
Microsoft Got Sued – And We Had No Tests

How Test Automation Helped With
a Scary System Rewrite

Paul Rohorzka
paul.rohorzka@techtalk.at



June 26th, 2019



www.techtalk.at



Paul Rohorzka

paul.rohorzka@techtalk.at

@paulroho

Software Architect

Software Gardener

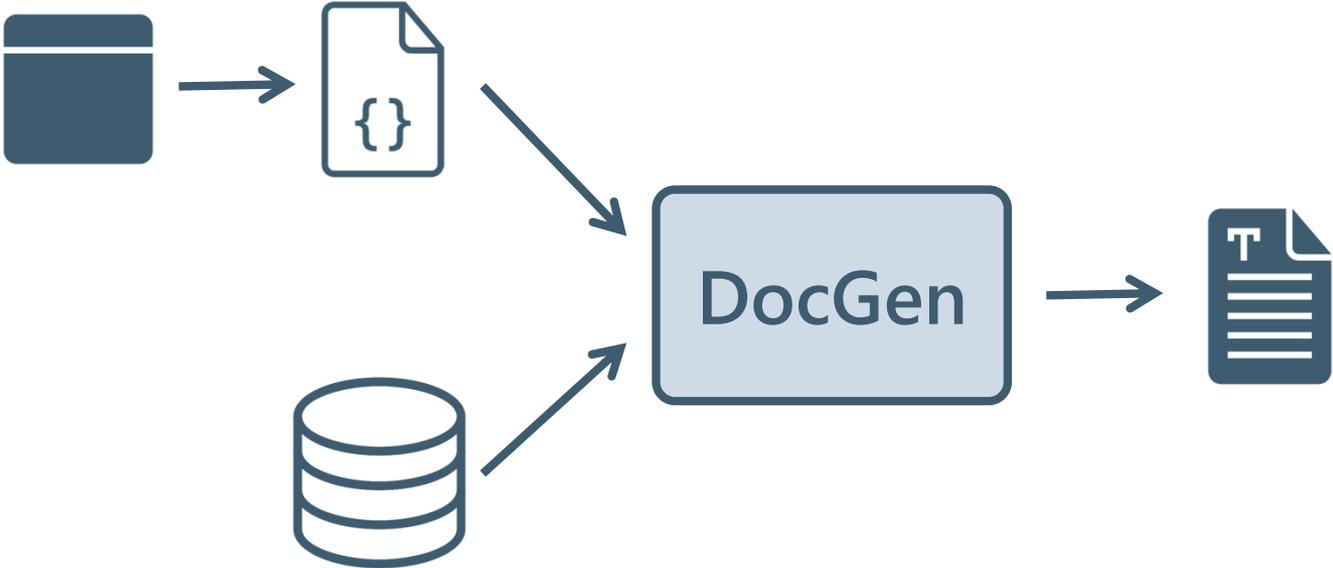
Agile Development Practices

Test Infected



Background

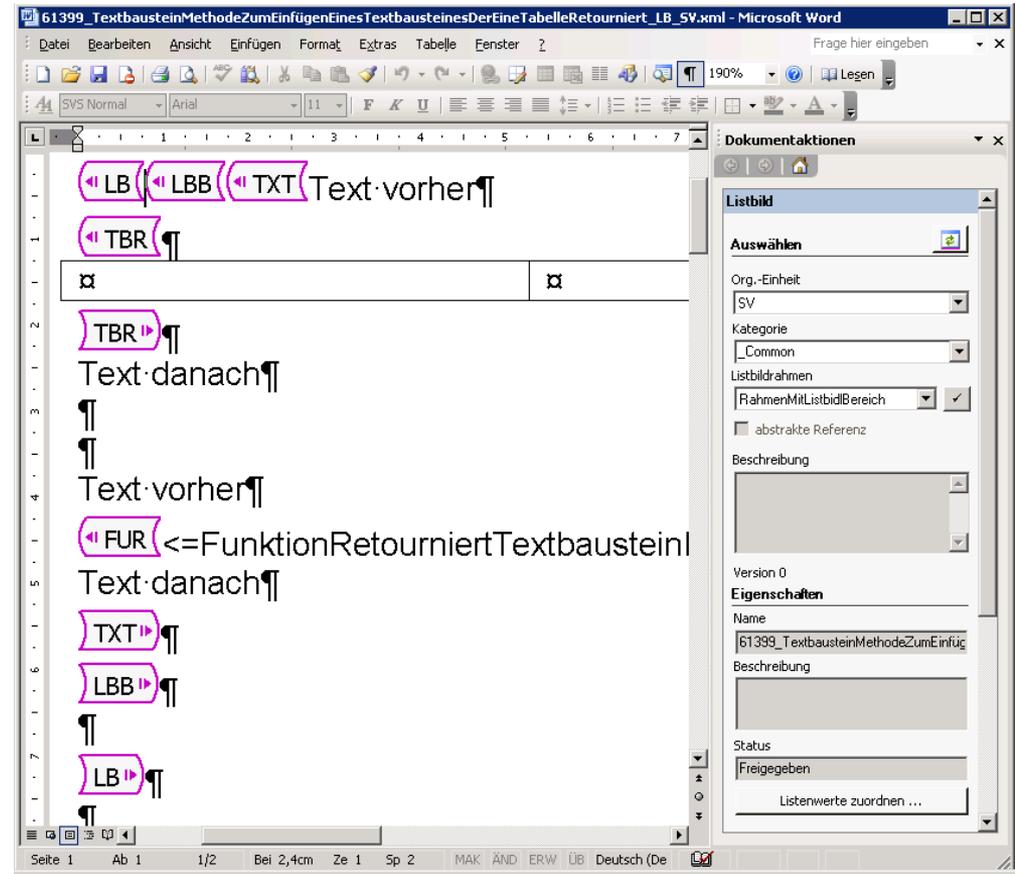
Generating a Document



Foundational Feature: Custom Xml

```
<w:body>
  <wx:sect>
    <ns0:LB beschreibung="" version="0" status="N...
      <ns0:LBB identifier="LBB1">
        <ns0:TXT bearbeitungsBerechtigung="Jede...
          <w:p wsp:rsidR="001C6EC4" wsp:rsidRD...
            <w:pPr>
              <w:pStyle w:val="SVSNormal" />
            </w:pPr>
            <w:rPr>
              <w:noProof />
            </w:rPr>
          </w:p>
          <w:r>
            <w:rPr>
              <w:noProof />
            </w:rPr>
            <w:t>Text vorher</w:t>
          </w:r>
        </w:p>
      <ns0:TBR zielObjektVersion="0" bearb...
        <w:p wsp:rsidR="001C6EC4" wsp:rsid...
          <w:pPr>
            <w:pStyle w:val="SVSNormal" />
          </w:pPr>
```

