Техники тест-дизайна

КУРС ВАДИМА КСЕНДЗОВА
«ТЕСТИРОВАНИЕ ПО»

Тест дизайн — один из первоначальных этапов тестирования программного обеспечения, этап планирования и проектирования тестов. Тест дизайн представляет собой продумывание и написание тестовых случаев (test case), в соответствии с требованиями проекта, критериями качества будущего продукта и финальными целями тестирования.

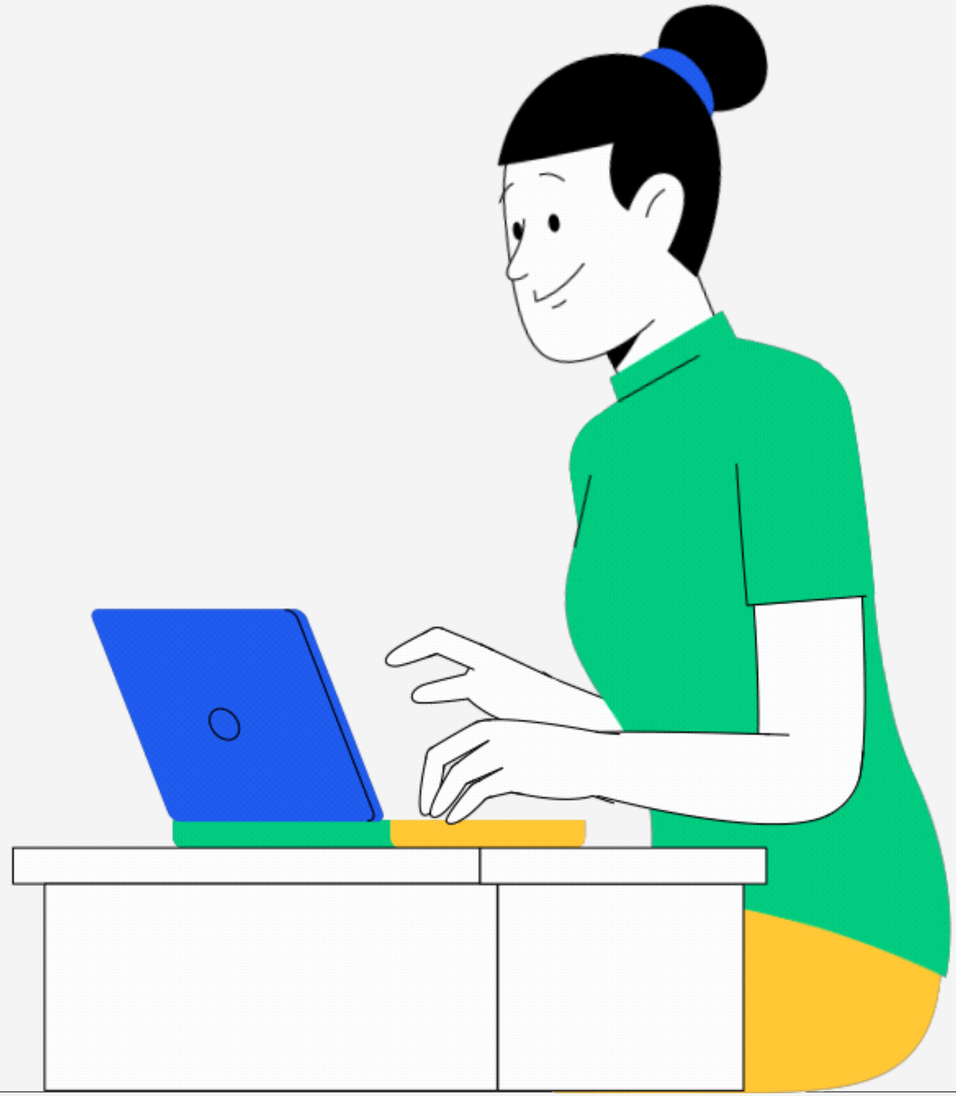
Цели тест-дизайна

ОБЕСПЕЧИТЬ ПОКРЫТИЕ ФУНКЦИОНАЛА ПРИЛОЖЕНИЯ ТЕСТАМИ:

- Тесты должны покрывать весь функционал;
- Тестов должно быть минимально достаточно.

