

**NAV
TECH
DAYS
2017**

mibuso.com

BEST PRACTICES TO GET AUTOMATED TESTS RUNNING ON YOUR SOLUTION

LUC VAN VUGT
THE LEARNING NETWORK

JAMES CROWTER
TECHNOLOGY MANAGEMENT

WHEN YOU ARE PASSIONATE ABOUT MICROSOFT DYNAMICS NAV | www.navtechdays.com

**NAV
TECH
DAYS
2017**
mibuso.com

History is made @NAVTechDays

And you're all part of it ...

History is made @NAVTechDays



**UNOFFICIAL
ATTEMPT**

The talest
presentation duo ever
at NAV TechDays
!!!!!!!!!!!!

History is made @NAVTechDays



James Crowter
&
Luc van Vugt

**UNOFFICIAL
ATTEMPT**

History is made @NAVTechDays



**UNOFFICIAL
ATTEMPT**

You can take
pictures now

History is made @NAVTechDays



**UNOFFICIAL
ATTEMPT**



Introduction - And You

Do you know the Testability
Framework that is build in NAV?

i.e. technical features like test
codeunits, ASSERTERROR, etc.



Introduction - And You

Did you already build and execute
test yourself?



Agenda

Why automated testing

How to write automated tests

What about MS automated tests

Take-aways / Do's & Don'ts

Why automated testing

a business justification

The NAV developers role is hard !

Everyone expects what you do, to be perfect first time every time

Even when they don't know what they really want

They expect a lifetime warranty, even for what they don't currently use

Oh and the timescales better be yesterday, instantly is to long

Then your expected to remember, every last detail, forever

And that even if you did write the original code!



And our world is
changing

Many more deployment cycles than historically

- Monthly CU's need implementing more regularly as 'fixes'
- No choice on Tenerife SaaS platform with limited time to prepare

Annual & now Biannual Full Version updates force refactoring

More dependencies with Office and API integration

More complexity with more ISV's via Extensions and AppSource

Its always testing that gets squeezed

Timescale reasons

- Signoff of requirements or design
- Development overran (its not a precise science)
- Preparation of test data
- Rework was required
- The client doesn't have the required people available
- Its too close to the mandatory go-live to have time

Budget Reasons

- The test budget was used in the analysis or development phase
- Sales didn't factor in the need to test at all

Philosophy Reasons

- The client doesn't see the need to spend the time
- Develop fast & fix it in next release, (Microsoft style?)



Its your reputation that suffers

- Your clients and colleagues expect instant perfection
 - Blaming the specification is just an excuse
 - You have to know the full implications of every change you make
 - Any errors they find in testing are all your fault
- But How?
 - When you have little extra time
 - And clients demand ever more productivity

Automated Testing is your
answer

Build it once –
use forever

Its not hard –
its just a small
new skill

Go on, on top
of/compared
to

VS Code,
Docker,
Extensions V2

it's easy!

01

More reliable
design with up
front test
definition

02

Defined signed
off use cases

03

More
permutations
are quickly
possible

04

Run whenever it
helps without
waiting for data
prep

05

Its definitive
rather than
subjective

06

Unintended use
is highlighted
even years later

07

Your test library
to go with your
code library

Lose less sleep and reduce your stress

Once they've experienced it they won't want anything else

Who enjoys testing, not you but not them either

Management like the reduced risk and higher satisfaction

Once learning curves over, its can often be quicker than manual

Its lasts forever, every object set release can have a full test

It enables complex repeatable code,

Spend your future adding new features instead of chasing bugs

Selling it to your clients & colleagues

So enough justification – let 's see how it works!