paul.rohorzka@techtalk.at



2009

Microsoft Got Sued

The story: i4i **sued Microsoft** for patent infringement in Word 2003 and later versions. The feature in question was "**custom XML.**"

i4i wins lawsuit, Microsoft fined \$290 million, and Microsoft is told they **cannot sell Word 2007** with the custom XML feature beginning January 11.

Microsoft appeals lawsuit and **loses on appeal.**

Microsoft then issues a patch stripping Word and other Office programs of custom XML editing capabilities; the injunction is circumvented.



2007



2010



2013



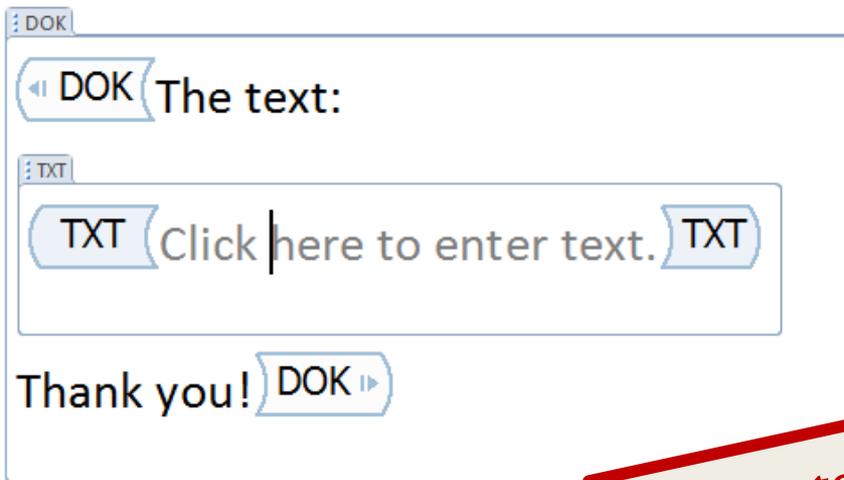
2016

In Production  In Production

Microsoft Got Sued
And ...

**Removed
THE
Foundational Feature**

Alternative: ContentControls



**Migrate?!?
Rewrite?!?**

WordOpenXML
FlatOpc

Microsoft Got Sued
We Had To Rewrite
But ...

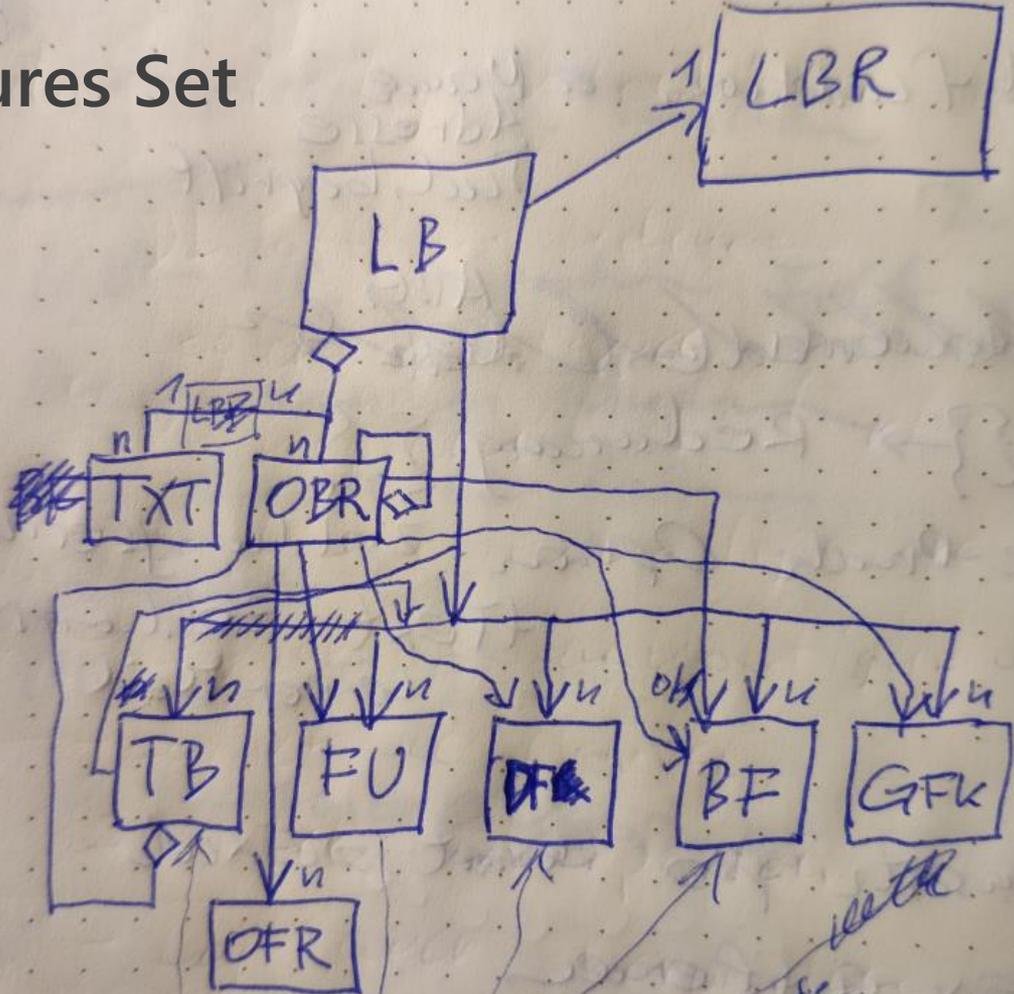
We Had No Tests

No Tests (Describing The Core Domain)

