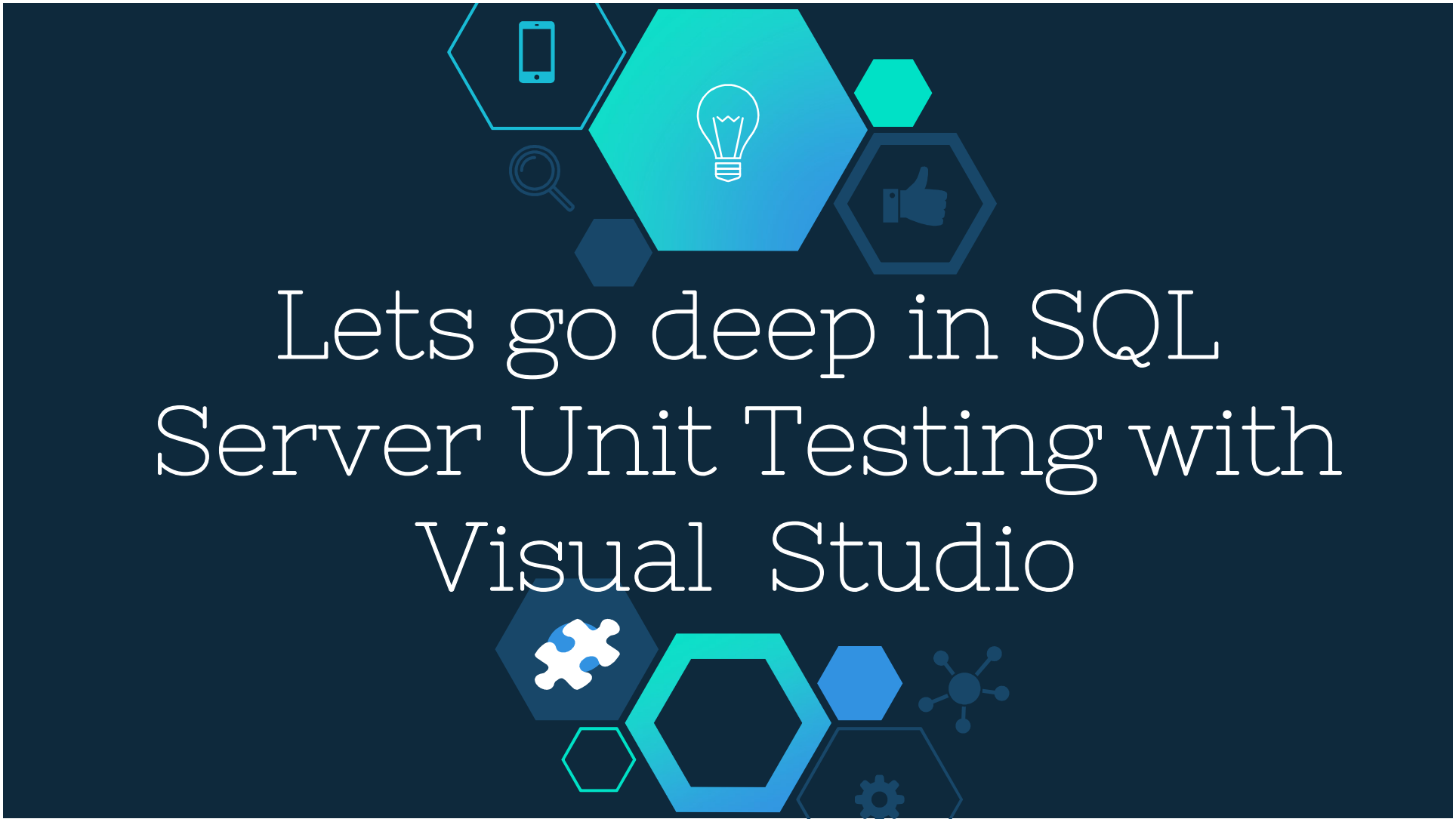

Lets go deep in SQL
Server Unit Testing with
Visual Studio

FEEDBACK FORMS

PLEASE FILL OUT AND PASS TO YOUR ROOM HELPER BEFORE YOU LEAVE THE SESSION

Geovanny Hernandez

DB Engineer - THE WORKSHOP™

https://www.geohernandez.net/

