

Рассмотрим темы:

- ✓ Методы исследования продукта
- ✓ Оформление тестов (тест-кейсы, чек листы)
- ✓ Техники тест дизайна
 - Классы эквивалентности
 - Граничные значения
 - Таблица решений
 - Диаграмма состояний
 - Сценарии использования
 - Попарное тестирование

Исследование продукта

Мы устроились на работу, и нам дали задачу протестировать уже готовый продукт.

Что делать?

- ✓ исследовать продукт - что он умеет;
- ✓ правильно задавать вопросы;
- ✓ использовать инструменты для записи результатов.

- ✓ Спецификация есть – изучаем ее
- ✓ Спецификации нет – задаем вопросы



Как правильно задавать вопросы

Вопросы бывают открытые и закрытые .

| Закрытые вопросы | Открытые вопросы |
|---|--|
| <p><i>Возможен только однозначный ответ:</i></p> <p>дата; время; название; количество; ДА или НЕТ? Например: Сколько лет вы работаете в тестировании? Вы проводили нагрузочное тестирование? Система должна обрабатывать TXT-файлы?</p> | <p><i>Нужен развернутый ответ</i></p> <p>Открытые вопросы обычно начинаются со слов:</p> <p>Что? Как? Почему? Каким образом? При каких условиях?</p> |

Фактически закрытый вопрос означает, что вы сами уже додумали себе требования и их же и будете тестировать. А тестировщик НИКОГДА не додумывает. Ведь иначе он будет тестировать вообще не то, что нужно.

Например, система принимает на входе файл с текстом и проверяет его на опечатки. Тестировщик додумал себе, что это простой блокнот, и уточняет у заказчика:

- Система должна обрабатывать TXT -файлы?
- Да, должна.
- Ок, спасибо.

Полностью уверенный в том, что узнал всё, что ему было нужно, тестировщик готовит свои тесты. А на самом деле заказчик ждет, что система будет уметь работать со всеми текстовыми форматами, включая Excel, а еще она сможет считывать текст с картинок! И именно в этом ее фишка. То есть главный функционал. А тестировщик картинки с текстом вообще не проверяет, так как считает это негативным тестированием.

- ✓ Всегда старайтесь задавать максимально открытый вопрос.
- ✓ Если видите, что ваш вопрос «закрытый», подумайте, что именно вы хотели узнать, -об этом и спросите.

Как оформить результат, чтобы не забыть всё, что мы узнали?

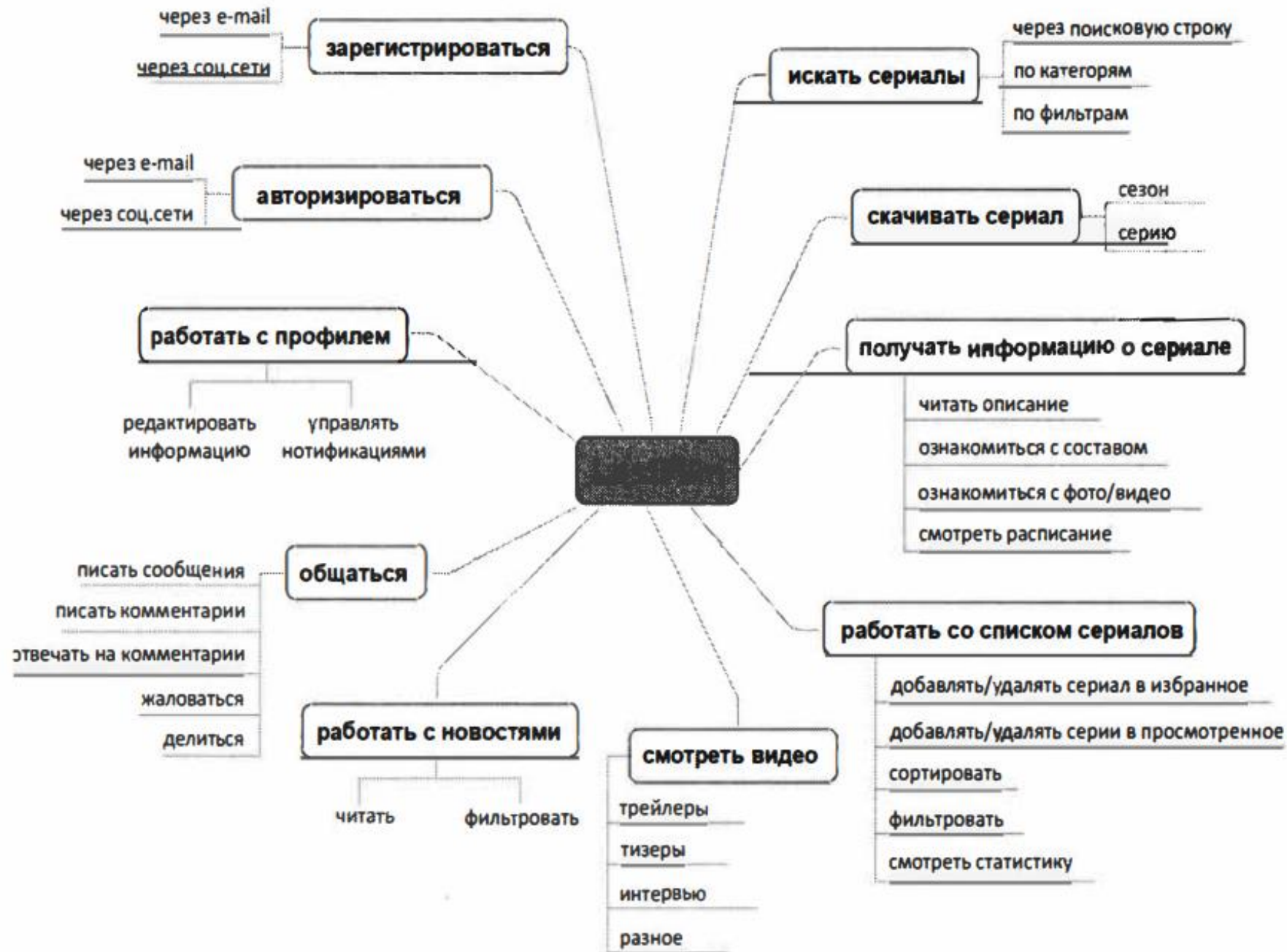
- ✓ Записать в блокнот
- ✓ Нарисовать карту приложения

Как нарисовать карту приложения?

Есть несколько основных правил, которые помогут сделать карту простой и понятной.

Выделите основные функции, задав себе вопросы:

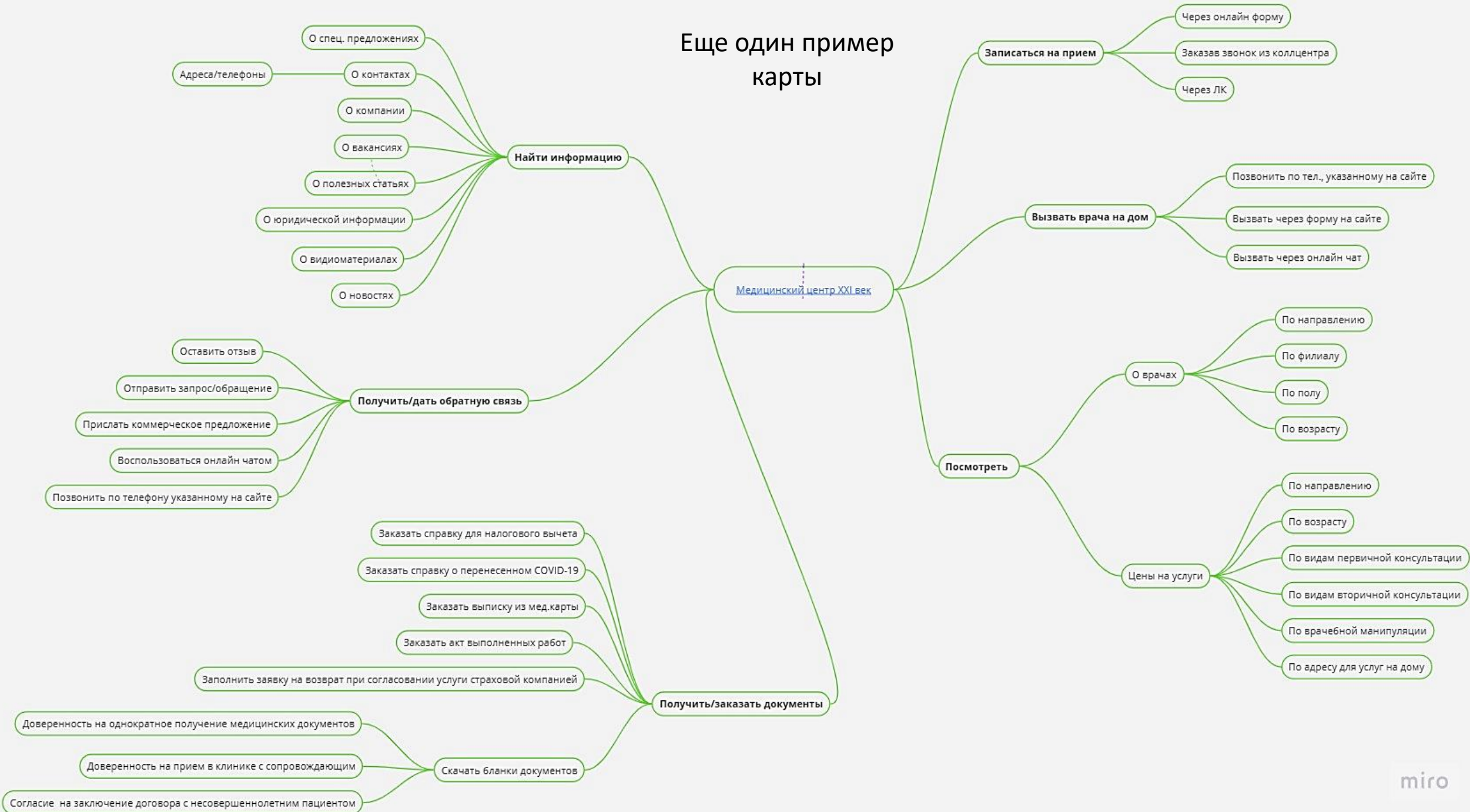
- ❖ *зачем пользователю наш продукт?*
- ❖ *что он там делает?*
- ❖ *что мы хотим, чтобы он делал?*



По важности

Регистрация
здесь не самая
важная функция,
поэтому она в
конце

Еще один пример карты



После изучения функционала продукта

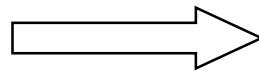
- проектируем тесты;
- оформляем тест-кейсы;
- пишем чек-листы.

Проектирование тестов

При продумывании тестов нужно помнить о том, каким должен быть тест и о приоритетах выполнения.

Каким должен быть тест?

- Тест должен быть конкретным
- У теста должен быть результат



Из описания теста должно быть понятно:

- что сделать;
- как сделать.

Приоритеты выполнения тестов

ВСЕГДА начинаем с позитивного тестирования и только потом проводим негативное. Потому что задача тестировщика - не найти как можно больше багов, а дать наиболее полную информацию по приложению

Как оформлять тесты?

1. Тест-кейс (ТК) - это подробное описание проверки.

ТК1. Авторизация через email

Предварительные шаги

Зарегистрироваться с email test@test.ru и паролем 1

Шаги

1. Открыть сайт <https://www.example.com/>.
2. Нажать на кнопку «Вход» в правом верхнем углу окна сайта.
3. В поле «Email» ввести значение: test@test.ru.
4. В поле «Пароль» ввести значение: 1.
5. Нажать кнопку «Войти».

Ожидаемый результат

- Открылась главная страница сайта.
- В верхнем правом углу страницы отображается значок личного кабинета (недоступен без авторизации)

Структура тест-кейса

1. шаги — это инструкция по вводу;
2. исполнение шагов — это ввод;
3. ожидаемый результат — это ожидаемый вывод;
4. фактический результат — это фактический вывод.

Исполнение тест-кейса завершается сравнением вывода фактического и вывода ожидаемого.

1. Положительный исход (PASS), если ФР равен ОР,
2. Отрицательный исход (FAIL), если ФР не равен ОР: найден баг!

Иногда возникает ситуация, когда мы заблокированы, так как не можем пройти ВСЕ шаги тест-кейса. Например, мы не можем продвинуться дальше, если кнопки "Завершить заказ" из шага 14 не существует на соответствующей веб-странице. В таком случае мы рапортуем баг (в данном случае баг об отсутствии кнопки "Завершить заказ") и откладываем исполнение тест-кейса до устранения бага.

Атрибуты тест-кейса

- **Уникальный идентификатор тест-кейса** — необходим для удобной организации хранения и навигации по нашим тест-наборам.
- **Приоритет** — важность тест-кейса, отражается по шкале от 1 до n, где 1 — это высший приоритет, а n — это низший приоритет.
- **Название** — основная тема, или идея тест-кейса. Кратное описание его сути.
- **Предусловия** — описание условий, которые не имеют прямого отношения к проверяемому функционалу, но должны быть выполнены. Например, «пользователь авторизован»
- **История редактирования** — кто создал/изменил? Когда? Зачем? Почему?
- **Шаги** — описание последовательности действий, которая должна привести нас к ожидаемому результату
- **Ожидаемый результат** — результат: что мы ожидаем увидеть после выполнения шагов.

| | | |
|---|-----------------|---|
| TC ID/Priority | CCPG0001 | 1 |
| IDEA: Оплата может быть произведена картой VISA SETUP and ADDITIONAL INFO: Эккаунт: testuser1/paSSwOrd Наименование товара: book117 Данные карты: Номер: 9999-5148-2222-1277 Окончание действия: 12/07 CVV2: 778 SQL1: select result from cc_transaction where id = <номер заказа>; | | |
| Revision History | | |
| Created on: 11/17/2003 by O.Тарасов | Новый тест-кейс | |

Пример оформления тест-кейса
из книги Романа Савина
«Тестирование Дот Ком»

| Execution part | |
|--|-----------------|
| PROCEDURE | EXPECTED RESULT |
| 1. Открой www.main.testshop.rs 2. Введи имя пользователя. 3. Введи пароль. 4. Нажми кнопку "Войти". 5. Введи наименование товара в поле поиска. 6. Нажми кнопку "Найти". 7. Кликни линк "Добавить в корзину". 8. Кликни линк "Корзина". 9. Кликни линк "Оплатить". 10. Выбери вид карты. 11. Введи номер карты. 12. Введи срок окончания действия. 13. Введи CVV2. 14. Нажми кнопку "Завершить заказ". 15. Запиши номер заказа | > "10" |
| 16. Запроси базу данных с SQL1 и запиши результат | |

Особенности тест-кейсов

- **В них максимум информации.** Их легко можно отдать новичку, и он проведет проверки, не задавая вопросов.
- **Они независимые.** Хорошие тест-кейсы не зависят друг от друга, поэтому можно легко редактировать один, не боясь, что развалятся все остальные.
- **Они небольшие.** Один тест-кейс – одна проверка.

Тест-кейсы нужны, когда у нас:

- **жизненно важные системы**, ошибка в которых может привести к гибели людей (авиастроение, медицина, ПО для атомных станций). Надо тестировать очень тщательно;
- **сложная система или сложная часть системы.** Чтобы каждый раз не вспоминать «а как мне это сделать?», лучше написать тест-кейс.
- **в команде постоянно появляются новые люди.**

Тест-кейсы не нужны, когда у нас:

- **простые системы** (веб-сайты одностраничники, мобильные приложения и т. п.);
- **если в команде всего один или два тестировщика, знающие свой продукт.** Время, потраченное на создание и поддержку тест-кейсов, никогда не окупится.