Group By: Namespace; Item(s) selected: 225

Test Name	Project
Dal.SqlCe.Tests	
Dal.SqlCe.Tests.SqlCeDbManagerTests	
Dal.Tests	
Dal.Tests.DbFactoryProviderTests	
DesignerProxy.Tests.Migration	
DesignerProxy.Tests.Offline	
DMS.Tests	
DMS.Tests.DAL Tests	
DMS.Tests.DAL Tests.BenutzerfrageDataAccessTests	
DMS.Tests.DAL Tests.DataFieldDataAccessTests	
DMS.Tests.DAL Tests.DataTypeDataAccessTests	
DMS.Tests.DAL Tests.FunktionDataAccessTests	
DMS.Tests.DAL Tests.ScriptingAssemblyDataAccessTests	
Test	
Test.Base	
Test.Base.Xml.Scripting	
Test.Dal	
Test.Runtime.DocumentGenerator	
Runtime.Tests	

Crazy Features Set



We Had To Rewrite A Crazy System

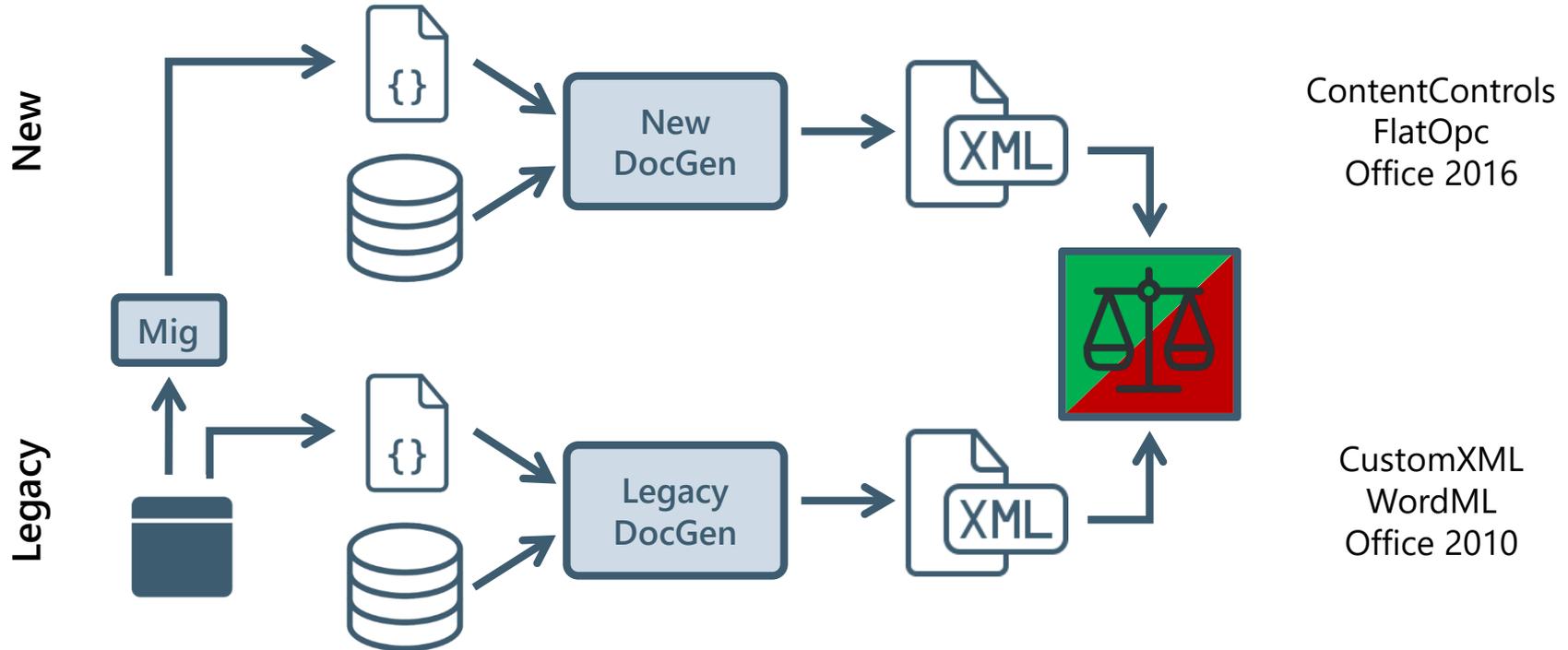
But Nobody Could Tell **Exactly**
What It Was **Supposed To Do**



Make sure it produces
documents identical
to the old system.

Test Automation to the Rescue!

"... Documents Identical to the Old System"



Compare Documents?