How to write automated tests

learn how to walk

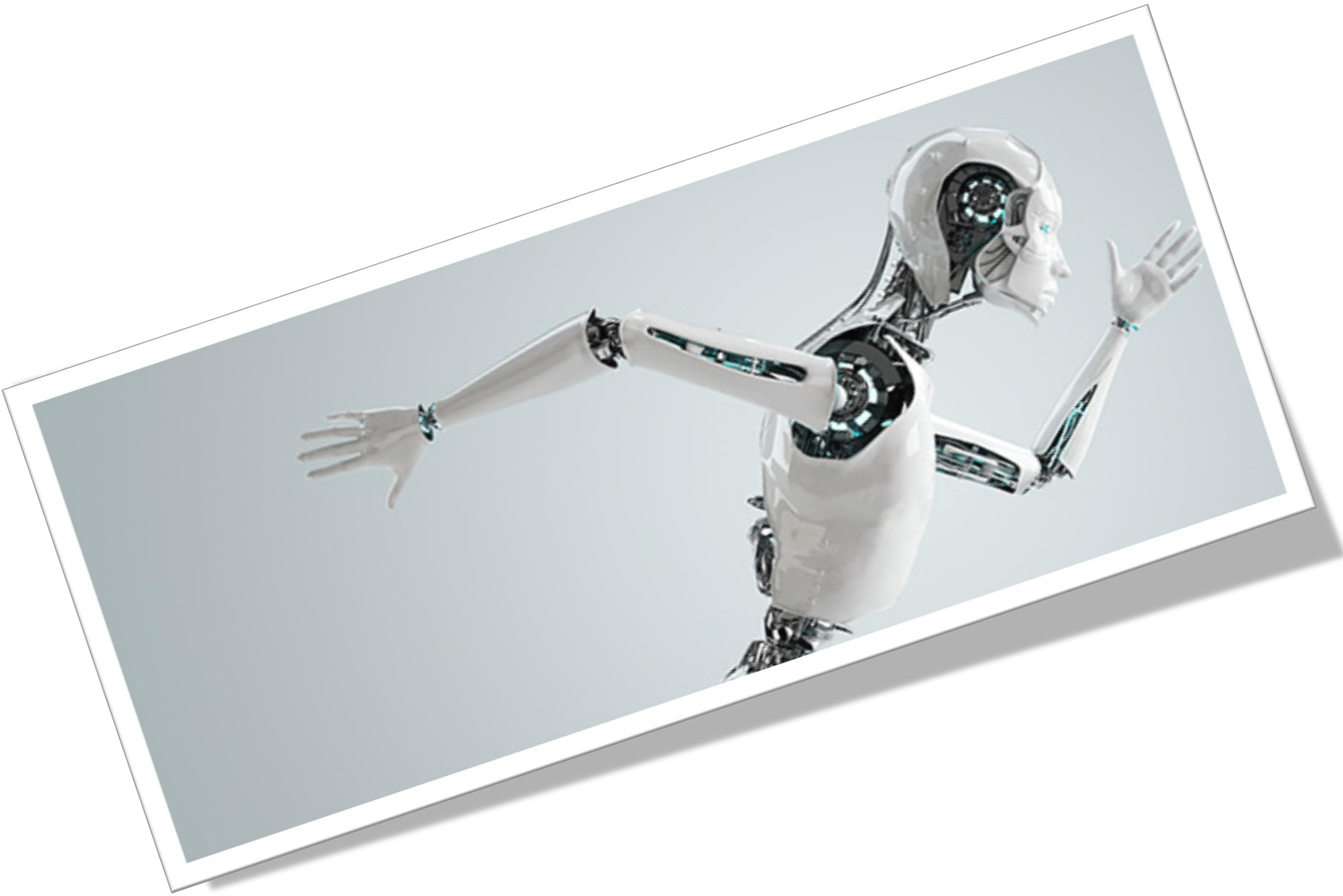
How to write automated tests

Where to start

Testability Framework

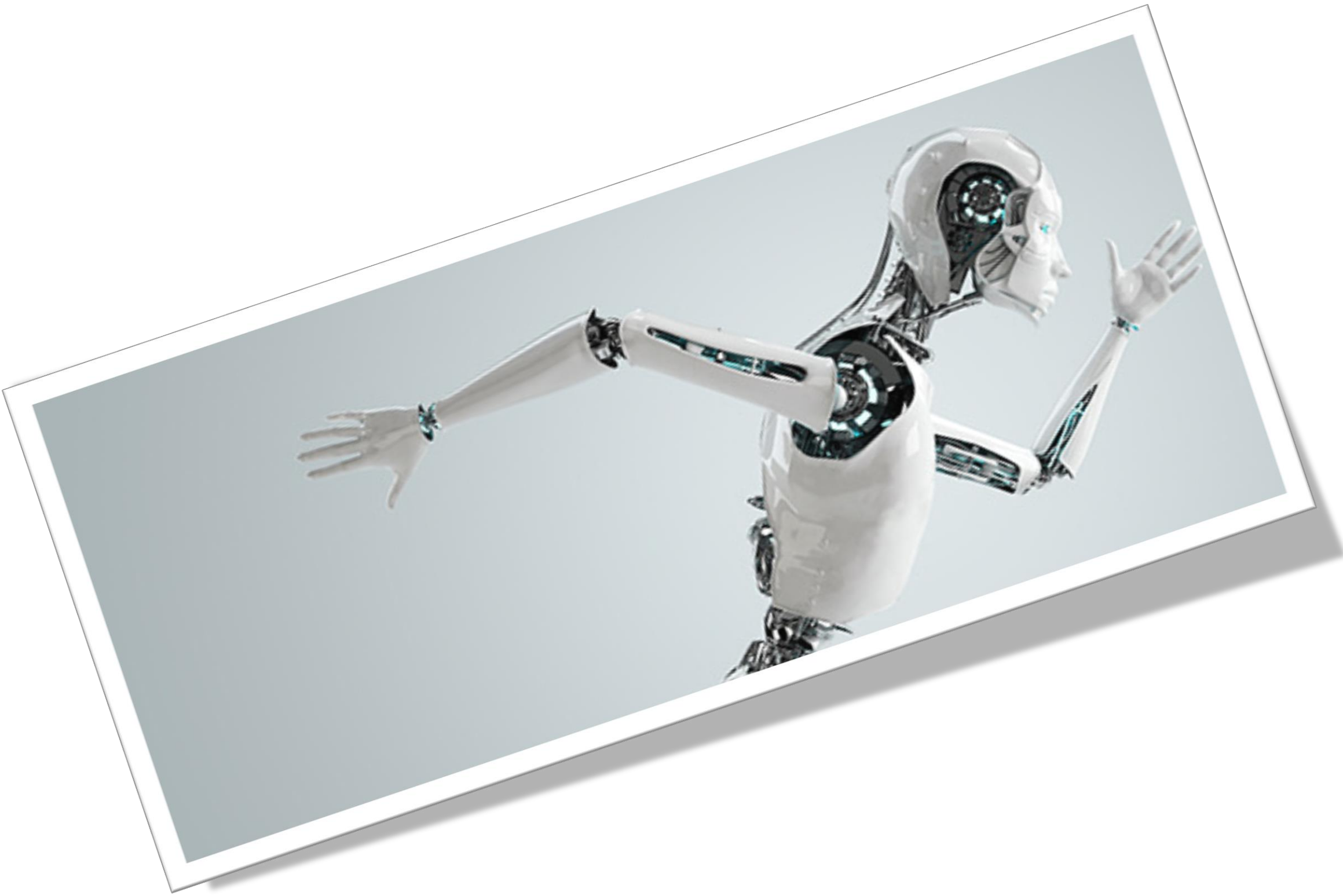
Test Design

Automated Testing - When



Each time code is changed

Automated Testing – Where to Start



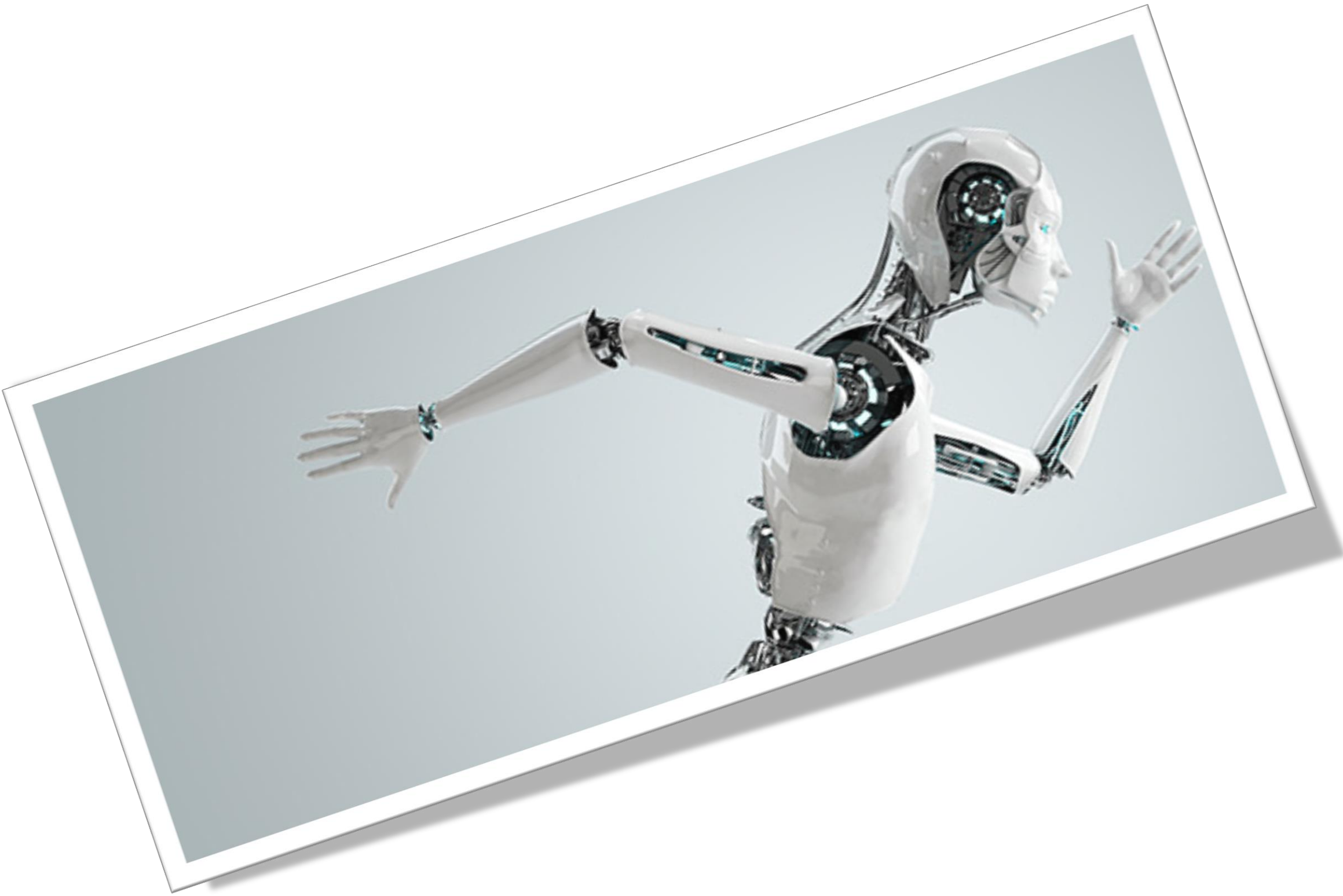
Bug fixing

Changing business critical feature

Developing (complex) feature

Code refactoring

Automated Testing



Repeatable

&

Fast

Testability Framework

Testability Framework ...

History

NAV Test Framework (NTF)

- UI based automated tests
- C#
- Time consuming

Multiple

- Builds per day
- Versions
- Localization

Introduction

- In platform
 - NAV 2009 SP1
- Headless (at first)
- Approx. 10 times faster

Purpose

Testing app code

- No load test
- No config test
- No license test

Running isolated

- Only result matters, i.e. success or failure

Testability Framework ...

... for Dummies

1



Test Codeunits

2



ASSERTERROR

3



UI Handlers

4



Test Runner / Test Isolation

Test Isolation

Test Runner

TestIsolation property

Disabled (default value)

- Do not roll back any changes to the database
Tests are not isolated from each other

Codeunit

- Roll back all changes to the database after each test codeunit executes

Function

- Roll back all changes to the database after each test function executes

Test Function

TransactionModel property

AutoCommit (default value)

- Commit is issued at the end of test function
Standard Error handling

AutoRollback

- Transaction is rolled back after test execution
Calls to COMMIT will fail with an error

None

- Test function does not have open write transaction
At end of each trigger changes are committed to database
If error occurred changes are rolled back at end of transaction.
Used primarily with TestPages

Testability Framework ...

