2.

# Agenda

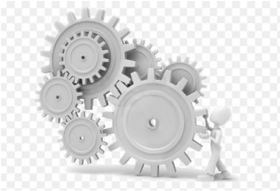
- **What kind of testing exists?**
- Test types
- Test levels
- Practice

# What kind of testing exists?

Manual testing

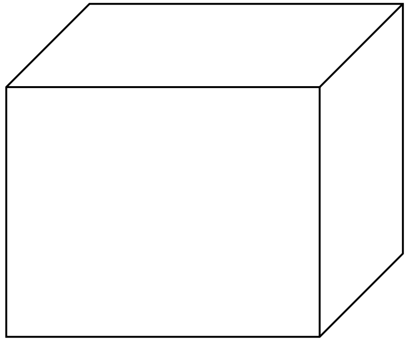
Automated testing

/



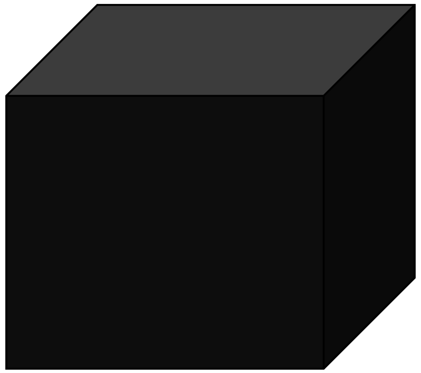
Semi-automated testing