File Contents ❌

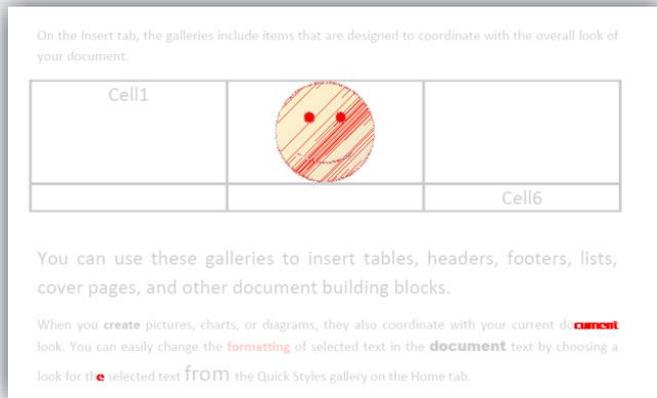


WordML +
CustomXML

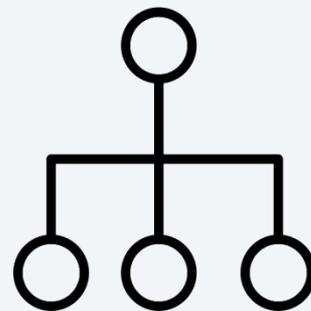


FlatOpc +
ContentControls

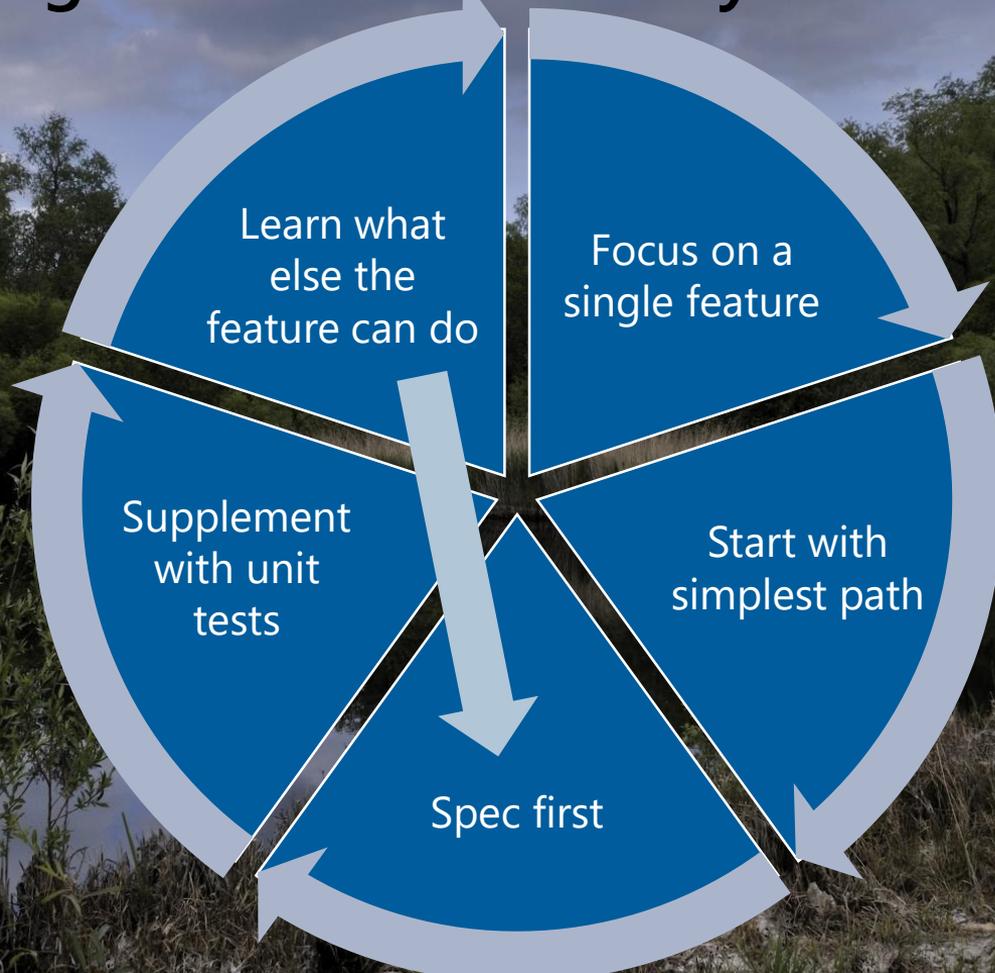
Printout/Rendering ❌



Logical Content! ✓



Approaching Uncharted Territory



Specs

@Runtime

Funktionalität: Funktionen rendern die auf eine OFR zugreifen

@BacklogId-61404

☐ Szenariogrundriss: Runtime rendert Funktionen welche Daten aus einem Objekt retournieren

- Wenn das Szenario '<Szenario>' von der neuen Runtime generiert
- Und mit dem Ergebnis der alten Runtime verglichen wird,
- Dann sollen die beiden Dokumente als identisch angesehen werden.

☐ → Beispiele:

→ Kein Objektbereich im Dokument

- | Szenario
- | In einer Funktion wird direkt (ohne CurrentObjekt) auf Daten eines Objekts zugegriffen.

☐ → Beispiele:

→ Ein Objektbereich im Dokument

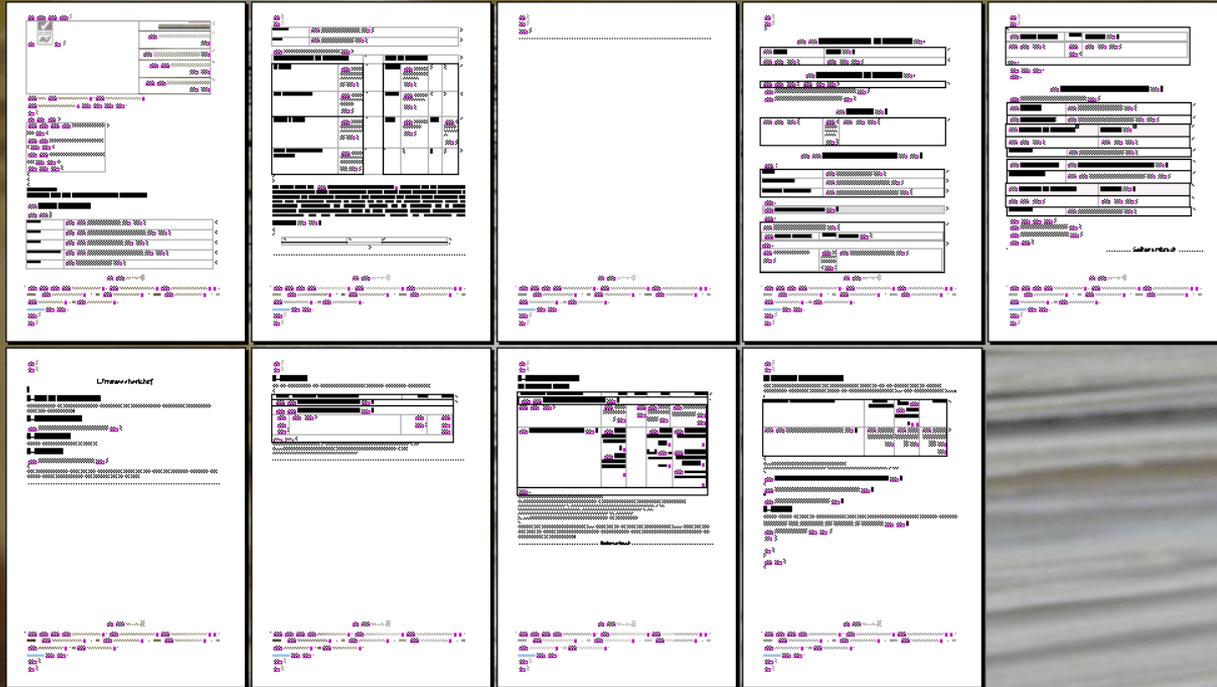
- | Szenario
- | Eine Funktion ist in einem Objektbereich eingebettet der ein einzelnes Objekt liefert.
- | Eine Funktion ist in einem Objektbereich eingebettet der eine Liste von Objekten liefert.

