

# Лекция 8

{ Тема }

**ТЕХНИКИ  
ТЕСТ-ДИЗАЙНА,  
ОСНОВАННЫЕ  
НА СПЕЦИФИКАЦИИ**



# Test design techniques

ПЛАН  
ЛЕКЦИИ



1

Тестирование по сценариям использования

2

Тестирование по пользовательским историям

Комбинаторные виды тестирования

3

Доменный анализ



# Use case testing



Тестирование  
по сценариям  
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).

Каждый сценарий использования **имеет предусловия**, необходимые для успешного выполнения **сценария**. Каждый сценарий использования завершается **постусловиями**, являющимися наблюдаемыми результатами и итоговым состоянием системы после выполнения сценария.

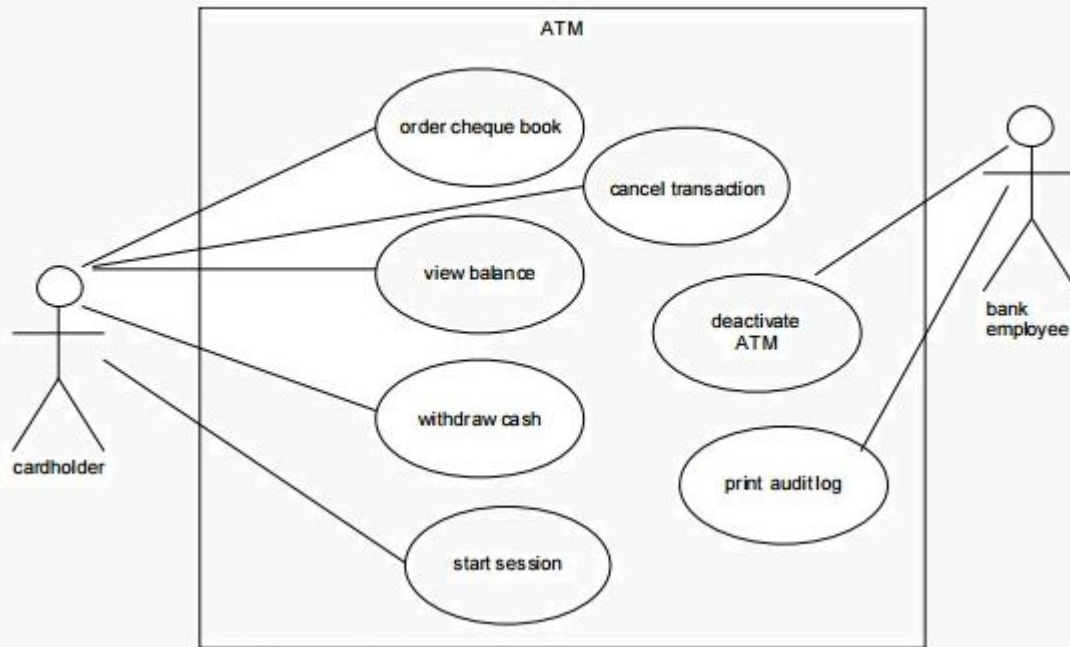
Сценарий использования **обычно имеет основной (наиболее вероятный) сценарий и альтернативные сценарии**. Сценарии использования описываются как «потоки процессов» в системе, основанные на типовом предполагаемом использовании.