Так как тест-кейсы очень сложно поддерживать, то чаще используют чек-листы или комбинацию «чек-листы & тест-кейсы».

В последнем случае большинство проверок пишут в виде чек-листов, а особо сложные уже в виде тест-кейсов, чтобы каждый раз не вспоминать, как этот хитрый сценарий работает.

2 . Чек–листы

Чек-лист - это список проверок. Это своего рода напоминание: «не забудь проверить то и то».

Пример чек-листов в жизни:

- ✓ список покупок
- ✓ список вещей в отпуск

Как оформить чек-лист?

Оформление может
быть любым:

- список
- таблица
- схема
-

| Имена | | |
|-----------|---|--|
| Мужское | | |
| Женское | ✓ | |
| Унисекс | ✓ | |
| Составное | | |
| Пустое | | |

| Проверка | Пример | Результат |
|----------|--------|----------------------|
| Мужское | Иван | Успешная регистрация |
| Женское | Мария | Успешная регистрация |
| Унисекс | Саша | Успешная регистрация |
| Пустое | | Ошибка: введите имя! |

Это чек-лист

И это чек-лист

И даже это чек-лист



Оформление чек-листа

| Неправильно | Правильно |
|--|---|
| <p>1. Открыть страницу регистрации https://www.example.com/. Ввести в поле «имя» обычное имя, в поле «емайл» — корректный емайл, в поле «пароль» — корректный пароль.</p> <p>2. Открыть страницу регистрации https://www.example.com/. Ввести в поле «имя» распространенное имя, в поле «емайл» — корректный емайл, в поле «пароль» — корректный пароль.</p> <p>3. Открыть страницу регистрации https://www.example.com/. Ввести в поле «имя» редкое имя, в поле «емайл» — корректный емайл, в поле «пароль» — корректный пароль.</p> <p>4. ...</p> | <p>Открыть страницу регистрации https://www.example.com/.</p> <p>Проверки для поля «имя»:</p> <ol style="list-style-type: none">1. Обычное.2. Распространенное.3. Редкое.4. ... |

Лучше сразу писать
конкретные
примеры

Если «Регистрация прошла успешно, на почту отправлено письмо-приветствие» - результат «ОК». (вынесли из таблицы общий результат, что бы не засорять таблицу)

| Описание | Пример | Результат |
|-------------------------------------|--|-----------|
| Все поля заполнены правильно | Имя: Ольга Email: ok.molechka@gmail.com Пароль: 1 | ОК |
| Проверка поля «Имя» | | |
| Свое имя | Ольга | ОК |
| Короткое имя | Ева | ОК |
| Длинное имя (составное) | Розалинд Аруша Аркадина Алталун Флоренс Луна | ОК |
| ... | ... | ... |
| Проверка поля «Email» | | |
| Корректный email (популярный домен) | olga@mail.ru | ОК |
| Точка внутри email | ok.molechka@gmail.com | ОК |
| Кириллический email | олечка@мустики.рф | ОК |
| Пустая почта | | Ошибка |
| Одно слово вместо домена | olga@fdgfdg | Ошибка |
| ... | | |

Но лучше результат описывать конкретно, или добавить графу для комментариев.

| Описание | Пример | Результат |
|------------------------------|--|--|
| Все поля заполнены правильно | Имя: Ольга Email: ok.molechka@gmail.com Пароль: 1 | Регистрация прошла успешно, на почту отправлено письмо-приветствие |

| Проверка поля «Имя» | | |
|-------------------------|--|---|
| Свое имя | Ольга | Успешная регистрация |
| Короткое имя | Ян | Ограничений на минимальное имя нет |
| Длинное имя (составное) | Розалинд Аруша Аркадина Алталун Флоренс Луна | В личном кабинете имя отображается нормально, влезает в отведенное под него место |
| ... | ... | ... |

Особенности чек-листов

- минимум информации.
- независимые.

Плюсы

- Меньше текста - проще и быстрее читать.
- Проще поддерживать - за счет предыдущих плюсов в основном. Если меньше текста, он медленнее устаревает.
- Быстрее писать - аналогично: меньше текста, больше дела!

Минусы

- Не все поймут, как проводить тест.

Чек-листы применимы, когда:

- вы - единственный тестирующий на проекте. Тогда нет смысла тратить время на подробное описание шагов, которые вы и так уже все наизусть знаете;
- система не суперсложная, и без указания конкретных шагов понятно, что и как делать.

Есть еще ЧИТ-ЛИСТЫ 😊

- Чек-лист - список проверок
- Чит-лист - список универсальных проверок

Подробнее о чит-листах

<https://habr.com/ru/articles/733986/>

<https://habr.com/ru/articles/715262/>

Чит-лист для проверки поля email

- 1) Пустое поле email -> Сообщение о незаполненном поле email
- 2) Email в нижнем регистре -> Операция проводится успешно
- 3) Email в верхнем регистре -> Операция проводится успешно
- 4) Email с цифрами в имени аккаунта -> Операция проводится успешно
- 5) Email с цифрами в доменной части -> Операция проводится успешно
- 6) Email с дефисом в имени аккаунта -> Операция проводится успешно
- 7) Email с дефисом в доменной части -> Операция проводится успешно
- 8) Email со знаком подчеркивания в имени аккаунта -> Операция проводится успешно
- 9) Email со знаком подчеркивания в доменной части -> Операция проводится успешно
- 10) Email с точками в имени аккаунта -> Операция проводится успешно
- 11) Email с несколькими точками в доменной части -> Операция проводится успешно
- 12) Email без точек в доменной части -> Должно появиться сообщение о неправильном или некорректном e-mail введенном в поле
- 13) Превышение длины email (>320 символов) -> Должно появиться сообщение о неправильном или некорректном e-mail введенном в поле
- 14) Отсутствие @ в email -> Должно появиться сообщение о неправильном или некорректном e-mail введенном в поле
- 15) Email с пробелами в имени аккаунта -> Должно появиться сообщение о неправильном или некорректном e-mail введенном в поле.

ИТАК.....

Тест-кейс

Шаги

1. Открываем сайт <https://www.example.com/>
2. Нажимаем кнопку «Вход» в правом верхнем углу.
3. Устанавливаем курсор на поле «email».
4. Вводим значение «olga@mail.com».
5. Устанавливаем курсор на поле «Пароль».
6. Вводим значение «12345».
7. Нажимаем «Сохранить».

Ожидаемый результат

Открылась главная страница сайта.
В верхнем правом углу отображается приветствие — «Здравствуйте, Ольга!»

Чек-лист

- Авторизоваться через email

ТЕСТ ДИЗАЙН (ТД)

Этап процесса тестирования ПО, на котором проектируются и создаются тестовые случаи (тест кейсы, чеклисты или тест сценарии), в соответствии с определёнными ранее критериями качества и целями тестирования.

Или проще:

Тест-дизайн - набор техник, которые позволяют приложить мало усилий и получить много результата.

Цель техник тест дизайна - Определить тестовые условия, тестовые случаи (проверки) и тестовые данные.

- **Тестовый случай:** проверить уникальность поля Login.
- **Тестовые условия:** Приложение корректно установлено и доступно для тестирования.
- **Тестовые данные:** в базе данных существует пользователь с Login=Ivan.

Основные техники тест-дизайна

| Английское название | Русский эквивалент |
|----------------------------|---------------------------------|
| Equivalence Class Testing | Классы эквивалентности |
| Boundary Value Testing | Граничные значения |
| State-Transition Testing | Диаграмма состояний и переходов |
| Decision Table Testing | Таблица решений |
| User Case Testing | Варианты использования |
| Allpairs Algorithm testing | Попарное тестирование |
| Orthogonal Arrays Testing | Ортогональная матрица |
| ... | ... |

Классы эквивалентности

Два значения называются **эквивалентными**, когда приводят к одному результату. В таком случае неважно, взять значение А или Б для теста – так как они идентичны.

