

Тест-дизайн и тест-аналитика
Урок 3

Попарное тестирование





Оглавление

На этом уроке	3
Как работает pairwise	3
Пример: прогноз погоды	3
PІCT и другие инструменты	8
Pairwise и негативное тестирование	9
Пример: окружение для кроссбраузерного тестирования	10
Контрольные вопросы	11
Глоссарий	12
Практическое задание	12
Дополнительные материалы	12



На этом уроке

1. Узнаем как работает pairwise, его примеры и инструменты.
2. Познакомимся с pairwise и негативным тестированием.

Как работает pairwise

ISTQB (Международная квалификационная комиссия по тестированию программного обеспечения) определяет попарное тестирование как технику тест-дизайна методом чёрного ящика. Проверки создаются так, чтобы выполнить все возможные комбинации каждой пары входных параметров.

Результат работы системы зависит от многих факторов. Часто — от комбинации разных входных параметров. Например, в системе есть фильтр из нескольких параметров, по результату сочетаний которых будет получен определённый результат. Если параметров много, на тестирование всех возможных сочетаний уйдёт много времени.

Большинство дефектов возникает при комбинации только двух параметров. Поэтому для создания оптимального покрытия проверками функционала с большим количеством входных параметров генерируются их определённые комбинации.

Тесты разрабатываются таким образом, что для каждой пары входных параметров существуют все возможные комбинации этих пар.

Pairwise не обеспечивает исчерпывающее тестирование,
но эффективен для поиска ошибок.

Пример: прогноз погоды

User story: Я как пользователь хочу узнать прогноз погоды в указанном городе с нужной степенью детализации.



Прогноз погоды

Город	▼
RU	US
Осадки	
По дням	По часам

Макет будущего сайта

Пользовательский сценарий:

Пользователь: заходит на сайт

Пользователь: выбирает город из списка

Пользователь: выбирает язык (по умолчанию выбран русский)

Пользователь: выбирает детализацию по осадкам (по умолчанию выключено)

Пользователь: выбирает детализацию по времени (по умолчанию по дням)

Система: показывает пользователю прогноз в соответствии с указанными значениями

Также известно, что используется три алгоритма расчёта прогноза.

1. Для городов федерального значения (Москва, Санкт-Петербург, Севастополь)
2. Для городов областного, республиканского, краевого, окружного значения (Волгоград, Казань, Краснодар)
3. Для городов районного значения (Бологое, Суоярви)

Выделим классы эквивалентности для всех параметров

Город	Язык	Осадки	Детализация
Федерального значения	Русский	Да	По дням
Областного значения	Английский	Нет	По часам
Районного значения			

Сколько нужно тестов для прогноза погоды? Если протестировать все возможные варианты комбинаций параметров, получится $3 * 2 * 2 * 2 = 24$ теста. Довольно



много и бессмысленно — количество тестов можно сократить техникой **pairwise** (попарным тестированием).

Так как большинство дефектов возникают при комбинации только двух параметров, нет смысла проверять все возможные сочетания. Если ошибка возникнет в сочетании «Москва» + «Русский», вероятнее всего, она будет возникать и в различных значениях детализации по осадкам, и со всем детализациями по времени. Такой подход позволяет сократить количество проверок.

Создадим попарные наборы для фильтра прогноза погоды.

Сначала создадим таблицу из 24 строк со всеми возможными комбинациями.

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Да	По дням
2	Москва	Русский	Да	По часам
3	Москва	Русский	Нет	По дням
4	Москва	Русский	Нет	По часам
5	Москва	Английский	Да	По дням
6	Москва	Английский	Да	По часам
7	Москва	Английский	Нет	По дням
8	Москва	Английский	Нет	По часам
9	Волгоград	Русский	Да	По дням
10	Волгоград	Русский	Да	По часам
11	Волгоград	Русский	Нет	По дням
12	Волгоград	Русский	Нет	По часам
13	Волгоград	Английский	Да	По дням
14	Волгоград	Английский	Да	По часам
15	Волгоград	Английский	Нет	По дням
16	Волгоград	Английский	Нет	По часам
17	Бологое	Русский	Да	По дням
18	Бологое	Русский	Да	По часам
19	Бологое	Русский	Нет	По дням



