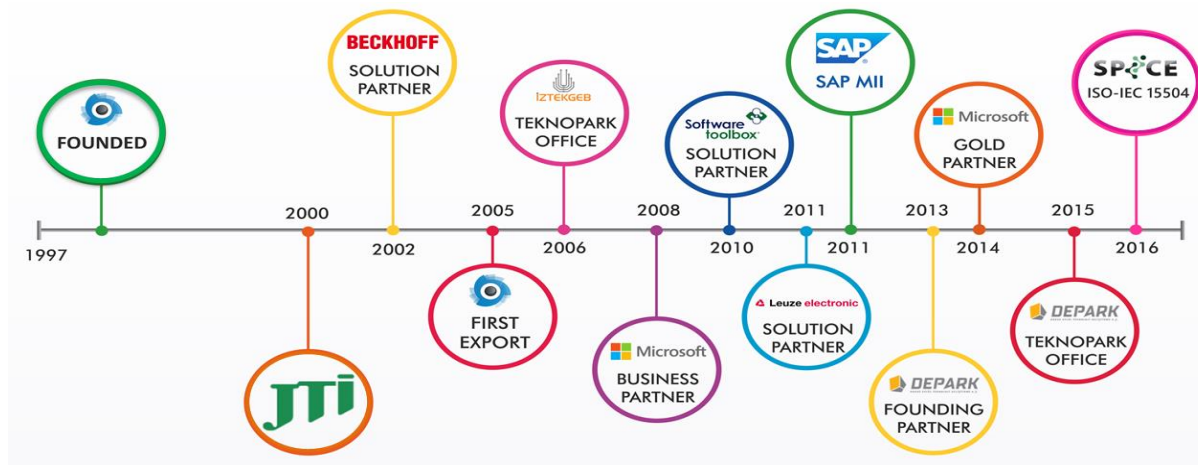




Automated Functional Testing with tSQLt

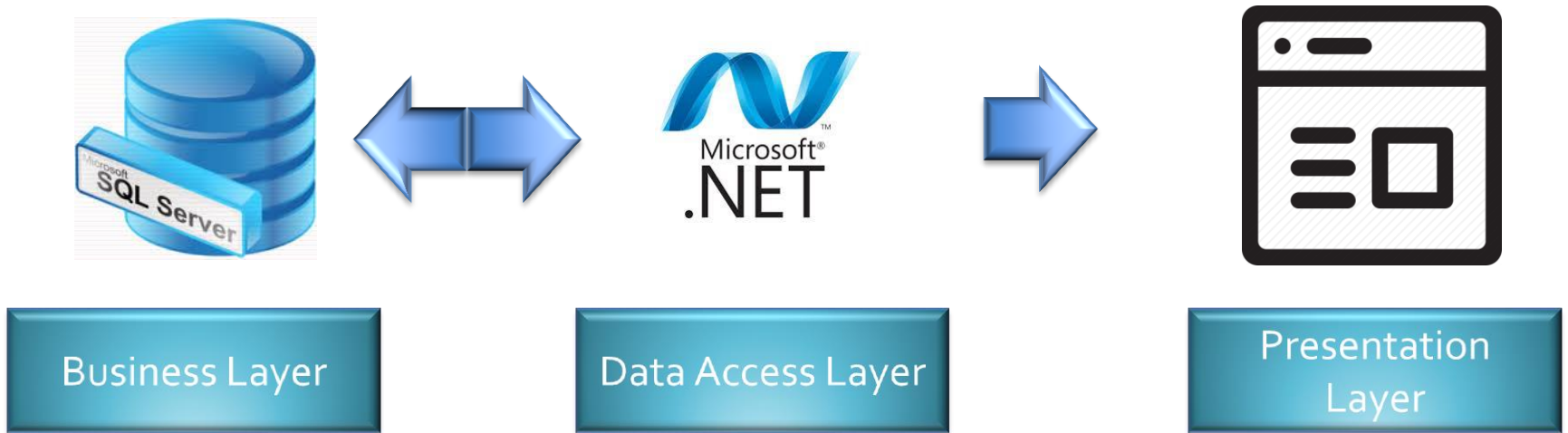
About Siskon



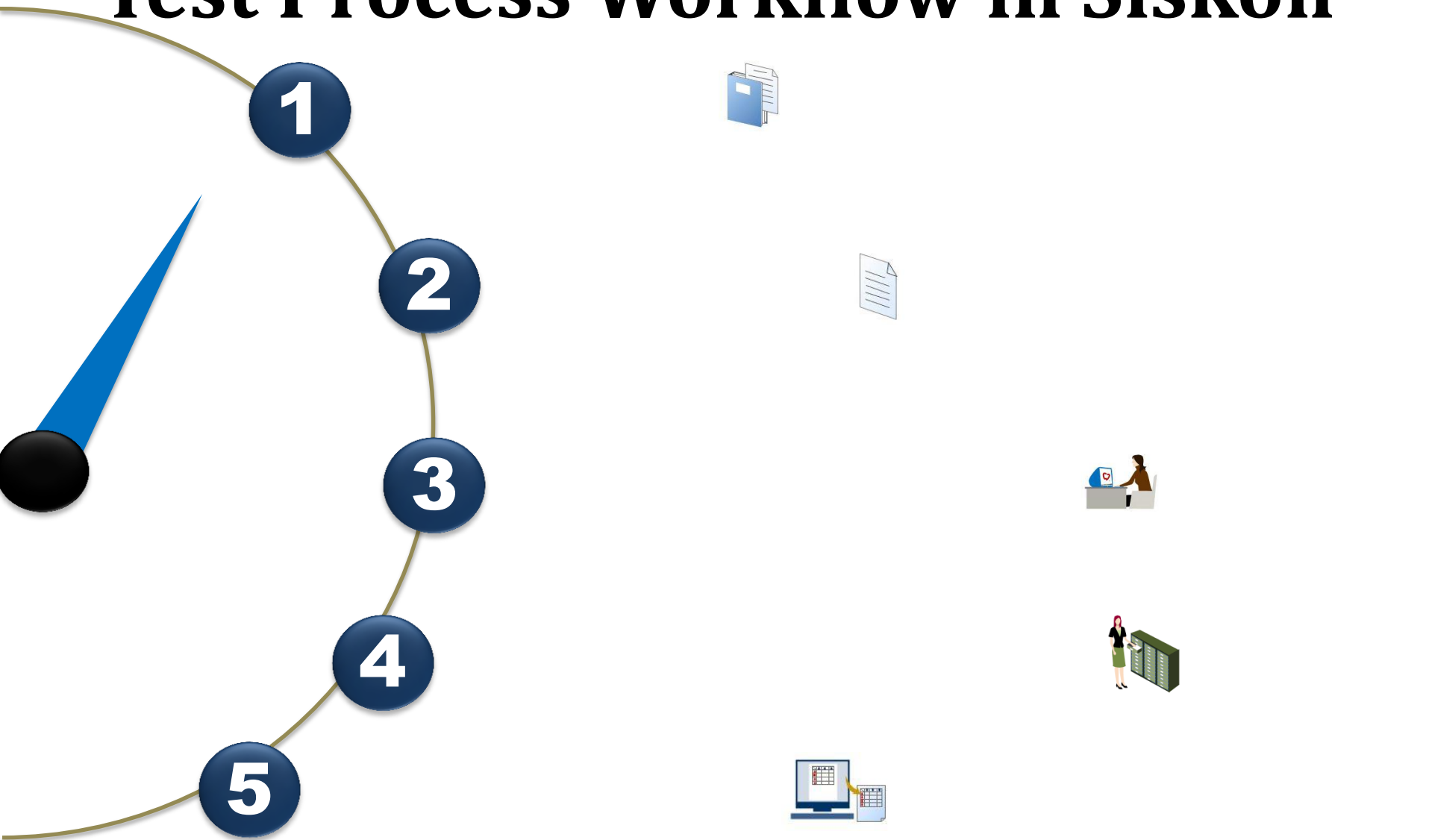
Overview of Presentation

- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓
- ✓

Architecture of our project



Test Process Workflow in Siskon



Executable Test Scripts



Workflow of Test Scripts Execution Process

Prepate test scripts with tSQLt



Generate test scripts using Microsoft Project



Execute test scripts in Microsoft SQL Server





Requirement Analysis

Rule Applicability

Rule
1 - Global Admin

Rule Type
BIZ - Business

ValidDateFrom
01/01/2017

ValidDateTo
 01/05/2017

New
Save
Delete

Rule	RuleType	ValidDateFrom	ValidDateTo
1-Global Admin	BIZ - Business	01/01/2017	01/01/2018
1-Global Admin	NTF - Notification	01/01/2017	31/12/2017
2-System Conf.	FUN - Functional	01/01/2017	01/01/2018

