

Основы тест-аналитики

Диаграммы переходов состояний



На этом уроке

1. Рассмотрим использование техник:
 - a. диаграмма и таблица переходов состояний
 - b. таблицы принятия решений
2. Составим диаграммы и таблицы переходов состояний, таблицы принятия решений.

Оглавление

[Диаграмма переходов состояний](#)

[Составление тестов](#)

[Таблицы переходов состояний](#)

[Составление тестов](#)

[Таблицы принятия решений](#)

[Глоссарий](#)

[Домашнее задание](#)

[Дополнительные материалы](#)

[Используемые источники](#)

Диаграмма переходов состояний

Диаграмма переходов состояний - это техника тест-дизайна для систем, которые изменяют характеристики в зависимости от внешних стимулов. Иными словами, система принимает то или иное состояние, в котором находится до тех пор, пока не получит стимул для дальнейшего изменения.

Хрестоматийный пример переходов состояний приведен в книге Л. Копланда "A Practitioner's Guide to Software Test Design". Рассмотрим пользовательский сценарий "Покупка билета на самолет"

Вариант 1

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создает бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: оплачивает билет

Система: выпускает билет

Контролер: принимает билет

Система: погашает билет

Вариант 2

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создает бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: не оплатил билет до окончания таймера

Система: отменяет бронь

Вариант 3

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создает бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: отменил заказ до окончания таймера

Система: отменяет бронь

Вариант 4

Пользователь: оформляет заказ на билет (предоставляет данные)

Система: создает бронь

Система: запускает таймер, до истечения которого нужно оплатить билет

Пользователь: оплачивает билет

Пользователь: возвращает билет

Система: оформляет возврат

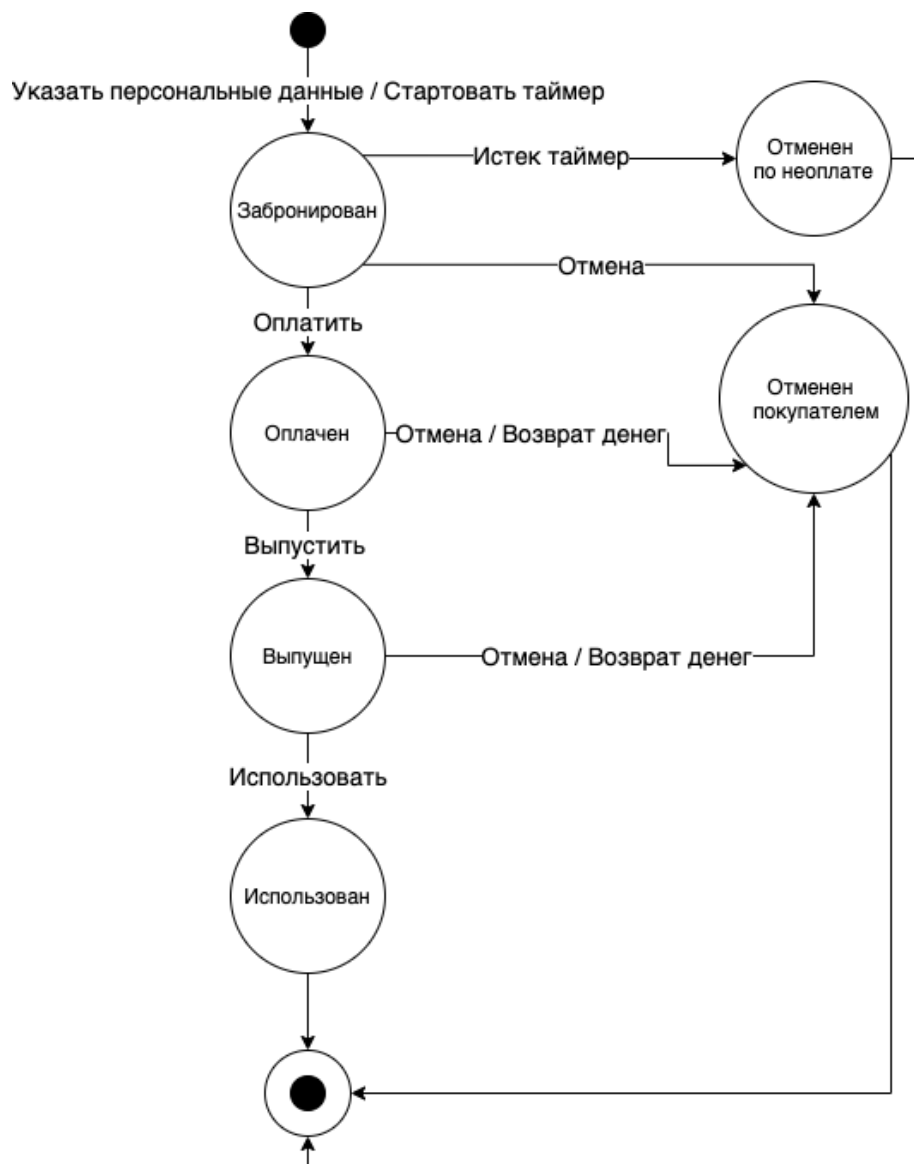
Помимо агентов: системы, пользователя и контролера - в сценарии есть объект - билет, над которым выполняются различные действия. Также мы видим ряд условий: был ли оплачен заказ, истек ли таймер. В этой ситуации четырех позитивных тестов (по 1 на каждый вариант использования) недостаточно, т.к. ошибки скрываются там, где условия нарушаются. Для более наглядного представления сценариев и выявления негативных кейсов используются диаграммы переходов состояний и таблицы переходов состояний.

