

NAUMEN

# Тест-анализ и тест- дизайн

Арина Кислова

Инженер по тестированию ПО

20.09.2

2

## **План лекции**

- **Определения тест-дизайна и тест-анализа**
- **Исследование продукта**
- **Техники проектирования тестов**
- **Техники проектирования тестовых данных**

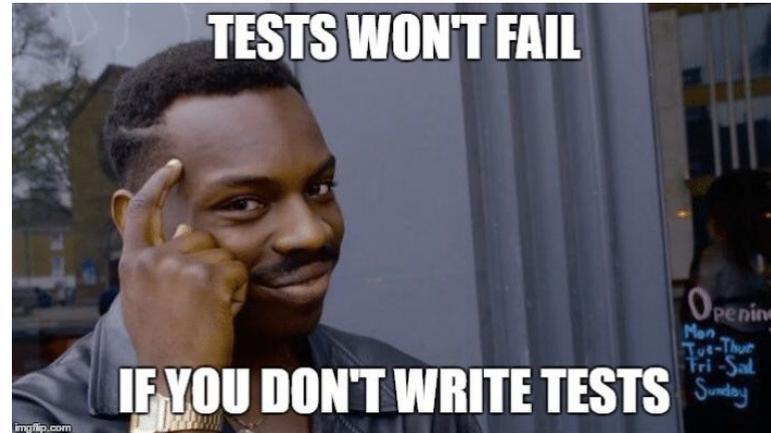
# Тест-анализ (что будем тестировать?)

- процесс анализа базиса тестирования ( то есть требований к продукту или системе) и определения целей тестирования

- зачем создан продукт, кто и как им пользуется
- как работает продукт
- что нам необходимо тестировать
- риски

## Тест- дизайн (как будем тестировать?)

Процесс проектирования и создания тест-кейсов (тестовых сценариев), в соответствии с определёнными ранее критериями качества и целями тестирования



# Тест-кейс

документ, описывающий последовательность действий, необходимых для проверки функционала.

## Тестирование добавления статьи в Избранное

Предварительные шаги:

1. Создать статью Article 1. Перевести в статус Опубликована.
2. Авторизоваться на портале

Шаги:

1. Открыть статью Article 1
2. Щелкнуть по иконке "Звездочка"

Ожидаемый результат:

На главной странице статья Article 1 отображается в разделе

**Тест-кейс = проверка**

# Процесс тестирования задачи





# Исследование продукта

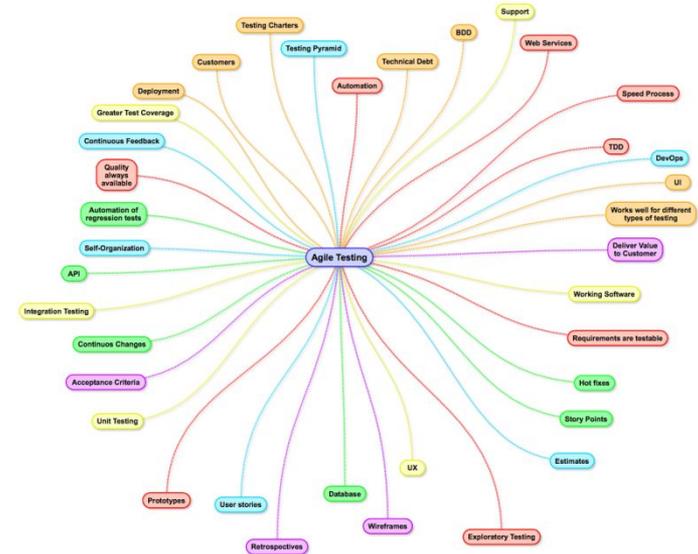
## Зачем исследовать?