# Use case testing



Тестирование  
по сценариям  
использования



# Use case testing

Тестирование  
по сценариям  
использования

<b>Name</b>	The Use Case name. Typically the name is of the format <action> + <object>.
<b>ID</b>	An identifier that is unique to each Use Case.
<b>Description</b>	A brief sentence that states what the user wants to be able to do and what benefit he will derive.
<b>Actors</b>	The type of user who interacts with the system to accomplish the task. Actors are identified by role name.
<b>Organizational Benefits</b>	The value the organization expects to receive from having the functionality described. Ideally this is a link directly to a Business Objective.
<b>Frequency of Use</b>	How often the Use Case is executed.
<b>Triggers</b>	Concrete actions made by the user within the system to start the Use Case.
<b>Preconditions</b>	Any states that the system must be in or conditions that must be met before the Use Case is started.
<b>Postconditions</b>	Any states that the system must be in or conditions that must be met after the Use Case is completed successfully. These will be met if the Main Course or any Alternate Courses are followed. Some Exceptions may result in failure to meet the Postconditions.
<b>Main Course</b>	The most common path of interactions between the user and the system. 1. Step 1 2. Step 2
<b>Alternate Courses</b>	Alternate paths through the system. AC1: <condition for the alternate to be called> 1. Step 1 2. Step 2  AC2: <condition for the alternate to be called> 1. Step 1
<b>Exceptions</b>	Exception handling by the system. EX1: <condition for the exception to be called> 1. Step 1 2. Step 2  EX2 <condition for the exception to be called> 1. Step 1

# Use case testing



Тестирование  
по сценариям  
использования

Тесты, построенные на основе сценариев использования, являются наиболее полезными для выявления дефектов в потоках процессов во время реального использования системы.

Сценарии использования очень **полезны для разработки приёмочных тестов** с участием заказчика или пользователей.

Также они могут выявить дефекты интеграции, вызванные взаимодействием различных компонентов, не выявляемые индивидуальным тестированием компонентов.

Проектирование тестовых сценариев на основе сценариев использования может быть объединено с другими методами, основанными на спецификации.

**Покрытие считается 100%, если тестами покрыты основной путь use case-а и все альтернативные пути.**



# Practice 1

<b>ID</b>	12223.33
<b>Название</b>	Покупка на коммерческом сайте
<b>Действующее лицо</b>	Покупатель
<b>Краткое описание</b>	Позволить покупателю оформить заказ после оплаты товара, добавленного в Корзину
<b>Приоритет</b>	Очень высокий
<b>Частота использования</b>	25% пользователей сайта, до 1000 покупателей каждый день
<b>Предусловия</b>	1) Наличие одного или нескольких товаров в корзине; 2) пользователь авторизован; 3) покупатель нажал на кнопку Оформить Заказ
<b>Основной сценарий</b>	1. Система собирает информацию об адресе, способе оплаты и способе доставки. 2. Система отображает всю информацию для подтверждения. 3. Покупатель подтверждает заказ и производит оплату.
<b>Исключение 1</b>	Покупатель пытается оформить заказ с пустой Корзиной. Система показывает сообщение
<b>Исключение 2</b>	Покупатель предоставляет неправильную информацию об адресе, способе доставки или оплаты. Система показывает сообщение
<b>Исключение 3</b>	Покупатель отказывается от оплаты во время подтверждения заказа. Система удаляет информацию о заказе
<b>Постусловия</b>	Статус заказа в системе "Подтвержден"

## Задание

Для use case из примера написать тесты, необходимые для достижения 100%-го покрытия.

# User story testing



Тестирование  
по пользова-  
теским  
историям

В некоторых гибких методологиях разработки ПО спецификации готовятся в виде пользовательских историй (User Stories).

**Пользовательская история** описывает часть функциональности (или полностью всю), которую можно спроектировать, реализовать, протестировать и продемонстрировать за одну итерацию.

Эти пользовательские истории описывают как функциональные, так и не функциональные требования, а также включают описание приемочных критериев.

**На основании приемочных критериев и принимается решение о готовности реализованной функциональности.**





# User story testing



Тестирование  
по пользова-  
теским  
историям



Для **минимального покрытия user story** необходимо, чтоб были проверены все критерии приемки.

При тестировании пользовательских историй чаще всего находятся различные проблемы функционального характера.

Также могут быть обнаружены проблемы интеграции реализованной функциональности с уже существующей.

Нередко могут быть обнаружены проблемы с производительностью, пользовательским интерфейсом и обработкой ошибок.



# User story testing

Тестирование  
по пользова-  
теским  
историям

## User Story



**As a** <role>  
**I want** <goal>  
**so that** <benefit>

**Acceptance criteria:**  
(Conditions of Satisfaction)

...