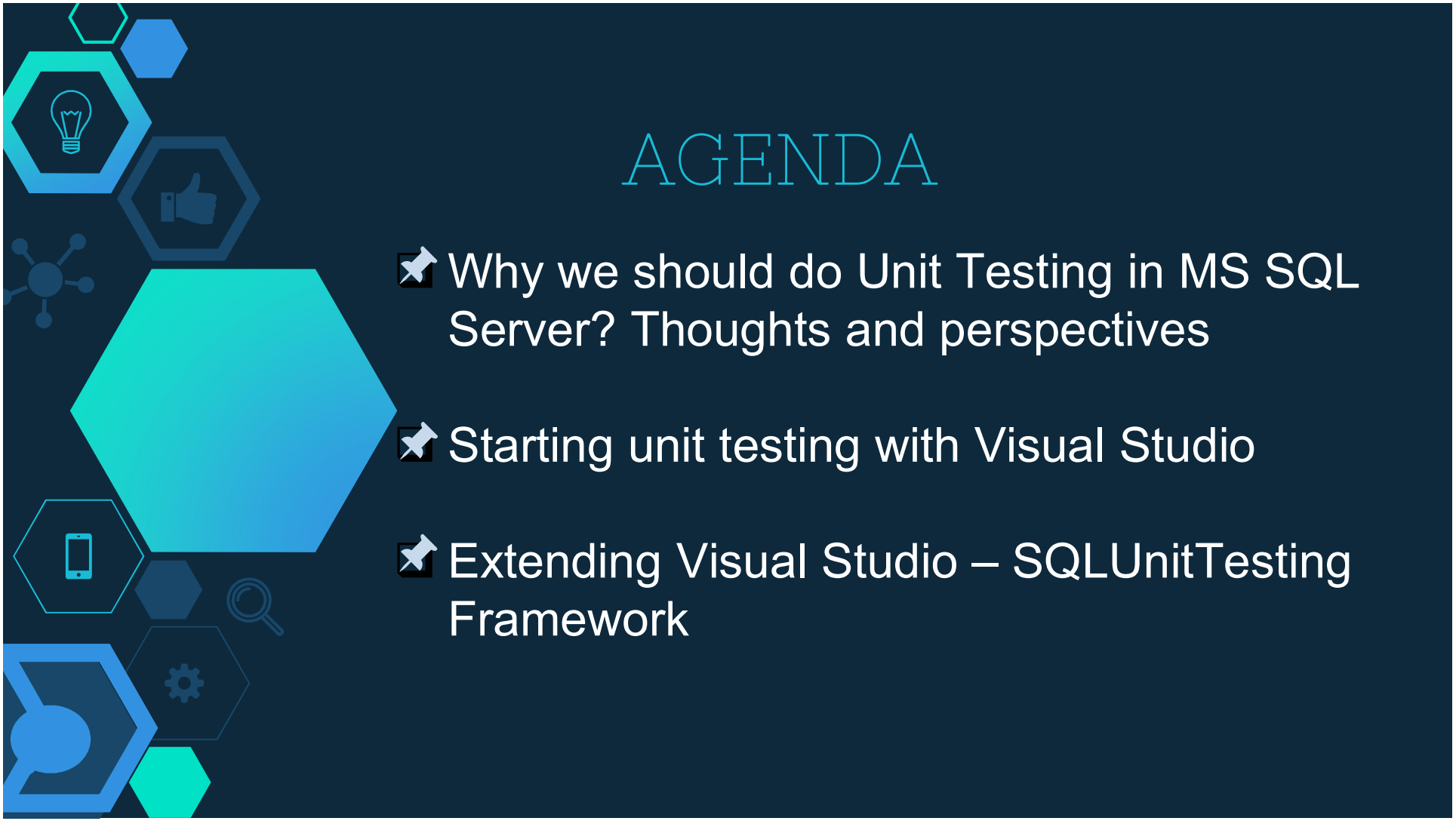
me@geohernandez.net

DB Engineer / MCT

Database Engineer and Microsoft Certified Trainer with more than twelve years of experience in software development, passionate for performance tuning in MS SQL Server, Unit Testing and Python.

