Мануал техник тест-дизайна   
  
Техники тест-дизайна - это методы и подходы, которые используются для создания тестовых случаев и наборов данных для тестирования программного обеспечения и других систем.  
  
Основная цель техник тест-дизайна - это создание тестовых случаев, которые обеспечивают максимальное покрытие функциональности и поведения тестируемой системы, при минимальных затратах на время и ресурсы.

# Определение основных техник

## Причина и следствие (Cause-Effect):

Эта техника предполагает идентификацию потенциальных причин (факторов) и их влияния на систему, а затем создание тест-кейсов для проверки этих причин и соответствующих следствий. Например, для использования виджетов Google Analytics, необходимо настроить интеграцию и заполнить аккаунт данными.  
  
Примерный алгоритм использования техники:   
  
1. Выделяем причины и следствия в спецификациях.   
2. Связываем причины и следствия.   
3. Учитываем «невозможные» сочетания причин и следствий.   
4. Составляем «таблицу решений», где в каждом столбце указана комбинация входов и выходов, т.е. каждый столбец – это готовый тестовый сценарий.   
5. Расставляем приоритеты.

## Таблица принятия решений (Decision Table):

Эта техника используется для моделирования различных входных параметров и их комбинаций с ожидаемыми результатами. Составляется таблица (матрица) с возможными условиями и вариантами реализации условий. Примером может быть удовлетворение необходимых условий для использования проекта ( первый логин, пароль, соглашение, смена пароля...).

## Попарное тестирование (Pairwise Testing):

Эта техника позволяет создавать тест-кейсы, которые проверяют все возможные комбинации пар входных данных, при этом минимизируя количество тестовых случаев.   
Суть этого метода, также известного как pairwise testing, в том, что каждое значение каждого проверяемого параметра должно быть протестировано на взаимодействие с каждым значением всех остальных параметров. После составления такой матрицы мы убираем тесты, которые дублируют друг друга, оставляя максимальное покрытие при минимальном необходимом наборе сценариев.   
Попарное тестирование позволяет обнаружить максимум ошибок без избыточных проверок.   
Например, тестирование опций Price strategy, Динамических тегов тестирование Filters.  
Ресурсы:   
<https://pairwise.teremokgames.com/>  
<https://inductive.no/pairwiser-tool/>  
<https://pairwise.yuuniworks.com/>

## Классы эквивалентности (Equivalence Partitioning):

Эта техника предполагает разделение диапазона входных данных на классы эквивалентности, где каждый класс представляет группу данных, которые должны вести себя одинаково.  
Метод эквивалентного разбиения позволяет минимизировать число тестов, не создавая сценарий для каждого возможного значения, а выбрав только одно значение из целого класса и приняв за аксиому, что для всех значений в данной группе результат будет аналогичным.  
 Например, при тестировании полей ввода : возможно создать классы эквивалентности для валидных, невалидных (пустых) значений, используя минимальное-максимальное значение, тип ввода данных, количество символов.

## Анализ граничных значений (Boundary Value Analysis):

Эта техника подразумевает тестирование значений, находящихся на границах диапазонов входных данных, так как ошибки часто возникают именно в этих точках.   
Техника граничных значений основана на предположении, что большинство ошибок может возникнуть на границах эквивалентных классов. Часто эти техники применяются вместе для тестирования.  
Например, тестирование Price position и Gross Margin градусников, значений Red, Green, Orange, Yellow на табличном виджете ДС, тестирование значений min, max, deviate from.

## Схема состояний и переходов (State & Transition Diagram):

Эта техника применяется для моделирования состояний системы и переходов между ними. Например, для тестирования системы добавления/ удаления матчей (рекомендаций), тестирование действий что обрабатываются очередью, запускается триггером.  
Ресурсы:   
<https://app.creately.com/>  
<https://staruml.io/>  
<https://www.drawio.com/>

## Предугадывание ошибок (Error Guessing):

Эта техника основывается на опыте и интуиции тестировщика. Тестировщик предполагает, где могут возникнуть ошибки, и создает тест-кейсы для проверки этих предполагаемых ошибок. Например, если вы видите, система обычно дает сбой при большой нагрузке, вы можете создать тесты, чтобы проверить его стабильность в таких условиях.

## Тестирование сценариев использования (Use Case Testing):

Этот подход основывается на том, что для каждого сценария использования системы, существует набор действий и реакций. На примере проекта можно протестировать доступность опций для каждой роли пользователя (User, Admin, RA, User only reader), статусы аккаунта.