Integrated Tests – Focusing Is Hard

- Szenariogrundriss: Runtime berücksichtigt beim Generieren des Listbilds den Listbildrahmen - bekannte Unterschiede zur alten Runtime
 - Wenn das Szenario '<Szenario>' von der neuen Runtime generiert
 - Und mit dem Ergebnis der alten Runtime verglichen wird,
 - # In der alten Runtime wird in manchen Fällen (z.B. bei leeren Bereichen) einer der leeren Absätze am Ende des Dokuments
 - # nicht gelöscht. Es handelt sich dabei wahrscheinlich um den Absatz, der das VSTO-OCX-Control enthält.
 - Aber beim Vergleichen der letzte Absatz des von der alten Runtime generierten Dokuments ignoriert wird (falls dieser leer ist),
 - Dann sollen die beiden Dokumente als identisch angesehen werden.
- @BacklogId-60734
- Beispiele:
 - | Szenario |
 - | Lists von Listbild und Listbildrahmen werden berücksichtigt |

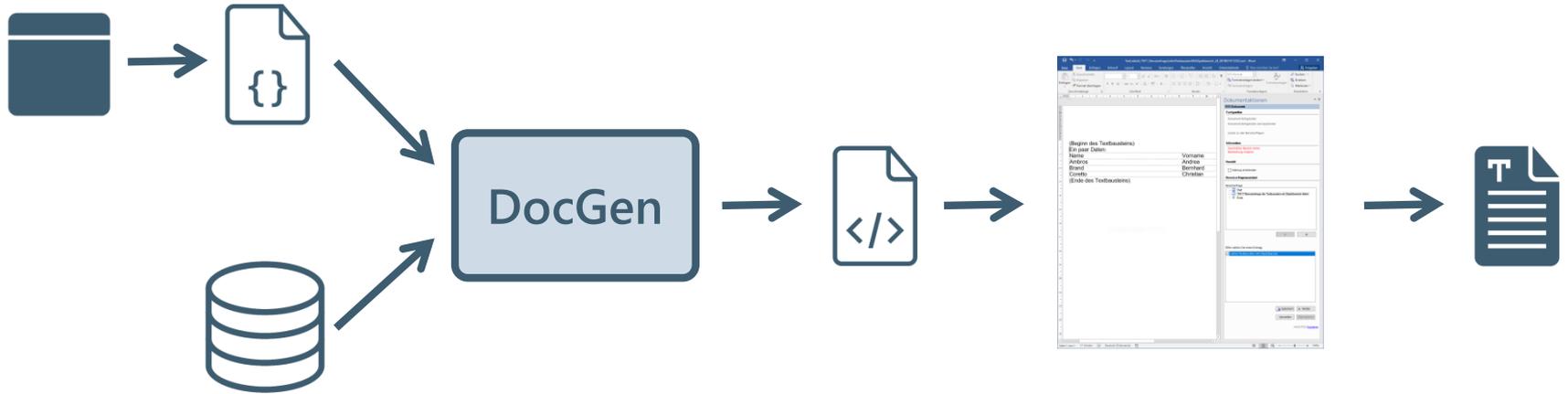
Paragraph Panic!

