

# Основные темы

- Общие понятия и принципы тестирования
  - Способы получения исходных данных для тестирования
  - Способы получения ожидаемого результата
-

# Общие понятия

- *Тестирование* – процесс выполнения ETL-процесса с целью выявления ошибок.
  - Шаги процесса задаются тестами (тестовыми вариантами).
  - Каждый тест определяет:
    - свой набор исходных данных и исходное состояние среды;
    - набор ожидаемых результатов работы.
-

# Уровни тестирования

- Компонентное (модульное, unit) тестирование

Тестирование отдельных модулей. Для получения максимального результата тестирование проводится одновременно с разработкой модулей.

- Интеграционное тестирование

Тестирование взаимодействия различных модулей между собой.

- Системное тестирование

Тестирование системы в целом. Данный уровень тестирования подразумевает имитацию реальной работы системы.

---

# Стратегии тестирования

- "Черный ящик" (поведенческое тестирование, функциональное тестирование)

Стратегия основанная на требованиях. Тестовые варианты проектируются исходя из предъявляемых к продукту требований. Для применения этой стратегии (в принципе, но не на практике) не обязательно знать, как тестируемый объект устроен внутри.

- "Белый ящик" ("Прозрачный ящик", структурное тестирование)

Стратегия основанная на структуре проверяемого объекта. Тестовые варианты проектируются таким образом, что бы каждый оператор выполнялся хотя бы один раз, каждая ветка алгоритма выполнялась хотя бы один раз и т.д.

- "Серый ящик" (гибридное тестирование)

Стратегия является комбинацией поведенческого и структурного тестирования.

---

# Классификация тестирования по "возрасту" функционала

- Тестирование нового функционала (на проекте BIIS называют функциональное тестирование)

Проверка нового создаваемого в рамках задачи функционала.

- Тестирование существующего функционала (регрессионное тестирование)

Проверка, что внесенные в систему или модуль изменения не затронули уже работающий функционал.

---

# Классификация тестирования по объекту проверки

- Тестирование алгоритмов преобразования данных

Проверка, что все данные преобразуются в соответствии с проектной документацией.

- Тестирование полноты данных

Проверка, что все ожидаемые данные загружаются в хранилище данных.

- Тестирование качества данных

Проверка того, как ETL-процесс обрабатывает не стандартные ситуации (например: значения NULL)

- Тестирование производительности

Проводится с целью определения скорости работы системы или её части под определенной нагрузкой.

---

# Способы получения исходных данных для теста

- Копирование порции данных с источника
- Моделирование синтетических данных

	Копирование	Моделирование
Достоинства	<ul style="list-style-type: none"><li>• Быстрый способ получения тестовых данных</li><li>• Позволяет выявить ошибки, для типов транзакций не описанных в требованиях</li></ul>	<ul style="list-style-type: none"><li>• Ускоряет процесс загрузки</li><li>• Позволяет хранить исходные данные, и следовательно повторять тесты</li><li>• Полное покрытие требований тестами</li><li>• Удобство использования дебагера</li></ul>

# Методика моделирования синтетических данных

- Для минимизации тестовых данных используются:
    - Способ разбиения по эквивалентности
    - Способ анализа граничных значений
    - Способ диаграмм причин-следствий
    - Предположение об ошибке
  - Теоретические материалы и примеры использования
    - База знаний (BIIS / Тестирование ETL-процессов / Способы получения исходных данных для теста)
-



# Пример применения №1

- Поле номер абонента (CTN)

Должно определяться по следующему алгоритму:

если subscriber\_id менее 11 символов - то выбираем subscriber\_id

если subscriber\_id 11 символов и более, то

если в начале 7, то ее отрезаем

если в начале не 7, то кладем subscriber\_id без изменений

# Анализ граничных значений

- В данном случае 11 является граничным значением.
- Рассматривая диапазон  $[11; \infty]$ , получаем 2 тестовых варианта

Исходные данные
Длина строки subscriber_id = 11 символов
Длина строки subscriber_id = 10 символов

# Метод эквивалентности

- Разбиваем значения источника на 2 класса эквивалентности

Исходные данные
Строка subscriber_id начинается с 7
Строка subscriber_id не начинается с 7

# Диаграммы причинно-следственных связей

- Причинами являются выбранные варианты исходных данных

	Причины
1	Длина строки subscriber_id = 10 символов
2	Длина строки subscriber_id = 11 символов
3	Строка subscriber_id начинается с 7
4	Строка subscriber_id не начинается с 7