PASS Volunteer / Data enthusiast

Originally from Nicaragua(central America) lives in Malaga, Spain with his wife and kids. Volunteer in the chapter of PASS Malaga, active member in MSDN forums and enjoys the beautiful beaches in La Costa del Sol.


AGENDA
Why we should do Unit Testing in MS SQL Server? Thoughts and perspectives
Starting unit testing with Visual Studio
Extending Visual Studio – SQLUnitTesting Framework

What is it a Unit Test ?

Should we apply unit testing on legacy code?

Why are we excluding the Unit Testing on Databases?

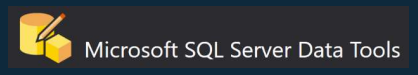
- It is hard to do automated tests on the DB
- The DB only stores data (Are you sure? ….)


Let's do it!

Unit Testing on Visual Studio

Requisites:

. Visual Studio 2015 or up 

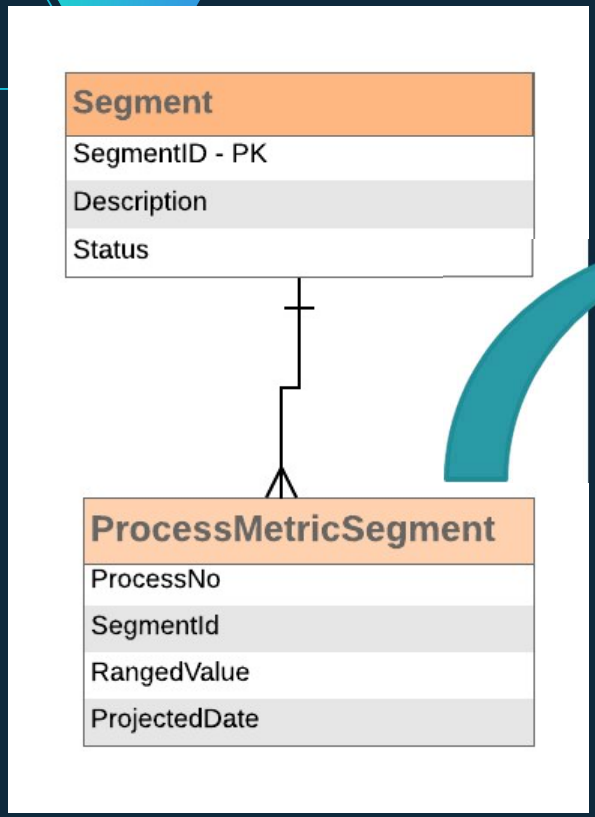
. SSDT 16.5 or up 

. MS SQL Server 2012 or up 

A unit test then it should be

- Automated and repeatable
- Run quickly
- Isolated
- Still valid through the time
- Deterministic

DBUnitTestDemo_DBFirst



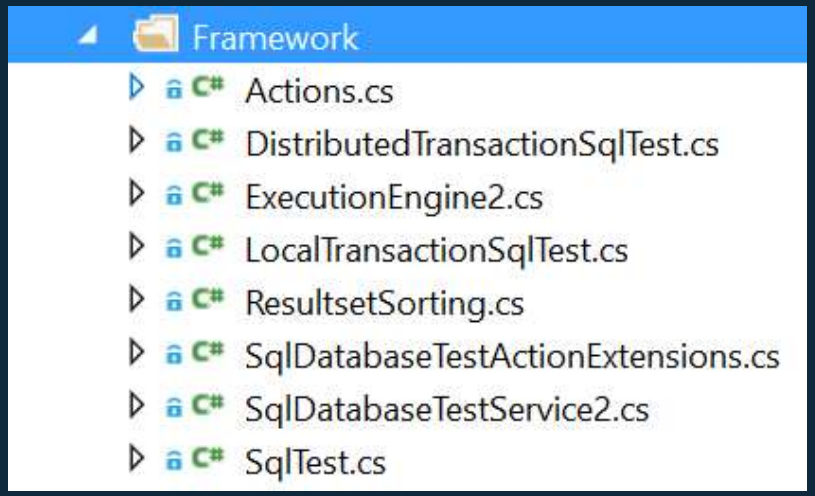

```
EXEC [dbo].[usp_SetProcessBySegment]
    @range_start =1 ,
    @range_end =3 ,
    @initial_date ='20190101',
    @segmentid = 1
```