- Декомпозиция функционала
- Оценка взаимосвязей
- Планирование работ
- Сбор статистики
- Понимание продукта
- Поиск потерянного



# Процесс декомпозиции

- Сбор входящей информации
  - спецификация,
  - продукт,
  - общение с пользователями, ПМ, разработчиком
  - опыт
  - работа с кодом
  - внешние стандарты и регламенты
- Наглядное представление информации
- Поиск потерь



# Правила декомпозиции

1. На одном уровне декомпозиции находятся элементы, различающиеся по одному признаку

Удобно выделять уровни по схеме:

Функция-Параметр-Значение

1. Нижний уровень декомпозиции = один тест-кейс
2. Баланс между полнотой и простотой описания системы
3. Расстановка приоритетов
4. Вычленяемые подсистемы в сумме должны составлять всю систему, при этом взаимно исключать друг друга



# Программы

<https://www.xmind.net/> - Бесплатные майнд-карты

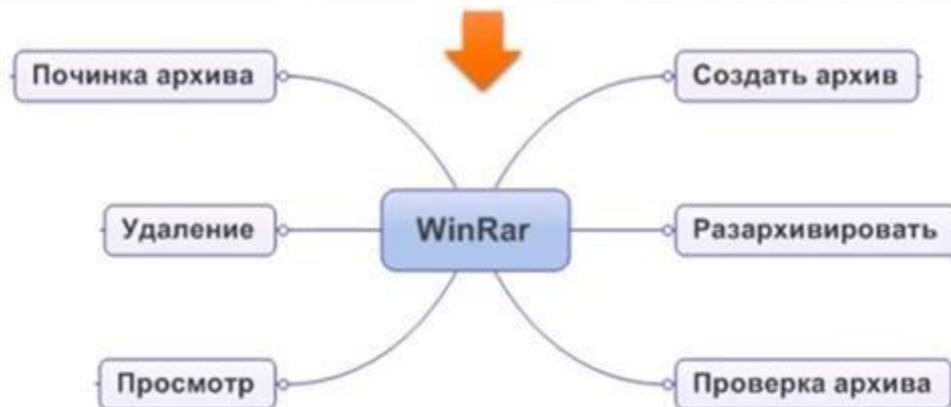
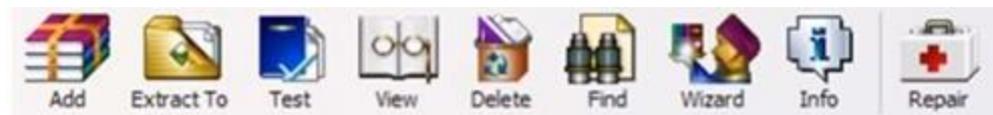
<https://www.mindmeister.com/ru> - Майнд-карты с более широким функционалом, в ограниченном формате можно работать бесплатно

<https://miro.com/app/dashboard/> - Онлайн-доска



# Исследуем тестируемое ПО

## Шаг 1 - Что делает наш продукт



Список действий

# Исследуем тестируемое ПО

Шаг 2 - Какие параметры влияют на действия

**Создать архив**

- Имя архива
- Расположение архива
- Формат архива
- Метод компрессии
- Удаление файлов после архивации
- Создание SFX архива
- Разделение архива на части

