

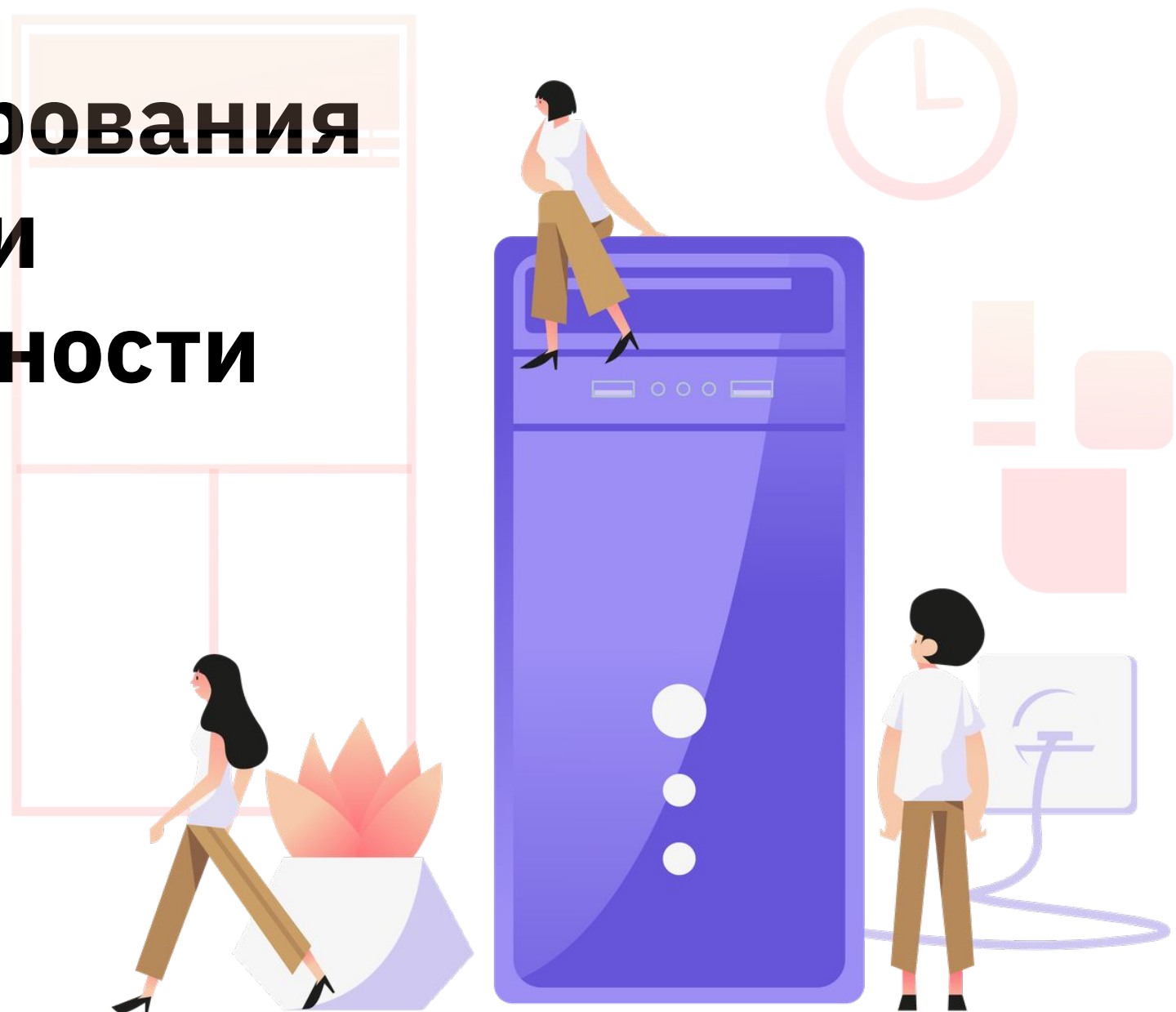
Уровни тестирования

Основы ручного тестирования

Что будет на уроке

1. Уровни тестирования.
2. Тест-кейс VS чек-лист.
3. Демонстрация работы по тест-кейсу.

Уровни тестирования по значимости функциональности



По значимости функций



Дымовое тестирование

Тестирование критического пути

Расширенное тестирование

Дымовое тестирование

Проверка самой важной, ключевой функциональности, неработоспособность которой делает бессмысленной идею использования приложения.