Диаграмма переходов состояний состоит из элементов:

- **Точка входа.** Пользователь еще не начал взаимодействовать с системой

- **Круг с названием состояний** - состояние, в которое система пришла вследствие действий пользователя
- **Стрелки** обозначают переход от одного состояния к другому
- **События** - то, что приходит в систему извне и запускает изменение состояния (например, пользователь оплатил заказ). Пишется над стрелкой
- **Действие** - то, что происходит внутри системы из-за смены состояния (например, запуск таймера). Пишется над стрелкой, после **события**. Отделяется от события косой чертой /
- **Точка выхода** - взаимодействие с системой окончено, изменение состояния невозможно

В случае с билетом диаграмма переходов состояний выглядит следующим образом:



Состояния: Забронирован, Оплачен, Выпущен, Использован, Отменен по неоплате, Отменен покупателем.

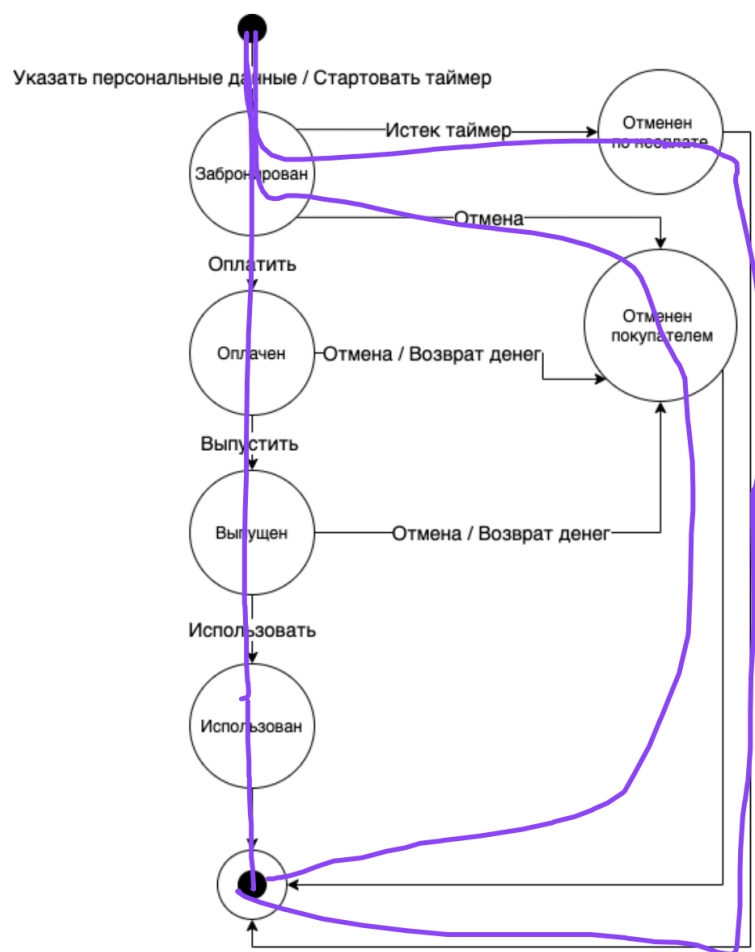
События: Указать персональные данные, Оплатить, Использовать, Отмена, Истек таймер.