**Разархивировать**

**Проверка архива**

Параметры действий

# Исследуем тестируемое ПО

## Шаг 3 - Какие значения могут принимать параметры

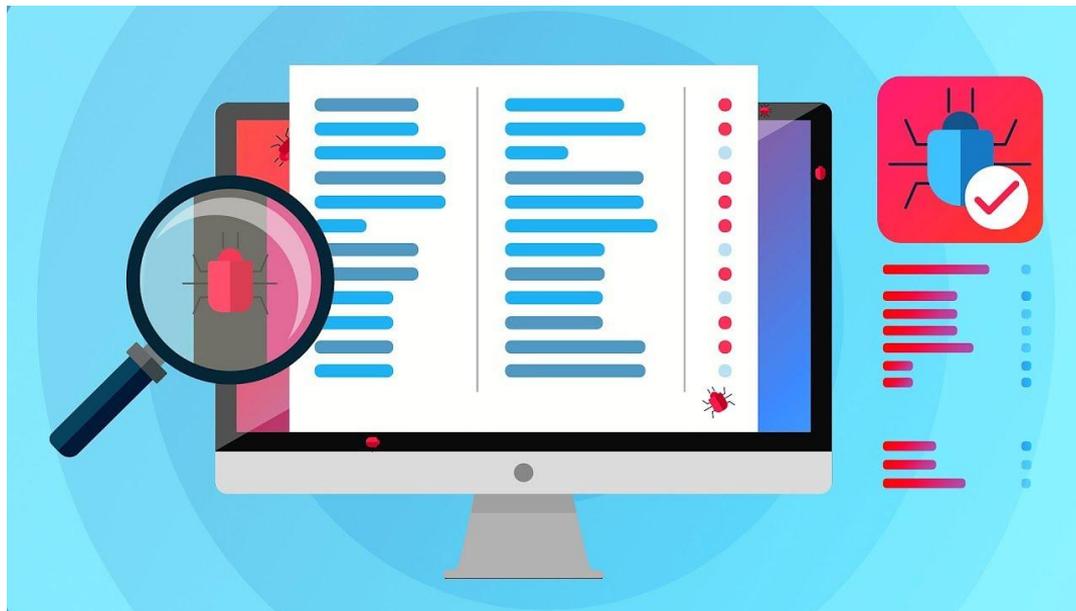


### Значение параметров действий

# Цель тест-дизайна

Минимум тестов

Максимум результата



# Тестирование по сценариям использования

**Сценарий использования (use case)** - последовательность действий пользователя и реакций системы на эти действия

## Структура сценария:

Название

Цель

Действующие лица

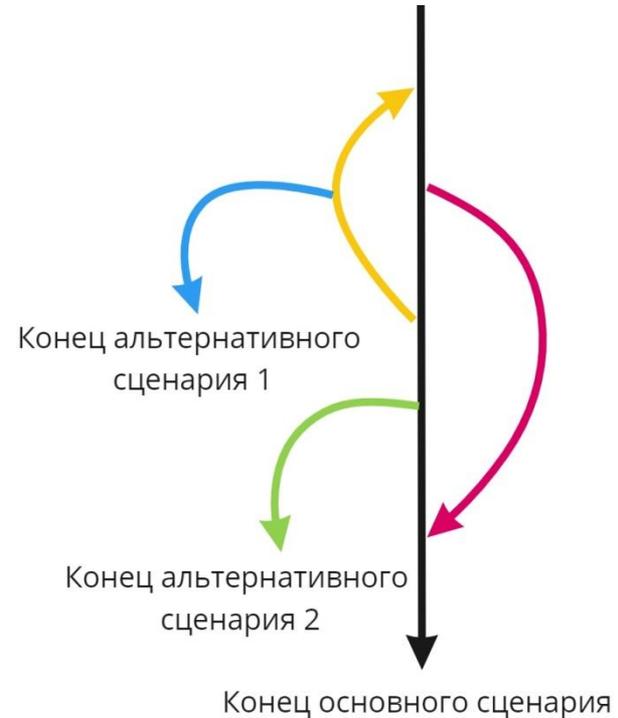
Предусловия\*

Постусловия\*

Основной сценарий

Альтернативные сценарии\*

\*- необязательный



# Тестирование по сценариям использования

<b>Название</b>	Авторизация пользователя
<b>Действующие лица</b>	Пользователь, Система
<b>Цель:</b>	П: авторизоваться в системе и начать работать; С: идентифицировать пользователя и его права.
<b>Предусловия:</b>	Учетная запись пользователя активирована

# Тестирование по сценариям

## ИСПОЛЬЗОВАНИЯ

<b>Основной сценарий</b>	<ul style="list-style-type: none"><li>• П запускает систему. С открывает сессию пользователя, предлагает ввести логин и пароль.</li><li>• Пользователь вводит логин и пароль.</li><li>• Система проверяет логин и пароль.</li><li>• Система создает запись в истории авторизаций (IP адрес пользователя, логин, дата, рабочая станция).</li><li>• Система выдает пользователю сообщение по поводу успешной авторизации (<i>ссылка на сообщение</i>).</li></ul>
<b>Результат</b>	Пользователь успешно авторизован и может работать с системой.

# Тестирование по сценариям

## ИСПОЛЬЗОВАНИЯ

Альтернативный сценарий 1	Пользователь на форме входа выбирает кнопку: «Напомнить пароль».
Результат 1:	Система высылает на почту пароль
Альтернативный сценарий 2	Количество неудачных попыток авторизоваться достигло максимального, установленного в настройках.
Результат 2:	Пользователь не может войти. Система выдает сообщение "Вход с IP адреса Пользователя заблокирован на время, установленное в настройках."