...

**As an** Account Manager  
**I want** a sales report of my account  
to be sent to my inbox daily  
**So that** I can monitor the sales  
progress of my customer portfolio

### Acceptance criteria:

1. The report is sent daily to my inbox
2. The report contains the following sales details: ...
3. The report is in csv format.

# User story testing

Тестирование  
по пользова-  
теским  
историям

The screenshot shows a Jira user story card with the following details:

- Assignee:** bryce
- Due-date:** Next Friday
- Status:** Dev Board - In Progress
- Estimate:** 9 points

**User Story:** As a writer, I want to receive notifications when others add comments so that I am up to date

**Description:**

**Acceptance Criteria:**

- Given** I don't have app open **when** my phone is locked **then** I should receive a banner notification.
- Given** I have app open **when** I am writing on the doc **then** the bell icon should update to show unread notification with count.
- Given** a user was mentioned in a comment using @ mention **when** the second user is reading other comments **then** a flash message should show up on the same comment thread with a message notifying about the new comment

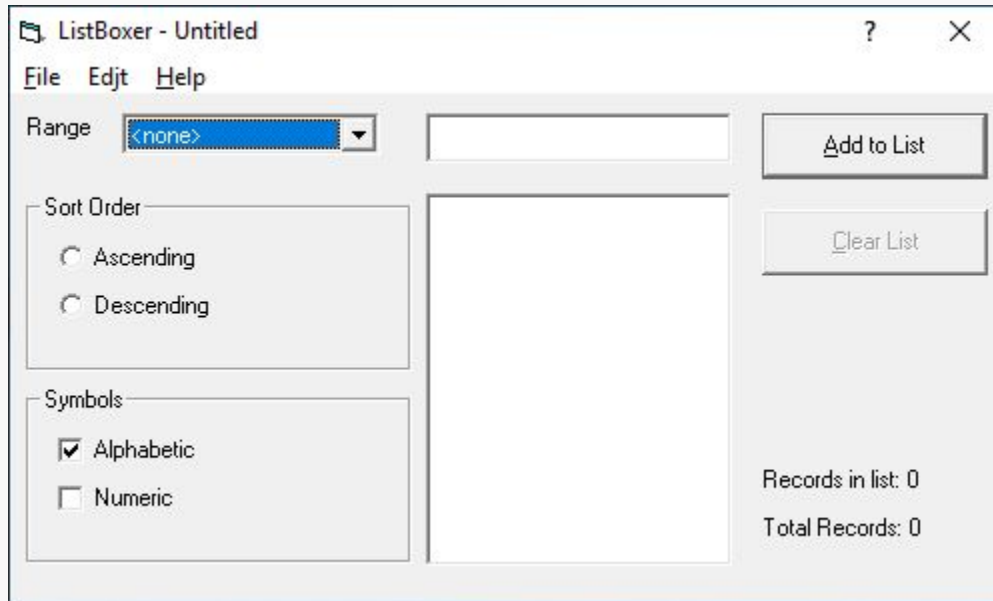
**Attachments:**

- notification\_mockup.png (330.34 KB) by Larson Paddy

**Comments:**

- Larson Paddy** (4 minutes ago): I've attached a mockup of notification @bryce. Is there anything I missed in the mockup?
- Bruce Wayne** (2 minutes ago): Thanks for this, @paddy. This looks neat.

# Practice 2



## Задание

Придумать user story для программы ListBoxer и написать acceptance criteria для этой пользовательской истории.