Эквивалентный класс — это одно или больше значений ввода, от которых ожидается сходное поведение, то есть они должны обрабатываться аналогичным образом. ..

А если реакция на все значения из одного класса эквивалентности одинаковая, то нет необходимости проверять все значения из этого класса.

Таким образом можно минимизировать количество тест-кейсов.

Пример 1:

В поле процент прибыли пользователь вводит число. Разрешенные значения находятся в диапазоне от 1 до 100 по требованию заказчика.

Шаг 1: Определите эквивалентные классы:

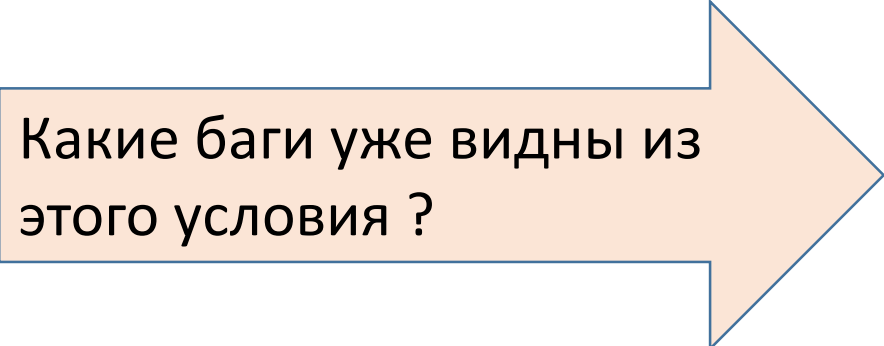
- Положительные числа (1–100)
- Отрицательные числа и ноль (≤ 0)
- Больше допустимого (> 100)
- Буквы (a-z, a-Z, кириллица, арабский и т. д.)
- Специальные символы (!@# и т. д.)

Шаг 2: Выберите одно значение для каждого класса.

Итого у нас есть 5 тест-кейсов для проверки: 50, -10, 146, aBc, !

Пример 2:

Предположим, что книжный интернет магазин запускает новую кампанию "Больше тратишь — больше скидка".



Какие баги уже видны из этого условия ?

| Потраченная сумма, руб- | Скидка, % |
|----------------------------|--------------|
| 200 — 500 | 2 |
| 500—1000 | 3 |
| 1000 — 5000 | 4 |
| 5000 и более | 5 |


Баг1: Непонятно, по какой ставке рассчитывается скидка, если потрачены следующие суммы: ровно 500 руб., ровно 1000 руб., ровно 5000 руб., так как каждая из этих сумм находится не в одной, а в двух корзинах со скидками.

Баг 2: Что означает "Потраченная сумма"? Это количество дензнаков, выплаченных только за книги, или полная сумма к оплате, включая оплату книг и расходы на доставку?

Баг 3: Для полноты картины нужно дописать эквивалентный класс от 0 до 199,99, на значения которого никакая скидка не распространяется.

| Стоимость купленных книг, руб. | Скидка, % |
|--------------------------------|-----------|
| 0—199,99 | 0 |
| 200,00 — 499,99 | 2 |
| 500,00 — 999,99 | 3 |
| 1000,00 — 4999,99 | 4 |
| 5000,00 и более | 5 |

Каждое значение внутри каждого класса является эквивалентным всем другим значениям этого класса. Потому что ко всем значениям класса должна применяться одинаковая логика кода.



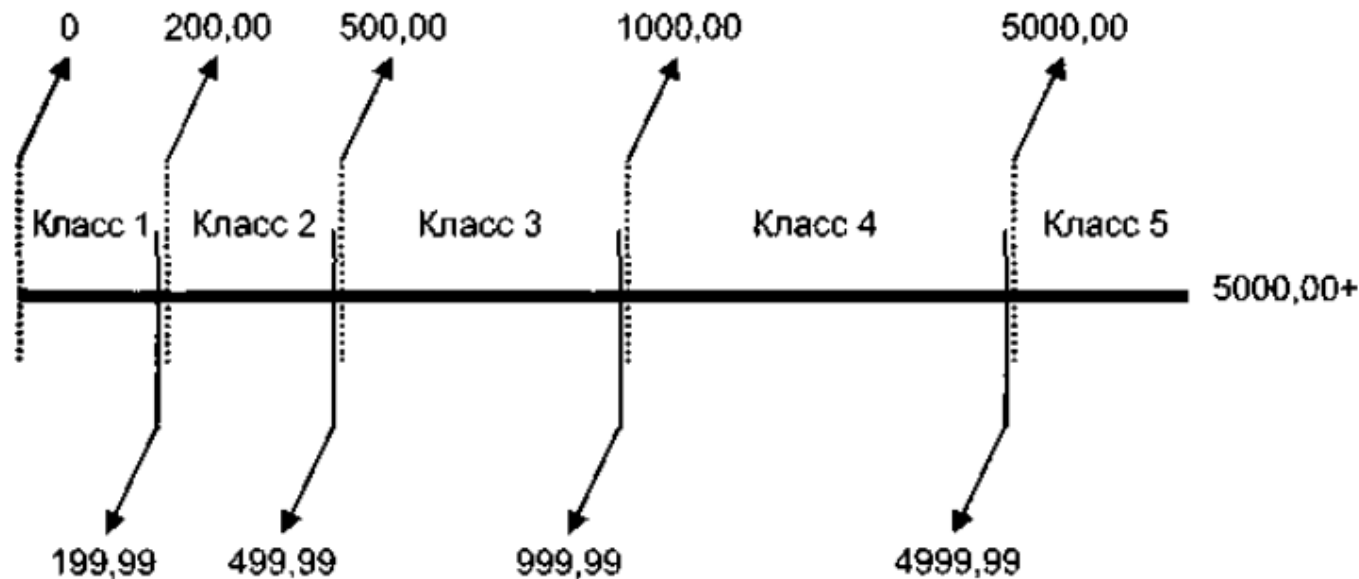
| | |
|----------|-------------------|
| Класс 1: | 0—199,99 |
| Класс 2: | 200,00 — 499,99 |
| Класс 3: | 500,00 — 999,99 |
| Класс 4: | 1000,00 — 4999,99 |
| Класс 5: | 5000,00 и более |

Полезность раскладывания значений ввода на эквивалентные классы состоит в том, что мы отсеиваем огромное количество значений ввода, использовать которые для тестирования просто бессмысленно. Отсев происходит путем применения знаний о тестировании **граничных значений**.

Граничные значения

Эта техника является "братом" разбиения на классы эквивалентности. Смысл этого подхода заключается в выборе значений на границах эквивалентных классов с минимальным шагом.

Вернемся к
примеру
книжного
интернет-
магазина



Поведение на границах эквивалентных областей имеет наибольшие шансы быть некорректным, таким образом **границы являются потенциальным источником дефектов.**

Для каждого эквивалентного класса может быть лишь один из трех вариантов:

- Есть только нижний предел (класс 5).
- Есть нижний и верхний пределы (класс 2, класс 3, класс 4).
- Есть только верхний предел (не рассматриваемый в данном примере класс, который ограничен только сверху отрицательным значением, непосредственно предшествующим классу 1).

Тестирование граничных значений

1. Тестируется нижний предел данного класса (если он имеется).
2. Затем тестируется верхний предел данного класса (если он имеется).
3. Затем тестируется любое значение внутри данного класса.
4. Затем тестируется верхний предел класса, непосредственно предшествующего данному классу (если предшествующий класс имеется).
5. Затем тестируется нижний предел класса, непосредственно следующего за данным классом (если следующий класс имеется).

Можно
записать
граничные
значения
иначе

Min -1
Min
Average
Max
Max +1

Какие из этих тестов
являются
позитивными, а какие
— негативными ?