# Тесты по сценариям использования

## Преимущества

- проверяют работоспособность основной функциональности
- воспроизводят действия пользователя при работе с системой

## Ограничения

- не проверяют подробно отдельные элементы интерфейса
- не проверяют комбинации данных при заполнении форм
- только позитивные сценарии

# Тестирование переходов между состояниями

техника тест-дизайна, основанная на анализе состояний системы и переходов между состояниями



# Тестирование переходов между состояниями

## Части диаграммы

- **состояние**
- **переходы**
  - событие - пользователь что-то сделал
  - действие - реакция приложение на события
- **точка входа и точка выхода**

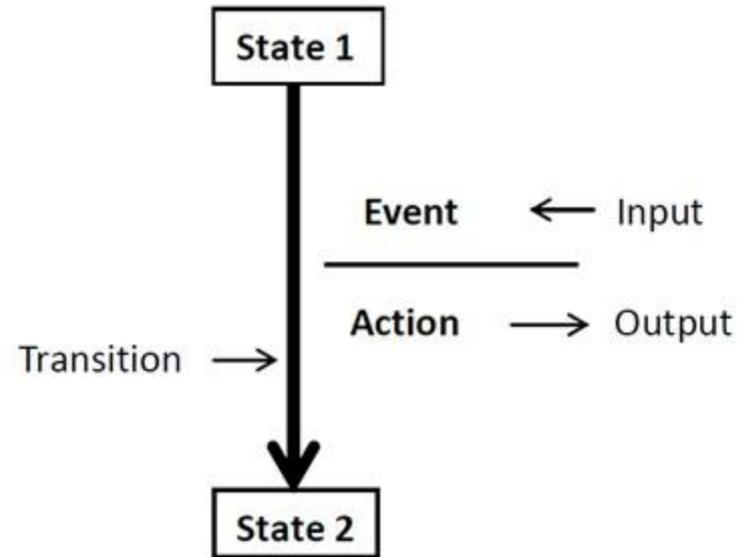
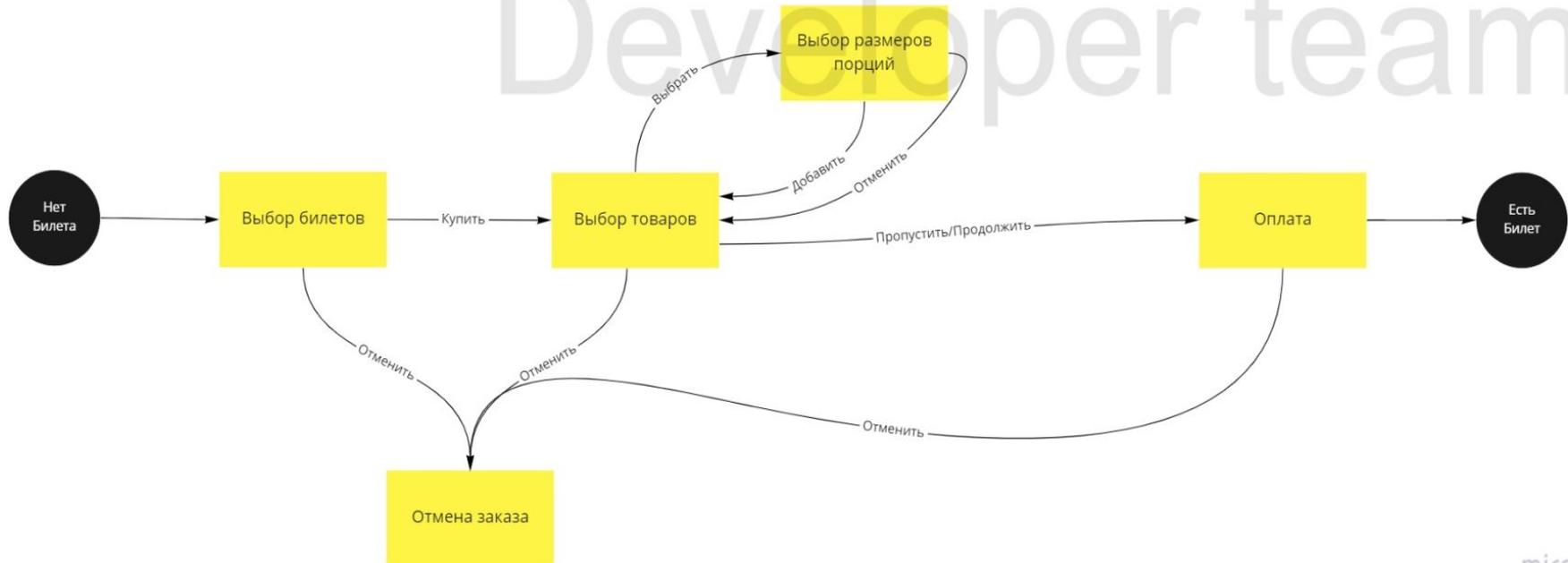