... for Dummies

1



Test Codeunits

2



ASSERTERROR

3



UI Handlers

4



Test Runner / Test Isolation

Test Design

Test Design

Architecture

Functional test vs. unit test

NAV vs. structured programming

Headless vs. UI

Positive & Positive-Negative

Data agnostic

Terminology & Patterns

- xUnit Test Patterns

Patterns

Test Case Patterns

- 4 Phase testing
- ATDD
 - Acceptance Test-Drive Development
 - preferred pattern

Test Data Patterns

- Built Fixture
- Lazy Setup
- Fresh Setup

Test Case Patterns - 4 Phase testing

Setup

Exercise

Verify

Teardown

```
265 UpdateBusRelCodeforBankAccount(DefaultBusRelCodeforBankAccs);
266
267 [Test] VendorFromContactWithSetup()
268 // Covers document number TC0032 - refer to TFS ID 21737.
269 // Test Vendor successfully created from Contact with Bus. Relation Code for Vendors on
270
271 // 1. Setup: Set value of Bus. Relation Code for Vendors on Marketing Setup and Create C
272 Initialize;
273 BusinessRelation.FINDFIRST;
274 DefaultBusRelCodeforVendors := UpdateBusRelCodeforVendors(BusinessRelation.Code);
275 LibraryMarketing.CreateCompanyContact(Contact);
276
277 // 2. Exercise: Create Vendor from Contact.
278 Contact.CreateVendor;
279
280 // 3. Verify: Verify Contact Business Relation and Vendor successfully created.
281 ContactBusinessRelation.SETRANGE("Link to Table",ContactBusinessRelation."Link to Table")
282 ContactBusinessRelation.SETRANGE("Contact No.",Contact."No.");
283 ContactBusinessRelation.FINDFIRST;
284
285 Vendor.GET(ContactBusinessRelation."No.");
286
287 // 4. Teardown: Set Default value of Bus. Relation Code for Vendors on Marketing Setup.
288 UpdateBusRelCodeforVendors(DefaultBusRelCodeforVendors);
289
290 [Test] BankFromContactWithSetup()
291 // Covers document number TC0032 - refer to TFS ID 21737.
292 // Test Bank Account successfully created from Contact with Bus. Relation Code for Bank
```


Test Case Patterns - ATDD

Given

- *Setup*

When

- *Trigger*

Then

- *Verification*

```
222 STRSUBSTNO(AppInRoundingError,DetailedCustLedgEntry.FIELDCAPTION("Entry Typ
223 DetailedCustLedgEntry.FIELDCAPTION("Entry No."),DetailedCustLedgEntry."En
224
225 [Test] ClearAppliesToIDFromCLEWhenChangeValueOnGenJnlLine()
226 // [SCENARIO 118226] Applies-to ID is cleared from Customer Ledger Entry when
227 Initialize;
228
229 // [GIVEN] Customer Ledger Entry and General Journal Line with the same Appli
230 FindOpenInvCustLedgEntry(CustLedgEntry);
231 SetAppliesToIDOnCustLedgEntry(CustLedgEntry);
232 CreateGenJnlLineWithAppliesToID(
233 GenJnlLine,GenJnlLine."Account Type"::Customer,CustLedgEntry."Customer No."
234
235 // [WHEN] Change "Applies-to ID" in General Journal Line
236 GenJnlLine.VALIDATE("Applies-to ID",LibraryUtility.GenerateGUID);
237 GenJnlLine.MODIFY(TRUE);
238
239 // [THEN] "Applies-to ID" in Customer Ledger Entry is empty
240 CustLedgEntry.FIND;
241 Assert.AreEqual('',CustLedgEntry."Applies-to ID",STRSUBSTNO(AppliesToIDIsNotE
242 Assert.AreEqual(0,CustLedgEntry."Amount to Apply",AmountToApplyErr);
243
244 [Test] ClearAppliesToIDFromCLEWhenDeleteGenJnlLine()
245 // [SCENARIO 118226] Applies-to ID is cleared from Customer Ledger Entry when
246 Initialize;
```


Test Data

aka Fixture or Test Fixture

Setup at different stages in tests

- CRONUS
 - **Prebuilt Fixture**
- Data shared by multiple tests
 - **Lazy Setup** or **Shared Fixture**
 - master, supplemental & setup data
- Test specific data
 - **Fresh Setup** or **Fresh Fixture**
 - document & custom master data



```
25 [Test] ClearAppliesToIDFromCLEWnenChangeValueOnGenJnlLine()  
26 // [SCENARIO 118226] Applies-to ID is cleared from Customer Ledger Entry when ch  
27 Initialize;  
28  
29 // [GIVEN] Customer Ledger Entry and General Journal Line with the same Applies-
```

```
73 BusinessRelation.FINDFIRST;  
74 DefaultBusRelCodeforVendors := UpdateBusRelCodeforVendors(BusinessRelation.Code);  
75 LibraryMarketing.CreateCompanyContact(Contact);  
76  
77 // 2. Exercise: Create Vendor from Contact
```

Test Design

But even more important is a ...

Test Plan



Test Plan

Create before app coding

Review to get agreement

Based on requirements/use cases

- Use Cases = Test Plan
- Minimalism



Test Plan

Create before app coding

Re

Bc

Create before app coding

Some Test Case Examples

based on a Test Plan for
Seminar Registration application

Test Plan Seminar Registration - WIP.xlsx - Microsoft Excel

LOAD TEST TEAM HULPMIDDELEN VOOR TA... ONTWERPEN Luc van V...

Area	Object/Process	Test Case Level 1	Test Case Level 2	Comment	Scenario	UI/Non-	Priori	Statu	Comment2
1	Area								
2	Setup	Seminar Setup	Entering setup data through page		0001	UI	1	Done	TestPage / AutoRollback / Teard
3									
4		Seminar	Create a new seminar	Manual allowed	0002	Non-UI	1	Done	
5			Using number series	Manual not allowed	0003	Non-UI	3	Pending	
6			Not using number series 1	AssistEdit	0004	Non-UI	1	Pending	
7			Not using number series 2		0005	UI	1		
8			Using number series		0006	Non-UI	2		
9			Last Date Modified		0007	Non-UI	3	Done	
10			Dimensions	Should delete comment lines	0008	Non-UI	3		
11			Delete		0009	Non-UI	2		
12			Search Name		0010	Non-UI	3		
13			Gen. Prod. Posting Group		0011	UI	3		
14		Page	Dimensions		0012	Non-UI	3		
15			Delete						
16		Seminar Journal Template	Delete		0015	Non-UI	1	Done	
17					0016	Non-UI	3		
18		Report Selections			0017	Non-UI	3		
19				Check also result of InitRecord	0018	UI	3	Done	
20	Operation - Data Entry	Seminar Registration	Create a new seminar registration	Manual allowed	0019	Non-UI	1	Pending	
21			Using number series	Manual not allowed	0020	Non-UI	3		
22			Not using number series 1	AssistEdit	0021	Non-UI	2		
23			Not using number series 2	Inheritance of fields - Lines exist	0022	Non-UI	3		
24			Using number series	Inheritance of fields - No Lines Exist	0023	Non-UI	1	Pending	
25			Select Seminar No. 1	Dimensions	0024	Non-UI	2		
26			Select Seminar No. 2	Instructor Name	0025	Non-UI	2		
27			Select Seminar No. 3	Dimensions					
28			Select Instructor No. 1	Empty - Clearing of related fields	0026	Non-UI	2		
29			Select Instructor No. 2	Inheritance of fields - SeminarRoom, "Maximum Participants" >= "Maximum Participants"	0027	Non-UI	3	Done	
30			Select Room No. 1	Inheritance of fields - SeminarRoom, "Maximum Participants" < "Maximum Participants"	0028	Non-UI	1		
31			Select Room No. 2	Dimensions					
32			Select Room No. 3	Lines exist (Unregistered)					
33			Select Room No. 4						
34			Set Seminar Price 1						
35									
36									
37									

Statistics Test Plan

GEREED SCROLL-LOCK

[SCENARIO 0005]

