

# Функциональное тестирование: наивный подход

**№ урока:** 7    **Курс:** Тестирование ASP.NET Core MVC приложений

**Средства обучения:** Visual Studio 2019 Community Edition

## Обзор, цель и назначение урока

Вспомнить, что такое функциональный тест. Разобраться со структурой функциональных тестов, рассмотреть признаки хорошего функционального теста и типичные ошибки. Рассмотрение демо проекта.

## Изучив материал данного занятия, учащийся сможет

- Разбираться в том, как правильно писать функциональные тесты.
- Разбираться в том, какие признаки хорошего функционального теста.
- Разобраться в том, как избежать типичных ошибок при реализации функционального теста.
- Рассмотреть на практике, как и с чего начинать писать функциональные тесты.

## Содержание урока

1. Что такое функциональное тестирование
2. Типы функционального тестирования
3. Признаки хорошего функционального теста
4. Структура функционального теста
5. Типичные ошибки функционального тестирования
6. Демо

## Резюме

- **Функциональное тестирование** – тестирование реализации функциональных требований программы.
- **Хороший функциональный тест** – понятный для нетехнических специалистов, максимально устойчивый к рефакторингу и содержит оптимальное количество деталей тестирования.
- **Плохой функциональный тест** – перегруженный деталями (или наоборот, недогруженный), непонятный для всех и зависит от технологий проекта.
- **Так как функциональные тесты тесно связаны с требованиями программы** они должны быть понятны тем, кто эти требования пишет, на практике это бизнес специалисты. Основной недостаток наивного подхода именно в том, что он абсолютно непонятен для нетехнических специалистов.
- **Функциональные тесты** могут проверять требования и бизнес-процессы.
- **Требования** – формализованный список того, как приложение должно работать. В фокусе находятся намерения системы. Требования зачастую очень обширные и детальные до мелочей.
- **Бизнес-процессы** – в свою очередь проверяется некий логический связанный аспект функционирования бизнес-логики с точки зрения его участников.
- **Структура функциональных тестов:** Given, When, Then.

- **Структура функциональных тестов очень похожа** на структуру unit и интеграционных тестов, отличие только в ключевых словах. Смысл, который стоит за ними абсолютно такой же – сначала задаём предусловия, выполняем действие, проверяем результат.
- **Интеграционные тесты и функциональные тесты очень похожи** по своей структуре – они оба проверяют приложение и его взаимодействие. Отличие в том, что интеграционные тесты сконцентрированы на технических аспектах работы, в то время как функциональные тесты сконцентрированы на проверке того, ЧТО именно делает приложения.
- **В случае функционального тестирования случайно сгенерированные данные мало используются** – так как все предусловия мы задаём заранее, возможно использование заранее готовых данных и предусловий.

### Закрепление материала

- Что такое функциональный тест?
- Назовите признаки хорошего функционального теста.
- Что может проверять функциональный тест?
- Какова типичная структура функционального теста?
- Каковы типичные ошибки в функциональных тестах?

### Дополнительное задание

#### Задание

Напишите функциональные тесты на некоторый функционал в ASP.NET Core MVC программе, без реализации этого функционала.

### Самостоятельная деятельность учащегося

#### Задание 1

Выучите основные понятия, рассмотренные на уроке.

#### Задание 2

Дополнительно найдите сопутствующую информацию по теме. Составьте сводную таблицу по основным понятиям, найдите схожие и отличающиеся моменты в описаниях, постарайтесь сформулировать своё понимание данной темы.

#### Задание 3

Напишите функциональные тесты на бизнес-логику вашего проекта из предыдущих уроков. Используйте любой тестовый фреймворк на языке программирования C# и любые библиотеки, помогающие написанию функциональных тестов.

### Рекомендуемые ресурсы

Test ASP.NET Core MVC apps

<https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/test-asp-net-core-mvc-apps>

Functional testing

<http://softwaretestingfundamentals.com/functional-testing>

User story vs requirements

<https://blog.testlodge.com/user-story-vs-requirements>