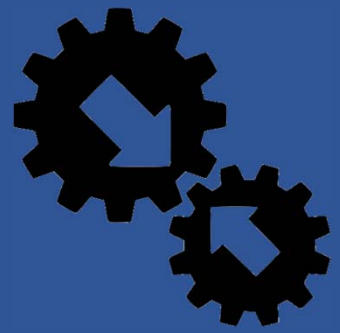
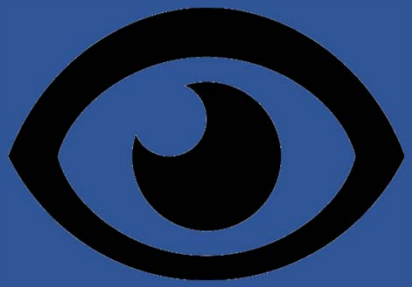
Start to See with tSQLt

… the next 60 minutes

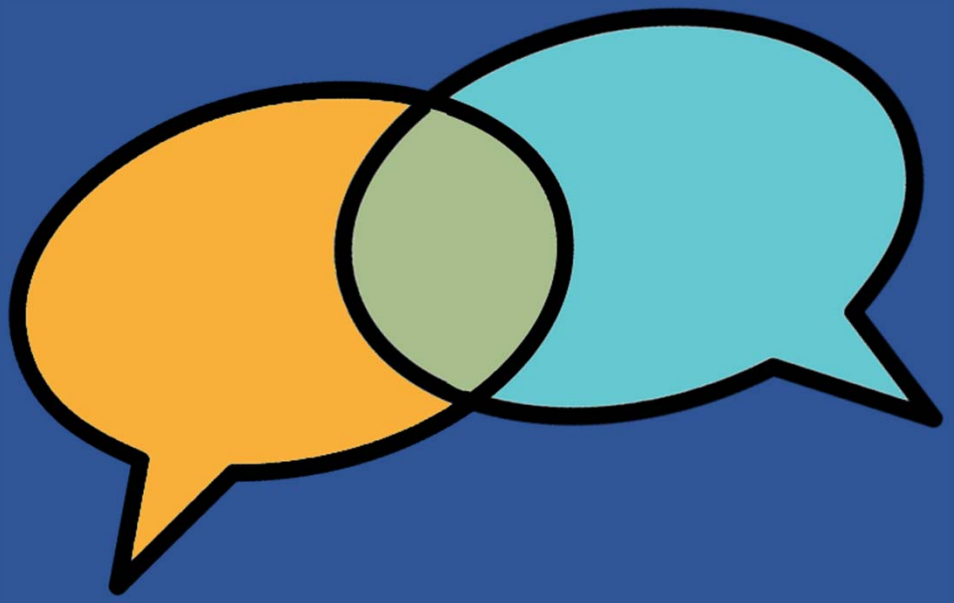




Source: https://www.onlinewebfonts.com

Presentation Rules

- Always ask questions
- Interrupt me
- This is a two-way conversation, let's learn from each other's experiences