Вернемся к примеру с книжным магазином

Протестируем эквивалентный класс 2.

Для покупок от 200,00 до 499,99 руб. (включительно) будет дана скидка 2%.

Количество всех вариантов для данного диапазона:

30 тыс (по количеству копеек в 299,99 руб. плюс один случай, когда потрачено 200,00 руб.)

Следуем методике тестирования эквивалентного класса, нам нужно лишь

5 вариантов данных:

- | | |
|--|------------------|
| 1. 200,00; 2. 499,99; 3. 315,11; 4. 199,99; 5. 500,00. | Позитивные тесты |
|--|------------------|

Как ошибки в тестировании граничных значений указывают на ошибки в коде

| | |
|---|---|
| Тест-кейс | Код с выделенной жирным шрифтом частью, которая проверяется данным тестом |
| | Возможная проблема кода, разоблачаемая тестом, и пример проблемы |
| | Ожидаемый результат |
| а. Сначала тестируется <i>нижний</i> предел данного класса (если нижний предел имеется): 200 | ЕСЛИ сумма > 200,00 И сумма < 499,99, ТО скидка = сумма/100 x 2 |
| | Ошибка в знаке равенства и/или сумме нижнего предела. |
| | <i>Пример (знак равенства перед 200,00 пропущен):</i> <i>ЕСЛИ сумма > 200,00 И сумма < 499,99,</i> <i>ТО скидка = сумма/100 x 2</i> |
| | 2% от 200 |

| | |
|--|---|
| б. Затем тестируется <i>верхний</i> предел данного класса (если верхний предел имеется): 499,99 | ЕСЛИ сумма > 200,00 И сумма < 499,99 , ТО скидка = сумма/100 x 2 |
| | Ошибка в знаке равенства и/или сумме верхнего предела. |
| | <i>Пример (499,00 вместо 499,99): ЕСЛИ сумма ></i> <i>200,00 И сумма < 499,00, ТО скидка = сумма/100</i> <i>x 2</i> |
| | 2% от 499,99 |
| в. Затем тестируется любое значение внутри данного класса: 315,11 | ЕСЛИ сумма > 200,00 И сумма < 499,99 , ТО скидка = сумма/100 x 2 |
| | Ошибка в знаках больше (>) и меньше (<). <i>Пример</i> <i>(больше вместо меньше и меньше вместо больше):</i> <i>ЕСЛИ сумма < 200,00 И сумма > 499,00: ТО скидка</i> <i>= сумма/100 x 2</i> |
| | 2% от 315,11 |

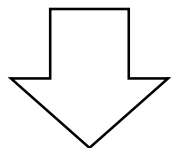
| | |
|--|---|
| г. Затем тестируется <i>верхний</i> предел класса, непосредственно <i>предшествующего</i> данному классу (если предшествующий класс имеется): 199,99 | ЕСЛИ сумма > 200,00 И сумма < 499,99, ТО скидка = сумма/100 x 2 |
| | Тонкий момент. Здесь мы проверяем две вещи: 1. Наличие скачка от верхнего предела предыдущего класса к нижнему пределу нашего класса. Это делается для следующей ситуации. Допустим, программист напечатал 100,00 вместо 200,00 : ЕСЛИ сумма > 100,00 И сумма < 499,99, ТО скидка = сумма/100 x 2. Если сделана такая ошибка, то она не будет обнаружена ни тестом <i>а</i> , ни тестом <i>б</i> , ни тестом <i>е</i> . |
| | 2. Логическое "И", так как если бы у нас было "ИЛИ": ЕСЛИ сумма > 200,00 ИЛИ сумма < 499,99, ТО скидка = сумма/100 x 2, то к данному классу принадлежало бы любое в принципе возможное значение |
| | Скидка не равна 2% от 199,99 |

!!!

| | |
|---|---|
| <p>д. Затем тестируется <i>нижний</i> предел класса, непосредственно <i>следующего</i> за данным классом (если следующий класс имеется): 500,00</p> | <p>ЕСЛИ сумма > 200,00 И сумма < 499,99, ТО скидка = сумма/100 x 2</p> |
| | <p>1. Наличие скачка от верхнего предела нашего класса к нижнему пределу следующего за ним класса. Это делается для следующей ситуации. Допустим, программист напечатал 599,99 вместо 499,99: ЕСЛИ сумма > 200 И сумма < 599,99, ТО скидка = сумма/100 x 2</p> |
| | <p>Если сделана такая ошибка, то она не будет обнаружена ни тестом <i>a</i>, ни тестом <i>б</i>, ни тестом <i>в</i>, ни тестом <i>г</i>.</p> |
| | <p>2. Проверяется логическое "И", так как если бы у нас было "ИЛИ": ЕСЛИ сумма > 200,00 ИЛИ сумма < 499,99, ТО скидка = сумма/100 x 2, то к данному классу принадлежало бы любое в принципе возможное значение</p> |
| | <p>Скидка не равна 2% от 500,00</p> |

!!!

Итак, 5 вариантов данных
для одного класса
эквивалентности.



1. 200,00;
2. 499,99;
3. 315,11;
4. 199,99;
5. 500,00.

А сколько будет
вариантов данных
для 5 классов ?

| | |
|----------|-------------------|
| Класс 1: | 0—199,99 |
| Класс 2: | 200,00 — 499,99 |
| Класс 3: | 500,00 — 999,99 |
| Класс 4: | 1000,00 — 4999,99 |
| Класс 5: | 5000,00 и более |

| Класс | Значение | Ожидаемая ставка скидки, % |
|---------|----------|----------------------------|
| Класс 1 | 0 | 0 |
| | 100,00 | |
| | 199,99 | |
| Класс 2 | 200,00 | 2 |
| | 315,11 | |
| | 499,99 | |
| Класс 3 | 500,00 | 3 |
| | 659,23 | |
| | 999,99 | |
| Класс 4 | 1000,00 | 4 |
| | 3265,26 | |
| | 4999,99 | |
| Класс 5 | 5000,00 | 5 |
| | 5075,00 | |

Для тестирования одного класса эквивалентности понадобилось 5 тестов, а если тестировать 5 классов понадобится **всего 14 тестов**, так как некоторые граничные условия будут учтены при тестировании других классов

Пример: СИСТЕМА ПОДБОРА КРЕДИТА

Возраст до 18 лет



Кредит не разрешен

Возраст от 18 до 65 лет



- Потребительский кредит
- Кредит на недвижимость
- Кредит на покупку авто

Возраст старше 65 лет



Потребительский
кредит

Пример: СИСТЕМА ПОДБОРА КРЕДИТА

Класс эквивалентности #1: от 1 до 17

Возраст до 18 лет



Кредит не разрешен

Класс эквивалентности #2: от 18 до 65

Возраст от 18 до 65 лет



- Потребительский кредит
- Кредит на недвижимость
- Кредит на покупку авто

Класс эквивалентности #3: от 66 до 99

Возраст старше 65 лет



Потребительский
кредит

Пример: СИСТЕМА ПОДБОРА КРЕДИТА

Класс эквивалентности #1: от 1 до 17

Возраст до 18 лет



Кредит не разрешен

Min -1: пустое поле «Возраст»

Min: 1

Average: 11

Max: 17

Max +1: 18

> Сообщение об ошибке

> «Кредит не разрешен»

> «Кредит не разрешен»

> «Кредит не разрешен»

> «Потребительский кредит

Кредит на недвижимость

Кредит на покупку авто»

Пример: СИСТЕМА ПОДБОРА КРЕДИТА

Класс эквивалентности #2: от 18 до 65 лет

Возраст до 18 до 65 лет 

- Потребительский кредит
- Кредит на недвижимость
- Кредит на покупку авто

Min -1: 17

Min: 18

Average: 40

Max: 65

Max +1: 66

> «Кредит не разрешен»

> 3 кредита

> 3 кредита

> 3 кредита

> «Потребительский кредит»

