



## Современный Framework для тестирования: TUNIT



**Щелкунова Елена**

Программист

# Обо мне

- Елена Щелкунова, компания Directum
- Работаю программистом с 2010 года
- Работала в 7 фирмах за это время
- Full-stack разработчик (C# / JavaScript – React)
- Много работала с legacy-кодом в крупных проектах
- В текущем проекте пишем тесты на каждую фичу



# Microsoft Testing Platform

# Microsoft Testing Platform

- Альтернатива VsTest
- Open-source ([microsoft/testfx](https://github.com/microsoft/testfx))
- Native support начинается с .NET 10
- В .NET 9 и ранее нужно включать флажком:

```
<TestingPlatformDotnetTestSupport>true</TestingPlatformDotnetTestSupport>
```

- Разработано Microsoft
- Устанавливается как NuGet пакет
- Версия 1.0.0 вышла 24.01.2024г.

# Зачем?

- VsTest разрабатывался еще для старого .NET Framework
- Поддержка новых фич и форм-факторов .NET, включая native AOT
- Избавление от рефлексии (reflection)
- Hostable. Теперь этот exe-файл запускается на любом устройстве с .NET
- Быстрая (Performant)
- Расширяемая (сведения о тестовом процессе, фреймворке)
- Поддержка нескольких тестовых фреймворков

# Поддерживаемые тестовые фреймворки

- MSTest. Поддержка через [MSTest runner](#).
- NUnit. Поддержка через [NUnit runner](#).
- xUnit.net. [xUnit.net v3](#) и [xUnit.net v2](#)
- TUnit: полностью основан на Microsoft.Testing.Platform. [TUnit documentation](#).

# TUnit

Nov 5, 2025

 github-actions

 v1.0.0

 4bab947 

[Compare ▾](#)

# 1.0.0

It's been a long time coming! Here's version 1.0!

## What's Changed

### Breaking Changes

- Refactor TestContext and related interfaces for tidiness and cohesion

Snippets on the refactored API can be found here: <https://tunit.dev/docs/>

### Changes

- Implement GetNestedClassName method by [@thomhurst](#) in [#3708](#)

### Dependencies

- chore(deps): update tunit to 0.90.45 by [@t](#)
- chore(deps): update dependency benchmarks by [@t](#)
- chore(deps): update dependency benchmarks by [@t](#)

Full Changelog: [v0.90.45...v1.0.0](#)



 Sponsor

[Follow](#)

thomhurst Tom Longhurst

Developer of TUnit + Hobbyist Photographer

 PRO

 London, UK

 19:57 - 3h behind

 Owns this repository

 Committed to this repository in the past day

 Answered 1 discussion in this repository in the past week

 Started 3 discussions in this repository

# 1.12.65

## What's Changed

### Other Changes

- docs: fix incorrect API references in [MethodAssertionGenerator](#) 8
- feat: make [MethodAssertionGenerator](#) 8
- fix: make thread assertion tests deterministic

# Почему TUNIT?

- Тестовый фреймворк (аналог NUnit, xUnit, MSTest)
- Объединил в себе лучшие практики из NUnit, xUnit, MSTest
- Комьюнити-поддержка
- Open-source, MIT license
- Богатая документация, включая QuickStart и Migration Guides с NUnit, xUnit, MSTest

# Основные принципы TUnit

- Производительность
- Поддержка AOT и Trimming Поддерживает фичи языка C# 12+ и фичи .NET свежих версий
- Изоляция тестов
- Параллельное исполнение по умолчанию
- Поддержка асинхронности через async (в т.ч. для Assert)
- Assert'ы с помощью FluentAssertions
- Богатый набор атрибутов для управления жизненным циклом тестов (в т.ч. кастомные)
- Больше хуков для управления жизненным циклом, чем в xUnit