Действия: Стартовать таймер, Выпустить, Возврат денег.

Диаграмма наглядно иллюстрирует бизнес-правила и варианты взаимодействия пользователя с системой. При наличии сложной логики диаграммы обычно создаются аналитиками как часть документации к ПО.

Составление тестов

При составлении тест-кейсов по диаграммам переходов состояний, можно выделить 4 уровня покрытия:



1. Каждое состояние затрагивает хотя бы 1 тест

Тестовое покрытие неоптимально, т.к. тестируются не все действия

2. Каждое событие вызывается хотя бы в 1 тесте. Данный уровень покрытия также не оптимален

3. Каждый путь выполняется хотя бы в 1 тесте. Это предпочтительный вариант, но он бывает недостижим. Если в диаграмме есть циклы, то количество возможных путей становится бесконечным. Например, если есть 2 состояния: А и В, есть переходы из состояния А в состояние В и обратно из В в А, то возможные пути:

- a. А -> В
- b. А -> В -> А
- c. А -> В -> А -> В -> А -> В -> А -> В -> А ...

Важно тестировать циклы, т.к. в них могут накапливаться ошибки или происходить утечки памяти

4. Каждый переход используется хотя бы в 1 тесте. Такой уровень тестового покрытия оптимален, т.к. затронет все состояния, действия и события. Он может совпадать с покрытием всех возможных путей. В этом случае мы получим следующий набор тестовых сценариев:

- a. Использование оплаченного билета (*Забронирован - Оплачен - Выпущен - Использован*)
- b. Отмена брони по неоплате (*Забронирован - Отменен по неоплате*)
- c. Отмена брони покупателем (*Забронирован - Отменен покупателем*)
- d. Отмена покупателем после оплаты (*Забронирован - Оплачен - Отменен*)
- e. Отмена покупателем после выпуска (*Забронирован - Оплачен - Выпущен - Отменен*)

Недостаток составления тестов по диаграмме - отсутствие негативных сценариев. Что будет, если покупатель попытается использовать отмененный билет? Или оплатит билет после того, как истечет таймер? Данные переходы не предусмотрены на схеме, т.к. при большом количестве стрелок она станет нечитаемой. Для негативных тестов удобнее использовать таблицу переходов.

Таблицы переходов состояний

Диаграммы переходов состояний - не единственный способ показать поведение системы. Диаграммы более понятны, но таблицы переходов состояний лучше систематизируют сведения о системе. Они состоят из 4 колонок:

- Текущее состояние
- Событие
- Действие
- Новое состояние

Алгоритм составления таблицы переходов состояний:

1. Сопоставить состояния и события по принципу “все ко всем” и заполнить столбцы “Текущее состояние”, Событие”.

2. Там, где смена состояний вызывает действие - дописать в столбец "Действия"
3. В каждой строке указать новое состояние. Если событие не ведет к смене состояния, то новое состояние совпадает с текущим.

Таблица переходов состояний для покупки билета:

Текущее состояние	Событие	Действие	Новое состояние
null	Указать персональные данные	Стартовать таймер	Забронирован
null	Оплатить	--	null
null	Использовать	--	null
null	Отмена	--	null
null	Истек таймер	--	null
Забронирован	Указать персональные данные	--	Забронирован
Забронирован	Оплатить	--	Оплачен
Забронирован	Использовать	--	Забронирован
Забронирован	Отмена	--	Отменен покупателем
Забронирован	Истек таймер	--	Отменен по неоплате
Оплачен	Указать персональные данные	--	Оплачен
Оплачен	Оплатить	--	Оплачен
Оплачен	--	Выпустить	Выпущен
Оплачен	Использовать	--	Оплачен
Оплачен	Отмена	Возврат денег	Отменен покупателем
Оплачен	Истек таймер	--	Оплачен
Выпущен	Указать персональные данные	--	Выпущен
Выпущен	Оплатить	--	Выпущен
Выпущен	Использовать	--	Использован
Выпущен	Отмена	Возврат денег	Отменен покупателем
Выпущен	Истек таймер	--	Выпущен
Использован	Указать персональные данные	--	Использован
Использован	Оплатить	--	Использован