Пример дымового тестирования формы регистрации:

- 1. Поля «логин» и «пароль» обязательные.*
- 2. Пароль маскируется.*
- 3. Регистрация успешна при валидных логине и пароле.*

Тестирование критического пути

Функциональность, которую используют типичные пользователи в повседневной деятельности.

Пример тестирования критического пути приложения «Интернет-магазин»:

- 1. Товар добавляется в корзину.*
- 2. Товар можно оплатить картой.*
- 3. Работает отправка сообщения в службу поддержки.*

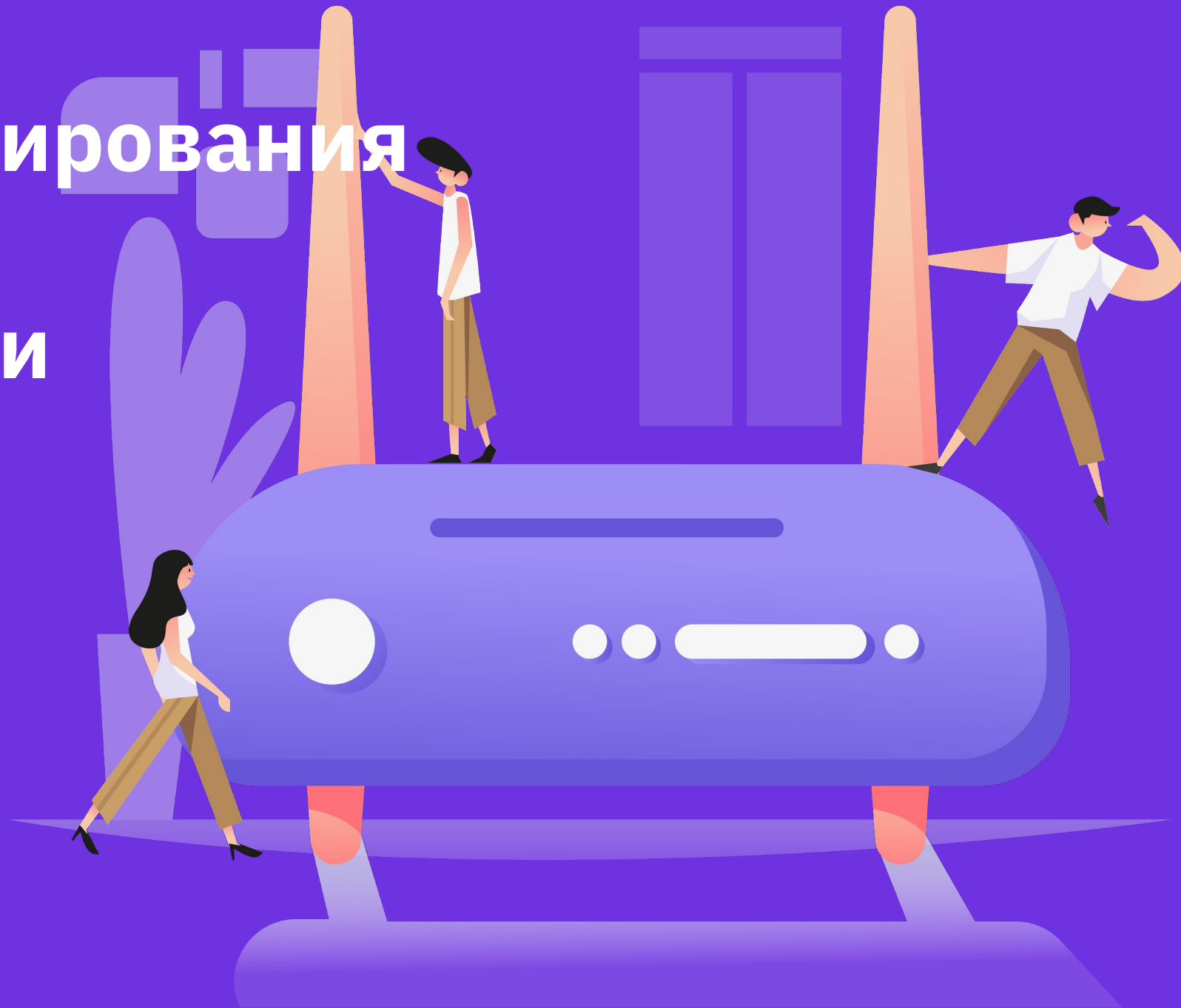
Расширенное тестирование

Проверяет всю заявленную в требованиях функциональность, в том числе низкого приоритета и незначительной важности.

Пример расширенного тестирования приложения «Интернет-магазин»:

1. Проверяется правильность заполнения раздела с документацией.