# Пример теста

```
[Test]  
public async Task User_Creation_Should_Set_Timestamp()  
{  
    // Arrange  
  
    var userService = new UserService();  
  
    // Act  
  
    var user = await userService.CreateUserAsync("john.doe@example.com");  
    // Assert - TUnit's fluent assertions  
  
    await  
    Assert.That(user.CreatedAt).IsEqualTo(DateTime.Now).Within(TimeSpan.FromMinutes(1));  
    await Assert.That(user.Email).IsEqualTo("john.doe@example.com");  
}
```

# Тест с входными аргументами

```
[Test]
[Arguments("user1@test.com", "ValidPassword123")]
[Arguments("user2@test.com", "AnotherPassword456")]
[Arguments("admin@test.com", "AdminPass789")]
public async Task User_Login_Should_Succeed(string email,
string password)
{
    var result = await authService.LoginAsync(email,
password);
    await Assert.That(result.IsSuccess).IsTrue();
}
```

# Тест с комбинацией входных параметров

```
[Test]
[MatrixDataSource]
public async Task Database_Operations_Work(
    [Matrix("Create", "Update", "Delete")] string operation,
    [Matrix("User", "Product", "Order")] string entity)
{
    await Assert.That(await ExecuteOperation(operation,
entity)).IsTrue();
}
```

# Пример с generic-классом

```
public class GenericTestExample<T>
{
    [Test]
    public void GenericTest()
    {
        // This is a test in a generic class
        Console.WriteLine($"Running test with type: {typeof(T).Name}");
    }

    [Test]
    [Arguments(5)]
    [Arguments("hello")]
    public void GenericTestWithArguments<TArg>(TArg value)
    {
        Console.WriteLine($"Running test with value: {value} of type:
{typeof(TArg).Name}");
    }
}
```

# Оркестрация тестов

```
[BeforeClass]
```

```
public static async Task SetupDatabase(ClassHookContext context)  
{ await DatabaseHelper.InitializeAsync(); }
```

```
[Test, DisplayName("Register a new account")]
```

```
[MethodDataSource(nameof(GetTestUsers))]
```

```
public async Task Register_User(string username, string password)  
{ // Test implementation }
```

```
[Test, DependsOn(nameof(Register_User))]
```

```
[Retry(3)] // Retry on failure
```

```
public async Task Login_With_Registered_User(string username, string password)  
{ // This test runs after Register_User completes }
```