First Contact

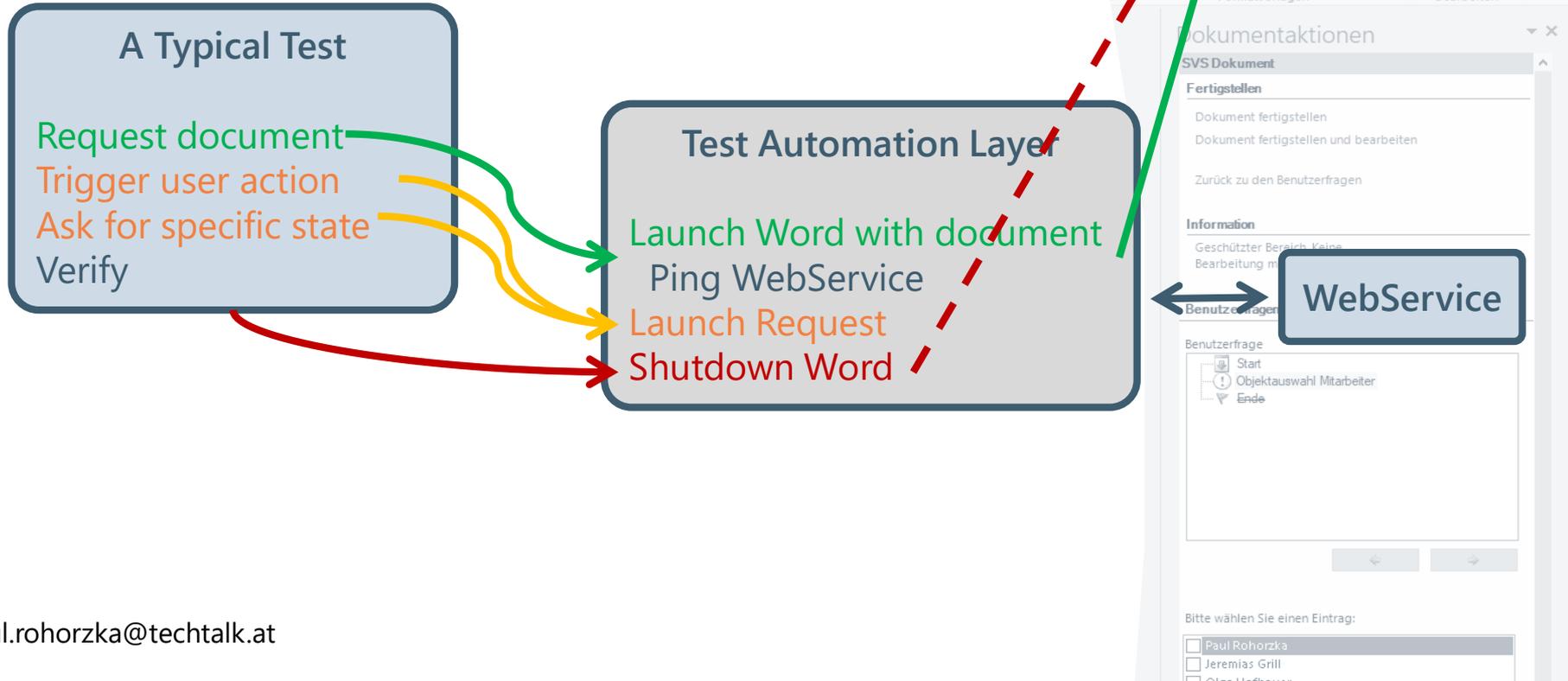


Now It Gets Scary

„User Questions“ – Half Baked Documents



Testing User Interaction With Word



Szenario: Die Antworten von Benutzerfragen mit Einfachauswahl werden in das Dokument eingefügt (mit weiterer Referenz auf die Benutzerfrage)

Angenommen das Szenario '*Benutzerfrage mit Einfachauswahl wird zur Objektauswahl verwendet (mit weiterer Referenz auf die Benutzerfrage)*' wird angenommen.
Und das Dokument wird im Benutzerfragen-Designer geöffnet.

Wenn die Benutzerfrage '*Objektauswahl Mitarbeiter*' mit der Option '*Maria Hofbauer*' beantwortet wird,

Dann sieht das Dokument so aus (ohne Header/Footer und BFL):

DOK

LBB

TXT

Paragraph

"Referenz auf die Benutzerfrage (außerhalb des Objektbereichs):"

BFR

Paragraph

"Maria Hofbauer"

Paragraph

Paragraph

"Objektbereich:"

OBR

Paragraph

"Mitarbeiter "

OFR

"Maria"

" "

OFR

"Hofbauer"

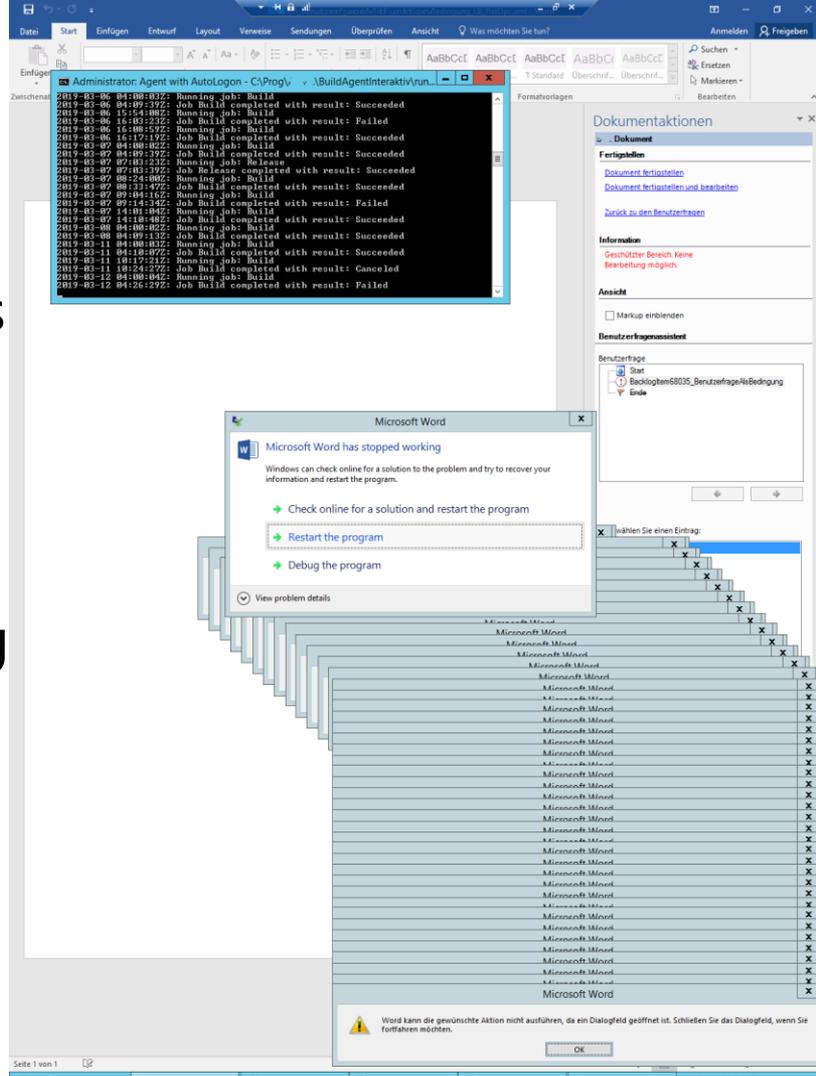
" "

OFR

Now We Could Write Tests Like This

UI-Level Testing Challenges

- Reliably controlling external process
 - Timeouts!
- Modal dialogs – blocking!
- Global dependencies
 - Dev Machine: Windows popping
 - Build-Server: Registry
 - Under the hood features
- Run Time (monitor it!)



Focusing on the Relevant Parts

Dann sieht das Dokument so aus (ohne Header/Footer und BFL):

```
→ ""
→ DOK
→   ... LBB
→     ... TXT
→       ... FUR
→         ... Paragraph
→           ... "(Beginn des Textbausteins)"
→             ... OBR
→               ... Paragraph
→                 ... OFR
→                   ... "Andrea"
→                     ... ""
→                       ... OFR
→                         ... "Ambros"
→                           ... Paragraph
→                             ... "(Ende des Textbausteins)"
→                               ""
```

◀ KB (KB ▶

◀ DOK (◀ LBB (◀ TXT (◀ FUR (Beginn des Textbausteins)▶

OBR (OFR Andrea) OFR Ambros)▶

(Ende des Textbausteins) FUR ▶ TXT ▶ LBB ▶

DOK ▶

◀ BFL ▶

◀ BF (BF ▶ BFL ▶

Expected actualDocAsTree to be

"DOK

LBB

TXT

FUR

Your Existing Tools Might Not Suffice

Paragraph

"(Beginn des Textbausteins)"

OBR

Paragraph

OFR

"Andrea"

" "

OFR

"Ambros"

Paragraph

Paragraph

"(Ende des Textbausteins)"" with a length of 367, but

"DOK

LBB

TXT

FUR

Paragraph

"(Beginn des Textbausteins)"

OBR

Paragraph

OFR

"Andrea"

" "

OFR

"Ambros"

Paragraph

"(Ende des Textbausteins)"" has a length of 409, differs near " " (index 240).