# Black and white box testing



**White box**

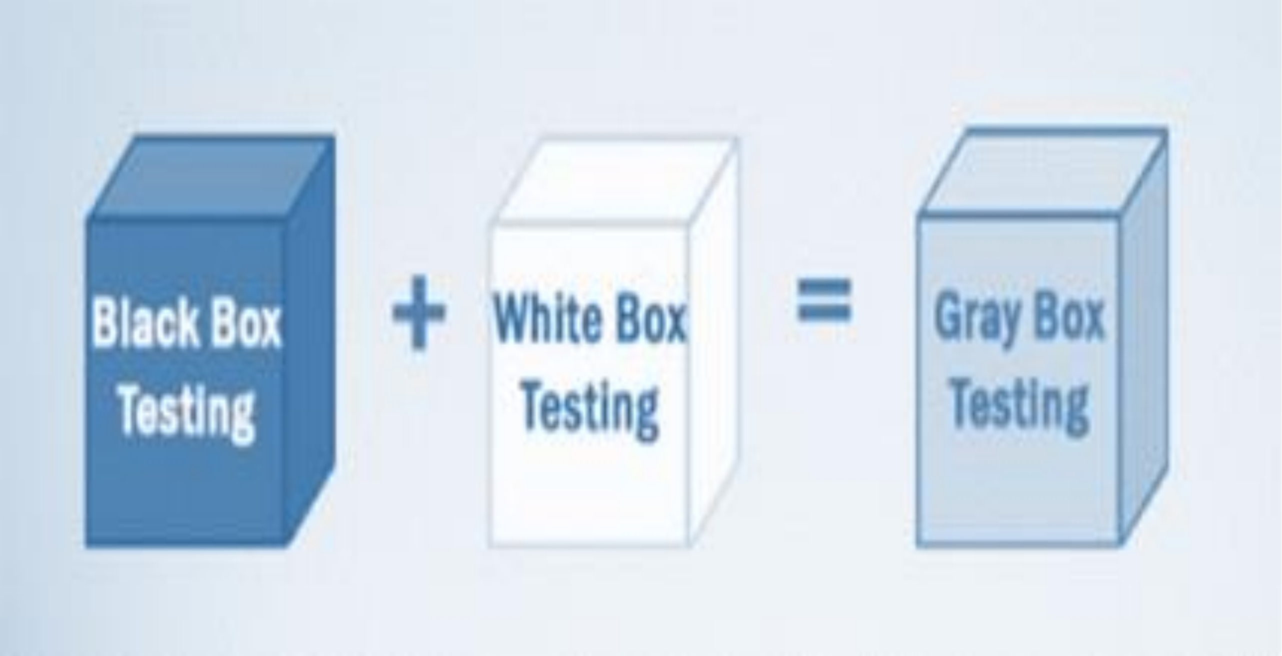
Tester uses
code to design tests



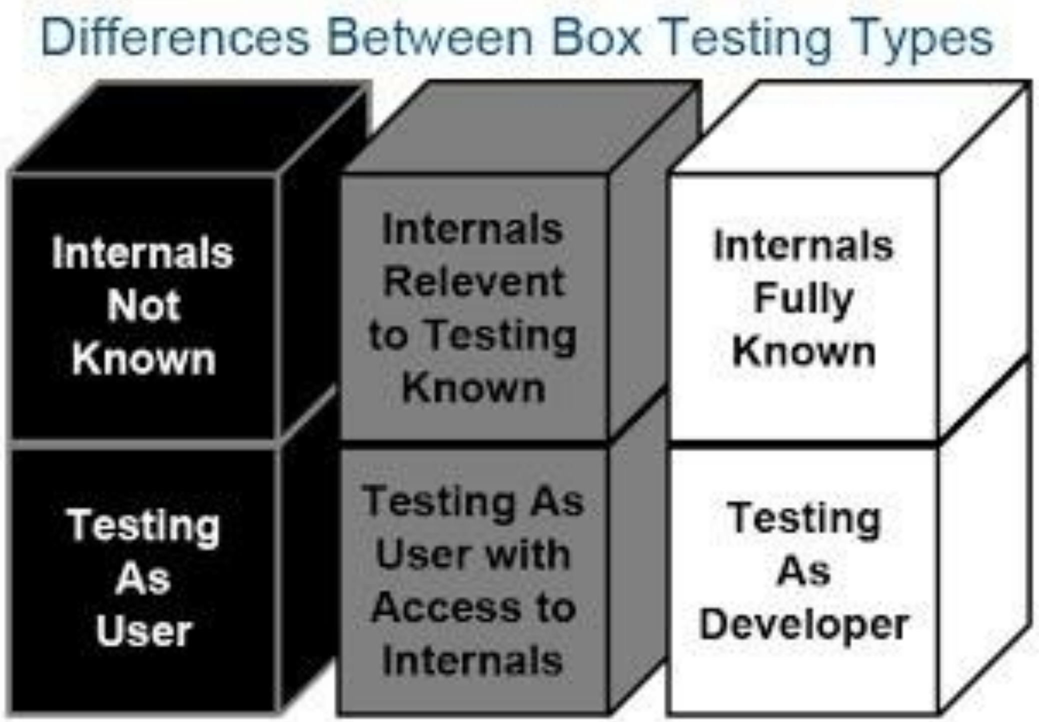
**Black box**

Tester does not have
access to code

# Gray box testing



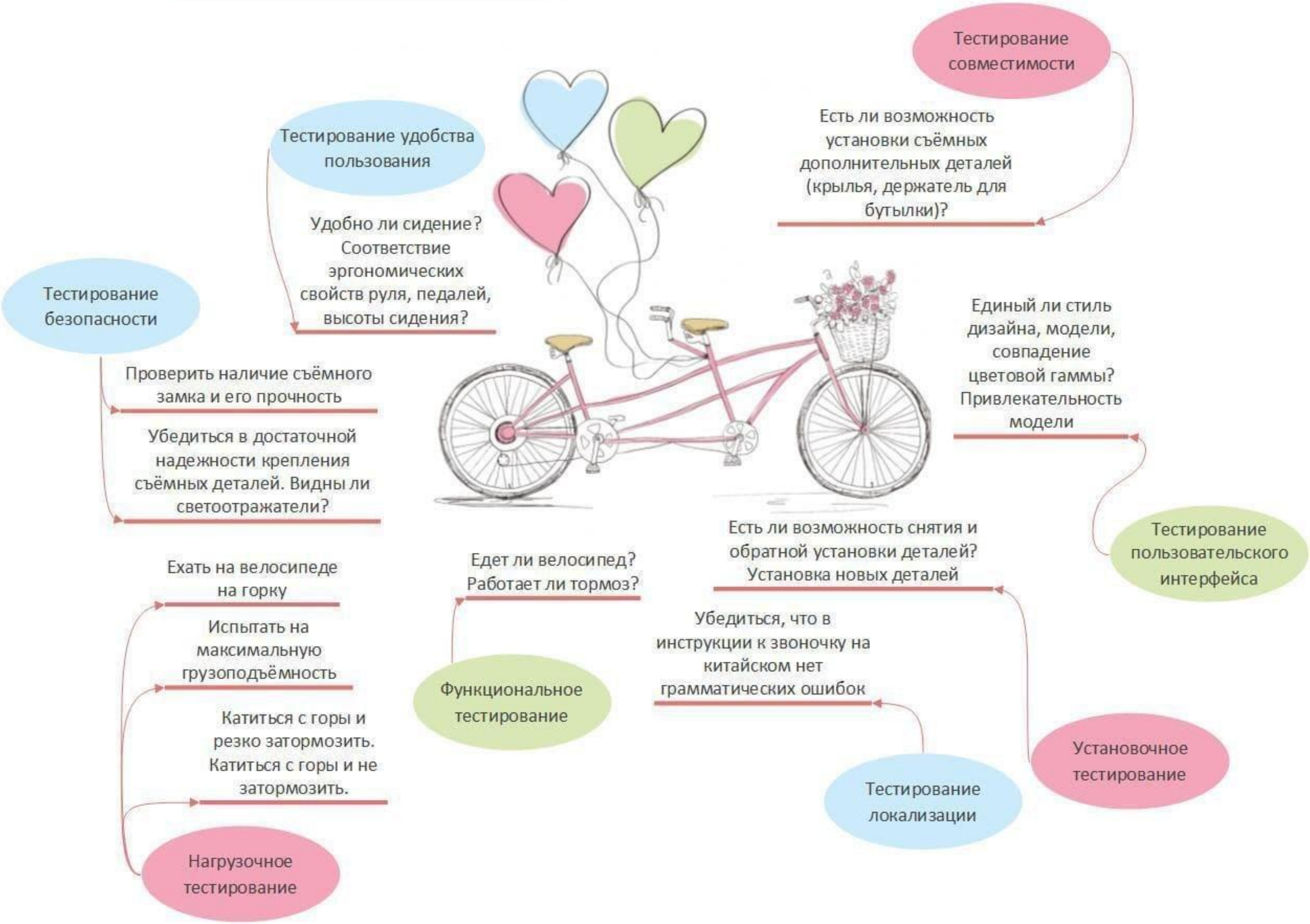
Black Box Testing + White Box Testing = Gray Box Testing