Determine Test Objectives



-
-

Input #	Input Variables
•	
•	

Output #	Output Variables

Guidelines for Functional Testing Technique Selection

C1: Variables Physical (P) Logical (L)	P	P	P	P	P	L	L	L	L	L
C2: Independent Variables?	Y	Y	Y	Y	N	Y	Y	Y	Y	N
C3: Single fault assumption?	Y	Y	N	N	-	Y	Y	N	N	-
C4: Exception handling?	Y	N	Y	N	-	Y	N	Y	N	Y
A1: Boundary value analysis		X								
A2: Robustness testing	X									
A3: Worst case testing				X						
A4: Robust worst case testing			X							
A5: Traditional equivalence testing	X		X			X		X		
A6: Weak equivalence testing	X	X				X	X			
A7: Strong equivalence testing			X	X	X			X	X	X
A8: Decision table					X					X

Reference: <http://perceval.gannon.edu/xu001/teaching/shared/TestingCraftApproach/xu/Lecture8.ppt>

Test Technique: Decision Table based testing

#	Business Requirements	Exists in decision table	Test technique

Test Technique: Decision Table based testing

2^n $2^4 = 16$

BR#	Condition #	Condition	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉	R ₁₀	R ₁₁	R ₁₂	R ₁₃	R ₁₄	R ₁₅	R ₁₆



Decision Table based testing - Abstract Test Cases

Decision Table based testing – Possible Input Values

Generate Automated Executable Test Scripts





Why T-SQL?

tSQLt allows you to implement unit tests in T-SQL. This is important as you do not have to switch between various tools to create your code and your unit tests. tSQLt also provides the following features to make it easier to create and manage unit tests:

- ❖ Tests are automatically run within transactions – this keeps tests independent and reduces any clean-up work you need
- ❖ Tests can be grouped together within a schema – allowing you to organize your tests and use common setup methods
- ❖ Output can be generated in plain text or XML – making it easier to integrate with a continuous integration tool
- ❖ Provides the ability to fake tables and views, and to create stored procedure spies – allowing you to isolate the code which you are testing

* Reference: <http://www.tsqlt.org>



Why Microsoft Project?



RuleApplicabilityForm - Project Professional

View JIRA Integration Project Reporting Test Scripts Developer TEAM Gantt Chart Tools Format Tell me what you want to do

Predecessor	Successor	JIRA_IssueType	Notes	Task Name
		Deliverable		Applicability Rule Definition Form
	4		[...]	test data for insert operation
				FUNC
2		TestScript		\$TC_No\$ - Rx= "\$Rx\$", RTx = "\$RTx\$", DFx= "\$DFx\$", DTx= "\$DTx\$" - Ry= "\$Ry\$", Rty = "\$RTy\$", [
		TestStep	declare @Rynvarchar(10)...	declare parameters
		TestStep	set @Ry='\$Ry\$'...	Set paramaters
		TestStep	set @Fields = 'RuleID RuleTyp	Insert an Applicability rule record
		TestStep	set @RuleApplicability= '<Rule.	Insert second Applicability rule record
		TestStep	set @ExpectedResult = @Expec	Check Expected vs Actual

Prepare Test Data

The screenshot displays a software interface with a grid background. A 'Notes' dialog box is open, containing the following text:

```
[  
{TC No:'TC2',RTy: 'BIZ' ,Ry:'1' ,DFy:'1/1/1930' ,DTy:'12/30/8999' ,  
  RTx: 'FN' ,Rx:'1' ,DFx:'12/30/8999' ,DTx:'1/1/1930',  
  Expected Output:'ValidDateFrom must be earlier than ValidDateTo !'},  
{...},  
{...},  
{...}  
]
```

The dialog box includes a 'Help' button on the left, and 'OK' and 'Cancel' buttons on the right. The background grid shows a table with columns labeled 'Predecessor' and 'Successor', and rows with values like '4' and '2'. A yellow border highlights the 'Notes' dialog box.