When deleting a seminar with comments
lines these lines should also be deleted

[SCENARIO 0030] - ASSERTERROR

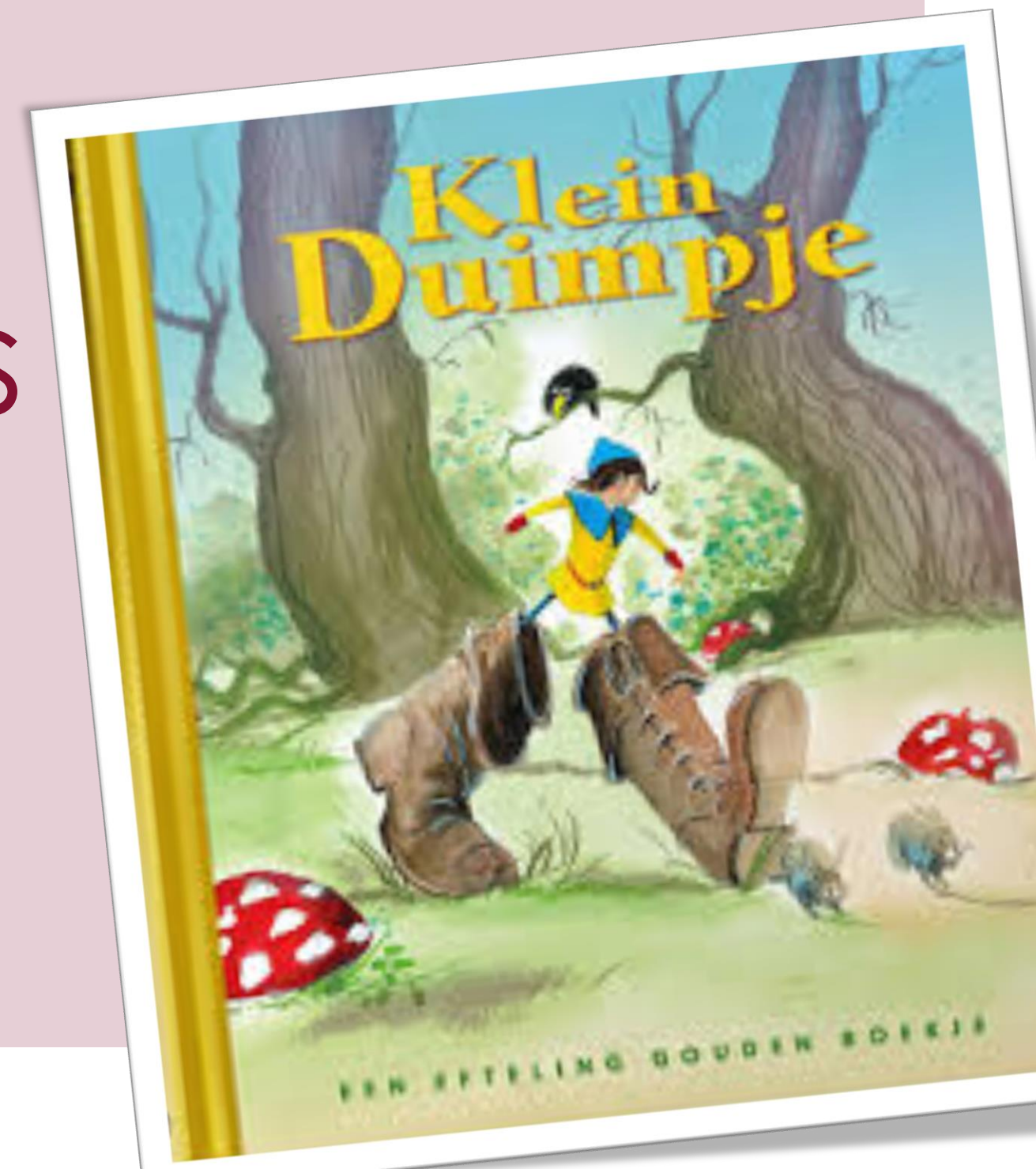
When deleting a seminar registration with status <> Canceled an error should be thrown

[SCENARIO 0028] - ConfirmHandler

Change Seminar Price on header having
registered lines

What about MS automated tests

walk with seven-mile boots



MS Test Suite

It's a humongous collection of tests

- NAV 2016 16.000+
- NAV 2017 19.000+

MS Test Suite

Imagine you writing this amount of tests

- Assume 10 minutes per test, writing it
- 6 per hour
- Approx. 3000 hours, being 2+ years of work

MS Test Suite

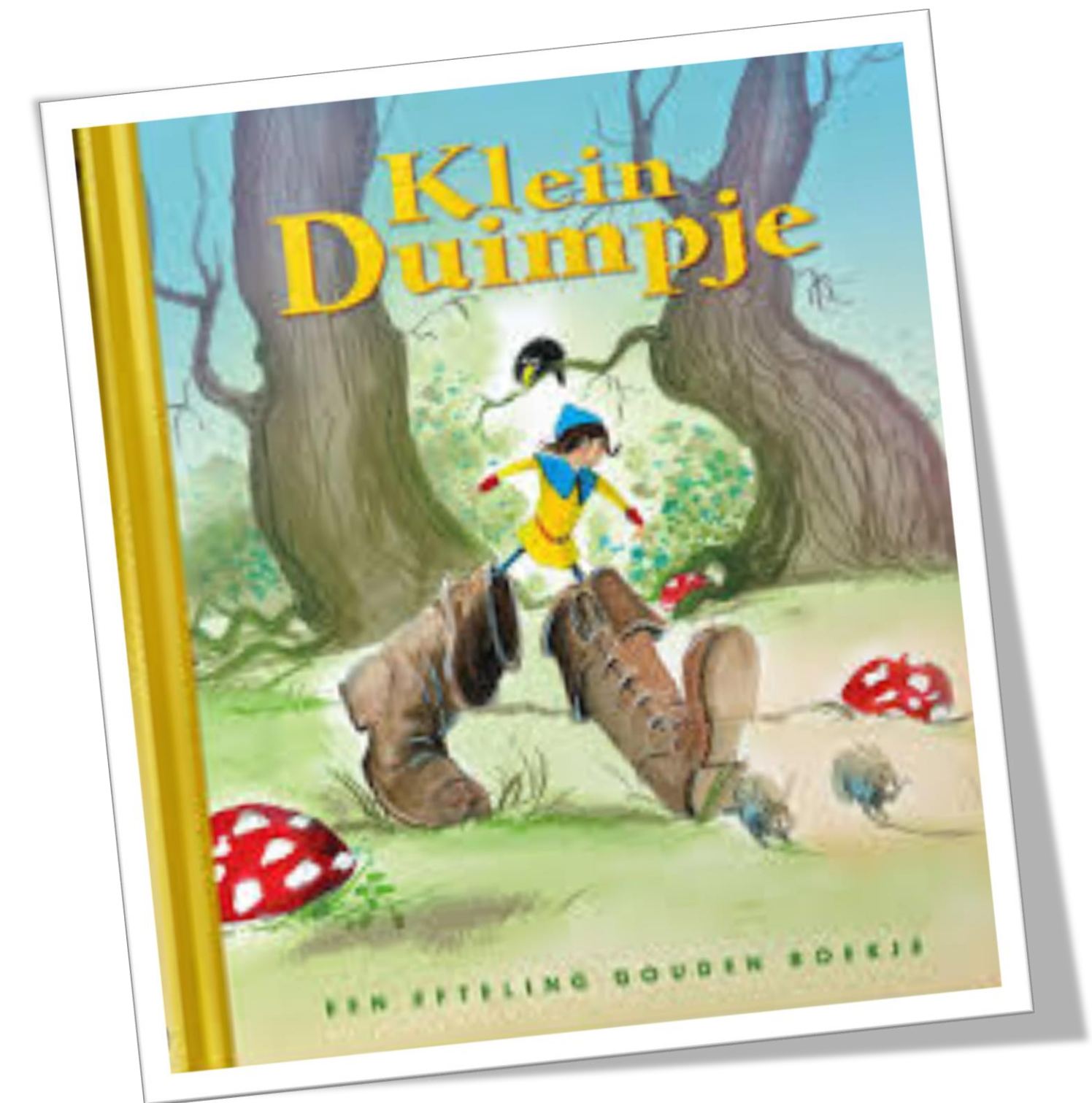
Image you could get this working within a couple of weeks

We did imagine

And we did get it working in approx. 6 weeks

... well, 90 % ...

... on NAV 2016 ..



What did we do?

Context

- The Learning Network
 - No. 1 schoolbooks reseller in the Netherlands
- NAV 2016
 - RTM (technically CU11)
- Customization
 - Approx. 400 standard
 - Approx. 630 own



**THE LEARNING
NETWORK**

What did we do?