Your Existing Tools Might Not Suffice

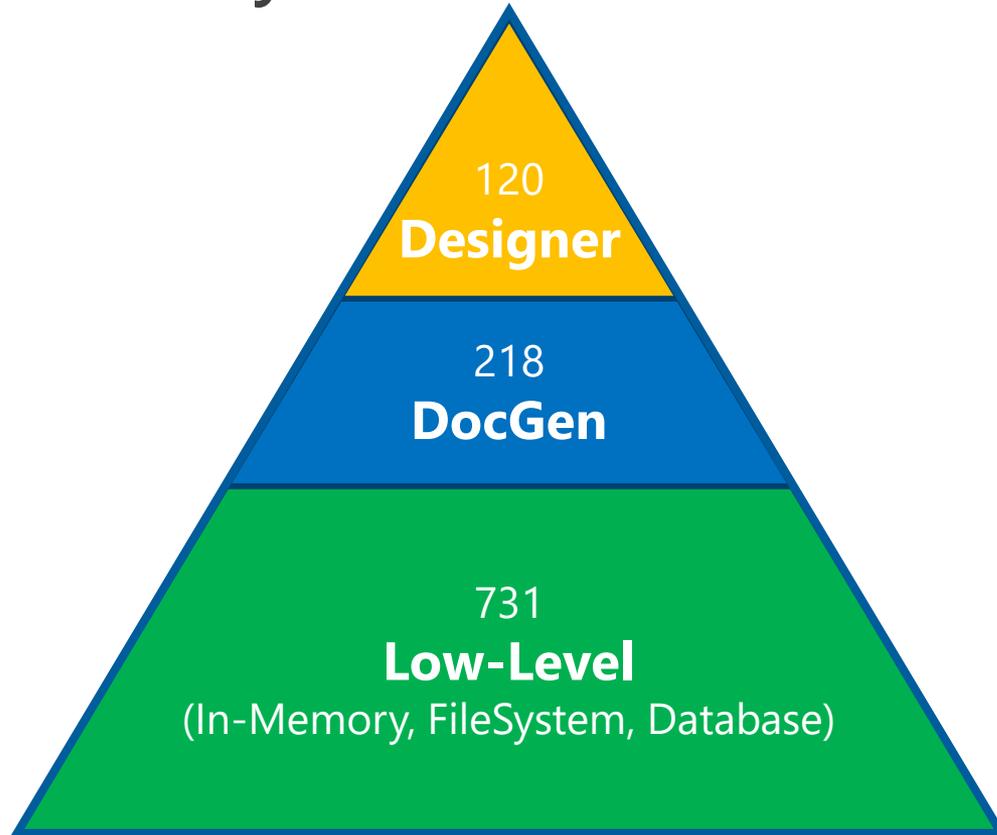
The lines are unequal:

```
"
-Expected-
DOK
  LBB
    TXT
      FUR
        Paragraph
          "(Beginn des Textbausteins)"
        OBR
          Paragraph
            OFR
              "Andrea"
              " "
            OFR
              "Ambros"
          Paragraph
          Paragraph
            "(Ende des Textbausteins)"
"
-Actual-
DOK
  LBB
    TXT
      FUR
        Paragraph
          "(Beginn des Textbausteins)"
        OBR
          Paragraph
            OFR
              "Andrea"
              " "
            OFR
              "Ambros"
          Paragraph
            "(Ende des Textbausteins)"
"
```

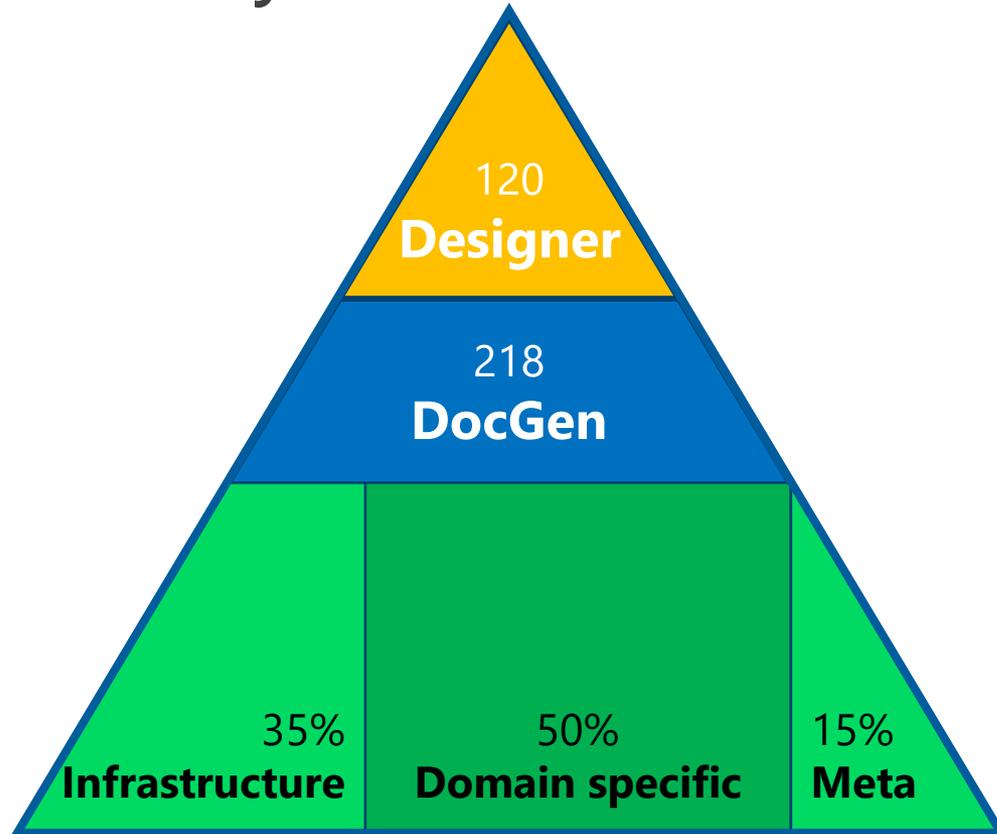
Meta Tests – Testing the Test Automation Layer

```
..... [TestMethod]
..... public void AParagraphWithMultipleTextRunsSeparatedByStructuredDocumentTags
..... {
.....     CheckTree_SkippingLastParagraph(
.....         doc => doc.DOK(
.....             dok => dok.LBB(
.....                 lbb => lbb.TXT(
.....                     obr => obr.Paragraph(
.....                         p => p.HasARun("The text before the OFR"),
.....                         p => p.OFR(),
.....                         p => p.HasARun("The text after the OFR")))
.....                     )),
.....         expectedTree: @"
DOK
..... LBB
.....     TXT
.....         Paragraph
.....             ""The text before the OFR""
.....             OFR
.....             ""The text after the OFR""");
..... }
```

