

TEST DRIVEN DEVELOPMENT FOR T-SQL IT'S NOT JUST TESTING FIRST!

Philip Japikse (@skimedic)

phil.japikse@telerik.com

www.skimedic.com/blog

MVP, MCSD.Net, MCDBA, CSM, CSP
Patterns & Practices Evangelist, Telerik



WHO AM I?

- Patterns & Practices Evangelist, Telerik, Inc.
- Microsoft MVP
- MCSD, MCDBA, CSM, CSP
- Developer().Speaker().Writer()
- Lead Director, Cincinnati .NET User's Group
- Founder, Agile Conferences, Inc.

UNIT TESTING MOTIVATION

- Cost of correcting a defect (by phase)¹:
 - *Requirements* = \$139
 - *Design* = \$455
 - *Coding* = \$977
 - Unit testing targets coding -> deployment and maintenance
 - *System Testing* = \$7,136
 - *Maintenance* = \$14,102

¹B.Boehm and V. Basili, "Software Defect Reduction Top 10 List", IEEE Computer, January 2001

WHY REALLY?

- The Team
 - Confidence
 - Courage
 - Cadence

“The main thing that distinguishes legacy code from non-legacy code is tests, or rather a lack of tests.” – Michael Feathers

UNIT TESTING

UNIT OF WORK/UNIT TESTS

- Unit of Work
 - Smallest testable part of an application.
- Unit Test
 - Code used to validate units of work

UNIT TEST “CODE” OF HONOR

- Be Independent of all other unit tests
- Return the System Under Test to it's original state
- Note: Integration tests should follow these rules as well

T-SQL PARADIGM SHIFT

- Think Lego blocks, not feature sets
- Single Responsibility Principle (SRP)
- Separation of Concerns (SOC)
- Don't Repeat Yourself (DRY)

TEST STRUCTURE

- Arrange
- Act
- Assert

TDD MANTRA

- **Red** – write a test for expected results
- **Green** – write just enough code to have the test pass
- **Refactor** –
 - Eliminate
 - Duplicate/Complex code
 - Add Use Cases

SOURCE CODE CONTROL

- Commit early and often
- Run tests on each commit (CI)
- Use Data Dude or Red Gate's SQL Source Control

HANDLING SQUIRRELS

- Computers can multi-thread, people cannot
- Keep list of To-Dos
 - As ideas come up, write them down
 - Tackle them in order of confidence
- Finish what you start!
 - Don't context switch

UNIT TESTING T-SQL

RESOURCES

- T-SQL Test Tool
 - <http://www.codeplex.com/tst>
- MbUnit
- Managed Code Interface for TST
 - <http://www.codeplex.com/mci4tst>

DATABASE INSTANCES

- Use an empty copy of target database for testing
 - Must match target database schema
 - Prevents noise from occluding the results
- Add necessary records in the setup
- Test the Unit of Code
- Remove records in teardown
 - Best to use transacted test framework

TEST STORAGE

- Store the Tests in the empty clone, not production
 - Use Comparison tool (like Red Gate SQL Compare) to make sure the test database exactly matches the deployed database
 - Eliminates promotion issues

TEST STORED PROCEDURES

- Start with “SQLTest_”
- Full range of asserts in TST.Assert.<AssertType>
- Group in Suites
 - SQLTest_SuiteX#TestA

RUNNING TESTS

- EXEC TST.Runner.RunAll 'DatabaseName'
- EXEC TST.Runner.RunSuite 'DatabaseName', 'SuiteName'
- EXEC TST.Runner.RunTest 'DatabaseName', 'TestName'

TRANSACTION ISSUES

- TST is Fully Transacted
 - Don't use your own in tests
- Begin/Commit in SUT
 - Works as expected
- Begin/Rollback in SUT, two options:
 - Change Begin/Rollback to Save/Rollback
 - Disable TST Rollback

DISABLE TST TRANSACTION

```
-- =====  
-- TSTConfig. TST calls this at the start of each test session  
-- to allow the test client to configure TST parameters.  
-- =====  
CREATE PROCEDURE dbo.TSTConfig  
AS  
BEGIN  
    EXEC TST.dbo.SetConfiguration  
        @ParameterName='UseTSTRollback',  
        @ParameterValue='0',  
        @Scope='Test',  
        @ScopeValue='SQLTest_RolledBackOperation'  
END
```

SETUP/TEARDOWN PROCEDURES

- Setup/Tear Down
 - SQLTest_SESSION_Setup (TearDown)
 - SQLTest_SETUP_SuiteName (TearDown)
- TearDown sprocs are considered an anti-pattern

AVAILABLE ASSERTS

- `Assert.LogInfo`
- `Assert.Pass/Assert.Fail`
- `Assert.Equals/Assert.NotEquals`
- `Assert.NumericEquals/Assert.NumericNotEquals`
- `Assert.FloatEquals/Assert.FloatNotEquals`
- `Assert.IsLike/Assert.IsNotLike`
- `Assert.IsNull/Assert.IsNotNull`
- `Assert.RegisterExpectedError`

ASSERTING TABLE DATA

- `Assert.TableEquals`
 - Uses `#ExpectedResult` and `#ActualResult`
 - Use `@IgnoredColumns` param to ignore columns
- `Assert.IsTableEmpty/Assert.IsTableNotEmpty`

ADDING TESTS TO .NET BUILD PROCESS

- Use TestFactory capabilities from MbUnit/Gallio
- Disable Cleaning Test Results
- Build MbUnit “Tests” using TST Data
- Clean Test Results

CONTACT ME

- phil@telerik.com
- www.skimedic.com/blog
- www.twitter.com/skimedic