Figure 3: State Transition Diagram notation

# Тестирование переходов между состояниями

Бронирование билетов - <https://kinoteatr.ru/>

Developer team



## Таблица переходов

<b>Текущее состояние</b>	<b>Событие</b>	<b>Реакция</b>	<b>Новое состояние</b>
выбор билетов	заполнить	валидация формы	подтверждение
выбор билетов	отменить	удаление информации о заказе	отмена заказа
...			
оплата	оплатить	денежный перевод	билет

# Стратегии выбора тестов

- использованы все состояния
- произошли все события
- пройдены пути от входы к выходу
- заполнены все переходы

дополнительно

- невалидные переходы
- нестандартные переходы и последовательности



# Тестирование переходов между состояниями

## Преимущества

- Моделирование улучшает понимание системы
- Хорошо демонстрирует пробелы в покрытии тестами, так как на схеме наглядно отображаются непроверенные переходы и состояния системы

## Ограничения

- Требуется больше времени и навыков на составление схемы
- Для больших и сложных по логике функциональностей потребуется много схем
- Тяжело поддерживать в актуальном состоянии

# Проектирование тестовых данных



# Классы эквивалентности и граничные значения

**Класс эквивалентности** - это набор данных, обработка которых приводит к одному и тому же результату.

**Эквивалентное разбиение** - техника тест-дизайна, при которой входные значения разделяются на классы, эквивалентные по воздействию на систему.

- для тестирования достаточно выбрать по одному представителю для каждого класса. Потому что, если один тест ловит ошибку, то и остальные тесты, содержащие данные из этого класса, ее обнаружат.

# Количество классов

слишком много классов - не уменьшаем время на тестирования

слишком мало классов - увеличиваем риск пропустить ошибку

# Как разбить на классы эквивалентности

- Если область определения параметра - диапазон, берется три класса эквивалентности: невалидный промежуток слева от диапазона, сам диапазон, невалидный промежуток справа от диапазона
- Если область определения - неупорядоченный набор данных, то берется два основных класса - валидные и невалидные значения, а дальше полученное разбиение можно дробить на более мелкие классы.

# Классы эквивалентности. Пример

Подсчет комиссии при отмене бронирования авиабилетов.

Комиссия зависит от времени до вылета, в которое совершена отмена:

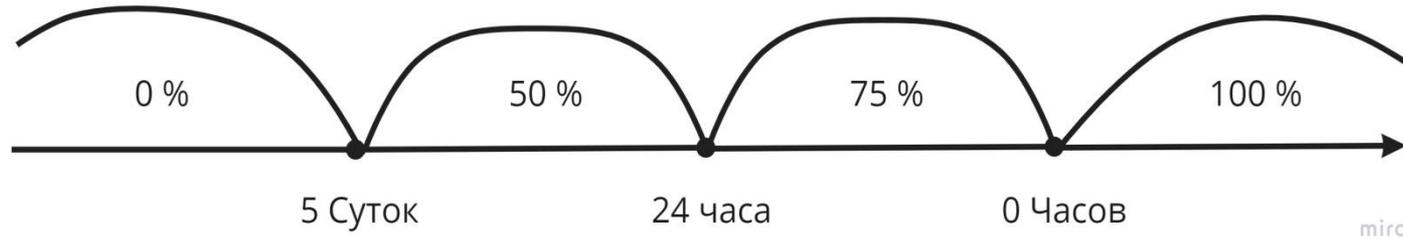
- За 5 суток до вылета комиссия составляет 0%
- Меньше 5 суток, но больше 24 часов – 50%
- Меньше 24 часов, но до вылета – 75%
- После вылета – 100%