20	Бологое	Русский	Нет	По часам
21	Бологое	Английский	Да	По дням
22	Бологое	Английский	Да	По часам
23	Бологое	Английский	Нет	По дням
24	Бологое	Английский	Нет	По часам

А теперь попробуем создать таблицу таким образом, чтобы перебрать все возможные комбинации каждой пары параметров. Начнем с пары «Город — Язык». Остальные ячейки пока заполнять не будем.

	Город	Язык	Осадки	Детализация
1	Москва	Русский		
2	Москва	Английский		
3	Волгоград	Русский		
4	Волгоград	Английский		
5	Бологое	Русский		
6	Бологое	Английский		

Теперь заполним колонку «Осадки», чтобы перебрать все возможные пары «Город — Осадки».

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Да	
2	Москва	Английский	Нет	
3	Волгоград	Русский	Да	
4	Волгоград	Английский	Нет	
5	Бологое	Русский	Да	
6	Бологое	Английский	Нет	

Далее проверим пару «Язык — Осадки». Видим, что у нас есть пары «Русский — Да» и «Английский — Нет», но не хватает пары «Русский — Нет», «Английский — Да». Это легко исправить: просто поменяем местами значения осадков в строке 1 и 2. Теперь у нас есть все возможные варианты пары «Язык — Осадки».



	Город	Язык	Осадки	Детализация
1	Москва	Русский	Нет	
2	Москва	Английский	Да	
3	Волгоград	Русский	Да	
4	Волгоград	Английский	Нет	
5	Бологое	Русский	Да	
6	Бологое	Английский	Нет	

Аналогично заполняем колонку «Детализация». Сначала сопоставляем её с городом, затем проверяем, чтобы были все пары «Язык — Детализация», «Осадки — Детализация» и при необходимости меняем значения в строках местами (но следим, чтобы не разрушались нужные пары).

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Нет	По дням
2	Москва	Английский	Да	По часам
3	Волгоград	Русский	Да	По часам
4	Волгоград	Английский	Нет	По дням
5	Бологое	Русский	Да	По дням
6	Бологое	Английский	Нет	По часам

Готово! Вместо 24 тестов осталось всего 6. Это в 4 раза сократит время на тестирование.

Pairwise используется только для неупорядоченных классов эквивалентности. Для упорядоченных классов эта техника может дать слишком много проверок на выходе, так как нужно учитывать значения внутри класса и на границах.

Особенности тестирования линейных классов эквивалентности рассмотрим на уроке «Доменный анализ».

PIST и другие инструменты

Составлять таблицы попарного тестирования вручную долго и есть риск ошибок. Можно использовать специальные программы, которые сами составляют таблицы из исходных параметров:



- [pairwiseTool](#)
- [Pairwise Pict Online](#)

Рассмотрим составление таблицы в [pairwiseTool](#). Создадим колонки с нужными значениями и нажмём Generate Pairwise:

	× Город	× Язык	× Осадки	× Детализация	
× Row 1	Москва	RU	Да	По дням	+
× Row 2	Волгоград	ENG	Нет	По часам	+
× Row 3	Бологое	{your value}	{your value}	{your value}	+
+	+	+	+	+	+

В итоге получится файл, похожий на то, что мы сделали сами:

	A	B	C	D
1	Город	Язык	Осадки	Детализация
2	Волгоград	RU	Да	По дням
3	Бологое	RU	Нет	По часам
4	Волгоград	US	Да	По часам
5	Москва	US	Нет	По дням
6	Бологое	US	Да	По дням
7	Волгоград	US	Нет	По часам
8	Москва	RU	Да	По часам

Обратите внимание, программа сгенерировала 7 строк, а не 6, как мы вручную. Почему так получилось? Дело в том, что в идеале матрица должна быть сбалансирована. Это значит, что если какая-то пара значений двух столбцов встречается несколько раз, то все возможные парные комбинации значений этих столбцов должны встретиться столько же раз. Матрица, составленная вручную, не сбалансирована, так как пара «Английский — Нет» встречается 2 раза, а «Английский — Да» — 1 раз.