Пример: СИСТЕМА ПОДБОРА КРЕДИТА

Класс эквивалентности #3: от 66 до 99 лет

Возраст старше 65 лет



Потребительский
кредит

Min -1: 65
Min: 66
Average: 80
Max: 99
Max +1: 100

> 3 кредита
> «Потребительский кредит»
> «Потребительский кредит»
> «Потребительский кредит»
> Система не позволяет ввести
больше двух символов

Граничные значения для поля ДАТА РОЖДЕНИЯ

Формат: ДД.ММ.ГГГГ

День (значение):


| | Январь, март, май, июль, август, октябрь, декабрь | Февраль (високосный год) | Февраль (не високосный год) | Апрель, июнь, сентябрь, ноябрь |
|---------|---|-----------------------------|-----------------------------------|-----------------------------------|
| Min – 1 | 00 | 00 | 00 | 00 |
| Min | 01 | 01 | 01 | 01 |
| Average | 15 | 15 | 15 | 15 |
| Max | 31 | 29 | 28 | 30 |
| Max +1 | 32 | 30 | 29 | 31 |

День (формат):

Min -1: 1 символ Min: 2 символа Average: 2 символа Max: 2 символа Max+1: 3 символа

E-MAIL

Формат: **name** **@** **hostname** **.** **domain**



| | Local-part | @ | Hostname | dot | domain |
|---------|-------------------------|--------------|----------------|--------------|---------------|
| Min – 1 | <i>Empty</i> | <i>Empty</i> | 1 character | <i>Empty</i> | 1 character |
| Min | 1 character | @ | 2 characters | . | 2 characters |
| Average | 30 different characters | @ | 150 characters | . | 6 characters |
| Max | 64 | @ | 243 characters | . | 11 characters |
| Max +1 | 65 | @ @ | 244 characters | .. | 12 characters |

E-MAIL

Формат: **local-part** @ **hostname.domain**
Size: 256 characters

ФОРМАТ

| | Local-part | @ | Hostname | dot | domain |
|---------|---------------|--------------|----------------|--------------|---------------|
| Min – 1 | <i>Empty</i> | <i>Empty</i> | 1 character | <i>Empty</i> | 1 character |
| Min | 1 character | @ | 2 characters | . | 2 characters |
| Average | 30 characters | @ | 150 characters | . | 6 characters |
| Max | 64 | @ | 243 characters | . | 11 characters |
| Max +1 | 65 | @ @ | 244 characters | .. | 12 characters |

РАЗМЕР

| | Size | Examples |
|---------|----------------|-----------------------------------|
| Min – 1 | 5 characters | @a.by; aa.by; a@.by; a@aby; a@a.b |
| Min | 6 characters | a@a.by |
| Average | 150 characters | ... |
| Max | 256 characters | ... |
| Max +1 | 257 characters | ... |

Тестирование ТАБЛИЦЫ РЕШЕНИЙ

Таблица решений содержит триггерные условия, обычно комбинации значений «истина» и «ложь» для всех входных условий, и результирующие действия для каждой комбинации условий. Каждый столбец таблицы соотносится с бизнес-правилом, определяющим уникальную комбинацию условий и результат выполнения действий, связанных с этим правилом.

Стандартом покрытия для тестирования таблицы решений обычно является наличие хотя бы одного теста для каждой колонки, что обычно включает в себя покрытие всех комбинаций триггерных условий.

СИСТЕМА ПОКУПКИ ПРОЕЗДНЫХ БИЛЕТОВ

ВИД ТРАНСПОРТА:

АВТОБУС



ТРОЛЛЕЙБУС



МЕТРО



ТРАМВАЙ



ПЕРИОД:

ДЕНЬ



НЕДЕЛЯ



ДЕКАДА



МЕСЯЦ



СПОСОБ ОПЛАТЫ:

КАРТОЙ



НАЛИЧНЫМИ



СУММА К ОПЛАТЕ: 10 РУБЛЕЙ

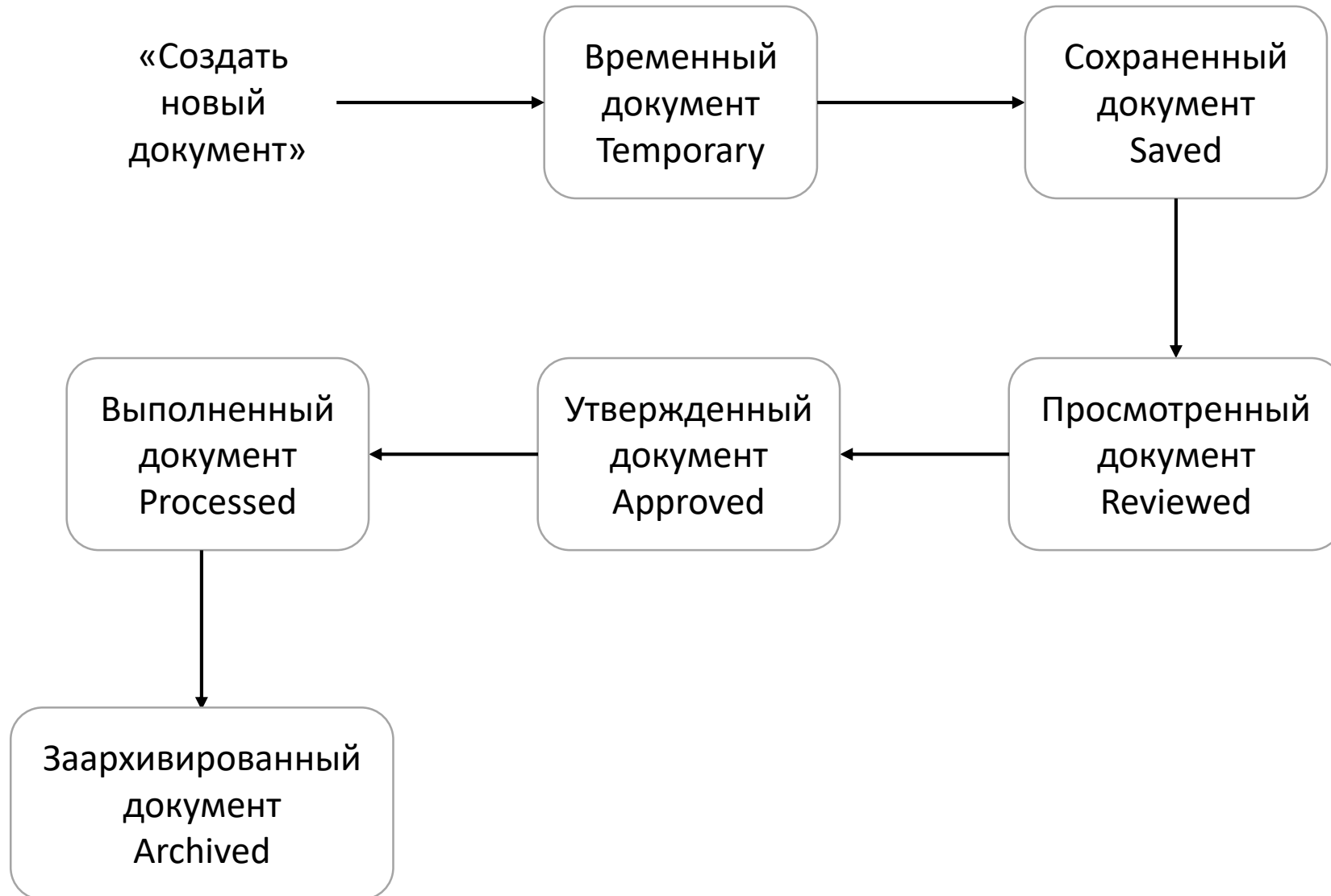
| ВИД ТРАНСПОРТА | ПЕРИОД | ОПЛАТА | ОЖИДАЕМЫЙ РЕЗУЛЬТАТ (СУММА) |
|----------------|--------|-----------|-----------------------------|
| Автобус | День | Карта | 5 рублей |
| Автобус | Неделя | Карта | 10 рублей |
| Автобус | Декада | Наличными | 13 рублей |
| Автобус | Месяц | Наличными | 50 рублей |
| Троллейбус | День | Наличными | 5 рублей |
| Троллейбус | Неделя | Карта | 10 рублей |
| Троллейбус | Декада | Наличными | 13 рублей |
| Троллейбус | Месяц | Карта | 50 рублей |
| Метро | День | Наличными | 5 рублей |
| Метро | Неделя | Карта | 10 рублей |
| Метро | Декада | Карта | 13 рублей |
| Метро | Месяц | Наличными | 50 рублей |
| Трамвай | День | Карта | 5 рублей |
| Трамвай | Неделя | Наличными | 10 рублей |
| Трамвай | Декада | Наличными | 13 рублей |
| Трамвай | Месяц | Карта | 50 рублей |