Executable Test Script Naming

	JIRA_Deli	Predecessor	Successor	JIRA_IssueType	Task Name
1	APP_Rule			Deliverable	Applicability Rule Definition Form
2			4		test data for insert operation
3					FUNC
4	2			TestScript	\$TC No\$-Rx= "\$Rx\$",RTx = "\$RTx\$",DFx= "\$DFx\$",DTx= "\$DTx\$" - Ry= "\$Ry\$",Rty = "\$RTy\$",DFy= "\$DFy\$",Dty= "\$DTy\$"
5				TestStep	declare parameters
6				TestStep	Set paramaters
7				TestStep	Insert an Applicability rule record
8				TestStep	Insert second Applicability rule record
9				TestStep	Check Expected vs Actual

Test Steps with SQL Scripts

RuleApplicabilityForm - Project Professional

View JIRA Integration Project Reporting

Predecessors	Successors	JIRA_IssueType	Notes
	4	Deliverable	[...]
2		TestScript	decl
		TestStep	set
		TestStep	set
		TestStep	set
		TestStep	set

Task Information

General | Predecessors | Resources | Advanced | Notes | Custom Fields

Name: declare parameters Duration: 1 day? Estimated


Notes:

```
declare @Ry nvarchar(10)
declare @RTy nvarchar(10)
declare @DFy datetime
declare @DTy datetime
declare @RTx nvarchar(10)
declare @Rx nvarchar(10)
declare @DFx datetime
declare @DTx datetime
declare @ExpectedResult nvarchar(100)
```

Help OK Cancel

"\$DTx\$" - Ry= "\$Ry\$", Rty = "\$RTy\$", [

Generate Executable Test Script



The screenshot shows the Microsoft SQL Server Management Studio interface. The main window displays a SQL script with the following content:

```
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
SET NOCOUNT ON
GO

EXEC tSQLt.NewTestClass 'APP_Rule';
GO

CREATE PROCEDURE [APP_Rule].[test FUNC TC1 - Rx= "1", RTx = "BIZ" , DFx= "12/30/8999", DTx= "" - Ry= "", Rty = "" , Dfy= "", Dty= ""]
AS
BEGIN
    SET NOCOUNT ON
    /**
    Generated: 26.04.2017 14:47:55
    Source: RuleApplicabilityForm
    **/
    /*
    Line:          5
    Step #:        1
    Test Step:     declare parameters
    */
    declare @Ry          nvarchar(10)
    declare @RTy         nvarchar(10)
    declare @DFy         datetime
    declare @DTy         datetime
    declare @RTx         nvarchar(10)
    declare @Rx          nvarchar(10)
    declare @DFx         datetime
    declare @DTx         datetime
    declare @ExpectedResult nvarchar(100)
```

Two black boxes highlight specific parts of the script: one around the `EXEC tSQLt.NewTestClass 'APP_Rule';` line, and another around the `CREATE PROCEDURE` definition. The interface also shows a sidebar with 'Plain SQL script' and a 'TEAM' label on the right.

Run Executable Test Script

SQL Query32.sql - si...Azizem.celik (118)*

```
exec tsqtl.run 'APP_Rule'
```

100 %

Results Messages

(1 row(s) affected)

```
[APP_Rule].[test FUNC TC5-Rx= "2",RTx = "FN" ,DFx= "1/1/1930",DTx= "12/30/8999" - Ry= "2",Rty = "FN",DFy= "12/30/8999",Dty= "12/31/8999"] failed: (Failure)
Expected: <Date period cannot be overlapped based on RuleID - RuleTypeID>
but was: <There is already applicability record of the rule between this date interval !>
```

```
+-----+
|Test Execution Summary|
+-----+
```