Differences Between Box Testing Types

| | | |
|---|---|---|
| Internals Not Known | Internals Relevent to Testing Known | Internals Fully Known |
| Testing As User | Testing As User with Access to Internals | Testing As Developer |

# Тестирование велосипеда



Тестирование совместимости

Есть ли возможность установки съёмных дополнительных деталей (крылья, держатель для бутылки)?

Тестирование удобства пользования

Удобно ли сидение? Соответствие эргономических свойств руля, педалей, высоты сидения?

Тестирование безопасности

Проверить наличие съёмного замка и его прочность

Убедиться в достаточной надежности крепления съёмных деталей. Видны ли светоотражатели?

Тестирование пользовательского интерфейса

Единый ли стиль дизайна, модели, совпадение цветовой гаммы? Привлекательность модели

Нагрузочное тестирование

Ехать на велосипеде на горку

Испытать на максимальную грузоподъёмность

Катиться с горы и резко затормозить. Катиться с горы и не затормозить.

Функциональное тестирование

Едет ли велосипед? Работает ли тормоз?

Установочное тестирование

Есть ли возможность снятия и обратной установки деталей? Установка новых деталей

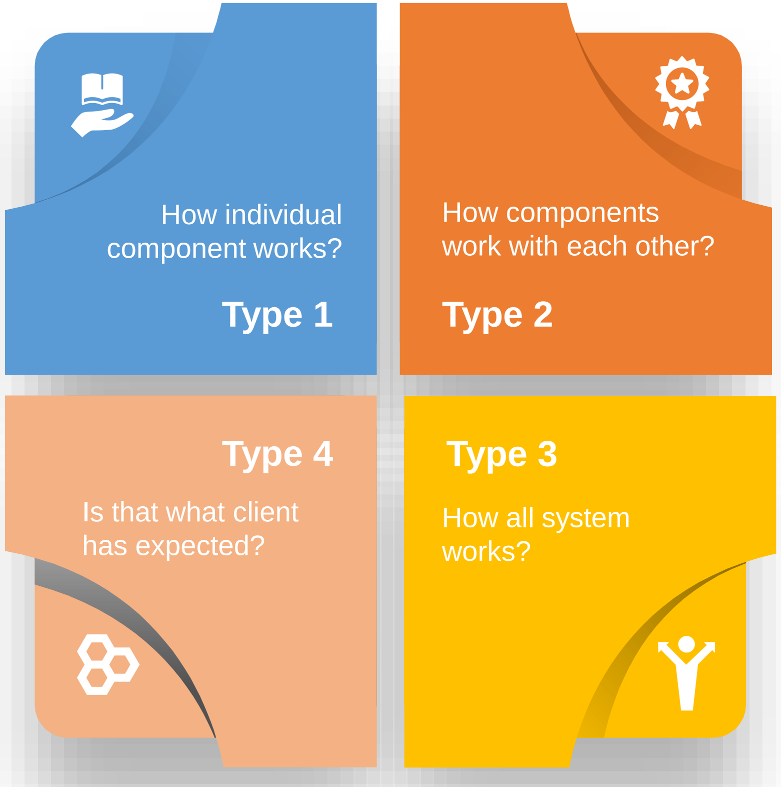
Тестирование локализации

Убедиться, что в инструкции к звоночку на китайском нет грамматических ошибок

# Agenda

- What kind of testing exists?
- **Test levels**
- Test types
- Practice

# Test levels



**Unit (component) testing**

How individual component works?