Тестирование ТАБЛИЦЫ СОСТОЯНИЙ

Система может показывать различные отклики в зависимости от текущих условий или предшествовавшей истории состояний. Данный метод позволяет тестировщику рассматривать систему с точки зрения её состояний, переходов между состояниями, входов или событий, активизирующих изменения состояний (переходы) и действия, к которым приводят эти переходы. Состояния системы или тестируемого объекта разделяемы, определяемы и конечны.

Таблица состояний демонстрирует связи между состояниями и входами и может подсказать возможные некорректные переходы.

СИСТЕМА ДОКУМЕНТООБОРОТА



| Текущее состояние | Событие | Действие | Следующее состояние |
|-------------------|----------------------------------|----------|---------------------|
| null | «Создать новый документ» | Create | Temporary |
| null | «Создать новый документ» | Discard | null |
| Temporary | Заполнение документа | Save | Saved |
| Temporary | Заполнение документа | Cancel | null |
| Saved | Просмотр документа | Review | Reviewed |
| Saved | Просмотр документа | Reject | Rejected to creator |
| Reviewed | Утверждение документа | Approve | Approved |
| Reviewed | Утверждение документа | Reject | Archived |
| Approved | Выполнение действий по документу | Process | Processed |
| Processed | Архивация документа | Archive | Archived |

СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

Сценарий использования (use case) описывает взаимодействия между участниками (включая пользователей и систему) приводящие к полезным результатам для заказчика или пользователя системы. Сценарии использования могут быть описаны на уровне абстракций (бизнес сценарий использования, уровень бизнес-процессов, не связанный с технологией) или на системном уровне (сценарий использования системы на уровне системного функционала).

Действия при разработке сценария

- ✓ Идентификация всех значений
- ✓ Выделение классов эквивалентности
- ✓ Построение таблиц с комбинациями значений из классов эквивалентности
- ✓ Написание тест-кейсов

ПОПАРНОЕ ТЕСТИРОВАНИЕ

ISTQB* определяет **попарное тестирование** как технику тест-дизайна методом черного ящика, при которой тест-кейсы создаются таким образом, чтобы выполнить **все возможные отдельные комбинации каждой пары входных параметров**.

Техника попарного тестирования очень помогает при разработке тестов для приложений, включающих множество параметров. Тесты разрабатываются таким образом, что для каждой пары входных параметров существуют все возможные комбинации этих параметров. Тестовые наборы (тест-сьюты, Test suite) охватывают все комбинации.

* **ISTQB** — International Software Testing Qualifications Board Glossary

Пример (<https://habr.com/ru/companies/otus/articles/592575/>)

Приложение для заказа автомобиля:

- С помощью приложения можно покупать и продавать машины. Приложение должно поддерживать оказание услуг в Дели и Мумбаи.
- В приложении должны содержаться регистрационные номера, которые могут быть валидными и невалидным. Оно должно разрешать продажу следующих марок автомобилей: BMW, Audi и Mercedes.
- Доступны два типа бронирования: бронирование через интернет и в офлайн-магазине.
- Заказы доступны к размещению только в рабочие часы.

Шаг 1. Перечислим задействованные переменные.

Категория заказа

- а. Купить
- б. Продать

Местоположение

- а. Дели
- б. Мумбаи

Марка автомобиля

- а. BMW
- б. Audi
- в. Mercedes

Регистрационные номера

- а. Валидные (5000)
- б. Не валидные

Тип заказа

- а. Заказ через Интернет
- б. Заказ в магазине

Время заказа

- а. Рабочие часы
- б. Нерабочие часы

Если тестировать все возможные допустимые комбинации: $2 \times 2 \times 3 \times 5000 \times 2 \times 2 =$ получаем **240 тысяч комбинаций !** Кроме того, недопустимых комбинаций вообще может быть бесконечное количество.

Шаг №2: Упростим.

Сокращаем регистрационные номера до двух типов:

- Валидный регистрационный номер
- Не валидный регистрационный номер

Категория заказа

- а. Купить
- б. Продать

Местоположение

- а. Дели
- б. Мумбаи

Марка автомобиля

- а. BMW
- б. Audi
- в. Mercedes

Регистрационные номера

- а. Валидные ~~(5000)~~
- б. Не валидные

Тип заказа

- а. Заказ через Интернет
- б. Заказ в магазине

Время заказа

- а. Рабочие часы
- б. Нерабочие часы

Теперь посчитаем количество возможных комбинаций: $= 2 \times 2 \times 3 \times 2 \times 2 \times 2 = 96$

Шаг 3. Упорядочивание задействованных переменных и значений.

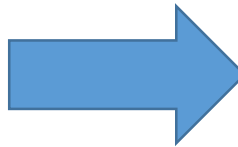
| Категория заказа | Местоположение | Марка авто | Регистрационный номер | Тип заказа | Время заказа |
|------------------|----------------|------------|-----------------------|----------------|----------------|
| Покупка | Дели | BMW | Валидные | через интернет | Рабочие часы |
| Продажа | Мумбаи | Audi | Невалидные | в магазине | Нерабочие часы |
| | | Mercedes | | | |

Теперь отсортируем переменные так, чтобы переменные с наибольшим количеством значений шли первыми, а с наименьшим — последними.

| Марка авто | Категория заказа | Местоположение | Регистрационный номер | Тип заказа | Время заказа |
|------------|------------------|----------------|-----------------------|------------|--------------|
| 3 | 2 | 2 | 2 | 2 | 2 |

Шаг 4. Расставляем переменные для создания набора тестов.

| Марка авто | Категория заказа | Ме |
|------------|------------------|----|
| BMW | | |
| BMW | | |
| | | |
| Audi | | |
| Audi | | |
| | | |
| Mercedes | | |
| Mercedes | | |



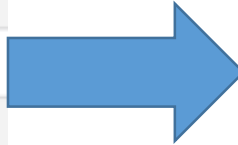
| Марка авто | Категория заказа | Мест |
|------------|------------------|------|
| BMW | Покупка | |
| BMW | Продажа | |
| | | |
| Audi | Покупка | |
| Audi | Продажа | |
| | | |
| Mercedes | Покупка | |
| Mercedes | Продажа | |

1

2

Шаг 4. Расставляем переменные для создания набора тестов.

| Марка авто | Категория заказа | Местоположение |
|------------|------------------|----------------|
| BMW | Покупка | Дели |
| BMW | Продажа | Мумбаи |
| | | |
| Audi | Покупка | Дели |
| Audi | Продажа | Мумбаи |
| | | |
| Mercedes | Покупка | Дели |
| Mercedes | Продажа | Мумбаи |



| Марка авто | Категория заказа | Местоположение |
|------------|------------------|----------------|
| BMW | Покупка | Дели |
| BMW | Продажа | Мумбаи |
| | | |
| Audi | Покупка | Мумбаи |
| Audi | Продажа | Дели |
| | | |
| Mercedes | Покупка | Дели |
| Mercedes | Продажа | Мумбаи |

3

4

Теперь у нас есть Покупка&Дели, но нет Покупка&Мумбаи. Есть Продажа&Мумбаи, но нет Продажа&Дели. Поменяем значения из второго набора в третьем столбце.