# Функциональности и примеры использования техник тест дизайна

## Login with Use Two Factor Authorization

*Таблица принятия решений*

| Условие | Значение | №1 | №2 | №3 | №4 |
| --- | --- | --- | --- | --- | --- |
| Email | 1,0 | 1 | 0 | 1,0 | 1 |
| Password | 1,0 | 1 | 1,0 | 0 | 1 |
| Verification | 1, 0, не пришел | 1 | 1, 0, не пришел | 1, 0, не пришел | 0 |
| Результат | Пользователь не авторизован | Пользователь вошел на сайт | Пользователь не авторизован | Пользователь не авторизован | Пользователь не авторизован |

1 - верный

0 - неверный

*Классы эквивалентности*

Для активации кнопки “Login” необходимо ввести email (мин 3 символа по типу \_@\_, макс 255, varchar, ,без пробелов) и пароль (мин 1, макс 255 varchar).

Для успешной авторизации, необходимо, чтоб введенные email и password совпадали с учетной записью не удаленного пользователя.  
Пароль имел минимум 6 символов.

1. **Допустимый email и пароль**: Валидный email и пароль, соответствующие условиям (минимум 3 символа для email, минимум 1 символов для пароля). Появляется кнопка “Login”. -> The password must be at least 6 characters.
2. **Допустимый email и пустой пароль**: Валидный email и пустой пароль. Кнопка “Login” неактивна.
3. **Пустой email и допустимый пароль:** Пустой email и валидный пароль. Кнопка “Login” неактивна.
4. **Допустимый email и пароль, несоответствующий учетной записи:** Валидный email и пароль, которые не совпадают с данными учетной записи неудаленного пользователя. Появляется кнопка “Login”. -> These credentials do not match our records.
5. **Недопустимый email и допустимый пароль:** Email, который не соответствует формату (менее 3 символов или более 255 символов) и валидный пароль.
6. **Валидный email и пароль, соответствующий удаленной учетной записи**: Валидный email и пароль, которые совпадают с данными удаленной учетной записи.  
   Пользователь авторизован.

## 2. Settings -> Gross margin thermometer *Анализ граничных значений*

1. Необходимо выяснить шаг - 0,01.2. Какие значения границ входят в какой параметр.  
3. Значения внутри класса должны иметь один параметр.  
4. Значения идут слева направо от большего к меньшему и не могут пересекаться.

