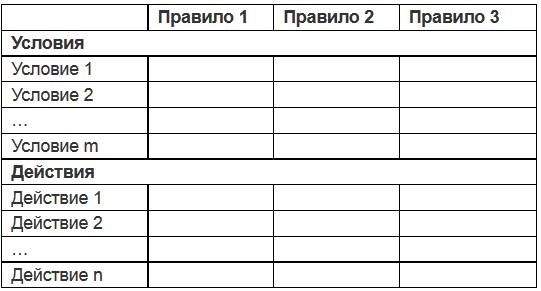
**Decision Table Testing**

1. **Суть метода**

**Decision Table** **Testing** - техника тестирования ПО методом черного ящика, помогающая пользователям автоматизированных инструментов тестирования в процессе принятия решений, без использования ручного кодирования.

Decision Tables(Таблицы принятия решения, DT) описывают логику приложения основываясь сущностях(свойствах/условиях) состояния системы. В таблицах решений представлен набор условий, одновременное выполнение которых должно привести к определенному действию.

Общий вид таблиц решений представлен на рисунке ниже:



Условия(conditions) от 1 до m - это разные свойства системы, они представляют в таблице входные данные, которые можно ввести в систему.

Действия(actions) от 1 до n - это действия которые могут произойти с указанной комбинацией условий. Каждое правило определяет уникальный набор входных данных всех свойств, которые приводят к исполнению конкретных действий.

1. **Пример использования Decision Table**

Пусть тестируем приложение для страховой компании. Это приложение вычисляет скидку на страхование автомобилей, взаимозависимости от того, был ли водитель хорошим студентом и состоит ли он в браке. Величина скидки вычисляется с помощью decision table.

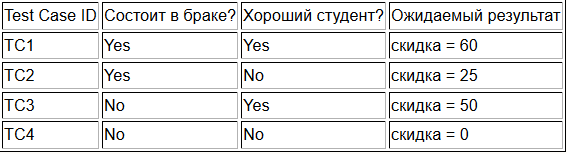
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Правило 1 | Правило 2 | Правило 3 | Правило 4 |
| **Условия** |  |  |  |  |
| Состоит в браке? | Yes | Yes | No | No |
| Хороший студент? | Yes | No | Yes | No |
| **Действия** |  |  |  |  |
| Скидка ($) | 60 | 25 | 50 | 0 |

Из таблицы видно, что выделено два условия и одно действие. Каждое из условия бинарное(принимает только два значения – true\false). В зависимости от комбинаций условия и правила получаем некоторый результат – действие. Для рассматриваемого примера таким действием является величина предоставляемой скидки.

Можно легко перевести таблицу в набор тест-кейсов. Правило для создания тест-кейсов просто – необходимо создать как минимум 1 тест-кейс на каждое правило в таблице. Тогда DT примет следующий вид:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Тест кейс 1 | Тест кейс 2 | Тест кейс 3 | Тест кейс 4 |
| **Входные данные** |  |  |  |  |
| Состоит в браке? | Yes | Yes | No | No |
| Хороший студент? | Yes | No | Yes | No |
| **Ожидаемый результат** |  |  |  |  |
| Скидка ($) | 60 | 25 | 50 | 0 |

Или перевернув таблицу, получим готовые тест-кейсы в следующем виде:



1. **Достоинства метода**
2. Удобный инструмент для фиксирования требований и описания функциональности приложения
3. Заставляет лучше проанализировать тестируемый продукт, систематизировать все знания по нему. Удобно описывать логику приложения
4. Служат основой для создания тест-кейсов
5. Таблица понятна всей проектной команде
6. Эффективна для определения комбинаций, которые в других случаях могут быть не замечены.
7. Применяется на любых уровнях тестирования
8. **Недостатки метода**
9. DT не очень хорошо масштабируемы. Может потребоваться расширить или упростить таблицу
10. Слишком большие таблицы решений необходимо разбить на более мелкие, чтобы удалить избыточность.
11. Трудно рассмотреть всевозможные альтернативы.
12. Необходимо знать какие условия важны для тестирования. Для этого необходимо знание предметной области.
13. Можно получить слишком большие таблицы. Так, для бинарных значений(true/false), получим уже 2n решений.