# Классы эквивалентности

Выделяем классы:

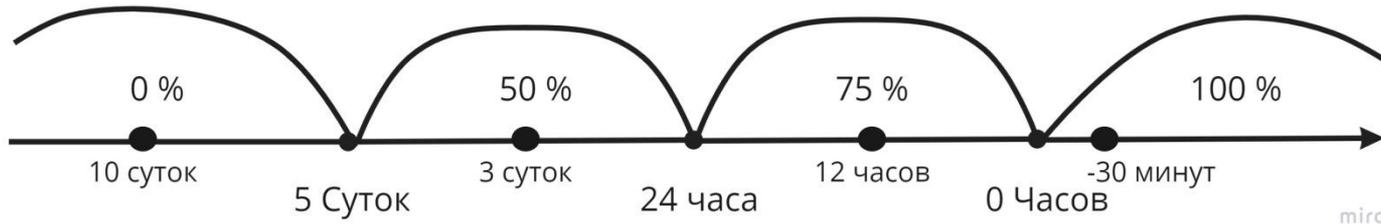
- 1 класс: время до вылета  $> 5$  суток
- 2 класс:  $5 \text{ суток} \geq \text{время до вылета} > 24 \text{ часа}$
- 3 класс:  $24 \text{ часа} \geq \text{время до вылета} > 0 \text{ часов}$
- 4 класс: время до вылета  $\leq 0$  часов (вылет уже состоялся)



# Классы эквивалентности

Выберем представителя от каждого класса:

- время до вылета = 10 суток (тест из 1-го класса)
- время до вылета = 3 суток (тест из 2-го класса)
- время до вылета = 12 часов (тест из 3-го класса)
- время до вылета = -30 мин (тест из 4-го класса)

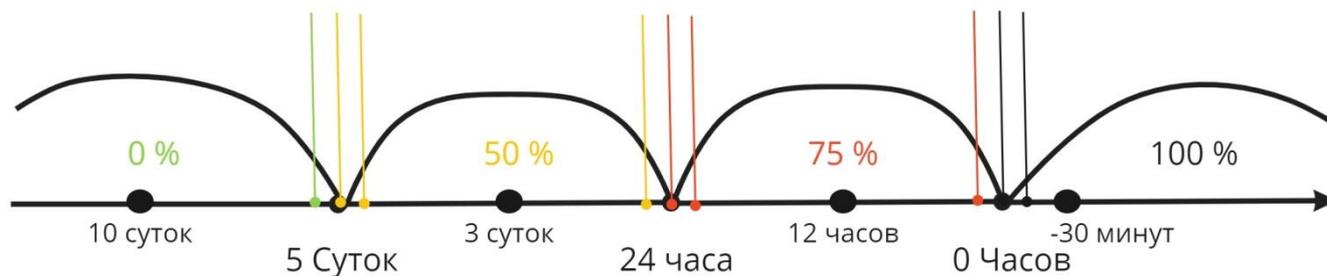


# Классы эквивалентности. Анализ граничных значений

Протестируем значения на границах:

- Отменим бронь за 5 суток + 1 секунду до вылета - комиссия 0%.
- Отменим бронь ровно за 5 суток до вылета - комиссия 50%.
- Отменим бронь за 5 суток – 1 секунду до вылета - комиссия 50%.
  
- Отменим бронь за 24 часа + 1 секунду до вылета - комиссия 50%.
- Отменим бронь ровно за 24 часа до вылета – комиссия 75%
- Отменим бронь за 24 часа - 1 секунду до вылета - комиссия 75%.
  
- Отменим бронь за 1 секунду до - комиссия 75%.
- Отменим бронь ровно во время вылета - комиссия 100%.
- Отменим бронь спустя 1 секунду после вылета - комиссия 100%.