Повторим шаги для столбцов 3 и 4.

| Марка авто | Категория заказа | Местоположение | Регистрационный номер | Т |
|------------|------------------|----------------|-----------------------|---|
| BMW | Покупка | Дели | Валидный | |
| BMW | Продажа | Мумбаи | Невалидный | |
| | | | | |
| Audi | Покупка | Мумбаи | Валидный | |
| Audi | Продажа | Дели | Невалидный | |
| | | | | |
| Mercedes | Покупка | Дели | Валидный | |
| Mercedes | Продажа | Мумбаи | Невалидный | |

Если сравнить столбцы 3 и 4, каждое значение из столбца 3 имеет пару с обоими значениями из столбца 4. Но если сравнить второй и четвертый столбец, у нас есть комбинации Покупка&Валидный и Продажа&Невалидный, но нет комбинаций Покупка&Невалидный и Продажа&Валидный. Следовательно, нам надо поменять местами последний набор значений в четвертом столбце.

| Марка авто | Категория заказа | Местоположение | Регистрационный номер | Тип заказа | Время заказа |
|------------|------------------|----------------|-----------------------|----------------|----------------|
| BMW | Покупка | Дели | Валидный | Через интернет | Рабочие часы |
| BMW | Продажа | Мумбаи | Невалидный | В магазине | Нерабочие часы |
| | | | | | |
| Audi | Покупка | Мумбаи | Валидный | В магазине | Рабочие часы |
| Audi | Продажа | Дели | Невалидный | Через интернет | Нерабочие часы |
| | | | | | |
| Mercedes | Покупка | Дели | Невалидный | В магазине | Рабочие часы |
| Mercedes | Продажа | Мумбаи | Валидный | Через интернет | Нерабочие часы |

С шестым столбцом (Время заказа) у нас проблемка: не хватает пар Покупка&Нерабочие часы и Продажа&Рабочие часы. Нам не удастся получить недостающие пары, поменяв значения местами, поскольку мы ранее уже поменяли местами все строки, и если мы снова начнём их менять, то есть риск пропустить другие возможные пары. Поэтому добавим еще два тестовых случая, которые содержат эти пары. Заполним пустые строки!

| Марка авто | Категория заказа | Местоположение | Регистрационный номер | Тип заказа | Время заказа |
|------------|------------------|----------------|-----------------------|----------------|----------------|
| BMW | Покупка | Дели | Валидный | Через интернет | Рабочие часы |
| BMW | Продажа | Мумбаи | Невалидный | В магазине | Нерабочие часы |
| | Покупка | | | | Нерабочие часы |
| Audi | Покупка | Мумбаи | Валидный | В магазине | Рабочие часы |
| Audi | Продажа | Дели | Невалидный | Через интернет | Нерабочие часы |
| | Продажа | | | | Рабочие часы |
| Mercedes | Покупка | Дели | Невалидный | В магазине | Рабочие часы |
| Mercedes | Продажа | Мумбаи | Валидный | Через интернет | Нерабочие часы |

заполним пустые ячейки на свое усмотрение, потому что другие значения переменных являются произвольными (обозначим знаком тильды ~)

| Марка авто | Категория заказа | Местоположение | Регистрационный номер | Тип заказа | Время заказа |
|------------|------------------|----------------|-----------------------|-----------------|----------------|
| BMW | Покупка | Дели | Валидный | Через интернет | Рабочие часы |
| BMW | Продажа | Мумбаи | Невалидный | В магазине | Нерабочие часы |
| ~BMW | Покупка | ~Дели | ~Валидный | ~Через интернет | Нерабочие часы |
| Audi | Покупка | Мумбаи | Валидный | В магазине | Рабочие часы |
| Audi | Продажа | Дели | Невалидный | Через интернет | Нерабочие часы |
| ~Audi | Продажа | ~Мумбаи | ~Невалидный | ~В магазине | Рабочие часы |
| Mercedes | Покупка | Дели | Невалидный | В магазине | Рабочие часы |
| Mercedes | Продажа | Мумбаи | Валидный | Через интернет | Нерабочие часы |

Итого мы получили всего **8 тест-кейсов вместо 96.**

Этот пример показывает, насколько эффективной может быть **техника попарного тестирования**. Она здорово повышает шансы найти баги, при этом сохранив время.

Для генерации комбинаций тестов методом попарного тестирования можно использовать [онлайн-инструменты](#)

Техника попарного тестирования помогает существенно уменьшить количество комбинаций проверок, **достаточных для обеспечения необходимого уровня качества** программного обеспечения.

Однако эта техника имеет и некоторые ограничения. Она не сработает, если:

- ✓ выбранные для тестирования значения некорректны;
- ✓ мало внимания уделяется комбинациям, которые могут привести к ошибке с высокой долей вероятности;
- ✓ взаимодействие между переменными недостаточно изучено.

ПОПАРНОЕ ТЕСТИРОВАНИЕ

| # | Тип подключения | Микрофон | Подсветка |
|---|-----------------|----------|-----------|
|---|-----------------|----------|-----------|

| | | | |
|---|-----------|---------|----------|
| 1 | Проводной | Включен | Включена |
|---|-----------|---------|----------|

| | | | |
|---|-----------|---------|-----------|
| 2 | Проводной | Включен | Выключена |
|---|-----------|---------|-----------|

| | | | |
|---|-----------|----------|----------|
| 3 | Проводной | Выключен | Включена |
|---|-----------|----------|----------|

| | | | |
|---|-----------|----------|-----------|
| 4 | Проводной | Выключен | Выключена |
|---|-----------|----------|-----------|

| | | | |
|---|--------------|---------|----------|
| 5 | Беспроводной | Включен | Включена |
|---|--------------|---------|----------|

| | | | |
|---|--------------|---------|-----------|
| 6 | Беспроводной | Включен | Выключена |
|---|--------------|---------|-----------|

| | | | |
|---|--------------|----------|----------|
| 7 | Беспроводной | Выключен | Включена |
|---|--------------|----------|----------|

| | | | |
|---|--------------|----------|-----------|
| 8 | Беспроводной | Выключен | Выключена |
|---|--------------|----------|-----------|

Пример 2. Возьмем наушники с характеристиками:

- Тип подключения (беспроводной или проводной)
- Микрофон (включен или выключен)
- Подсветка (включена или выключена)

Для данного примера число тестов сократилось вдвое!

| # | Тип подключения | Микрофон | Подсветка |
|---|-----------------|----------|-----------|
|---|-----------------|----------|-----------|

| | | | |
|---|--------------|----------|-----------|
| 1 | Беспроводной | Выключен | Выключена |
|---|--------------|----------|-----------|

| | | | |
|---|-----------|---------|-----------|
| 2 | Проводной | Включен | Выключена |
|---|-----------|---------|-----------|

| | | | |
|---|--------------|---------|----------|
| 3 | Беспроводной | Включен | Включена |
|---|--------------|---------|----------|

| | | | |
|---|-----------|----------|----------|
| 4 | Проводной | Выключен | Включена |
|---|-----------|----------|----------|

ТЕСТ ДИЗАЙН НА ОСНОВЕ ОПЫТА

Подходы основаны на предыдущем опыте использования аналогичных приложений / платформ.

- ✓ Предположения об ошибках (error guessing technique);
- ✓ Исследовательское тестирование (exploratory testing).

Вопрос:

Рассмотренные методы относятся к методам
Черного ящика или Белого ящика ?

- ✓ Классы эквивалентности
- ✓ Граничные значения
- ✓ Таблица решений
- ✓ Диаграмма состояний
- ✓ Сценарии использования
- ✓ Попарное тестирование