Использован	Использовать	--	Использован
Использован	Отмена	--	Использован
Использован	Истек таймер	--	Использован
Отменен по неоплате	Указать персональные данные	--	Отменен по неоплате
Отменен по неоплате	Оплатить	--	Отменен по неоплате
Отменен по неоплате	Использовать	--	Отменен по неоплате
Отменен по неоплате	Отмена	--	Отменен по неоплате
Отменен по неоплате	Истек таймер	--	Отменен по неоплате
Отменен покупателем	Указать персональные данные	--	Отменен покупателем
Отменен покупателем	Оплатить	--	Отменен покупателем
Отменен покупателем	Использовать	--	Отменен покупателем
Отменен покупателем	Отмена	--	Отменен покупателем
Отменен покупателем	Истек таймер	--	Отменен покупателем

Составление тестов

Алгоритм составления тестов по таблице:

1. Выделить зеленым все возможные переходы
2. Выявить невозможные переходы, которые несут риски и выделить их красным. Это будут негативные проверки. Для выявления рисков тестировщики советуются с разработчиками или аналитиками.

В данном случае существуют риски:

- покупатель использует неоплаченный, невыпущенный, ранее использованный или отмененный билет
- покупатель оплатит билет после отмены брони

Именно эти кейсы отмечены красным цветом.

Текущее состояние	Событие	Действие	Новое состояние
null	Указать персональные данные	Стартовать таймер	Забронирован
null	Оплатить	--	null
null	Использовать	--	null

null	Отмена	--	null
null	Истек таймер	--	null
Забронирован	Указать персональные данные	--	Забронирован
Забронирован	Оплатить	--	Оплачен
Забронирован	Использовать	--	Забронирован
Забронирован	Отмена	--	Отменен покупателем
Забронирован	Истек таймер	--	Отменен по неоплате
Оплачен	Указать персональные данные	--	Оплачен
Оплачен	Оплатить	--	Оплачен
Оплачен	--	Выпустить	Выпущен
Оплачен	Использовать	--	Оплачен
Оплачен	Отмена	Возврат денег	Отменен покупателем
Оплачен	Истек таймер	--	Оплачен
Выпущен	Указать персональные данные	--	Выпущен
Выпущен	Оплатить	--	Выпущен
Выпущен	Использовать	--	Использован
Выпущен	Отмена	Возврат денег	Отменен покупателем
Выпущен	Истек таймер	--	Выпущен
Использован	Указать персональные данные	--	Использован
Использован	Оплатить	--	Использован
Использован	Использовать	--	Использован
Использован	Отмена	--	Использован
Использован	Истек таймер	--	Использован
Отменен по неоплате	Указать персональные данные	--	Отменен по неоплате
Отменен по неоплате	Оплатить	--	Отменен по неоплате
Отменен по неоплате	Использовать	--	Отменен по неоплате
Отменен по неоплате	Отмена	--	Отменен по неоплате
Отменен по неоплате	Истек таймер	--	Отменен по неоплате

Отменен покупателем	Указать персональные данные	--	Отменен покупателем
Отменен покупателем	Оплатить	--	Отменен покупателем
Отменен покупателем	Использовать	--	Отменен покупателем
Отменен покупателем	Отмена	--	Отменен покупателем
Отменен покупателем	Истек таймер	--	Отменен покупателем

Диаграммы и переходы состояний не применимы, если система не изменяет состояние и не должна реагировать на внешние стимулы определенным образом.

Таблицы принятия решений

Таблицы принятия решения используются для тестирования программ со сложной бизнес-логикой. Они применимы в тех случаях, когда существует набор правил, и выходные данные зависят от комбинации этих условий.

User story: Я как заемщик хочу узнать, какую максимальную сумму ипотеки одобрит банк.