**Type 1**

**Integration testing**

How components work with each other?

**Type 2**

**Type 3**

How all system works?

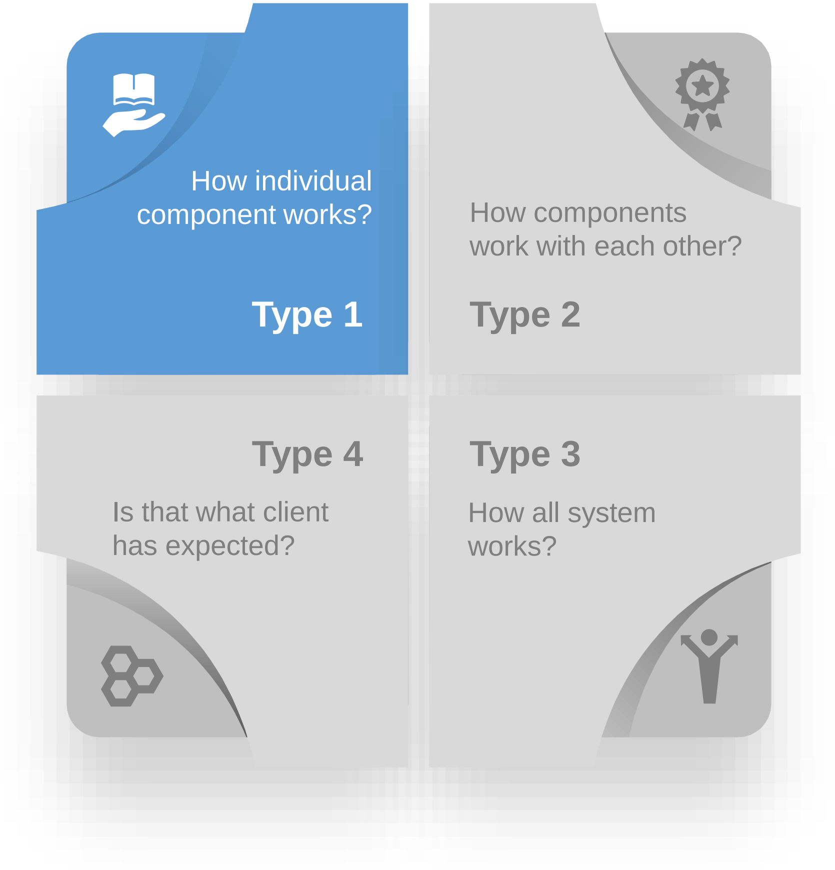
System testing

**Type 4**

Is that what client has expected?

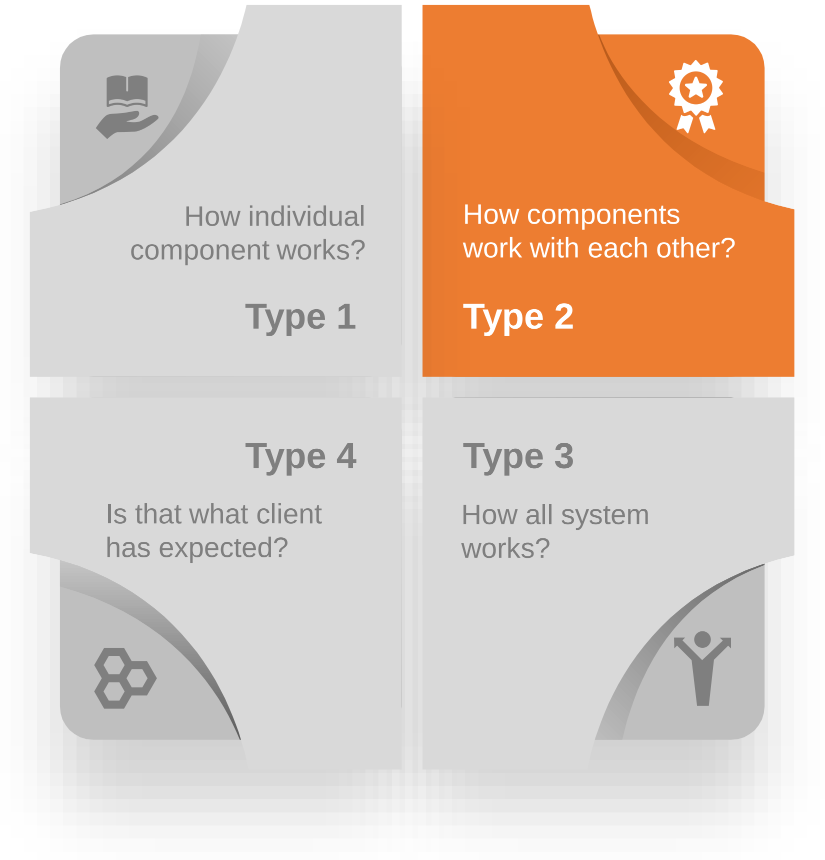
**Acceptance testing**

# Unit (component) Testing


How individual component works?
Type 1
How components work with each other?
Type 2
Type 4
Is that what client has expected?
Type 3
How all system works?