# Combinatorial techniques



Комбинаторные  
виды  
тестирования

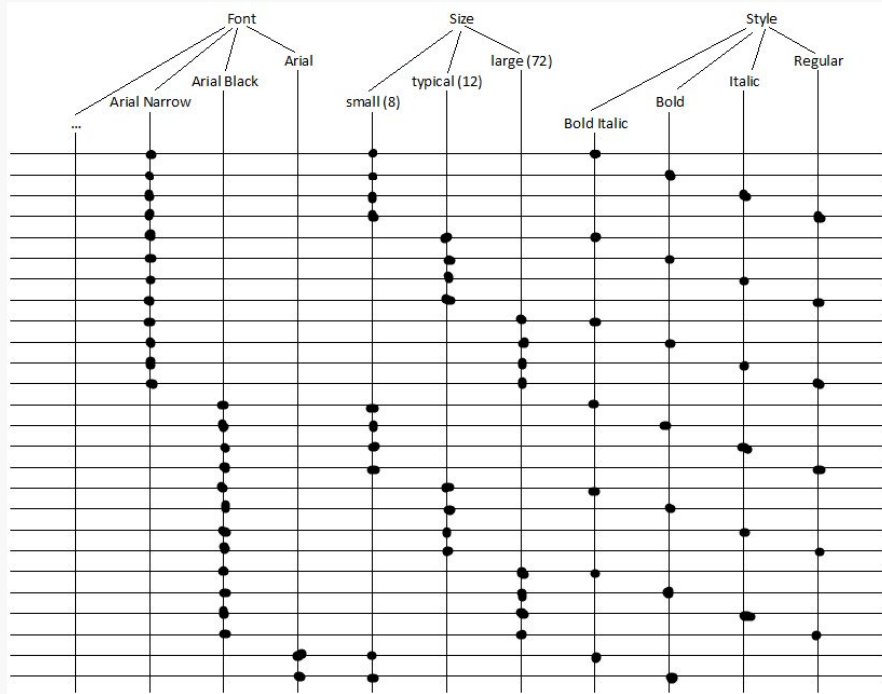


**Комбинаторные виды тестирования** используются, когда в тестируемой системе есть несколько параметров, каждый из которых имеет свои значения, и различные комбинации этих значений приводят (или могут привести) к различным результатам.

Для наглядного отображения комбинаций может быть использована **техника построения дерева классификации** (подробней [ТУТ](#)).



# Combinatorial techniques



Для системы, позволяющей выбирать Шрифт, Стил и Размер, дерево классификаций может выглядеть, как на картинке слева (это часть дерева).



# Combinatorial techniques



Комбинаторные  
виды  
тестирования



**Покрытие имеет несколько уровней.**

Если в тестах участвуют  $N$  параметров, то можно говорить о  **$N$ -wise покрытии**. Это когда все возможные комбинации из  $N$  параметров покрыты тестами.

Самый распространенный уровень покрытия - это **2-wise (pairwise)**. Это когда все возможные пары параметров покрыты тестами. Составление тестов для достижения 2-wise покрытия также называется попарное тестирование (**pairwise testing**).

Есть еще 1-wise покрыти (**singleton**). Это когда каждый параметр встречается в тестах хотя бы один раз.



# Practice 3



## Задание



Страховая компания предлагает полисы для жилья. Их есть несколько вариантов. Они зависят от типа здания, материала, из которого оно построено и от его расположения.

Тип здания:

- Частный дом
- Дом на 2 хозяина
- Многоквартирный дом
- Коттедж

Материалы:

- Дерево
- Бетон
- Кирпич
- Смешанный

Расположение:

- Город
- Пригород
- Сельская местность
- Пустыня

**Надо подготовить тестовые данные для 100%-го pairwise покрытия.**





# Practice 3

	<b>Тип здания</b>	<b>Материалы</b>	<b>Расположение</b>
1	Частный дом	Дерево	Город
2	Частный дом	Бетон	Пригород
3	Частный дом	Кирпич	Сельская местность
4	Частный дом	Смешанный	Пустыня
5	Дом на 2 хозяина	Бетон	Сельская местность
6	Дом на 2 хозяина	Кирпич	Пустыня
7	Дом на 2 хозяина	Смешанный	Город
8	Дом на 2 хозяина	Дерево	Пригород
9	Многоквартирный дом	Кирпич	Город
10	Многоквартирный дом	Смешанный	Пригород
11	Многоквартирный дом	Дерево	Сельская местность
12	Многоквартирный дом	Бетон	Пустыня
13	Коттедж	Смешанный	Сельская местность
14	Коттедж	Дерево	Пустыня
15	Коттедж	Бетон	Город
16	Коттедж	Кирпич	Пригород

Всего должно получиться 16 вариантов (тестов).