Use case

Пользователь: указывает дату рождения

Пользователь: указывает ежемесячный доход

Пользователь: указывает, может ли подтвердить доход справкой 2-НДФЛ или по форме банка

Пользователь: указывает срок ипотеки

Система: определяет максимальную сумму кредита

Правила:

1. Ипотека выдается под 6% годовых
2. Ипотека НЕ выдается:
 - а. заемщикам младше 18 лет
3. Сумма кредита вычисляется по формуле

$$\frac{\text{Ежемесячный доход} * \text{Срок в месяцах} * K}{1 + 0,005 * \text{Срок в месяцах}}$$

4. Если доход подтвержден справкой, $K = 0,3$.
5. Если доход не подтвержден справкой или менее 20 000 рублей, то $K = 0,1$.
6. Если заемщику больше 55 лет, то кредит выдается только при наличии созаемщика

При наличии сложных бизнес-правил самое важное - не запутаться и предусмотреть все возможные сценарии. В данном случае выходные данные (сумма кредита) зависят от 3 параметров:

- возраст заемщика
- ежемесячный доход
- возможность подтвердить доход справкой

Необходимо предусмотреть все возможные комбинации параметров, чтобы верно рассчитать итоговое значение. Для этого составим таблицу принятия решений.

Сначала определим все условия, применив классы эквивалентности

- Возраст: до 18, 18 - 55, больше 55
- Доход не менее 20000: да, нет
- Справка: есть, нет

И определим, какие решения принять. В данном случае это будет:

- максимальная сумма кредита. Если выдача кредита невозможна, то максимальная сумма будет равна 0.
- нужен ли созаемщик?
- какой применить коэффициент?

Алгоритм составления таблицы

1. Выписать все условия и действия

Условия		
Возраст заемщика		
Ежемесячный доход $\geq 20\ 000$		
Наличие справки		
Действия		
Максимальная сумма кредита		
Нужен созаемщик?		
Коэффициент К		

2. Посчитать общее количество столбцов в таблице. Для этого перемножить количество вариантов для каждого условия. В нашем случае это $3 * 2 * 2 = 12$. Значит, всего будет 12 комбинаций.
3. Делим 12 на 3 (количество вариантов в 1 условии), получилось 4. Значит, заполняем по 4 ячейки каждым вариантом

Условия												
Возраст заемщика	< 18	< 18	< 18	< 18	18-55	18-55	18-55	18-55	> 55	> 55	> 55	> 55
Ежемесячный доход $\geq 20\ 000$												
Наличие справки												
Действия												
Максимальная сумма кредита												
Нужен созаемщик?												
Коэффициент К												

4. Делим 4 (то, что получилось в п.3) на количество вариантов во 2 условии (2). Получается 2, значит, заполняем по 2 ячейки каждым вариантом ответа (“да” или “нет”)

Условия													
Возраст заемщика	< 18	< 18	< 18	< 18	18-55	18-55	18-55	18-55	> 55	> 55	> 55	> 55	
Ежемесячный доход >=20 000	Да	Да	Нет	Нет	Да	Да	Нет	Нет	Да	Да	Нет	Нет	
Наличие справки													
Действия													
Максимальная сумма кредита													
Нужен созаемщик?													
Коэффициент К													

5. Повторяем для 3 параметра. $2/2 = 1$. По 1 ячейке заполняем вариантами “есть” или “нет”

Условия													
Возраст заемщика	< 18	< 18	< 18	< 18	18-55	18-55	18-55	18-55	> 55	> 55	> 55	> 55	
Ежемесячный доход >=20 000	Да	Да	Нет	Нет	Да	Да	Нет	Нет	Да	Да	Нет	Нет	
Наличие справки	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	
Действия													
Максимальная сумма кредита													
Нужен созаемщик?													
Коэффициент К													

6. Опираясь на требования, заполняем строку “Коэффициент К”, т.к. он понадобится для дальнейших вычислений. Если ипотека недоступна, К = 0

Условия													
Возраст заемщика	< 18	< 18	< 18	< 18	18-55	18-55	18-55	18-55	> 55	> 55	> 55	> 55	
Ежемесячный доход >=20 000	Да	Да	Нет	Нет	Да	Да	Нет	Нет	Да	Да	Нет	Нет	
Наличие справки	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	
Действия													
Максимальная сумма кредита													
Нужен созаемщик?													
Коэффициент К		0	0	0	0	0,3	0,1	0,1	0,1	0,3	0,1	0,1	0,1