Задачи тест-дизайн

- Проанализировать требования к продукту;
- Оценить риски возможные при использовании продукта;
- Написать достаточное минимальное количество тестов.



Разделение на классы эквивалентности

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(black box).

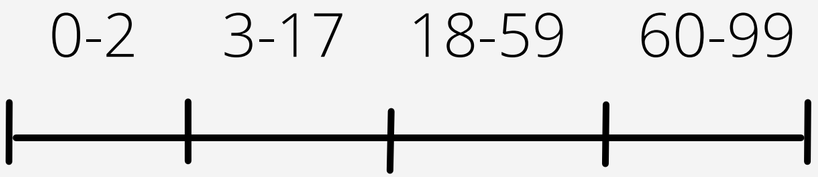
1

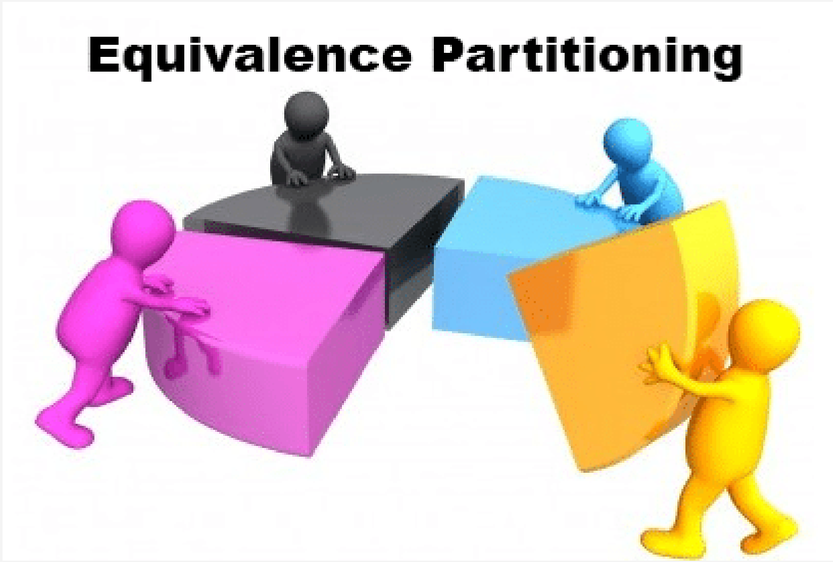
10

34

81

Мы тестируем функциональность приложения, позволяющего покупать авиабилеты. Стоимость билета будет зависеть от возраста пассажира.
У нас есть группы: младше 2, от 3 до 18 лет, старше 18 и младше 60 лет и люди старше 60 до 99 лет.





Анализ граничных значений

Техника граничных значений основана на предположении, что большинство ошибок может возникнуть на границах эквивалентных классов. Она тесно связана с вышеописанной техникой эквивалентного разбиения, из-за чего часто используется с ней в паре.

Граничное значение – входное или выходное значение, которое находится на краю разделения эквивалентности или на наименьшем дополнительном расстоянии по обе стороны граничного значения, например min и max значении диапозона.