Уровни тестирования по степени детализации

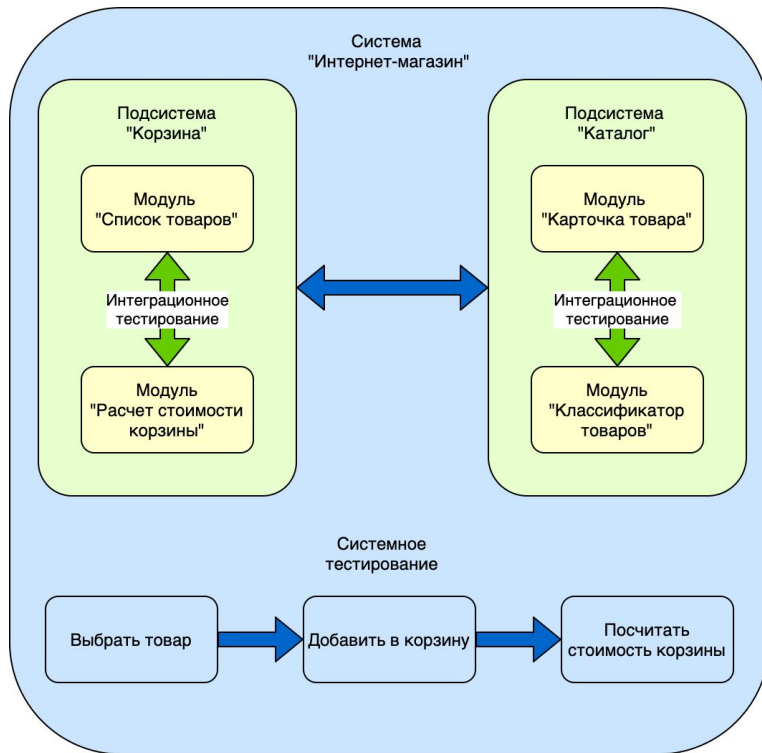


По степени детализации

Юнит-тестирование

Интеграционное тестирование

Системное и приёмочное
тестирование



Юнит-тестирование (модульное)

Проверка наименьших элементов ПО, которые тестируются по отдельности: модули, объекты, классы, функции.

Выполняется разработчиками, проводится методом белого ящика и всегда автоматизируется.

Интеграционное тестирование

Тестирование части системы, состоящей из двух и более модулей, направленное на проверку взаимодействия этих модулей.

Системное тестирование

Процесс тестирования системы в целом, чтобы проверить, соответствует ли она требованиям.

Тестирование проводится на уровне пользовательских интерфейсов.

Приёмочное тестирование

Формальный процесс тестирования, который проверяет соответствие системы потребностям, требованиям и бизнес-процессам пользователя.

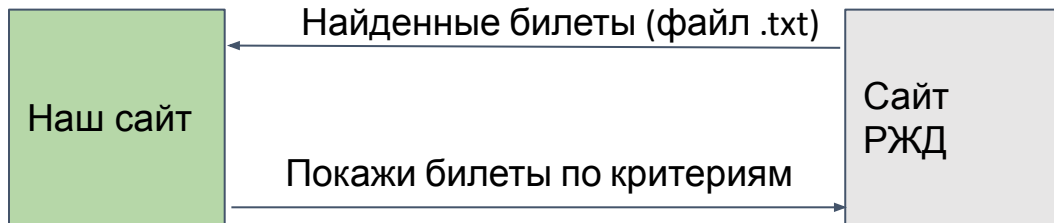
Проводится для вынесения решения о приёме приложения заказчиком.

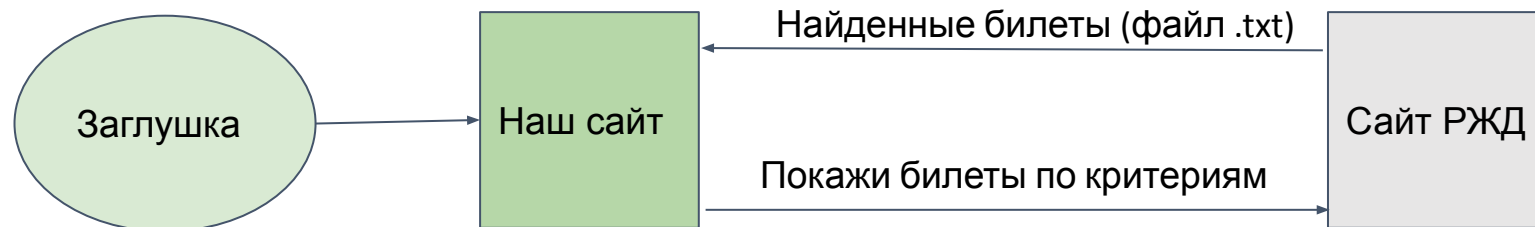
Уровни тестирования на примере продажи билетов



Пользовательский сценарий

1. Пользователь заходит на сайт.
2. Задаёт критерии поиска.
3. Нажимает кнопку «Найти билеты».
4. Сайт показывает билеты, найденные на сайте РЖД.