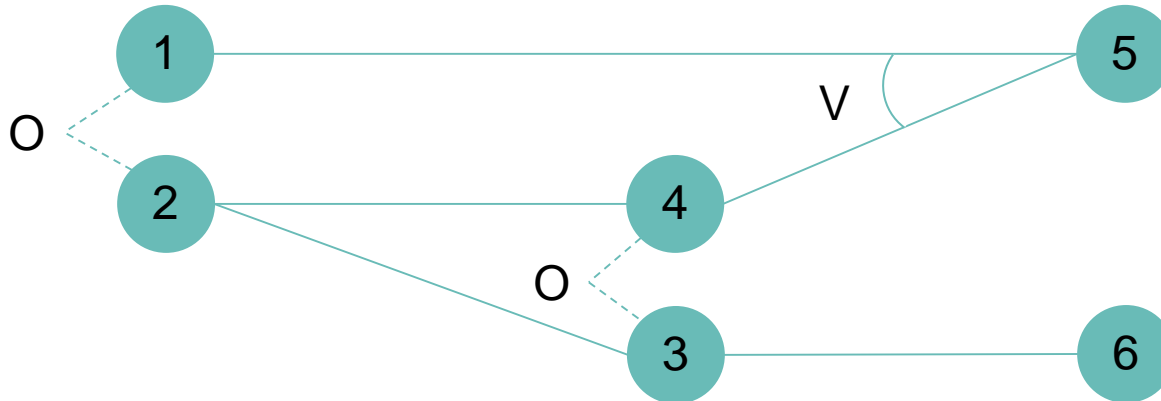
- Комбинации причин приводят к следствиям

	Следствия
5	Выбираем subscriber_id без изменений
6	Выбираем subscriber_id с отрезанной 7

---

# Граф причинно-следственных связей

- На основании выбранных причин и следствий формируем граф



- Каждая ветка графа является тестовым вариантом

Исходные данные	Ожидаемый результат
Длина строки subscriber_id = 10 символов	Выбираем subscriber_id без изменений
Длина строки subscriber_id = 11 символов и не начинается с 7	Выбираем subscriber_id без изменений
Длина строки subscriber_id = 11 символов и начинается с 7	Выбираем subscriber_id с отрезанной 7

# Пример применения №2

- **Требование**

В случае, если на источнике появляется новый регион, необходимо отправлять письмо на [biis\\_support@beeline.ru](mailto:biis_support@beeline.ru) с оповещением, что необходимо проверить -99 и проставить вручную актуальное значение.

- **Применяем способ разбиения по эквивалентности**

Варианты данных на входе разбиваем на два класса эквивалентности

Исходные данные	Ожидаемый результат
На источнике появился новый регион	Отправлено письмо с оповещением
На источнике не появился новый регион	Письмо с оповещением не отправлено

# Способы получения ожидаемого результата для теста

- **Существующие тесты**
  - **Результат выполнения похожих ранее созданных ETL-процессов.**
  - **Результат выполнения ETL-процесса до внесения изменений.**
  - **Построение прототипа (модели).**

Реализация заданных в требованиях алгоритмов простыми средствами (например SQL)
  - **Выбор простых тестовых вариантов.**

Характеризуется минимально необходимым количеством данных на входе ETL-процесса с легко прогнозируемым результатом на выходе.
  - **Анализ выходных данных.**

Иногда бывает проще проанализировать данные на выходе ETL-процесса, и оценить корректность его работы, чем заранее предсказать результат, по входным данным.
-

# Тестирование на проекте BIIS (AS IS)

- Уровень тестирования: Модульное тестирование
  - Стратегия тестирования: Гибридное тестирование
  - Тестирование нового функционала
    - Способ получения исходных данных
      - Копирование порции данных с источника
      - Моделирование синтетических данных
    - Способ получения ожидаемого результата
      - Построение прототипа (модели).
      - Выбор простых тестовых вариантов.(в случае невозможности построения прототипа)
  - Регрессионное тестирование
    - Способ получения исходных данных
      - Копирование порции данных с источника
    - Способ получения ожидаемого результата
      - Результат выполнения ETL-процесса до внесения изменений.
-



# Тестирование на проекте BIIS (TO BE)

- Уровень тестирования: Модульное тестирование
  - Стратегия тестирования: Гибридное тестирование
  - Тестирование нового функционала
    - Способ получения исходных данных
      - Моделирование синтетических данных (1 приоритет)
      - Копирование порции данных с источника (2 приоритет)
    - Способ получения ожидаемого результата
      - Выбор простых тестовых вариантов. (1 приоритет)
      - Построение прототипа (модели).
  - Регрессионное тестирование
    - Способ получения исходных данных
      - Сохраненные ранее тесты
    - Способ получения ожидаемого результата
      - Сохраненные ранее тесты
-