



# ТРЕУГОЛЬНИК И КВАДРАНТЫ ТЕСТИРОВАНИ Я

ОСНОВНЫЕ ПОНЯТИЯ

# ТРЕУГОЛЬНИК ИЛИ ПИРАМИДА ТЕСТИРОВАНИЯ

- ГРУППИРОВКА ТЕСТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО РАЗНЫМ УРОВНЯМ ДЕТАЛИЗАЦИИ, КОТОРАЯ ДАЕТ ПРЕДСТАВЛЕНИЕ СКОЛЬКО ТЕСТОВ ДОЛЖНО БЫТЬ В КАЖДОЙ ИЗ ЭТИХ ГРУПП.



- **Модульные тесты** составляют основную часть автоматизированного тестирования.
- **Интеграционные тесты** занимают середину пирамиды, без использования пользовательского интерфейса (UI); Тестируя за пределами пользовательского интерфейса, можно тестировать входы и выходы API или сервисов без всех сложностей, которые вводит пользовательский интерфейс.
- **Тесты пользовательского интерфейса** размещаются на вершине пирамиды. Большая часть кода и бизнес-логики должна быть уже протестирована до этого уровня. Тесты интерфейса пишутся, чтобы убедиться, что сам

# ПРИНЦИП ПИРАМИДЫ ТЕСТИРОВАНИЯ

Пирамида тестирования, в том числе, помогает наглядно объяснить причины, почему количество Unit тестов должно быть больше чем интеграционных. Чем ниже находятся на пирамиде тесты, тем:

- проще и быстрее они разрабатываются
- ниже затраты на поддержку тестов
- быстрее скорость запуска атомарного теста
- выше уровень изоляции компонент между собой
- меньше нужно денег на содержание инфраструктуры для запуска этих тестов

# ПЕРЕВЕРНУТАЯ ПИРАМИДА ТЕСТИРОВАНИЯ ИЛИ "РОЖОК МОРОЖЕННОГО"

## Основные тезисы:



1. Тесты пользовательского интерфейса должны автоматизироваться в большей мере.
2. Длинные тестовые прогоны, так как тестирование основано на взаимодействии с визуальными элементами пользовательского интерфейса и не обязательно имеет хуки в исходном коде.
3. Сложно поддерживать, так как тесты пользовательского интерфейса сложно писать и они очень сильно зависят даже от малейших изменений;
4. Больше подходит для сценариев позитивного пути. Тестирование отрицательных путей в сквозных тестах очень затратно и долго выполняется по сравнению с тестами более низкого уровня;
5. Ожидание написания модульных тестов до тех пор, пока функции не будут завершены, может привести к тому, что каждому придется несколько раз выполнить большую работу для решения проблемы.

**КВАДРАНТ ТЕСТИРОВАНИЯ** - НЕКИЙ ЧЕК-ЛИСТ, ПОКАЗЫВАЮЩИЙ, ЧТО МЫ ИДЕМ В ВЕРНОМ НАПРАВЛЕНИИ, ЧТО НАШИ ПРОЦЕССЫ В ДОСТАТОЧНОЙ СТЕПЕНИ АВТОМАТИЗИРОВАНЫ И МОЖЕМ ЗАНИМАТЬСЯ РУЧНЫМИ ИССЛЕДОВАТЕЛЬСКИМИ ТЕСТАМИ. ПОКАЗЫВАЕТ В ИТОГЕ НАСКОЛЬКО УСПЕШЕН НАШ ПРОДУКТ, С ТОЧКИ ЗРЕНИЯ БИЗНЕСА.



- Сектор Q1 представляет собой такую известную практику, как TDD. Все просто: сначала тесты, потом код.
- Сектор Q2 содержит в себе более высокоуровневые тесты, которые уже имеют отношение к бизнесу. Успешное прохождение этих тестов является обязательным, чтобы говорить о завершенности какого-либо функционала.
- Тесты из сектора Q3 подразумевают обязательное участие экспертов в доменной области, экспертов по usability, пользователей и так далее.
- Сектор Q4, говорит о тех тестах, которые не проверяют решение задач бизнеса, а говорят о том, насколько же хорошо написано наше приложение, используя такие термины как безопасность, производительность, устойчивость, масштабируемость и так далее.