# Классы эквивалентности. Анализ граничных значений



# Классы эквивалентности и граничные значения

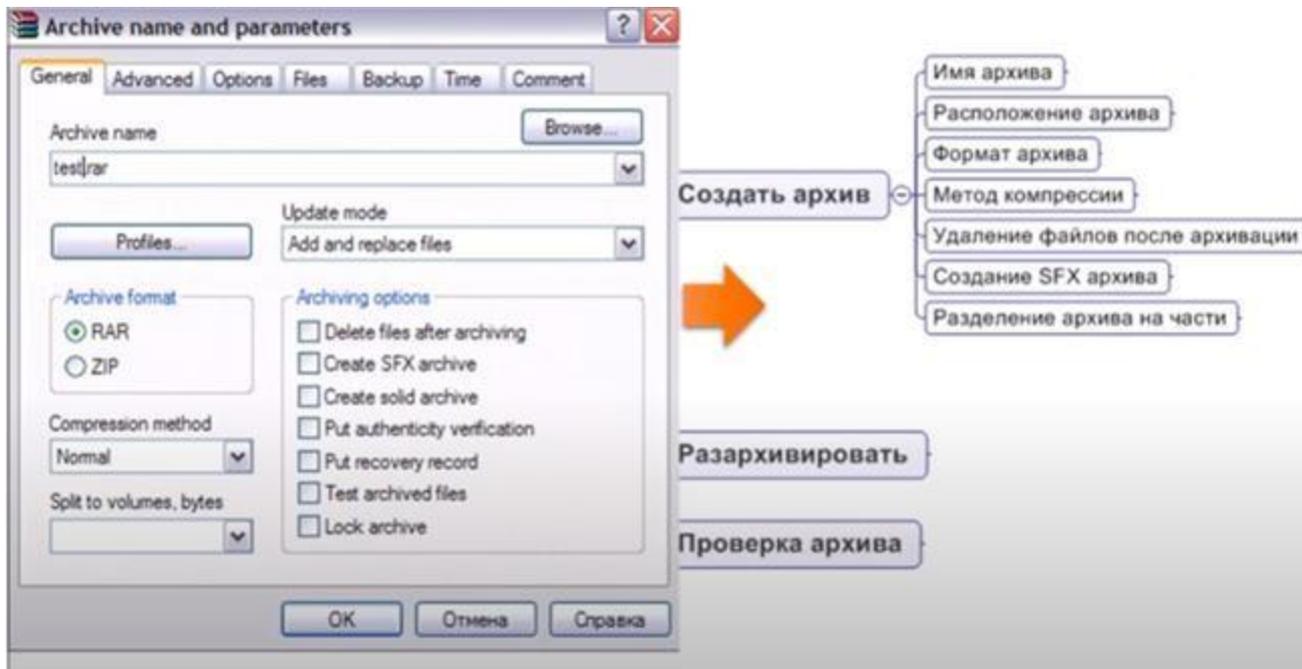
## Преимущества

- меньше тестов
- хорошее покрытие

## Ограничения

- Если группы немногочисленны или сфера связана с риском для жизни людей -> лучше полный перебор

# Тестовая комбинаторика



# Тестовая комбинаторика

Длина имени	Символы	Расположение	Файловая система	Размер файла
0	Английские	Локально	NTFS	0
1	Русские	Сеть	FAT32	1 байт
255	<b>Спец. символы</b>	USB		2 Гб
<b>256</b>		<b>CD-RW</b>		<b>2,01 Гб</b>

Позитивные и *исследовательские* значения параметров можно комбинировать

Негативные проверки: **всегда отдельно**

оптимизация набора за счет классов эквивалентности и границ значений

# Полный перебор значений

Длина имени	Символы	Расположение	Файловая система	Размер файла
1	Английские	Локально	NTFS	1 байт
255	Английские	Локально	NTFS	1 байт
1	Русские	Локально	NTFS	1 байт
255	Русские	Локально	NTFS	1 байт
...	...	...	...	...