Ran tool and saw 23 % succeed

- 3.690 of 16.128 on NAV NL

Analysis showed mainly missing setup data due to our customizations

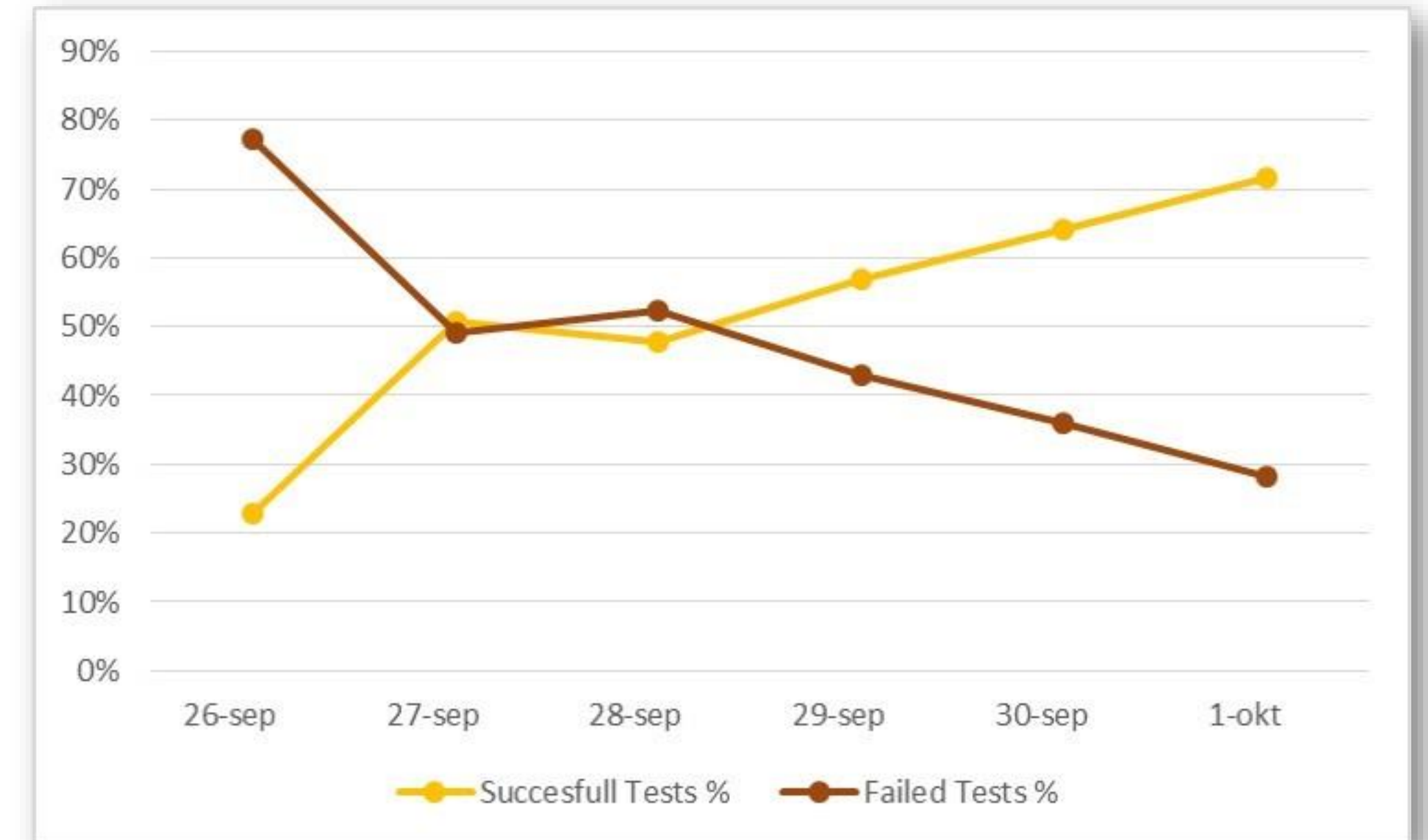
Fixing missing data increased successful tests to 72 % within a couple of days

- 11.464 of 16.128 on NAV NL

Additional effort raised it to 79 %

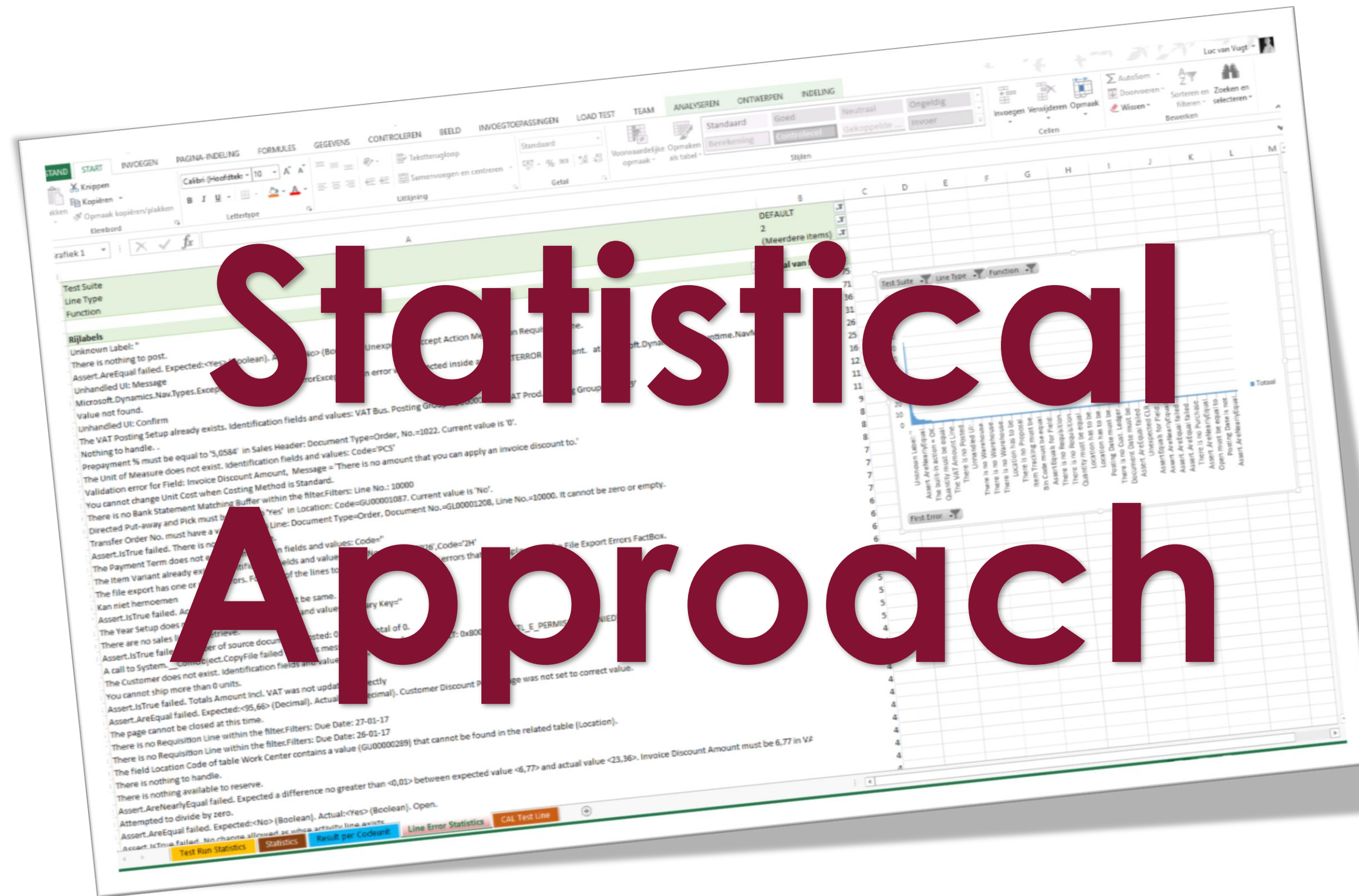
- 12.647 of 16.052 on NAV NL
- Another couple of days

Now at 90% and using successful tests in nightly run



* 16.052: we removed a number of failing tests

What did we do? – Action Plan



1. Run Test Tool
2. Collect and analyze results using Excel
3. Find most occurring error
4. Fix minimalistic
i.e. only code what is needed to fix,
nothing more

Eventually allowing us to focus on relevant areas

What did we do? – Setup

1. Connect to clean CRONUS database with DE
2. Import Test Toolkit
 - CALTestCodeunits.fob
 - CALTestLibraries.fob
3. Import your code
4. Recompile all
 - If not OK, fix the issues
5. Open Windows Client
6. Open Test Tool
7. Add test codeunits using *Get Test Codeunits* action
8. GO ... click Run action