Results | Messages

	ProcessNo	SegmentID	RangeValue	ProjectedDate
1	1	1	1	2019-01-02
2	2	1	2	2019-01-03
3	3	1	3	2019-01-04

Extending Unit Testing - SQLUnitTesting Framework

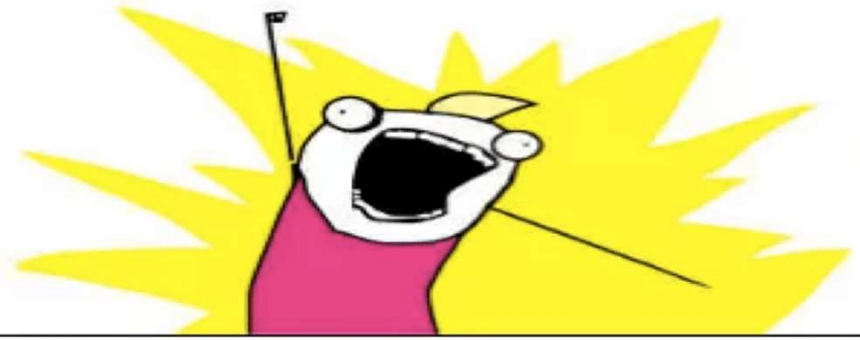
Extending Unit Testing - SQLUnitTesting Framework



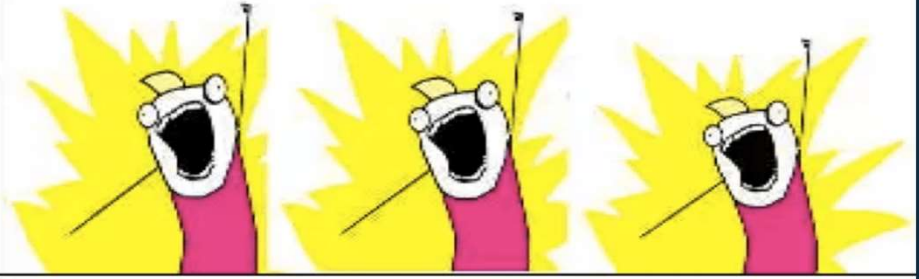

```
1  using Microsoft.Data.Tools.Schema.Sql.UnitTesting;
2
3  namespace SQLEasyTesting.Tests.Framework
4  {
       0 references | 0 changes | 0 authors, 0 changes
5      public static class Actions
6      {
           0 references | 0 changes | 0 authors, 0 changes
7          public static SqlDatabaseTestActions CreateBlock(string sql)
8          {
9              return new SqlDatabaseTestActions
10             {
11                 TestAction = CreateSingle(sql)
12             };
13         }
14
           1 reference | 0 changes | 0 authors, 0 changes
15         public static SqlDatabaseTestAction CreateSingle(string sql)
16         {
17             return new SqlDatabaseTestAction
18             {
19                 SqlScript = sql
```

https://github.com/SimpleSqlUnitTesting/SimpleSqlUnitTesting

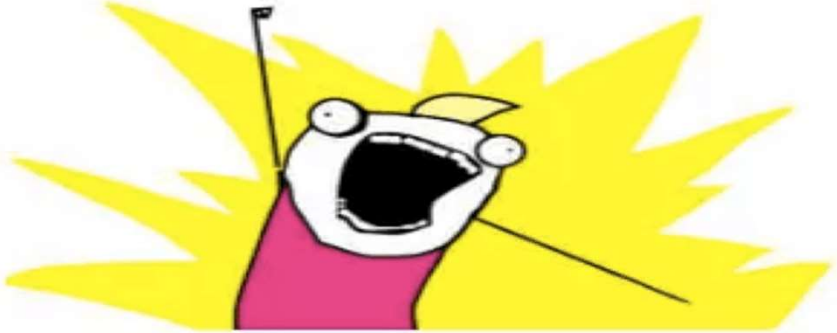
WHAT DO WE WANT?



TO DELIVER BETTER SOFTWARE!



WHEN DO WE WANT IT?



CONTINUOUSLY!!!

Remember …

- The DB objects are part of our software, therefore, there is no argument for NOT testing them.
- Unit Tests represent a valuable way of documenting our DB objects.
- Unit Testing is an approach to improve the way that we develop Database code.

Thank you for attending