Их можно составлять вручную, а можно при помощи [инструмента](#).

# Domain analysis



Доменный  
анализ



**Домен** - это определенный набор значений.

Набор может быть определен как диапазон значений одной переменной (одномерный домен, например, «мужчины в возрасте от 24 до 66 лет»), так и диапазоны значений взаимодействующих переменных (многомерный домен, например, «мужчины от 24 до 66 лет, а также весом от 69 до 90 кг»).

Когда домен одномерный, то применяются техники разбиения на классы эквивалентности и анализ граничных значений.

Когда домен двухмерный, то берутся значения за пределами домена, на границах домена и внутри домена.

Рассмотрим на примере.



# Practice 4

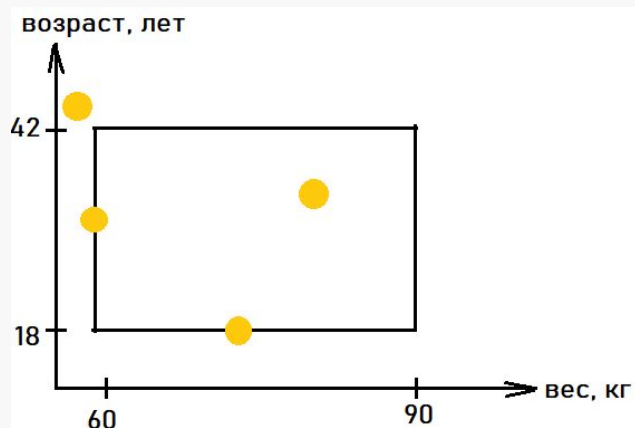


## Задание



В существующей системе для людей, ведущих здоровый образ жизни нужно сделать рассылку и пригласить на марафон мужчин от 18 до 42 лет, весом от 60 до 90 кг.

Вот наш домен:



Для проверки правильности рассылки нужно взять значения, отмеченные желтым:

возраст = 50, вес = 45

возраст = 30, вес = 60

возраст = 18, вес = 70

возраст = 30, вес = 70

Эти значения нужно использовать для проверки правильности рассылки.

# Practice 5



Задание



**Задание**

Система отслеживает параметры влажности и температуры в помещении.

Если влажность находится в пределах от 50% до 70% и температура от 16 до 22 градусов, то система выдает, что параметры оптимальны.

Если температура и\или влажность выходят за эти пределы, то слышен сигнал и выдается сообщение, что параметры вышли за оптимальные пределы.

Нужно подготовить тестовые данные для этой системы, используя доменный анализ.



# Home Task

## 1

Часть 1

Применить разбиение на классы эквивалентности, анализ граничных значений, тестирование таблицы решений и тестирование таблицы переходов для системы на следующем слайде.

Часть 2

Написать 2 use case для сайта deveducation и покрыть их тестами (для существующей функциональности).

Часть 3

Написать 2 user story для сайта deveducation и описать приемочные критерии для них (для существующей функциональности).

## 2

Часть 4

Предположим, что сайт deveducation должен работать на мобильных телефонах и планшетах (iOS, Android), а также на ПК (Windows OS), в браузерах: Safari, Chrome, Firefox, IE.

Нужно провести конфигурационное тестирование. Используйте pairwise testing и опишите все возможные конфигурации.

(Обратите внимание, что не все пары возможны в принципе. Например, нет ПК с iOS и Android.)



# Система для части 1



## Задание



Это система переводов. Пользователи могут получать или отправлять переводы.

Чтобы получить перевод, пользователь должен в точности указать Amount, Name, Address, которые отправитель указывал при отправлении этого перевода, а также должен указать верное значение в MTCN (это секретный код, отосланный получателю, после того, как перевод прошел успешно).



# Система для части 1

## Задание