Test Data



Test Data



Some test fixture fixes

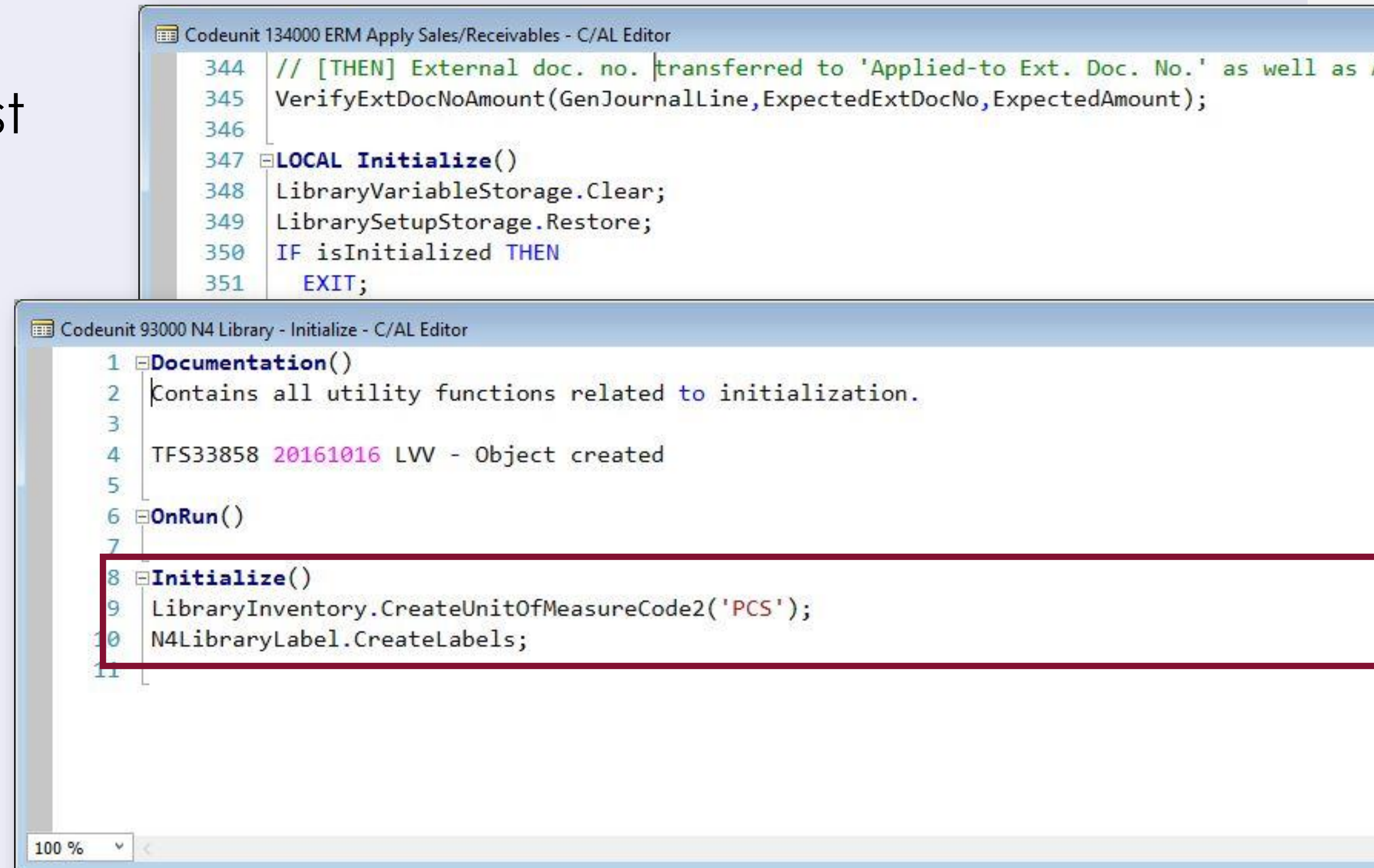
Test Fixture Fixes – Example 1

Failures

Unknown Label: "

The Unit of Measure does not exist

Lazy Setup



```
Codeunit 134000 ERM Apply Sales/Receivables - C/AL Editor
344 // [THEN] External doc. no. transferred to 'Applied-to Ext. Doc. No.' as well as
345 VerifyExtDocNoAmount(GenJournalLine,ExpectedExtDocNo,ExpectedAmount);
346
347 LOCAL Initialize()
348 LibraryVariableStorage.Clear;
349 LibrarySetupStorage.Restore;
350 IF isInitialized THEN
351 EXIT;

Codeunit 93000 N4 Library - Initialize - C/AL Editor
1 Documentation()
2 Contains all utility functions related to initialization.
3
4 TFS33858 20161016 LW - Object created
5
6 OnRun()
7
8 Initialize()
9 LibraryInventory.CreateUnitOfMeasureCode2('PCS');
10 N4LibraryLabel.CreateLabels;
11
```

Test Fixture Fixes – Example 1

Failures

Unknown Label: "

The Unit of Measure does not exist

Lazy?

We had to insert call to Initialize
in approx. 480 codeunits

```

ERM Apply Sales/Receivables - C/AL Editor
345 [THEN] External doc. no. transferred to 'Applied-to Ext. Doc. No.' as well as
346 VerifyExtDocNoAmount(GenJournalLine,ExpectedExtDocNo,ExpectedAmount);
347
348 LOCAL Initialize()
349 LibraryVariableStorage.Clear;
350 LibrarySetupStorage.Restore;
351 IF isInitialized THEN
352     EXIT;
353
Codeunit 25004 Library Initialize - C/AL Editor
Documentation()
2 Contains all utility functions related to initialization.
3
4 20161016 LWW - Object created
5 OnRun()
6
7
8 Initialize()
9 LibraryInventory.CreateUnitOfMeasureCode2('PCS');
10 N4LibraryLabel.CreateLabels;
11
100 %

```

Test Fixture Fixes – Example 2

Failure

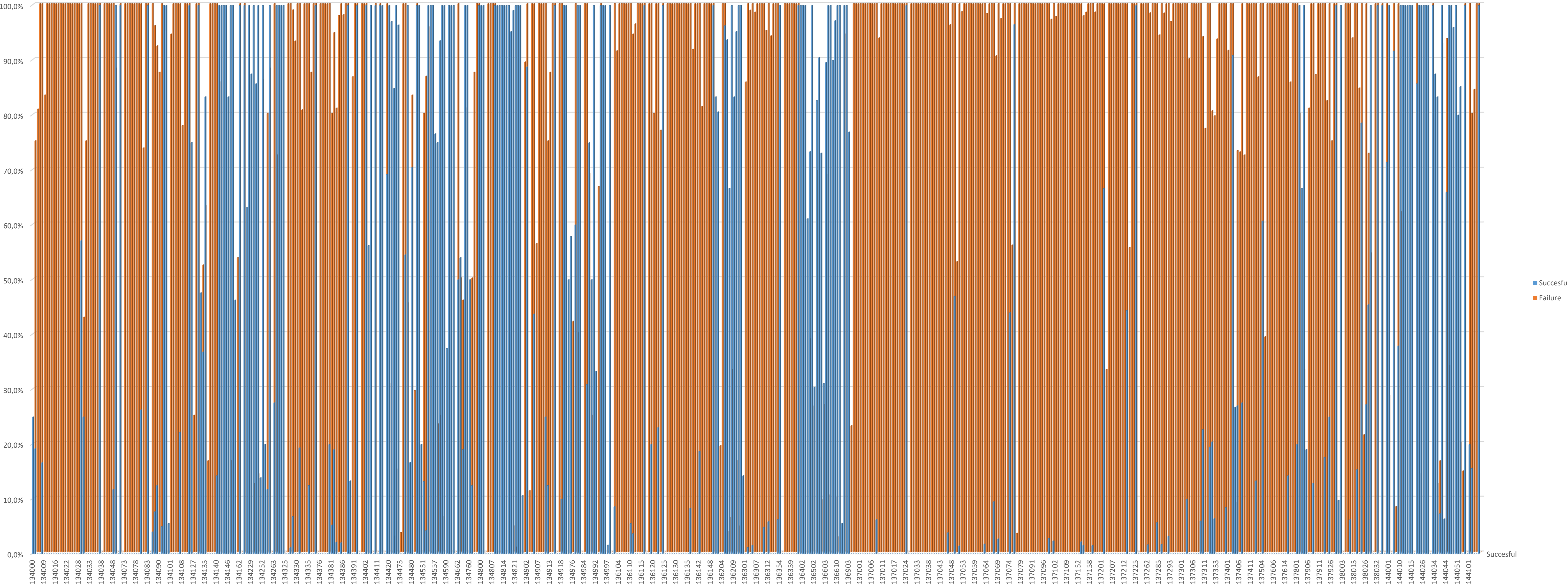
You cannot create this type of document when Vendor GL00000013 is blocked with type All.