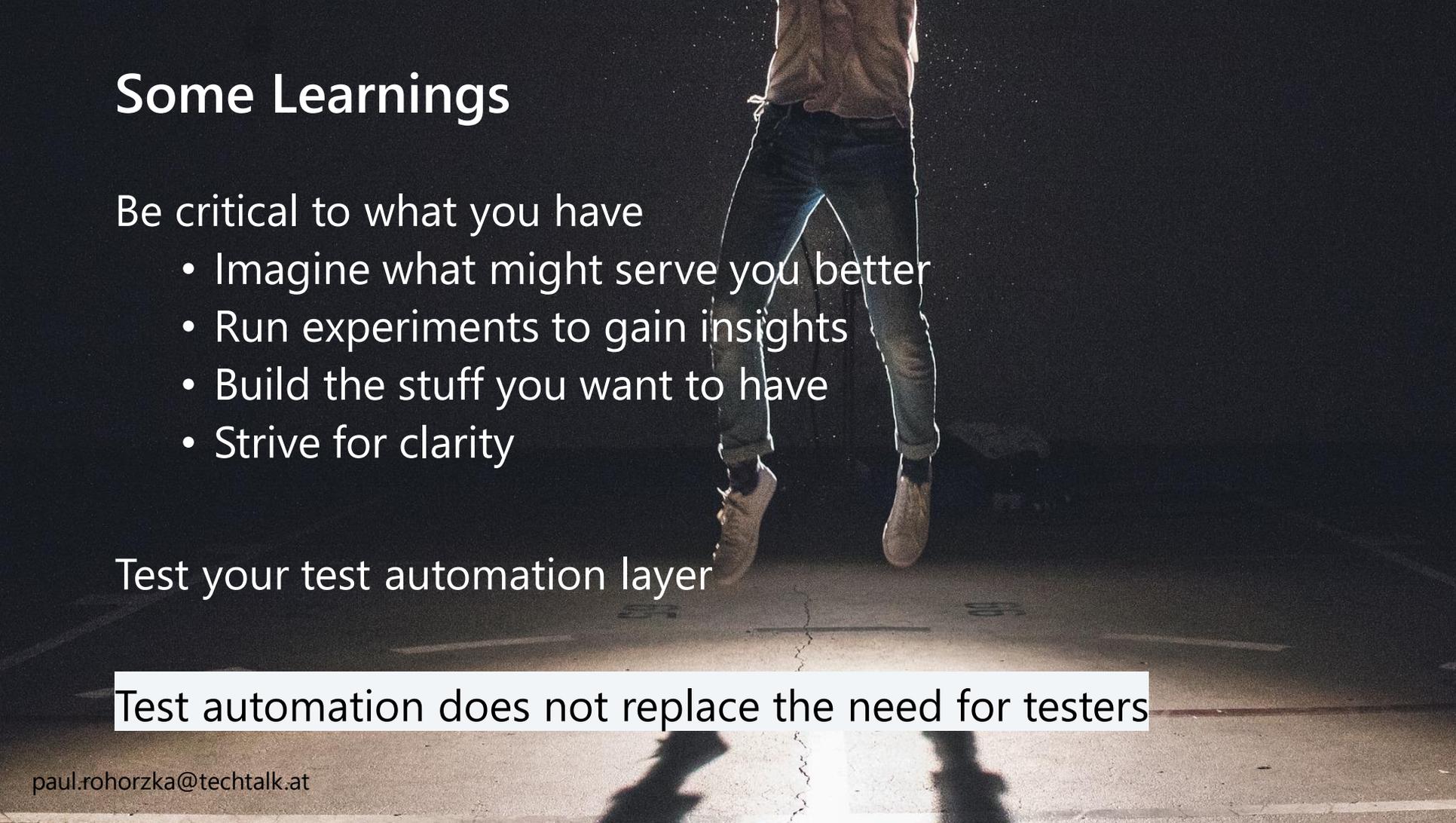
Test Automation Pyramid – Measured!



Test Automation Pyramid – Measured!



Some Learnings

A person is captured mid-jump in a dark environment. A bright spotlight illuminates the person's feet and the ground directly beneath them, creating a stark contrast with the surrounding darkness. The person is wearing a light-colored long-sleeved shirt, dark jeans, and light-colored sneakers. The background is mostly black, with some faint white lines on the floor suggesting a parking lot or a similar outdoor setting.

Be critical to what you have

- Imagine what might serve you better
- Run experiments to gain insights
- Build the stuff you want to have
- Strive for clarity

Test your test automation layer

Test automation does not replace the need for testers

Let The Tests Talk Your Language

paul.rohorzka@techtalk.at

TECHTALK 

The logo icon consists of two blue squares of different sizes, one above and to the right of the other, with a small blue square below the larger one, forming a stylized speech bubble or message icon.