# Кастомные атрибуты

```
public class LoadTestParallelLimit : IParallelLimit
{
    public int Limit => 10; // Limit to 10 concurrent executions }

[Test]
[ParallelLimiter<LoadTestParallelLimit>] // Custom parallel control
[Repeat(100)] // Run 100 times

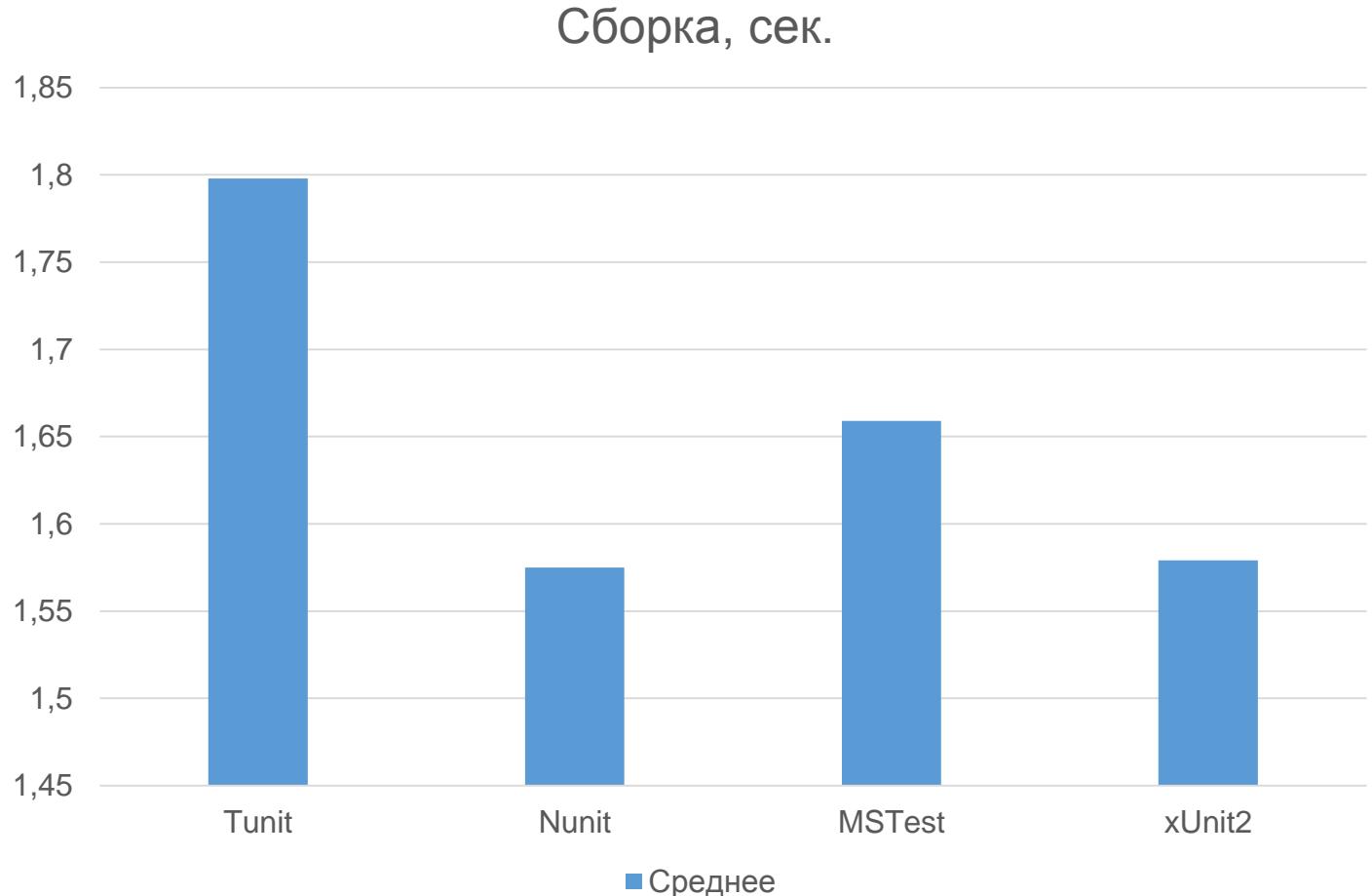
public async Task Load_Test_Homepage()
{
    // Performance testing
}

// Custom attributes
[Test, WindowsOnly, RetryOnHttpError(5)]

public async Task Windows_Specific_Feature()
{
    // Platform-specific test with custom retry logic
}
```