Fresh Setup

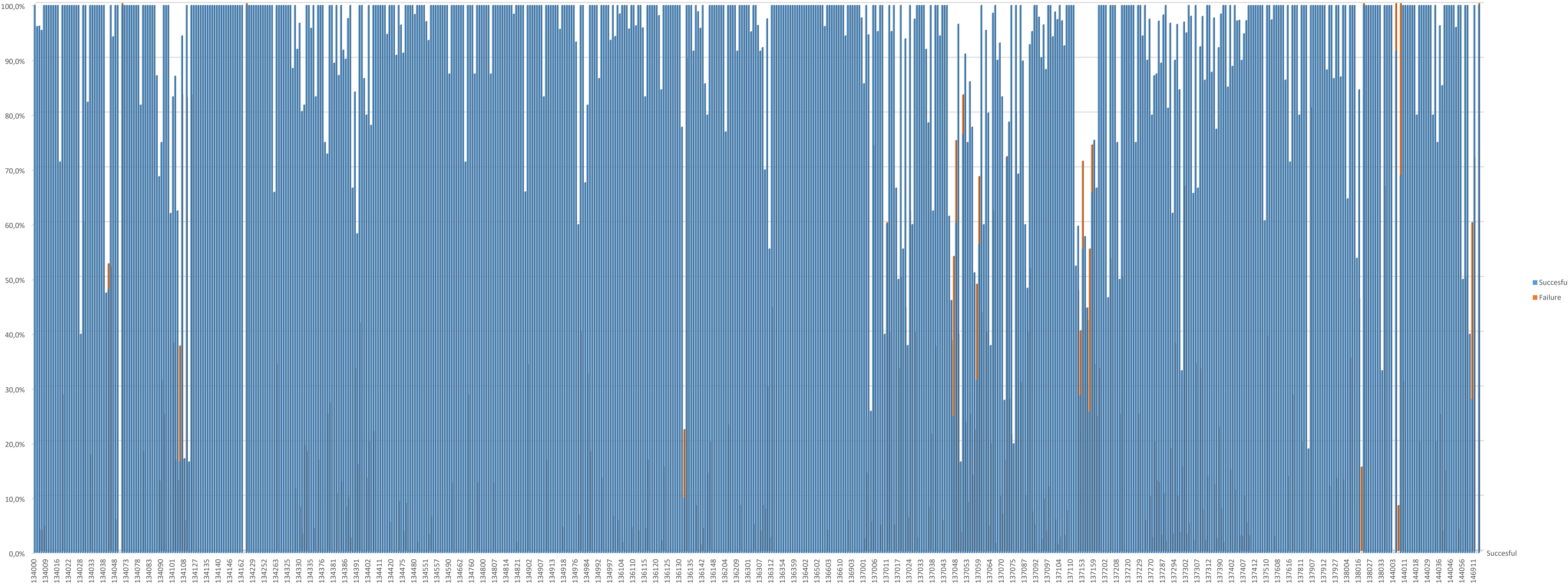
```
Codeunit 130512 Library - Purchase - C/AL Editor
171 CreateVendor(Vendor);
172
173 CreateVendor(VAR Vendor : Record Vendor) : Code[20]
174 LibraryERM.FindPaymentMethod(PaymentMethod);
175 LibraryERM.FindGeneralPostingSetupInvtFull(GeneralPostingSetup);
176 LibraryERM.FindVATPostingSetupInvt(VATPostingSetup);
177 LibraryUtility.UpdateSetupNoSeriesCode(
178     DATABASE::"Purchases & Payables Setup",PurchasesPayablesSetup.FIELDNO("Vendor N
```

```
Codeunit 93020 N4 Library - Purchase - C/AL Editor
1 Documentation()
2 TFS34780 20170308 LVV - Object created
3 TFS35374 20170530 LVV - Added functions (1) SetAttributesOnVendor
4
5 OnRun()
6
7 SetAttributesOnVendor(VAR Vendor : Record Vendor;PricesIncludingVAT : Boolean)
8 //TFS35374-BEGIN
9 Vendor.Blocked := Vendor.Blocked::" ";
10 //TFSTFS35374-END
11
```


RTM – 23 %



Now – 90 %



What did we gain?



YES YOU CAN!

A huge test collateral

- covering a major and essential part of our application
 - i.e. standard functionality we use and have customized
- being run every night
 - while we sleep peacefully
 - drop in success rate showing us something wrong with latest code changes
- great help for refactoring and coming upgrade to NAV 2018

Take-aways / Do's & Don'ts

Take-away

Test Creation

Testing is “all” about data

- most time consuming part

Plan, for existing code, what and when to write tests for

Using MS Tests

Testing is “all” about data

- most errors fixed in test fixture

Solving "first" failures

- will probably reveal next ones

The more fixed, the longer the test run will take

Testing will be done while you're sleeping

Test collateral will grow with each version upgrade

Dos & Don'ts – Test Creation

Dos

Design tests before app code is ready

Code minimalistic & generic

Pay attention to test fixture

Always verify result

Version management on test code

Use standard libraries

Don'ts

Do not wait till app code ready to design your tests

Run in production

Dos & Don'ts – Using MS Tests

Dos

Find most occurring errors

Fix minimalistic & generic

Fix in test fixture (... unless ...)

Version management on test code

Use standard libraries

Don'ts

Run in production

Fix errors in app code (... unless ...)