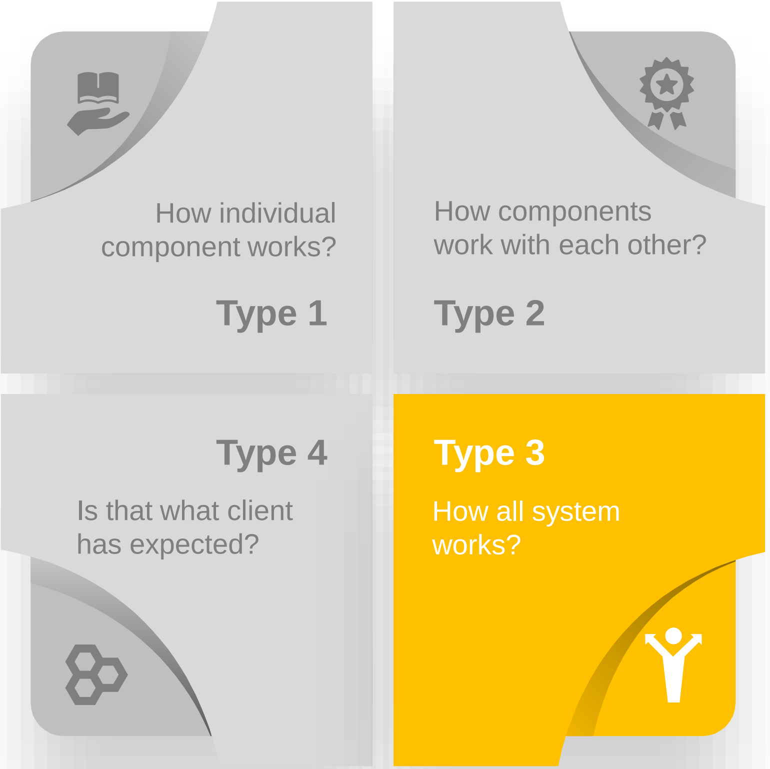
- tests a **unit** - the smallest testable part of any software: function, method, loop or statement
- automated
- usually performed by developers
- white-box technique

# Integration Testing


How individual component works?
Type 1
How components work with each other?
Type 2
Type 4
Is that what client has expected?
Type 3
How all system works?

- tests the **combination of units** works correctly
- automated
- usually performed by developers
- white-box technique

# System Testing



How individual component works?

Type 1

How components work with each other?

Type 2

Type 4

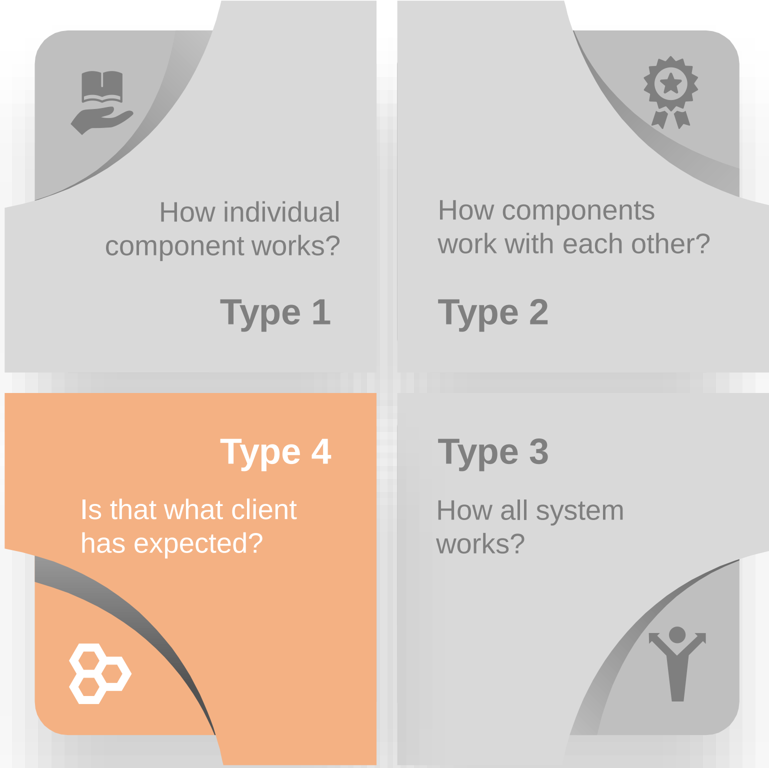
Is that what client has expected?