Unit-тест 1

Проверить, что сайт верно распознаёт количество записей в файле.

Unit-тест 2

Проверить, что сайт верно распознаёт города.

...

Интеграционный тест 1

Покажи билеты «Москва — Волгоград» на 31.12.2020.

Интеграционный тест 2

Покажи билеты «Мурманск — Волгоград» на 31.12.2020.

...

Системный тест 1

1. Зайти на сайт.
2. Задать критерии.
3. Посмотреть билеты.

...

Тест-кейс



Ценность тест-кейсов

1. Структурируют подход к тестированию.
2. Показатель соответствия ситуации плану.
3. Поддерживают взаимопонимание между заказчиком, разработчиками и тестировщиком.
4. Хранят информацию для длительного использования и обмена опытом между сотрудниками и командами.
5. Основа регрессионного тестирования.
6. Основа для метрик тестового покрытия.
7. Основа для увеличения тестового покрытия.

Атрибуты тест-кейса на этапе проектирования

1. Идентификатор тест-кейса.
2. Название.
3. Шаги.
4. Ожидаемый результат.
5. Приоритет тест-кейса.
6. Тестовые данные.
7. Предусловия.
8. Постусловия.

Атрибуты тест-кейса на этапе выполнения

1. Фактический результат.
2. Статус тест-кейса.
 - a. **passed** — все шаги проверены, ПО на 100% соответствует ожиданию.
 - b. **failed** — поведение не соответствует ожиданию, найден новый дефект.
 - c. **known bug** — поведение не соответствует ожиданию, найден известный дефект.
 - d. **broken** — тест-кейс нуждается в редактировании.
 - e. **blocked** — выполнение тест-кейса невозможно из-за дефекта, который его блокирует.
 - f. **skipped** — тест-кейс пропущен.

Правила работы с тест-кейсами

1. Один тест-кейс — одна проверка или один сценарий.
2. Заголовок точно описывает суть тест-кейса.
3. Точные названия для элементов приложения: кнопок, чекбоксов, элементов меню.
4. Простой технический стиль без объяснений базовых понятий работы ПО.
5. Нет пропущенных шагов: каждое действие вытекает из предыдущего.
6. Нет зависимостей от других тест-кейсов.
7. Нет дублей с другими тест-кейсами.
8. Обнаруженная ошибка становится очевидной.
9. Гибкость для модификации.

Ошибки в тест-кейсах

1. Заголовок тест-кейса отсутствует или сформулирован некорректно.
2. Ссылки ведут на разные или недействительные требования.
3. Используются личные формы глаголов: «нажми», «перейдите», «укажи».
4. Пунктуационные, орфографические, синтаксические ошибки.
5. «Выдумывание» особенностей поведения приложения без отсылки к требованиям.
6. Не описывается приготовление к выполнению тест-кейса.
7. Полное дублирование (копирование) одного тест-кейса на уровнях дымового тестирования, тестирования критического пути, расширенного тестирования.

Тест-кейс и чек-лист

	Тест-кейс	Чек-лист
Детализация	Проверки детализированы, расписаны по шагам.	Проверки формулируются в общем виде. Пункт чек-листа = заголовок тест-кейса.
Связность	Тест-кейсы не зависят друг от друга и выполняются в произвольном порядке.	Проверки могут быть связаны друг с другом, предполагается их последовательное выполнение.
Понятность	Понятен любому человеку.	Понятен человеку, который знаком с продуктом.
Эффект пестицида	Есть риск возникновения, т. к. каждый раз воспроизводятся одни и те же шаги.	Риск снижается, т. к. каждый выполняет проверку по-своему.

Наборы тест-кейсов

test case suite, test suite, test set

Совокупность тест-кейсов, выбранных с общей целью или по общему признаку.

Бывают последовательными и свободными.

Работа с тест-кейсами

Спасибо!

**Каждый день
вы становитесь
лучше :)**

