Technology Proposal: xUnit
MotoMoto

The New Panelists
12/15/2021

Blake Del Rey
Isabel Guzman
Jacob Sunia
James Austin Jr.
Naeun Yu

Team Leader: James Austin Jr.

Table of Contents

1. xUnit.net - Version 2.4.3

1.1.1 Overview

xUnit is an open-source automated unit testing tool for the .NET framework. xUnit is an excellent unit testing tool because it allows for running unit tests in parallel, allows for completely independent unit tests, and catches even the most generic test case exceptions.

1.1.2 Metric Scale

The metrics are scored between 0 and 1 according to how well the competitor aligns with our purpose stated above.

0	Poor
0.25	Some value
0.5	Average among competitors
0.75	Good but has disadvantages
1	Best among all competitors, no disadvantages

1.2 Purpose

We would like to request this technology for use in the development and maintenance of MotoMoto because it will allow our development team to automatically test every component of each build we create for our application in a quick and efficient manner. By using this technology we will be able to ensure that we catch broken code before we deploy it to production and before it impacts any unintended parts of our application.

1.3.1 Competition Analysis

The decision to request xUnit as our automated unit testing technology was made after we compared various quantifiable metrics between the three leading tools for unit testing in the .NET framework. The three technologies that were tested were MSTest version 2.2.8, NUnit version 3, and xUnit version 2.4.3:

Metrics:

Single Unit Test Speed:

- Average time in milliseconds to complete a single unit test (Averaged with data from 10,000 tests).
    - NUnit: 1880 milliseconds
    - xUnit: 1839 milliseconds
    - MSTest: 1924 milliseconds

  Observations: MSTest is significantly slower in running a single unit test than both NUnit and xUnit. While xUnit and NUnit are very similar in runtime, over time those milliseconds saved by using xUnit would save exponential amounts of time.

Parallel Unit Testing Speed:

- Completion of 100 unit tests performed in parallel with max threads set to 30
    - NUnit: 6231 milliseconds
    - xUnit: 6022 milliseconds
    - MSTest: 6422 milliseconds

  Observations: As expected based on the single unit test completion speed, when parallel running numerous unit tests, xUnit remains the fastest of the three testing frameworks. The difference between xUnit and NUnit in speed is much more apparent when parallel testing.

Sequential Unit Testing Speed:

- Completion time of 100 unit tests performed sequentially.
    - NUnit: 3 minute and 12 seconds
    - xUnit: 3 minutes and 1 second
    - MSTest: 3 minutes and 20 seconds

  Observations: Despite sequential unit testing being the least likely form of unit testing for us to use, this testing was important in solidifying that xUnit is in fact the fastest testing framework of it and it's competitors. Even with just 100 unit tests there is an almost 20 second difference in testing speed between xUnit and MSTest and an over 10 second difference in testing speed between xUnit and NUnit.

Analysis Table:

	Multiplier	MSTest 2.2.8	NUnit 3.0	**xUnit 2.4.3**
Single Unit Test Speed (10K tests)	1.2	0.5	0.75	1.0
Parallel Unit Testing Speed (100 tests, 30 threads)	1.1	0.25	0.5	1.0
Sequential Unit Testing Speed (100 tests)	1	0.5	0.75	1.0
Score		1.375	2.2	3.3

1.3.2 Key Advantages of xUnit

Based on the extensive speed testing displayed above it is beyond evident that xUnit is the fastest unit testing framework of the three industry standards that have been analyzed. Despite NUnit being fairly close in speed to xUnit in all three tests, by the data shown in the sequential testing we can see that after only 100,000 unit tests choosing xUnit will save over 305 hours when compared to NUnit and will save over 527 hours when compared to MSTest. For parallel testing, which is what will be the most used form of testing by MotoMoto, after 1,000,000 Unit Tests choosing xUnit will save over 58 hours when compared to NUnit and will save over 111 hours when compared to MSTest.

In addition to its speed advantages over NUnit and MSTest, xUnit has various other unquantifiable feature advantages when compared to its competitors. The first significant advantage it has is that it executes each test case as an individual class. This is significant because it means that the order by which each test case is run is completely inconsequential to the outcome of that test case. By avoiding test case dependency we will better be able to pinpoint where broken code is located if a test case is to fail. Another significant advantage of xUnit is that it is more versatile than its competitors due to its Theory attribute which allows for parameterized unit tests. Parameterized unit tests are incredibly useful because some components may only break with certain arguments passed into them and figuring out which arguments break a component may be useful. The last significant advantage of xUnit I would like to mention is that because it uses the Assert.Throws function to throw an error it can catch every type of error that is thrown whereas its competitors only catch the specified ExpectedException. For example if a competitor specified that it wants to catch error X and error Y is thrown, the test case would still pass, whereas with xUnit the test case wouldn't pass. For these reasons we believe xUnit to be the best unit testing technology for MotoMoto.