No	Test Case Name	Dur(ms)	Result
1	[APP_Rule].[test FUNC TC1-Rx= "1",RTx = "BIZ" ,DFx= "12/30/8999",DTx= "" - Ry= "",Rty = "",DFy= "",Dty= ""]	13	Success
2	[APP_Rule].[test FUNC TC2-Rx= "1",RTx = "FN" ,DFx= "12/30/8999",DTx= "1/1/1930" - Ry= "1",Rty = "BIZ",DFy= "1/1/1930",Dty= "12/30/8999"]	6	Success
3	[APP_Rule].[test FUNC TC3-Rx= "1",RTx = "FN" ,DFx= "3/2/2017",DTx= "12/30/8999" - Ry= "1",Rty = "BIZ",DFy= "1/1/1930",Dty= "1/2/1930"]	6	Success
4	[APP_Rule].[test FUNC TC4-Rx= "2",RTx = "FN" ,DFx= "1/2/1930",DTx= "3/2/2017" - Ry= "2",Rty = "FN",DFy= "12/31/8999",Dty= ""]	3	Success
5	[APP_Rule].[test FUNC TC6-Rx= "1",RTx = "BIZ" ,DFx= "1/2/1930",DTx= "3/2/2017" - Ry= "1",Rty = "FN",DFy= "12/31/8999",Dty= ""]	3	Success
6	[APP_Rule].[test FUNC TC7-Rx= "2",RTx = "FN" ,DFx= "3/2/2017",DTx= "" - Ry= "1",Rty = "FN",DFy= "1/1/1930",Dty= "1/2/1930"]	6	Success
7	[APP_Rule].[test FUNC TC8-Rx= "2",RTx = "BIZ" ,DFx= "1/2/1930",DTx= "1/1/1930" - Ry= "",Rty = "",DFy= "",Dty= ""]	3	Success
8	[APP_Rule].[test FUNC TC5-Rx= "2",RTx = "FN" ,DFx= "1/1/1930",DTx= "12/30/8999" - Ry= "2",Rty = "FN",DFy= "12/30/8999",Dty= "12/31/8999"]	10	Failure

Msg 50000, Level 16, State 10, Line 1

Test Case Summary: 8 test case(s) executed, 7 succeeded, 1 failed, 0 errored.



Reusability

	JIRA_Deli	Predecessor	Successor	JIRA_IssueType	Notes	Task Name
1						Utility test steps
2						Check Object is available
3			9	TestStep	declare @ObjectName nvarchar (max	Declare parameter
4		9		TestStep	set @Message = "+@ObjectName+' is	Check Object is available
5	APP_Rule			Deliverable		Applicability Rule Definition Form
6						DEPL
7			8		[...]	test data
8		7		TestScript		Object \$ObjectName\$ is available
9		3	4	TestStep	set @ObjectName = '\$ObjectName\$'	set paramaters

Common Steps

set @ObjectName = '\$ObjectName\$'



Only set parameters in main test script

Reusability

```
CREATE PROCEDURE [APP_Rule].[test DEPL Object RuleApplicability is available]
AS
BEGIN
SET NOCOUNT ON
/**
Generated: 03.05.2017 11:09:20
Source: RuleApplicabilityForm
**/
/*
Line:      3
Test Step: Declare parameter
*/
declare @ObjectName nvarchar (max)
declare @Message nvarchar (max)

/*
Line:      9
Step #:    1
Test Step: set paramaters
*/
set @ObjectName = 'RuleApplicability'

/*
Line:      4
Test Step: Check Object is available
*/
set @Message = ''+@ObjectName+' is not available in the database'
EXEC tSQLt.AssertObjectExists @ObjectName, @Message
```