7. Вычислим максимальную сумму. Для простоты составления формулы в строке “Ежемесячный доход” заменим “Да” на 50 000, а “Нет” - на 10 000. Формулу запишем в 1 ячейке и скопируем в остальные

Условия													
Возраст заемщика	< 18	< 18	< 18	< 18	18-55	18-55	18-55	18-55	> 55	> 55	> 55	> 55	
Ежемесячный доход >=20 000	Да	Да	Нет	Нет	50000	50000	10000	10000	50000	50000	10000	10000	
Наличие справки	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	
Действия													
Максимальная сумма кредита					=?=(F16*240*F22)/(1+0,005*120)								
Нужен созаемщик?													
Коэффициент К		0	0	0	0	0,3	0,1	0,1	0,1	0,3	0,1	0,1	0,1

Условия													
Возраст заемщика	< 18	< 18	< 18	< 18	18-55	18-55	18-55	18-55	> 55	> 55	> 55	> 55	
Ежемесячный доход >=20 000	Да	Да	Нет	Нет	50000	50000	10000	10000	50000	50000	10000	10000	
Наличие справки	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	
Действия													
Максимальная сумма кредита	0	0	0	0	2250000	750000	150000	150000	2250000	750000	150000	150000	
Нужен созаемщик?													
Коэффициент К	0	0	0	0	0,3	0,1	0,1	0,1	0,3	0,1	0,1	0,1	

8. Наконец, заполним строку “Нужен созаемщик?”

Условия													
Возраст заемщика	< 18	< 18	< 18	< 18	18-55	18-55	18-55	18-55	> 55	> 55	> 55	> 55	
Ежемесячный доход >=20 000	Да	Да	Нет	Нет	50000	50000	10000	10000	50000	50000	10000	10000	
Наличие справки	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	Есть	Нет	
Действия													
Максимальная сумма кредита	0	0	0	0	2250000	750000	150000	150000	2250000	750000	150000	150000	
Нужен созаемщик?					Нет	Нет	Нет	Нет	Да	Да	Да	Да	
Коэффициент К	0	0	0	0	0,3	0,1	0,1	0,1	0,3	0,1	0,1	0,1	

Таблица принятия решений готова. Мы рассмотрели все возможные комбинации условий и получили разные результаты для каждого случая.

Иногда случается так, что комбинация условий не описана в документации и тестировщик не может определить корректное решение. В этом случае требования неполные, и их нужно уточнять у аналитиков.

Домашнее задание

- Изучите документацию по созданию [тестов](#) и [тест-планов](#).
 - Проанализируйте, как может изменяться статус тест-кейса в процессе выполнения.
 - Нарисуйте диаграмму перехода состояний для изменения статуса тест-кейса.
 - Укажите действия и события. Составьте таблицу переходов состояний
 - В качестве результата предоставьте 2 ссылки:
 - Ссылку на диаграмму (можно залить на гугл-диск)
 - Ссылку на таблицу переходов состояний
- Задание “со звездочкой”. Изучая технику pairwise, мы говорили о том, что не обязательно перебирать все возможные комбинации значений, достаточно комбинаций всех пар. Однако, в технике “Таблица решений” мы перебираем все возможные условия. Как вы думаете, почему для задачи с ипотекой нельзя применить pairwise?

Глоссарий

Диаграмма переходов состояний - техника тест-дизайна, основанная на методе “черного ящика”, которая демонстрирует поведение системы при получении управляющих воздействий извне.

Таблица переходов состояний - см. Диаграмма переходов состояний.

Таблица принятия решений - техника тест-дизайна для объектов со сложной бизнес-логикой. Представляет набор условий, одновременное выполнение которых приводит к определенному действию.

Дополнительные материалы

1. Книга Lee Copeland “A Practitioner's Guide to Software Test Design”
2. <https://www.getsoftwareservice.com/state-transition-testing/> - ISTQB – What is State Transition Testing in Software Testing
3. <https://www.guru99.com/state-transition-testing.html> - What is State Transition Testing? Diagram, Technique, Example
4. <https://www.softwaretestinghelp.com/state-transition-testing-technique-for-testing-complex-applications/> - State Transition Testing Technique And State Transition Diagram With Examples
5. https://www.tutorialspoint.com/software_testing_dictionary/state_transition.htm - State Transition Testing

Используемые источники