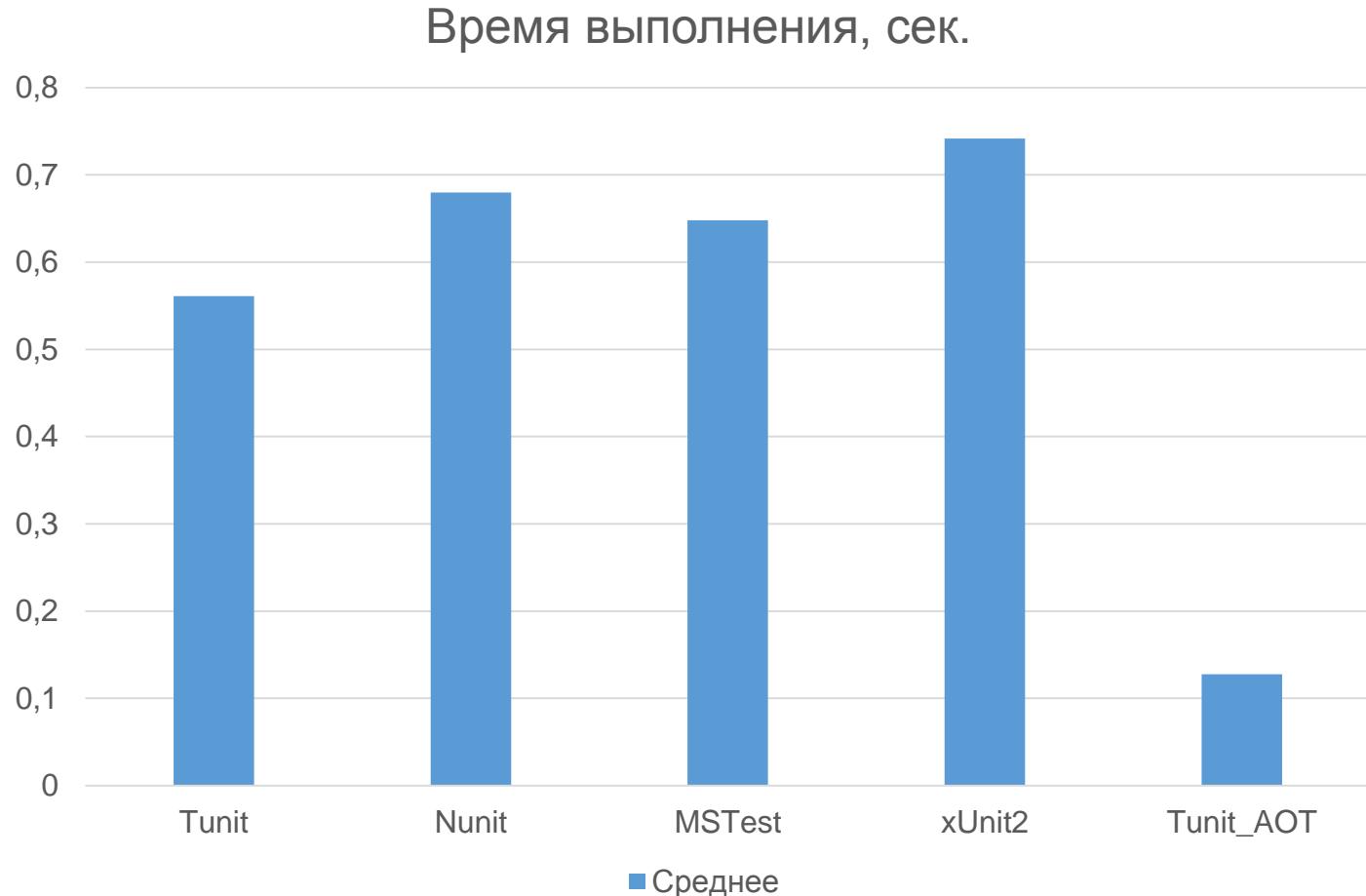
# Сборка тестового проекта

- BenchmarkDotNet v.0.16.6, Linux Ubuntu 24.04.3 LTS (Noble Numbat)
- AMD EPVC 7763 2.45GHz, 1 CPU, 4 logical and 2 physical cores
- .NET SDK 10.0.100-rc.2.25502.107
- Runtime = .NET 10.0