# Полный перебор значений

Максимальное покрытие

Много времени на прохождение всех тестов

Много времени на поддержку всех тестов

Количество тестов (перемножение количества значений всех параметров + негативные проверки)  $2*3*4*2*2 + 4 = 100$

## Минимальное количество проверок

Длина имени	Символы	Расположение	Файловая система	Размер файла
1	Английские	Локально	NTFS	1 байт
255	Русские	Сеть	FAT32	2 Гб
1	Спец. Символы	USB	NTFS	1 байт
255	Английские	CD-RW	FAT32	2 Гб

# Минимальное количество проверок

- минимум проверок: количество тестов = число вариантов у параметра с максимальным значением + негативные проверки =  $4 + 4 = 8$

Преимущества:

- быстро

Недостатки:

- сложно локализовывать ошибки

# Атомарные проверки

Длина имени	Символы	Расположение	Файловая система	Размер файла
1	Английские	Локально	NTFS	1 байт
255	Английские	Локально	NTFS	1 байт
1	<b>Русские</b>	Локально	NTFS	1 байт
1	Английские	<b>Сеть</b>	NTFS	1 байт
...	...	...	...	...

# Атомарные проверки

- легко локализовать
- количество тестов = сумма значений - сумма параметров +1+негативные проверки = 13

## **Риски качества**

Создание архива может не работать

Создание архива с длинным именем может не работать

Создание архива с длинным именем может не работать при сохранении на сетевой диск

# Попарное тестирование (pairwise)

метод генерации тестовых данных, основанный на том, что большинство дефектов появляются в результате взаимодействия двух факторов



# Попарное тестирование (pairwise)

Длина имени	Символы	Расположение	Файловая система	Размер файла	Пар
1	Английские	Локально	NTFS	1 байт	10
255	Русские	Локально	FAT32	2 Гб	10
255	Английские	Сеть	NTFS	2 Гб	8
1	Русские	Сеть	FAT32	1 байт	8
...	...	...	...	...	...

Перебор всех пар (~36)

Сложно локализуемое

Минимальные проверки – попарное тестирование – перемножение двух максимальных наборов значений

# Попарное тестирование. Инструменты

PICT

<https://github.com/Microsoft/pict/blob/master/doc/pict.md>

allpairs

<http://www.satisfice.com/tools.shtml>

PairwiseTool

<https://pairwise.teremokgames.com/>

# Хороший тестовый набор

1. Базовые проверки -> более сложные тесты -> тесты на исключительные ситуации
2. Не содержит дублирующих тестов
3. Гибок для модификации

# Итоги



Процесс заведения заявки на обслуживание

Техники тест-дизайна 1-го уровня

Техники тест-дизайна 2-го уровня

Техники работают вместе

## Домашнее задание. Срок 27.09.22

1. Проведите анализ формы отправки письма почтового клиента gmail.com и оформите результат исследования в виде майнд-карты или в табличном представлении.

P.S. Если захотите проанализировать что-то другое, а не форму отправки письма в gmail – Вэлком!

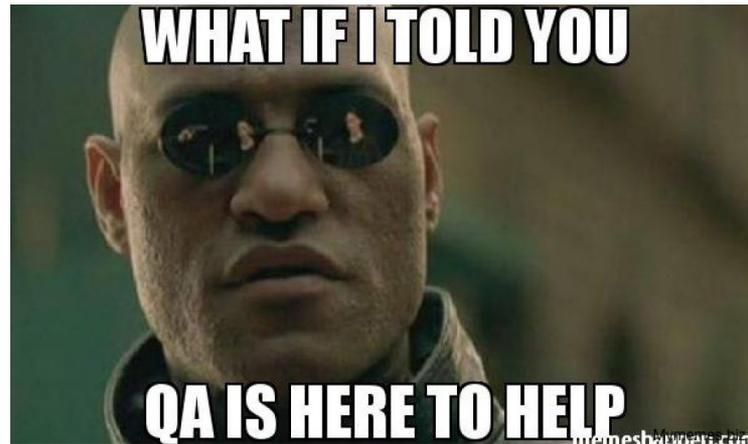
2. Выберите 3 наиболее важных параметра и для них подробно опишите все значения, необходимые для тестирования, воспользовавшись техниками генерации тестовых данных, изученных на лекции. бонусом будет, если воспользуетесь техникой pairwise.

Для остальных параметров значения можно описать по желанию

3. Объясните, почему выбранные параметры вы посчитали наиболее важными.



**Спасибо за внимание!**



# Итоги

## Техники

- **Цепочки действий**
  - сценарии использования,
  - диаграммы переходов между состояниями
  
- **Данные**
  - классы эквивалентности и граничные значения
  - полный перебор и попарное тестирование