Type 3

How all system works?

- ensures the **whole system** works as user expected
- both manual and automated
- usually performed by testers
- black-box technique

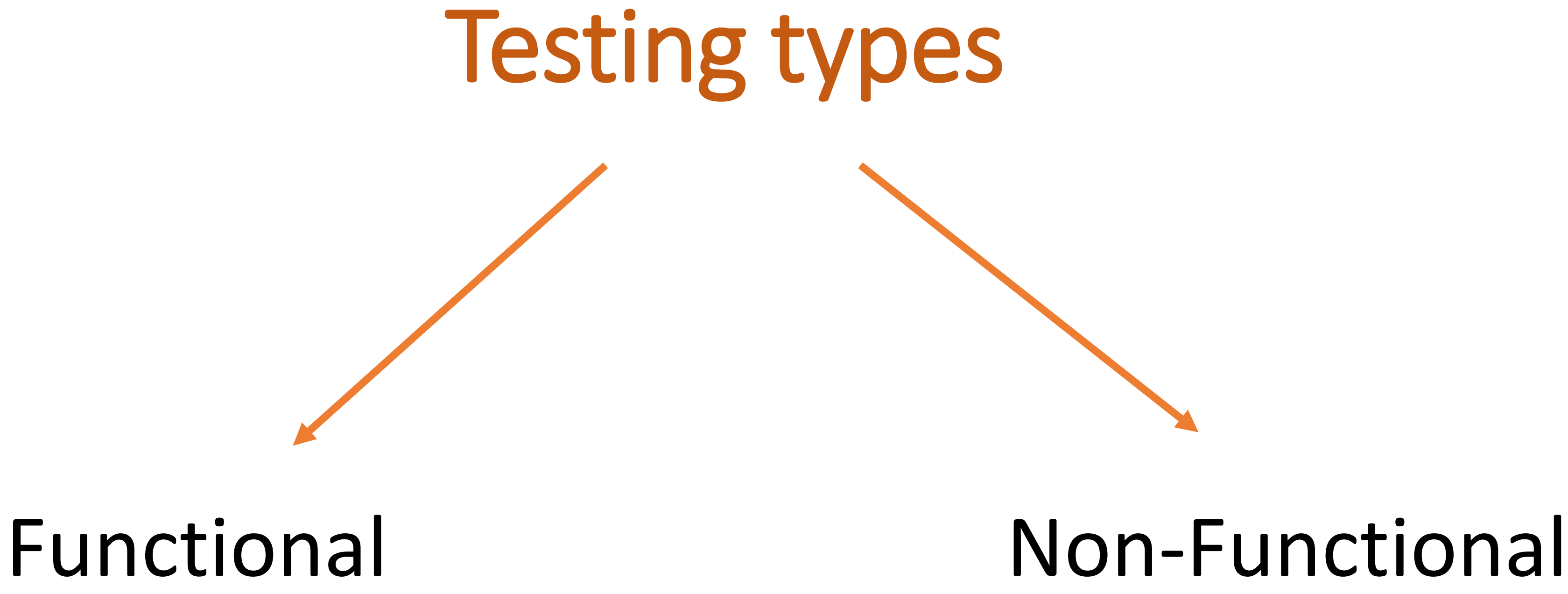
# User Acceptance Testing (UAT)


How individual component works?
Type 1
How components work with each other?
Type 2
Type 4
Is that what client has expected?
Type 3
How all system works?

- testing of the product done by the actual end users
- manual
- black-box technique

