизнес-логикой или пользователями	Регион → валюта; выходные дни → повышенная цена

 **Совет:**

Используйте все три техники вместе:

- сначала выделите домены,
- внутри — классы эквивалентности,
- на границах — граничные значения.