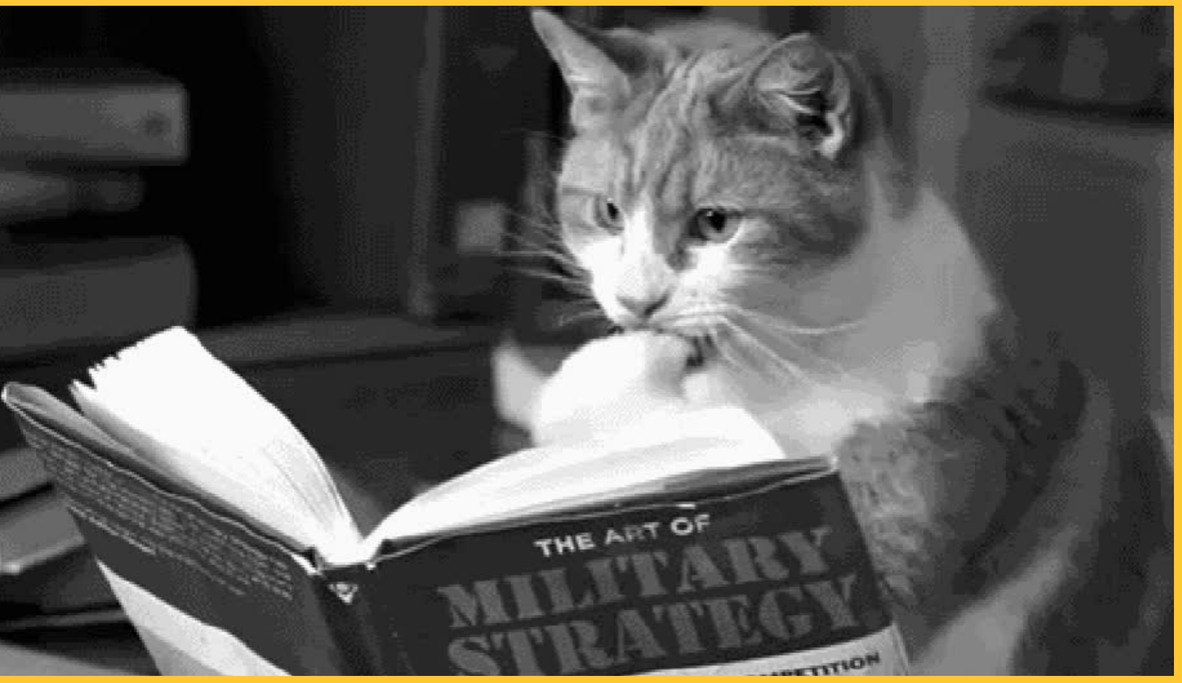
Name	Латиница	От 2-9
Y	Введем в поле имени 1 букву	
Ya	Введем в поле имени 2 буквы	
Yan	Введем в поле имени 3 буквы	
Aleksand	Введем в поле имени 8 букв	
Aleksandr	Введем в поле имени 9 букв	
Aleksandra	Введем в поле имени 10 букв	
Yanы	Введем 1 русскую букву	
Yan!	Введем один символ	
Y an	Ведем пробед	
Y2an	Введем цифру	

Таблица принятия решений – используется для тестирования с различными комбинациями входных данных, которые приводят к различным выходным данным в программе.
В двух словах, Decision Table Testing - это метод проверки черного ящика, в котором мы создаем таблицу решений для сложной бизнес-логики.
Этот метод применяется если поведение системы отличается для каждого набора входных данных.
Создание таблицы решений помогает команде тестирования в разработке тестов. Таблицы решений полезны не только при формулировании сложной бизнес-логики, но также полезны для тестировщиков, которые хотят понять, как различные комбинации входов влияют на результат.

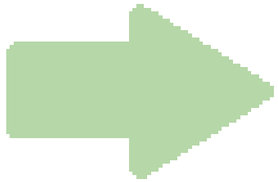
Условия входа	1 условие	2 условие	3 условие	4 условие
Email	False	False	True	True
Пароль	False	True	False	True
Вход	Ошибка	Ошибка	Ошибка	Вход выполнен

Попарное тестирование

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.



Google Chrome	Windows	RU
Google Chrome	Windows	EN
Google Chrome	Linux	RU
Google Chrome	Linux	EN
Opera	Windows	RU
Opera	Windows	EN
Opera	Linux	RU
Opera	Linux	EN