# Agenda

- What kind of testing exists?
- Testing levels
- **Testing types**
- Practice



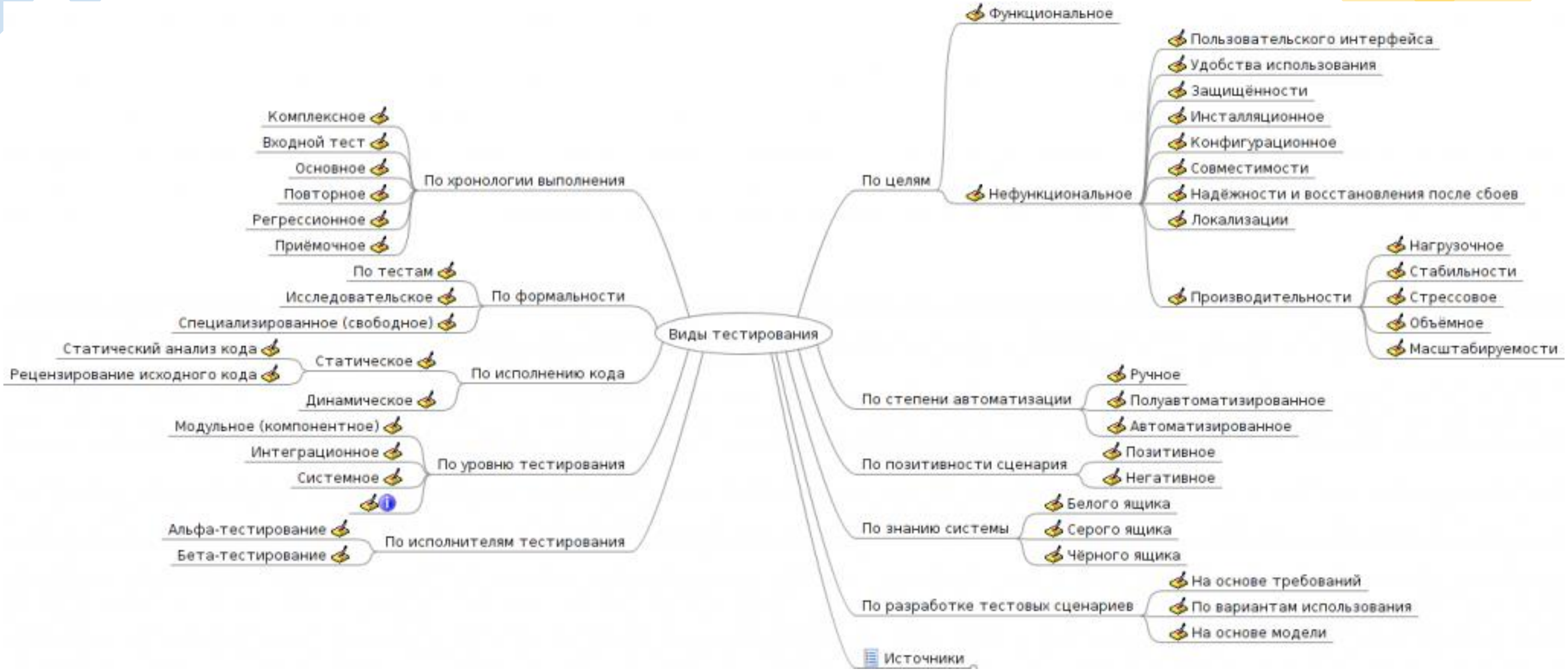
# Testing types

## Functional

verifies that each **function** of the software application operates **in conformance with** the requirement **specification**

## Non-Functional

verifies the readiness of a system as per **non-functional parameters** which are never addressed by functional testing



- Виды тестирования
  - По хронологии выполнения
    - Комплексное
    - Входной тест
    - Основное
    - Повторное
    - Регрессионное
    - Приёмочное
  - По формальности
    - По тестам
    - Исследовательское
    - Специализированное (свободное)
  - По исполнению кода
    - Статическое
      - Статический анализ кода
      - Рецензирование исходного кода
    - Динамическое
  - По уровню тестирования
    - Модульное (компонентное)
    - Интеграционное
    - Системное
  - По исполнителям тестирования
    - Альфа-тестирование
    - Бета-тестирование
  - По целям
    - Функциональное
    - Нефункциональное
      - Пользовательского интерфейса
      - Удобства использования
      - Защищённости
      - Инсталляционное
      - Конфигурационное
      - Совместимости
      - Надёжности и восстановления после сбоев
      - Локализации
      - Производительности
        - Нагрузочное
        - Стабильности
        - Стрессовое
        - Объёмное
        - Масштабируемости
  - По степени автоматизации
    - Ручное
    - Полуавтоматизированное
    - Автоматизированное
  - По позитивности сценария
    - Позитивное
    - Негативное
  - По знанию системы
    - Белого ящика
    - Серого ящика
    - Чёрного ящика
  - По разработке тестовых сценариев
    - На основе требований
    - По вариантам использования
    - На основе модели
  - Источники