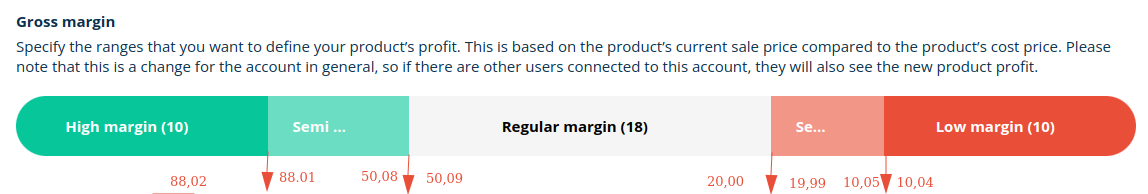
"High margin"- >88,01%

"Semi high margin"- <=88,01% and >50,09%

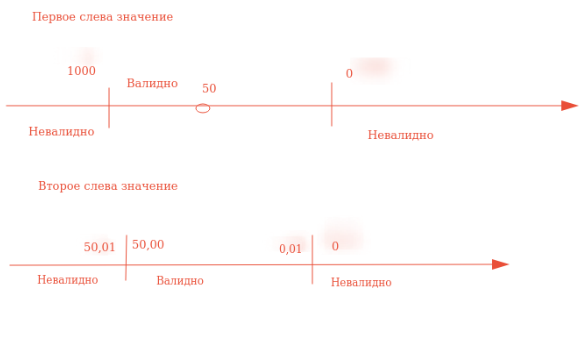
"Regular margin"- <=50,09% and >=20,00%

"Semi low margin"- <20,00% and >=10,05%

"Low margin"- < 10,05%

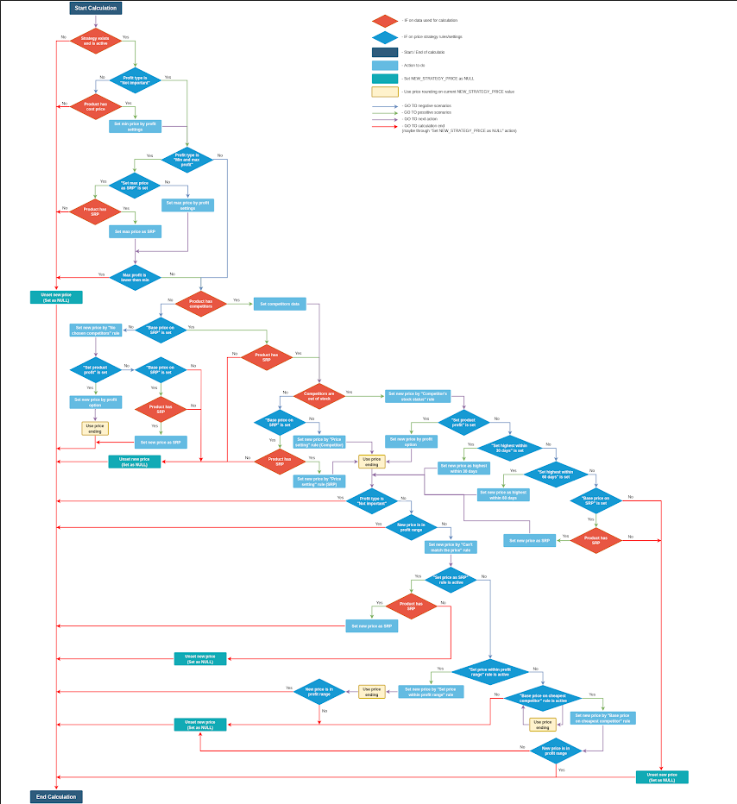


*Классы эквивалентности*

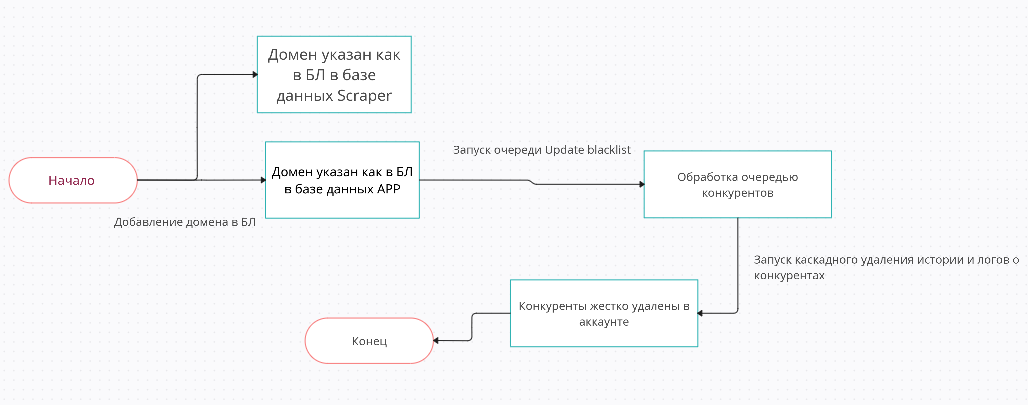
1. Для значений есть границы валидности от 0 до 1000, не могут быть отрицательными, пустым значением.  
2.Также граничное значение Слева не может быть больше граничного значения справа*.*  


## 

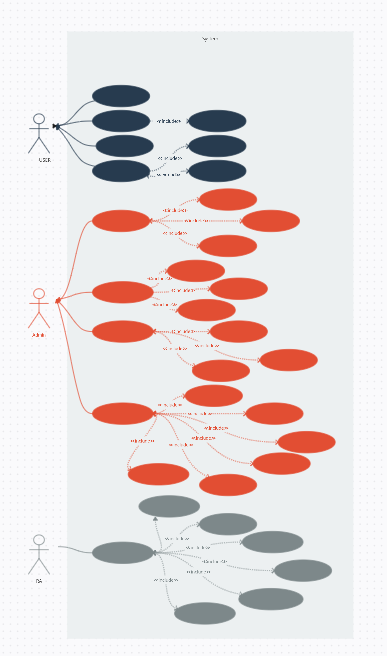
## Назначение New Strategy price *Схема состояний и переходов*



## Добавление домена в блеклист *Схема состояний и переходов*



## Схема USE CASE testing

Например, можно выделить определенный функционал и описать функции, которые доступны пользователям с разными ролями (USER, ADMIN, RA).  
  


# Чек-лист Логин-Логаут(шаблон)

## 1.Login

1.1. Логин в пользователя с ролью юзер, валидные данные. Two-factor OFF.

1.2. Логин в пользователя с ролью админ, валидные данные. Two-factor OFF.

1.3. Логин в пользователя с ролью юзер, невалидные данные.

1.4. Логин в пользователя с ролью админ, невалидные данные.

1.5. Логин в пользователя с ролью юзер, валидные данные. “Remember me”. Two-factor On.

1.6. Логин в пользователя с ролью админ, валидные данные. “Remember me”. Two-factor On.

1.7 Логин в пользователя с ролью юзер, валидные данные. “Forgot your password”.

1.8. Логин в пользователя с ролью админ, валидные данные. “Forgot your password”.

1.9. Первый логин в пользователя с ролью юзер, валидные данные.

1.10. Первый логин в пользователя с ролью админ, валидные данные.

1.1. Логин в пользователя с ролью юзер, валидные данные. Дефолтный аккаунт.

1.2. Логин в пользователя с ролью админ, валидные данные. Дефолтный аккаунт.

## 2.Logout

2.1. Logout пользователя с ролью юзер.

2.2. Logout пользователя с ролью админ.

2.3. Logout пользователя. Открыто несколько вкладок.

## 3.Switch account

3.1. Switch account для User. Один прикрепленный аккаунт.

3.2. Switch account для Admin. Один прикрепленный аккаунт.

3.3 Switch account для Admin. Нет прикрепленных аккаунтов.

3.4. Switch account поиск для User.

3.5. Switch account поиск для Admin.

## 