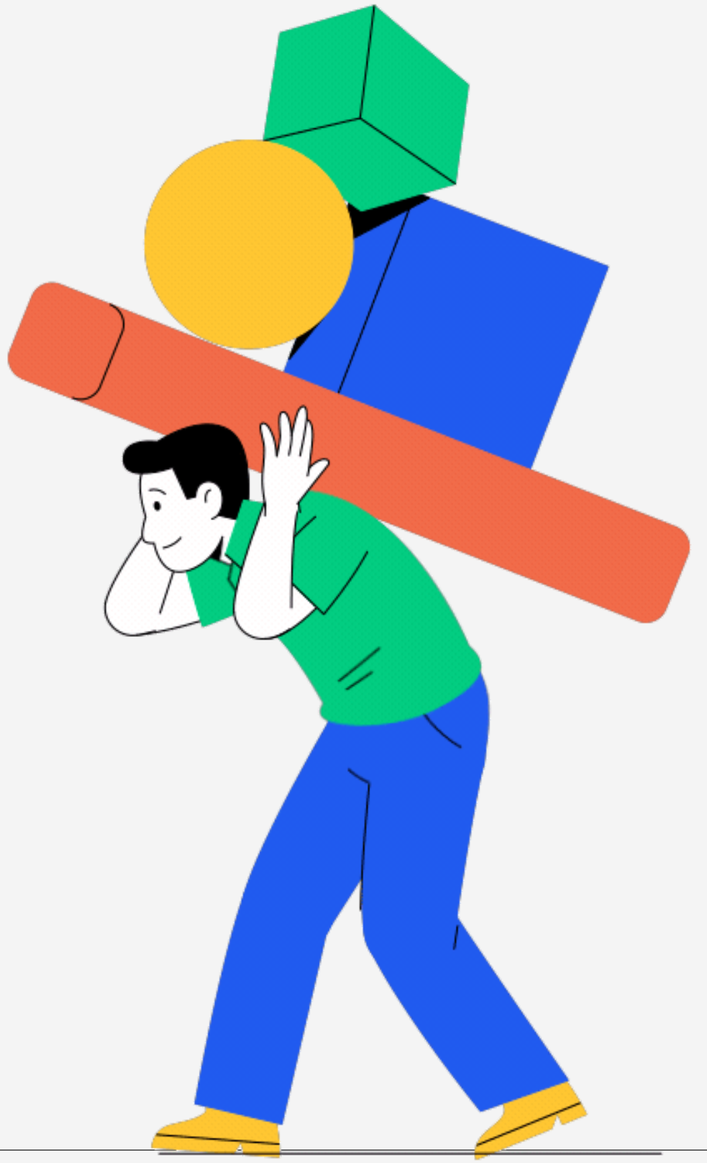


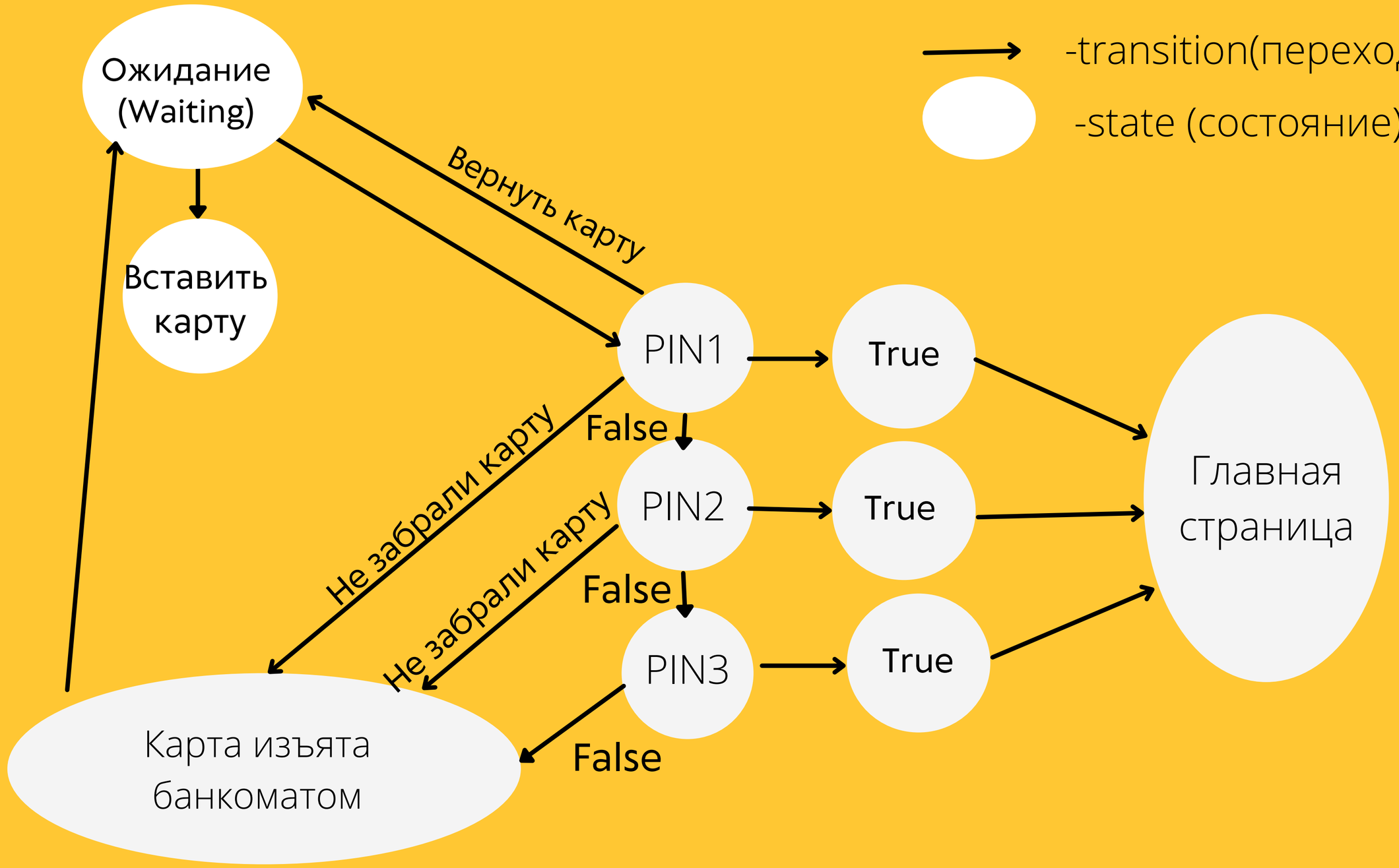
Google Chrome	Windows	RU
Google Chrome	Linux	EN
Opera	Windows	EN
Opera	Linux	RU