Pairwise и негативное тестирование

Составляя тестовые наборы для попарного тестирования, никогда не учитываются негативные сценарии, то есть те, которые способны привести к ошибке или невозможности обработать указанные значения. В случае с прогнозом погоды — оставить пустым значение поля «Город».

Нужно ли добавлять еще одну строку «Поле незаполнено» в генератор pairwise? **Нет.** Невалидные значения никогда не комбинируются друг с другом, так как если в тесте будет два или более невалидных параметра, тестировщик не установит причину возникновения ошибки.

Проверки с невалидными данными добавляются в таблицу по принципу «один невалидный параметр в строке». Остальные параметры могут быть любыми валидными.

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Нет	По дням
2	Москва	Английский	Да	По часам
3	Волгоград	Русский	Да	По часам
4	Волгоград	Английский	Нет	По дням
5	Бологое	Русский	Да	По дням
6	Бологое	Английский	Нет	По часам
7	Не заполнено	Английский	Нет	По часам
8	Бологое	Китайский	Да	Нет

Пример: окружение для кроссбраузерного тестирования

В требованиях прописаны следующие характеристики окружения:

- сайт поддерживается в браузерах последних версий: Chrome, FireFox, Safari;
- сайт локализован на русском и английском языках;
- поддерживаются разрешения экранов: 1920x1080, 1536x864, 1440x900, 1366x768, 1280x720.

При составлении попарных комбинаций лучше начинать с параметра, у которого больше всего вариантов значений. В нашем примере это «Разрешение». После него по количеству значений следует параметр «Браузер».



Начнём составлять таблицу:

Разрешение экрана	Браузер	Локализация
1920x1080	Chrome	
1920x1080	FireFox	
1920x1080	Safari	
1536x864	Chrome	
1536x864	FireFox	
1536x864	Safari	
1440x900	Chrome	
1440x900	FireFox	
1440x900	Safari	
1366x768	Chrome	
1366x768	FireFox	
1366x768	Safari	
1280x720	Chrome	
1280x720	FireFox	
1280x720	Safari	

Получили необходимые и возможные комбинации параметров «Разрешение» и «Браузер». Теперь добавляем сюда параметр «Локализации». Важно, чтобы каждый браузер хотя бы раз пересекался с каждым языком, и каждое разрешение также хотя бы раз пересекалось с каждым языком.

В результате получим:

Разрешение экрана	Браузер	Локализация
1920x1080	Chrome	русский
1920x1080	FireFox	английский



1920x1080	Safari	русский
1536x864	Chrome	русский
1536x864	FireFox	английский
1536x864	Safari	русский
1440x900	Chrome	русский
1440x900	FireFox	английский
1440x900	Safari	русский
1366x768	Chrome	русский
1366x768	FireFox	английский
1366x768	Safari	русский
1280x720	Chrome	английский
1280x720	FireFox	русский
1280x720	Safari	английский

Обратите внимание на языки в последнем разрешении — мы их поменяли местами, чтобы у каждого браузера были комбинации с каждым из языков.

Набор окружений готов — учтены все возможные попарные комбинации каждого двух параметров!

Контрольные вопросы

1. В чем суть техники попарного тестирования?
2. Для каких классов эквивалентности можно применить технику pairwise?
3. Можно ли использовать невалидные значения в технике pairwise?

Глоссарий

Попарное тестирование (pairwise) — техника подбора тестовых данных, основанная на утверждении, что большинство дефектов возникают при комбинации только двух параметров



Дополнительные материалы

1. [Pairwise testing. Part 1 - Orthogonal Arrays](#)
2. [Метод попарного тестирования](#)
3. [Классы эквивалентности для имен](#)
4. [Немного о простом. Тест-дизайн](#)