Predecessors test step

Main test step

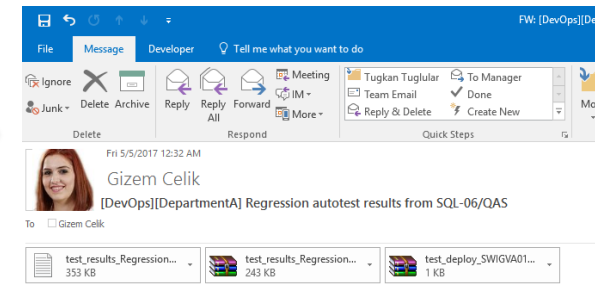
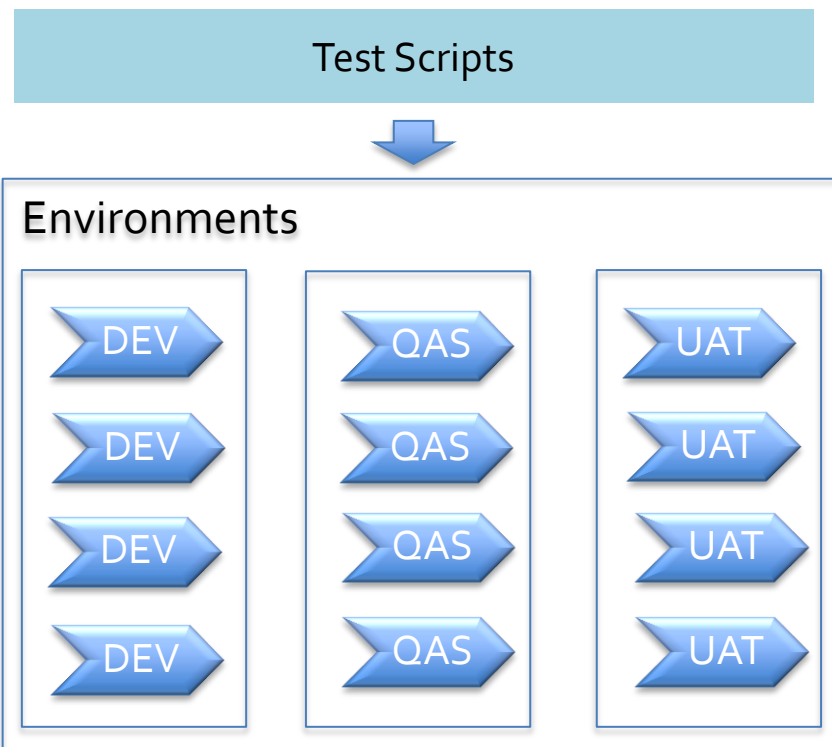
Successors test step

Comparison

-
-
-



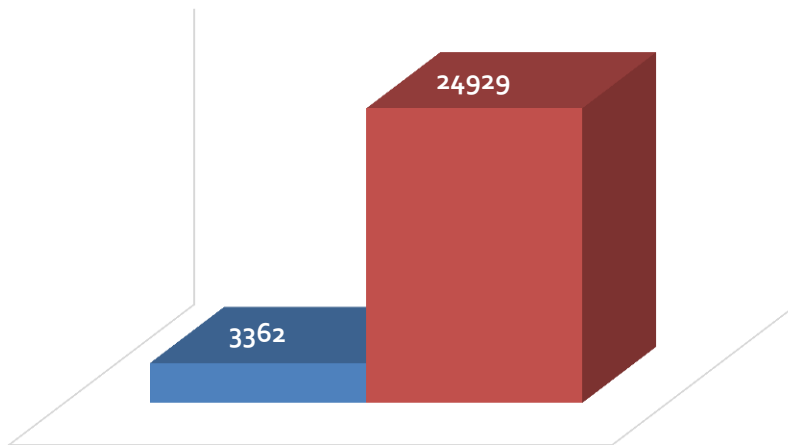
Comparison



Statistics

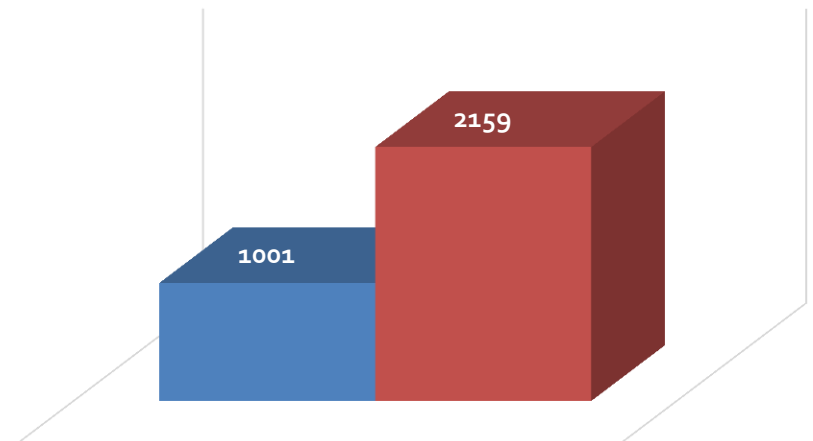
-
-
-
-
-
-
-

■ Microsoft Project ■ SQL Management Studio



TEST STEP COUNT

■ Microsoft Project ■ SQL Management Studio



TEST SCRIPT COUNT

Key Benefits

-
-
-
-

Thank You!