Waypoint ANALYTICAL

MVP Microsoft® Most Valuable Professional

@sqlstad

sqlstad.nl

sander@sqlstad.nl

github.com/sanderstad

Unit Testing/Integration Testing

Unit Testing vs Integration Testing

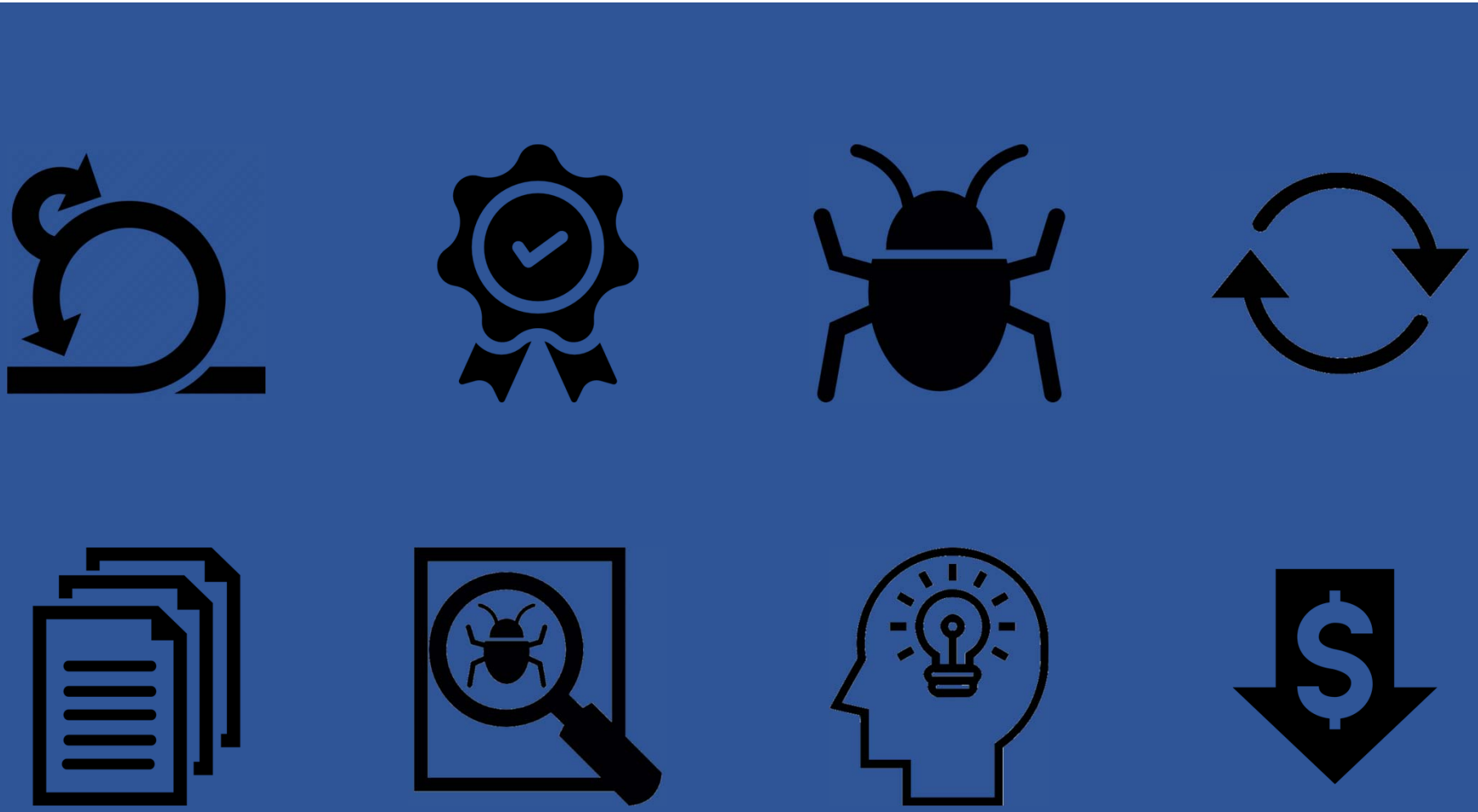
Unit Testing

- Tests a small piece of code
- Single component
- The scope is narrow
- Small effort to put together
- Mainly focuses on the testing the functionality of individual units only

Integration testing

- Test different pieces of code
- Integration modules is considered
- The scope of is wide
- Larger effort to put together
- Integration testing is to be carried out to discover the the issues arise when different modules are interacting

Why?



Source: https://www.onlinewebfonts.com

You CAN NOT write high quality software on a large scale without unit testing

Unit Testing Improves These Areas

- Visibility and reporting
- Control and correction
- Efficiency and speed
- Planning and predictability
- Customer satisfaction

What Makes A Good Unit Test Run

- Run the collection of all the tests
- Must be executed without any manual intervention
- Outcome has to be must be unambiguous and repeatable

Unit Test Anatomy

Assemble

Act

Assert

Assemble

```
-- Declare the variables
DECLARE @actual INT,
        @expected INT;

EXEC tSQLt.FakeTable @TableName = N'[dbo].[People]',
                     @Identity = 1,
                     @ComputedColumns = 1,
                     @Defaults = 1;

-- Get the expected value
SELECT @expected = IDENT_CURRENT('dbo.People');
```

Unit Test Anatomy

Assemble

Act

Assert

Act

```
-- Execute the procedure
EXEC dbo.People_Create @PeopleID = @actual OUTPUT,
@Firstname = ‘John’,
@LastName = ‘Doe’
PhoneNumber = ‘123-456 7890’
```

Unit Test Anatomy

Assemble

Act

Assert

Assert

```
EXEC tSQLt.AssertEquals @expected, @actual;
```

What is tSQLt

Unit testing framework for databases

What can we do with tSQLt?

- Mock tables, functions, views
- Assert tables, values, existence of objects
- Handle exceptions
- Return the results in XML form

Requirements

1. Enable CLR for the instance your running the tests on
2. Install the tSQLt framework into the database
3. At least one test class
4. At least one unit test

Tips

- Create multiple test classes
- Give the test a descriptive name
- Start each test with the name “test …..”
- Only one assert per unit test


DEMO

What did we use

- tSQLt
  - https://tsqlt.org/
- PsModuleDevelopment
  - https://psframework.org/documentation.html
- SSDT-With-tSQLt-Template
  - https://github.com/sanderstad/SSDT-With-tSQLt-Template
- PstSQLtTestGenerator
  - https://github.com/sanderstad/PStSQLtTestGenerator

Waypoint ANALYTICAL

MVP Microsoft® Most Valuable Professional

@sqlstad

sqlstad.nl

sander@sqlstad.nl

github.com/sanderstad