Все это можно просчитать и вручную, но не обязательно – гораздо удобнее автоматизировать процесс. Для этого существует программа попарного независимого комбинированного тестирования – Pairwise Independent Combinatorial Testing (PICT). Для проведения тестирования специалист создает текстовый файл с перечислением и их возможных значений, а затем запускает PICT через cmd – командную строку. Скомбинированные тесты отображаются в виде таблицы в самой консоли. Так же результаты по желанию можно выгрузить в файл Excel.
Очень легок в использовании Pairwise Tool.

Техника перехода состояний

Тестирование состояния перехода также является тестом черного ящика, в котором тестировщик видит поведение тестируемого приложения для различных входных условий в последовательности. Здесь тестер дает нам как положительный, так и отрицательный ввод тестовых значений, а затем делает запись поведения системы. Это также модель, на которой основаны система и тесты. Любое из того, откуда вы получаете разные выходные данные для одного и того же входа, зависящие от состояния, которое имело место ранее, называется сис-темой конечных состояний.




-transition(переход)
-state (состояние)
Ожидание (Waiting)
Вставить карту
Вернуть карту
PIN1
True
False
PIN2
True
False
PIN3
True
False
Не забрали карту
Не забрали карту
Главная страница
Карта изъята банкоматом

Use Case техника

Use case - это сценарии, описывающие то как actor (обычно человек, но может быть и другая система) пользуется системой для достижения определенной цели. Варианты использования описываются с точки зрения пользователя, а не системы. Внутренние работы по поддержанию работоспособности системы не являются частью варианта использования.

Name	**Вход на домашнюю страницу банкомата**
ID	1
Use Case Component	Банкомат
Actors	Действующие лица(клиенты банка)
Organization benefits	Польза от внедрения(можно снять деньги)
Frequency	Частота использования(часто)
Trigger	Условие срабатывания(когда вставил карточку в банкомат)
Pre-conditional	1. Карта обслуживается в банкомате этого банка. 2. Срок в норме. 3. Карта не деформирована. 4. Карта вставлена в банкомат.
Main Success Scenario (Основной сценарий)	1. Вводим пин-код; 2. Нажимаем enter; 3. Попадание на домашнюю страницу.
Sub-Variations Альтернативы	1. Вводим пин-код; 2. Нажимаем enter; 3. Неправильный пин-код; 4. Вводим второй раз пин-код; 5. Нажимаем enter 6. Попадание на домашнюю страницу
Extensions (Дополнительные условия)	1. При вводе 3-х раз неправильного пароля карточка изымается банкоматом.

Есть что потестить?

Тестирование на основе рисков или Предугадывание ошибок — основную роль играет опытность инженера, и его интуиция.
Ключевую роль здесь как нигде играет опыт инженера по качеству. На основе своего опыта, представлении как должно работать приложение, и предположения где может быть ошибка, строятся и проводятся тесты.

Отчет о выполнении тестирования

Предыдущий проектный опыт

Факторы, которые следует учитывать при угадывании ошибок

Checklist

Отчеты о рисках приложений

Спасибо за внимание!