

ПОРІВНЯЛЬНИЙ АНАЛІЗ ФРЕЙМВОРКІВ ЮНІТ-ТЕСТУВАННЯ В СЕРЕДОВИЩІ .NET

Вінницький національний технічний університет

Анотація

Проаналізовано основні сфери застосувань фреймворків юніт-тестування. Проведено аналіз найпопулярніших фреймворків юніт-тестування для мови C#. Визначено переваги та недоліки фреймворків юніт-тестування.

Ключові слова: тестування, юніт-тестування, TDD, Microsoft, .NET, C#, NUnit, xUnit.

Abstract

The common usage spheres of unit-testing frameworks are analyzed. Most popular unit-testing frameworks for C# are analyzed. The advantages and disadvantages of unit-testing frameworks are determined.

Keywords: testing, unit-testing, TDD, Microsoft, .NET, C#, NUnit, xUnit.

Вступ

Відомо, що вартість виправлення дефектів програмного забезпечення зростає експоненціально з переходом до кожної наступної стадії життєвого циклу розробки ПЗ [1]. За даними [2] 53% усіх проєктів завершуються з перевитратами. Тому, щоб зменшити витрати на виправлення помилок та прискорити процес розробки, необхідно починати процес тестування з ранніх стадій розробки ПЗ.

Сьогодні активного поширення набуває технологія розробки ПЗ через тестування (TDD, test-driven development), що базується на створенні автоматизованих модульних (юніт) тестів, які визначають вимоги до коду безпосередньо перед його написанням [3]. Такий підхід дозволяє не лише виявити помилки ПЗ на ранній стадії, а й в подальшому значно прискорити процес регресійного тестування при внесенні змін в існуючий код, оскільки великі обсяги програмного коду тестуються автоматично.

При створенні юніт-тестів розробки використовують спеціальні фреймворки (бібліотеки), які покликані спростити процес юніт-тестування.

Мета дослідження – визначення переваг та відмінностей найпопулярніших бібліотек юніт-тестування.

Об'єктом дослідження є процес юніт-тестування в середовищі .NET.

Предмет дослідження – основні якісні характеристики фреймворків юніт-тестування.

Результати дослідження

Юніт-тестування або модульне тестування – це метод тестування програмного забезпечення, який дозволяє перевірити на правильність окремі модулі вихідного коду програми. Ідея полягає в написанні тестів для кожної функції або методу. Перевагою такого методу є ізоляція окремих частин додатку і доказ, що кожний окремих модуль коректний і виконує очікувану поведінку [4]. Інші переваги юніт-тестування включають в себе легкий рефакторинг коду тестів, спрощене інтеграційне тестування, самодокументування тестів шляхом використання правильного синтаксису та простий дизайн кожного окремого тесту й системи тестування загалом. Кожна мова та платформа програмування надає готові реалізації бібліотек (фреймворків) юніт-тестування. Найпопулярнішими фреймворками для платформи .NET на сьогодні є MSTest, NUnit та xUnit:

- MSTest – стандартний фреймворк тестування Visual Studio, який встановлюється разом із редактором за замовчуванням. Відсутність додаткових дій для отримання функціоналу бібліотеки юніт-тестування однозначно є головною перевагою фреймворку від Microsoft.
- NUnit – фреймворк юніт-тестування, який створювався як інтерпретація з іншої мови

програмування Java, але врешті решт трансформувався повністю під потреби та логіку мови C#. Це рішення має довгу історію адаптовану під середовище .NET. NUnit має різні засоби запуску, але не є інтегрованою за замовчуванням в основний засіб програмування мовою C# – Visual Studio. Сьогодні бібліотека широко застосовується для модульного, інтеграційного та автоматизованого тестування.

- xUnit – бібліотека юніт-тестування, яка була створена одним з авторів NUnit. Вона відрізняється переосмисленням суті юніт-тестів, розділяючи їх на “факти” та “теорії”, що розділяє логіку на “завжди вірно” та “вірно для правильних вхідних даних”.

На рисунку 1 продемонстровано динаміку зацікавленості по невбудованим у Visual Studio фреймворкам юніт-тестування, яка свідчить, що користувачі віддавали перевагу NUnit, але постійне розширення та розповсюдження збільшує відсоток користувачів xUnit відносно інших бібліотек [5].