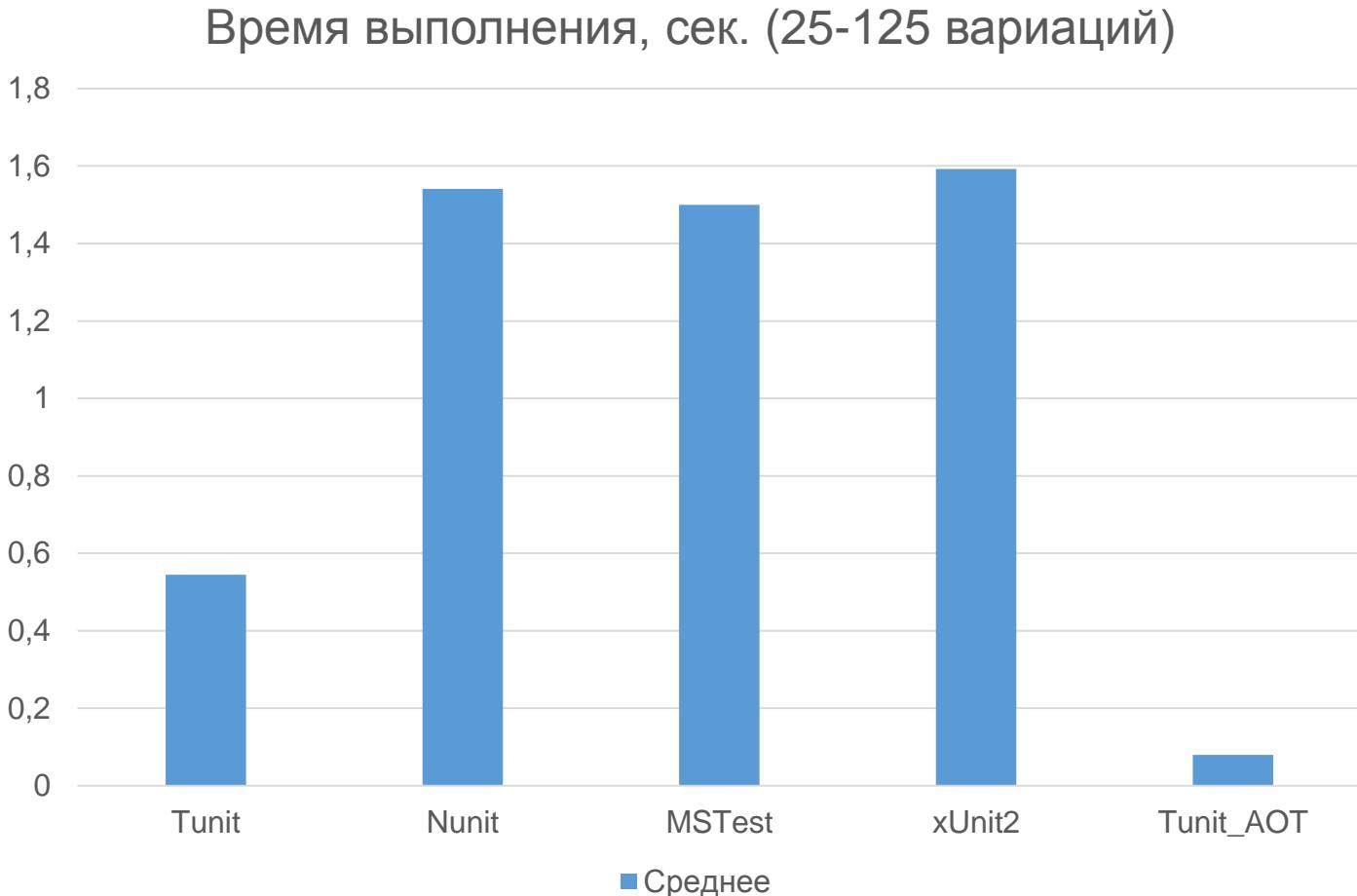
# Выполнение асинхронных тестов

- BenchmarkDotNet v.0.16.6, Linux Ubuntu 24.04.3 LTS (Noble Numbat)
- AMD EPVC 7763 2.45GHz, 1 CPU, 4 logical and 2 physical cores
- .NET SDK 10.0.100-rc.2.25502.107
- Runtime = .NET 10.0



# Сложные комбинации параметров

- BenchmarkDotNet v.0.16.6, Linux Ubuntu 24.04.3 LTS (Noble Numbat)
- AMD EPVC 7763 2.45GHz, 1 CPU, 4 logical and 2 physical cores
- .NET SDK 10.0.100-rc.2.25502.107
- Runtime = .NET 10.0



# Полезные ссылки

- [Readme.md TUNIT](#)
- [Обзор Microsoft Testing Platform](#)
- [Диаграмма жизненного цикла тестов](#)
- [Миграция с VSTest на Microsoft Testing Platform](#)
- [Философия и дизайн TUNIT](#)
- [Видео-обзор TUNIT для начинающих](#)

# Спасибо!