Fix test fixture manually

References

NAV

- <https://msdn.microsoft.com/en-us/dynamics-nav/testing-the-application>

xUnit Test Patterns

- <http://xunitpatterns.com>

Four-Phase Test

- <https://robots.thoughtbot.com/four-phase-test>

Acceptance Test–Driven Development

- https://en.wikipedia.org/wiki/Acceptance_test%E2%80%93driven_development

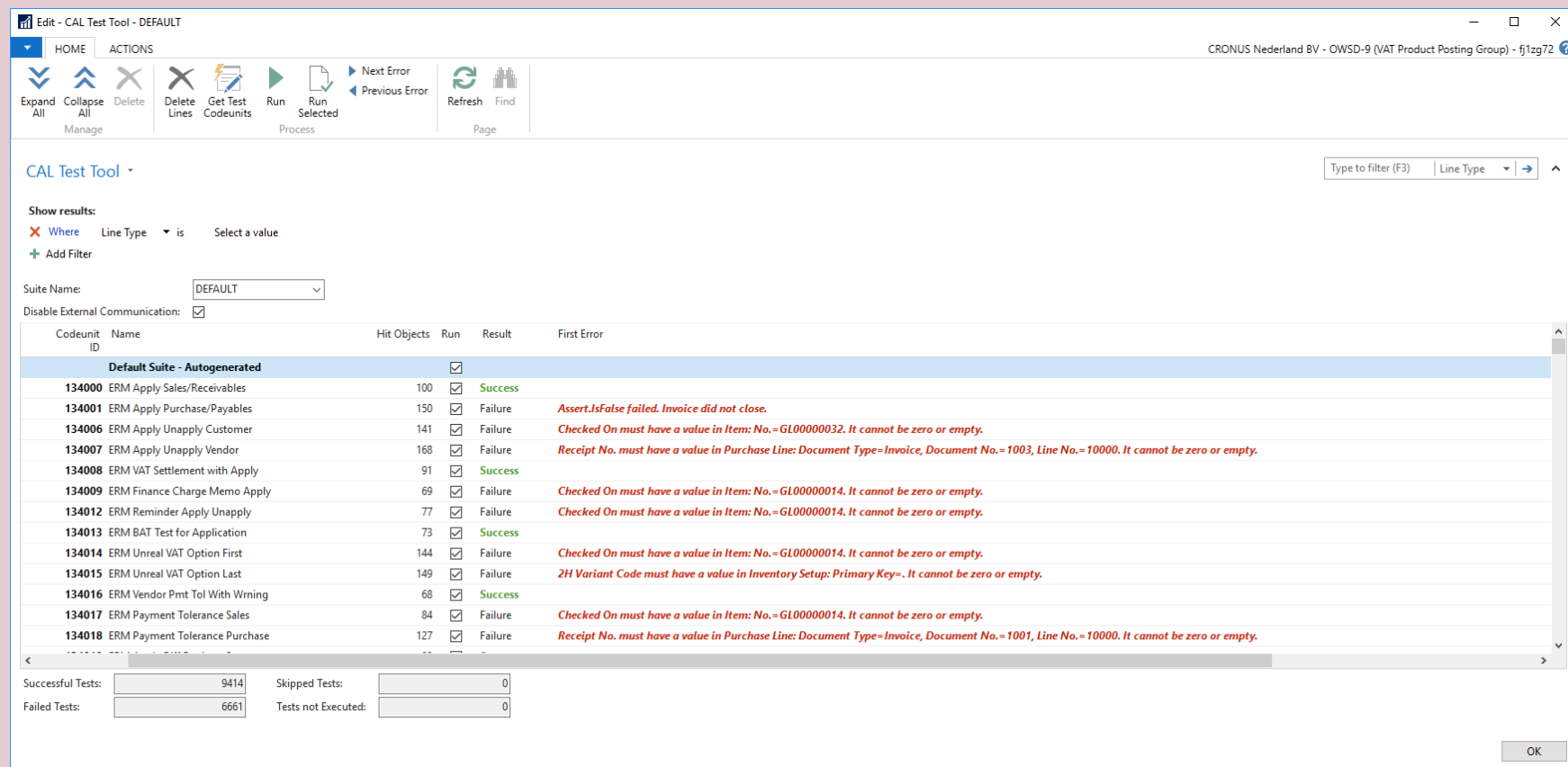
Blogs

- <http://www.fluxxus.nl>
- <http://thenavblog.com/>

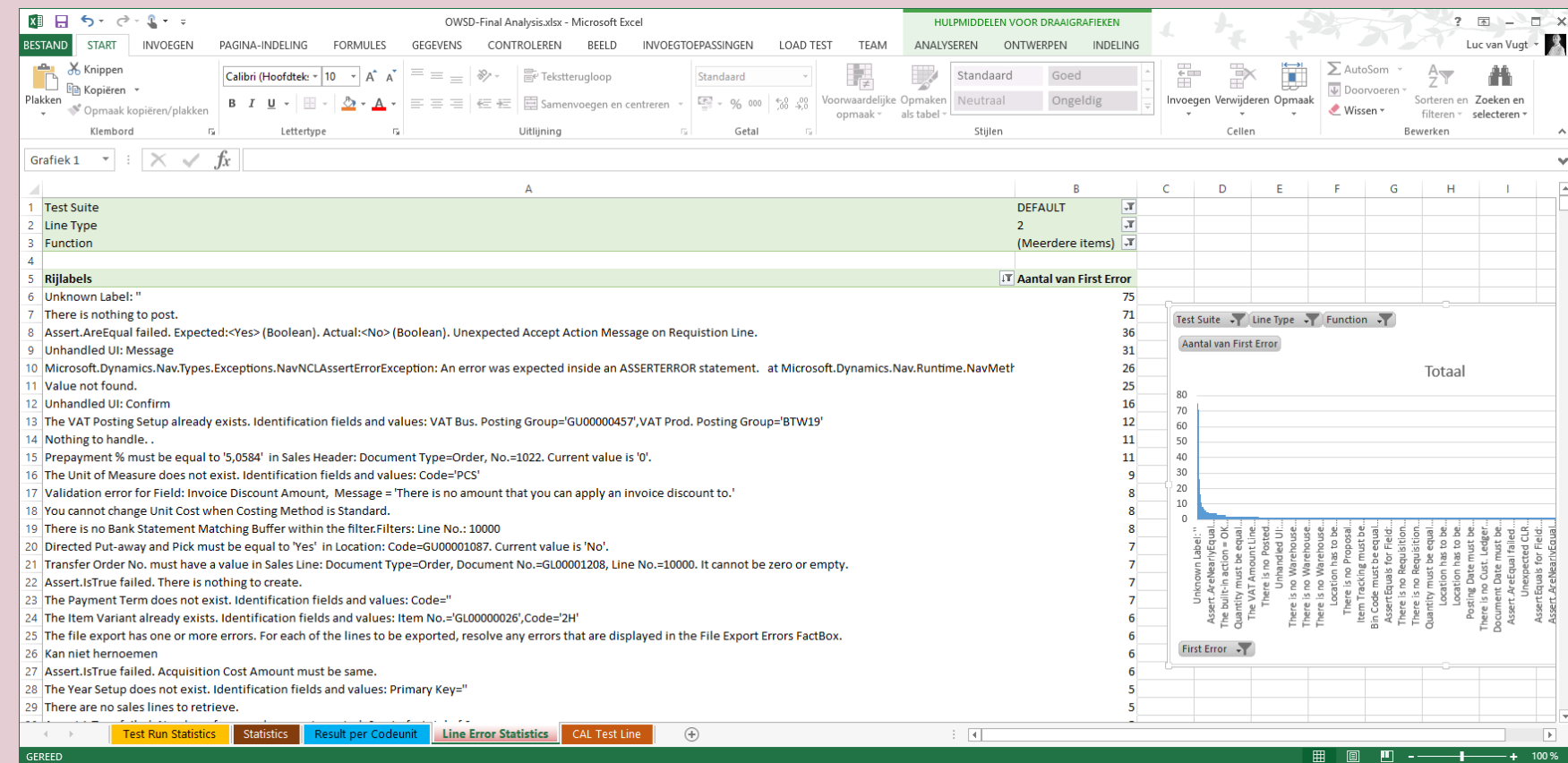


Tools Used

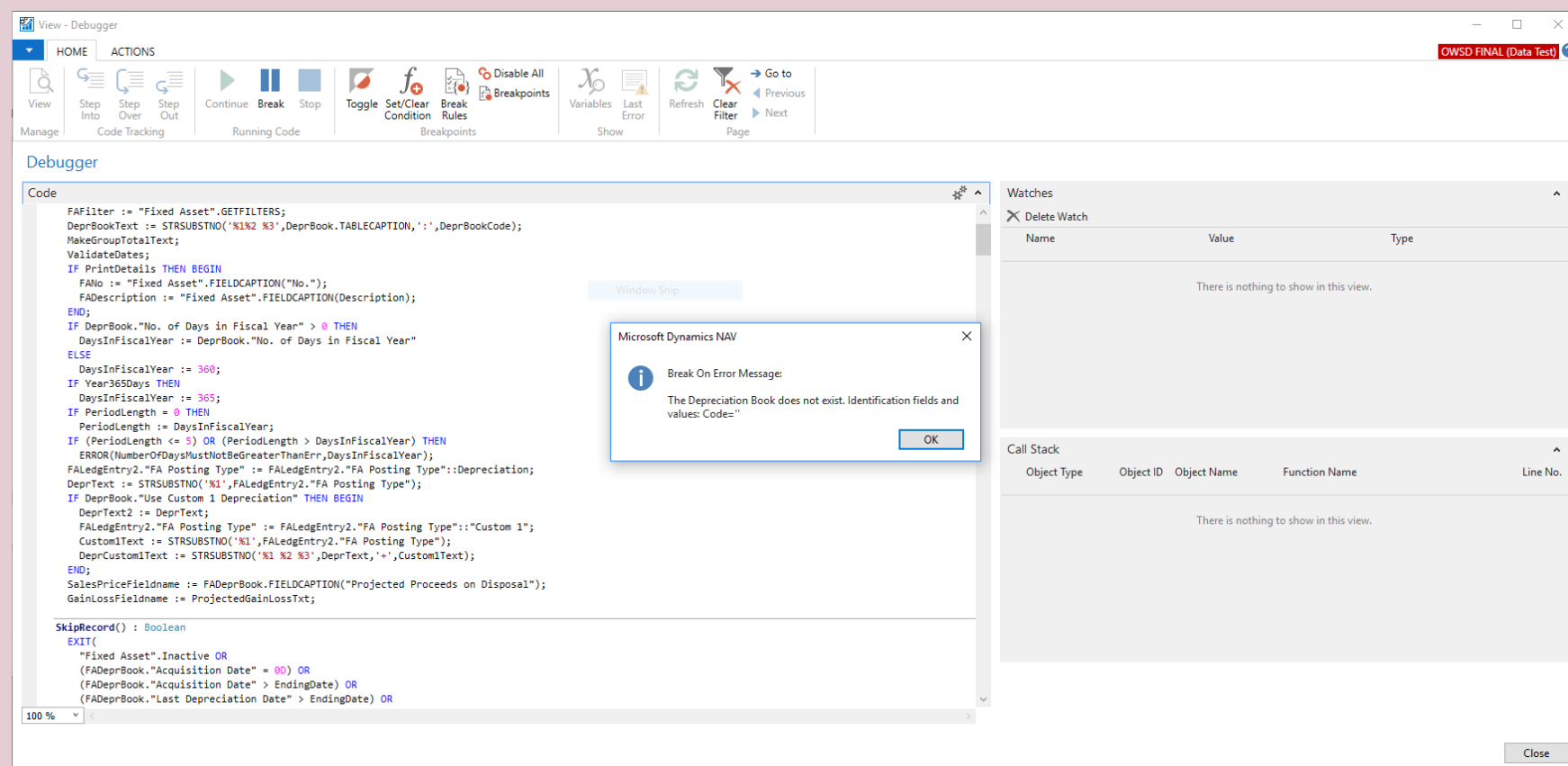
Dynamics NAV Test Tool



Excel



Dynamics NAV Debugger



SQL Server Management Studio

timestamp	VAT Bus_Posting Group	VAT Prod_Posting Group	VAT Calculation Type	VAT	Unrealized VAT Type	Adjust for Payment Discount	Sales VAT Account	Sales VAT Unreal_Account	Purchase VAT Account
0x000000000138F5A1			0	0.00000000000000000000	0	0	1G000000002		1G000000002
0x00000000000631F1		BTW19	0	0.00000000000000000000	0	0			
0x00000000000631E5		GEEN BTW	0	0.00000000000000000000	0	0			
0x0000000000063120		GEEN BTW	0	0.00000000000000000000	0	0			
0x0000000000063122	BINNENLAND	BTW19	0	19.00000000000000000000	0	0	1513		1518
0x0000000000063121	BINNENLAND	BTW6	0	6.00000000000000000000	0	0	1510		1515
0x0000000000063123	BINNENLAND	GEEN BTW	0	0.00000000000000000000	0	0	1514		1519
0x0000000000063125	EU	BTW19	1	19.00000000000000000000	0	0	1513		1518
0x0000000000063124	EU	BTW6	1	6.00000000000000000000	0	0	1510		1515
0x0000000000063126	EU	GEEN BTW	0	0.00000000000000000000	0	0	1514		1519
0x0000000000063128	EXPORT	BTW19	0	0.00000000000000000000	0	0	1513		1518
0x0000000000063127	EXPORT	BTW6	0	0.00000000000000000000	0	0	1510		1515
0x0000000000063128	EXPORT	GEEN BTW	0	0.00000000000000000000	0	0	1514		1519

NAV
TECH
DAYS
2017

mibuso.com

THANX



LUC VAN VUGT
THE LEARNING NETWORK

JAMES CROWTER
TECHNOLOGY MANAGEMENT