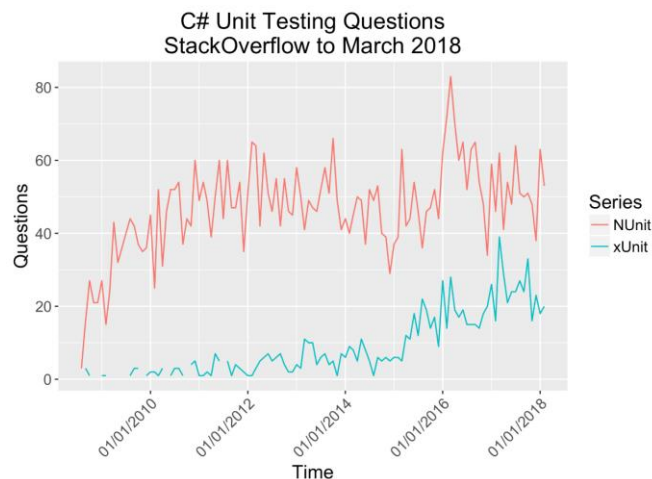


Рисунок 1 – Графік зацікавленості розробників на онлайн ресурсі по NUnit та xUnit

Головними критеріями вибору правильної бібліотеки юніт-тестування для окремої задачі є швидкість виконання, легкість інтеграції, набір необхідних функцій та повнота інформації, що супроводжує фреймворк. Необхідно зважати на ці критерії при порівнянні бібліотек.

Розглянемо три найпопулярніших фреймворки юніт-тестування: MSTest, NUnit та xUnit. Усі вони можуть обробляти величезні обсяги тестів та вхідних даних, хоча відносно інших основною проблемою MSTest відзначається невисока продуктивність бібліотеки. Для невеликих задач питання інтеграції сторонніх бібліотек ускладнює рішення. Розглядаючи Visual Studio як основну платформу для .NET розробки, MSTest є стандартною бібліотекою вбудованою в середовище програмування, а отже надає усі основні функції одразу після встановлення Visual Studio. В порівнянні, сторонні бібліотеки, NUnit та xUnit необхідно додатково встановлювати через пакетні менеджери. Будучи фреймворками з відкритим кодом, xUnit й NUnit надають ширший функціонал та можливості, такі як запуск тестів не лише з середовища програмування, а й окремо від нього, покращений паралелізм, гнучкість системи й специфічні покращення та семантичні зміни. Маючи базу користувачів та людей, які доповнюють документацію для фреймворків з відкритим кодом й компанію Microsoft, що стоїть за MSTest, документація усіх бібліотек є достатньою для роботи на будь-якому рівні, хоча xUnit як більш розширений та логічно змінений фреймворк є більш складним для початку роботи з ним.

Висновки

Таким чином, виявлено, що найпоширенішим фреймворком юніт-тестування є NUnit і він слугує гарним прикладом бібліотеки з відкритим кодом. Незалежно від обраної бібліотеки, вони усі надають базові функції для написання тестів. MSUnit найкраще інтегрований з Visual Studio, основним засобом для написання .NET додатків. NUnit сьогодні має найбільшу базу користувачів, а отже велику кількість рішень й перевірений статус. В той час xUnit представляє більш сучасний варіант юніт-тестування з дещо зміненими семантичними ідеями. Знаючи розбіжності бібліотек завжди можливо підібрати фреймворк, який найкраще вирішить поставлену проблему.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Поморова О.В. Сучасні проблеми оцінювання якості програмного забезпечення / О.В. Поморова, Т. О. Говорущенко // Радіоелектронні та комп'ютерні програми : наук.-техн. журнал. – Національний лісотехнічний університет України 294 Серія економічна Харків : Вид-во НАУ ім. М.Є. Жуковського "Харківський авіаційний інститут". – 2013. – № 5. – С. 319-327.
2. Мищенко В.О. CASE-оценка критических программных систем. – В 3-ох т. – Т. 1: Качество / В.О. Мищенко, О.В. Поморова, Т. А. Говорущенко; под ред. В.С. Харченко. – Харьков : Изд-во НАУ им. Н.Е. Жуковского "ХАИ", 2012. – 201 с. 1.
3. Murphy Craig. Improving Application Quality Using Test-Driven Development / Craig Murphy. [Electronic resource]. – Mode of access <http://www.Methodsandtools.co.in/archive/archive.php?id=20>.
4. Ошеров Р. The art of unit testing with examples in .NET – Гринвіч: Manning Publication Co, 2009. — 20 с.
5. Unit testing frameworks in C#: Comparing XUnit, NUnit and Visual Studio [Електронний ресурс] // Режим доступу: <https://raygun.com/blog/unit-testing-frameworks-c/>

Василевський Володимир Олегович, студент групи ІПІ-16б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: vovavasilevskiy@gmail.com

Романюк Оксана Володимирівна, к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: romaniukoksanav@gmail.com

Vasylevskiy Volodymyr, student of group ІPI-16b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: vovavasilevskiy@gmail.com

Oksana Romaniuk, Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: romaniukoksanav@gmail.com