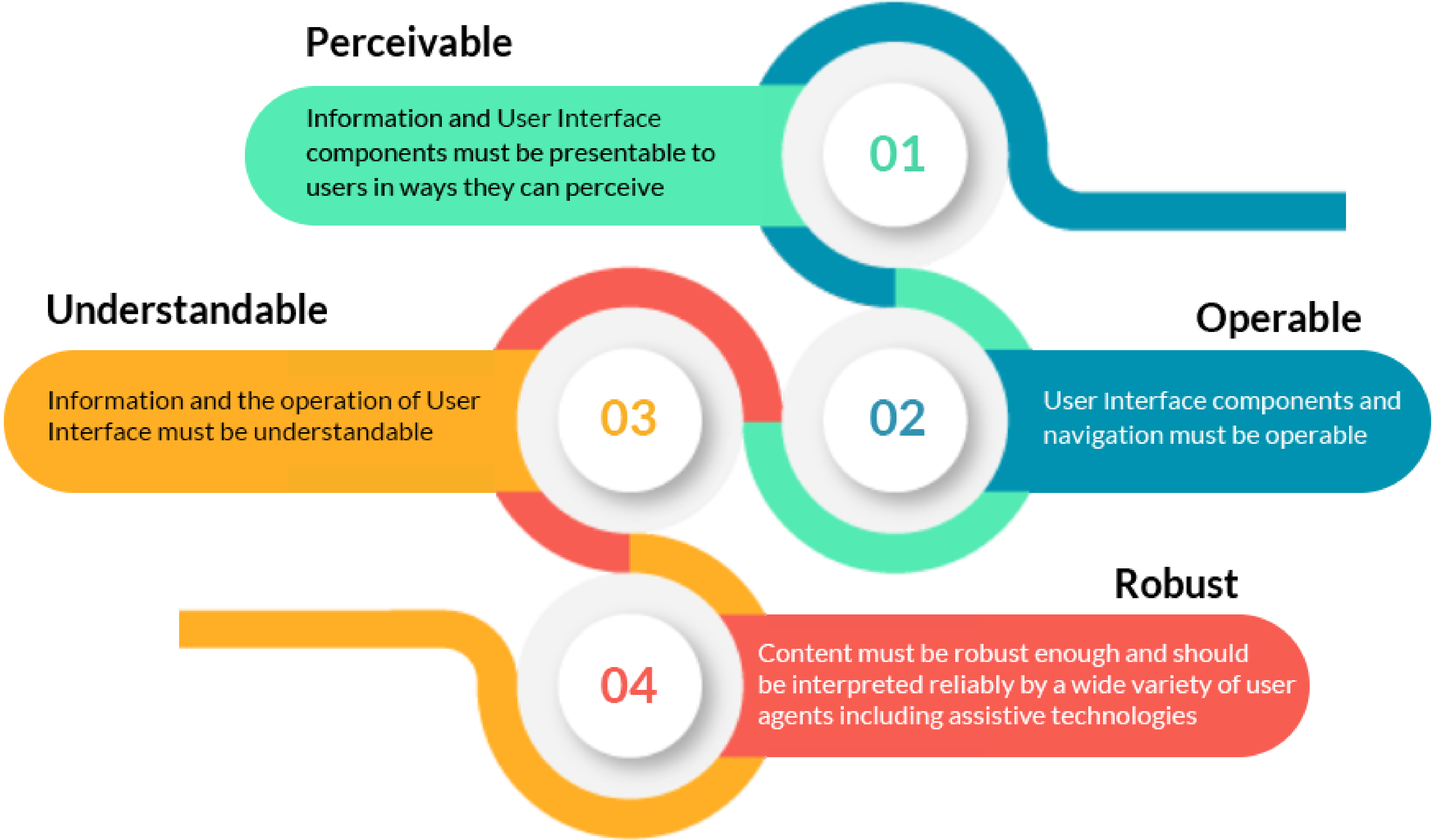
# Change related functional testing types

- **Sanity** – testing deep into certain functionality
- **Smoke** – high-level testing of main functionalities
- **Regression** - ensures that old code still works once the new code changes are done
- **End-to-end** (e2e) – complex testing of a real user scenario from start to finish, including integration with external systems
- **Ad-hoc** – informal, unstructured testing type with an aim to break the system, no test cases planned
- **Exploratory** - is all about discovery, investigation and learning, no test cases planned in advance, no documentation

# Non-Functional test types

- **Performance** – measures response times with a single user or several users exercising the system
- **Load** – checks behavior of the system under a specific load (many users work simultaneously like in real life)
- **Stress** - checks the upper limits of capacity within the system giving the extreme load
- **Security** - identifies the threats in the system and measure its potential vulnerabilities, helps in detecting all possible security risks in the system
- **Usability** - done from an end-user's perspective to determine if the system is easily usable, is user-friendly
- **Accessibility** – a subset of Usability Testing, which checks that the application is usable by people with disabilities like hearing, color blindness, old age and other disadvantaged groups.
- **Failover** and **Recovery** – determines whether operations can be continued after a disaster or after the integrity of the system has been lost.

# Accessibility testing



**Perceivable**

01

Information and User Interface components must be presentable to users in ways they can perceive

**Operable**

02

User Interface components and navigation must be operable

**Understandable**

03

Information and the operation of User Interface must be understandable

**Robust**

04

Content must be robust enough and should be interpreted reliably by a wide variety of user agents including assistive technologies

# Test types by scenario purpose

determines that your application works as expected according to requirements documentation

ensures that your application can gracefully handle invalid input or unexpected user behavior

# Questions?

# Agenda

- What kind of testing exists?
- Test types
- Test levels
- **Practice**

# Practice

# HomeWork