DITA and Markdown

Alex Jitianu

alex_jitianu@oxygenxml.com

@AlexJitianu

tekom Jahrestagung 2019
STUTTGART, 12. – 14. NOVEMBER

tcworld conference 2019
STUTTGART, NOVEMBER 12 – 14

Your opinion is important to us! Please tell us what you thought of the lecture. We look forward to your feedback under

http://swd04.honestly.de

or scan the QR code

The feedback tool will be available even after the conference!

Agenda

- Content and markup
- Structured authoring
- Markdown
- DITA
- Markdown meets DITA in a docs project

About the Tools

It all starts with the content

Create a Google account

How to create or set up your Google Account on your mobile phone.

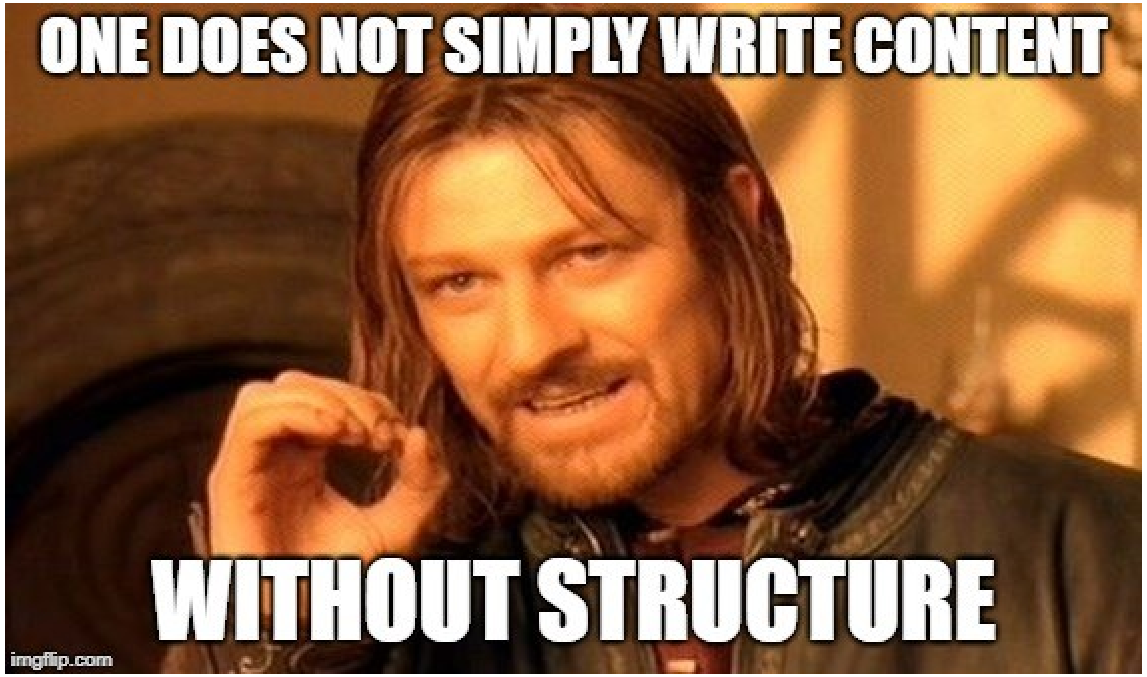
From a Home screen, swipe up to access Apps.

Tap Settings > Accounts

Tap Add account > Google.

A typical task

Content alone is not enough



Why do we need structure?

- Defines the organization/model of content
- Helps us enforce the defined model
- Increases consistency
- Automatic processing
- Faster publishing workflows

What Meaning Lies Beneath

Create a Google account

Title

How to create or set up your Google Account on your mobile phone.

Short description

From a Home screen, swipe up to access Apps.

Tap Settings > Accounts

Tap Add account > Google.

Procedure

Encode it with a Markup Language

Create a Google account

Title

How to create or set up your Google Account on your mobile phone.

Short description

From a Home screen, swipe up to access Apps.

Tap Settings > Accounts

Tap Add account > Google.

Procedure

Markdown

- Easy to learn
- Minimalistic
- Many authoring tools available
- Publishing tools

```
Create a Google account
=============

How to create or set up your **Google Account** on
your mobile phone.

* From a Home screen, swipe up to access Apps.
* Tap **Settings** > **Accounts**
* Tap **Add account** > **Google**.
```

XML

- e(**X**)tensible (**M**)arkup (**L**)anguage
- Define your own tags/markup
- Powerful mechanisms to enforce valid content structure (DTD/Schema/Schematron)

DITA

- DITA is an XML-based open **standard** for structuring, developing, managing, and publishing content.
- Semantic markup (separates formatting from content)
- Strong content reuse concepts
- Restrictions and specializations
- Huge ecosystem of publishing choices

Side by side

```
Create a Google account
=============

How to create or set up your **Google Account**
on your mobile phone.

1. From a Home screen, swipe up to access Apps.
1. Tap **Settings** > **Accounts**
1. Tap **Add account** > **Google**.
```

```
<topic id="create_google_account">
  <title>Create a Google account</title>
  <shortdesc>How to create or set up your <b>Google Account</b> on your
mobile phone.</shortdesc>
  <body>
    <ol>
      <li>From a Home screen, swipe up to access Apps.</li>
      <li>Tap <b>Settings</b> / <b>Accounts</b></li>
      <li>Tap <b>Add account</b> / <b>Google</b></li>
    </ol>
  </body>
</topic>
```

Side by side

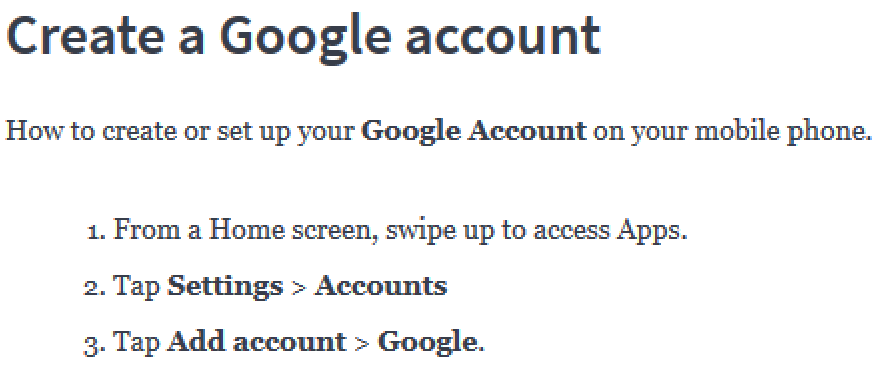
```
Create a Google account
============

How to create or set up your **Google Account**
on your mobile phone.

1. From a Home screen, swipe up to access Apps.
1. Tap **Settings** > **Accounts**
1. Tap **Add account** > **Google**.
```

```
<task id="create_google_account">
  <title>Create a Google account</title>
  <shortdesc>How to create or set up your <term>Google Account</term>
on your mobile phone.</shortdesc>
  <taskbody>
    <steps>
      <step>
        <cmd>From a Home screen, swipe up to access Apps.</cmd>
      </step>
      <step>
        <cmd>Tap <menucascade>
          <uicontrol>Settings</uicontrol>
          <uicontrol>Accounts</uicontrol>
        </menucascade></cmd>
      </step>
      <step>
        <cmd>Tap <menucascade>
          <uicontrol>Add account</uicontrol>
          <uicontrol>Google</uicontrol>
        </menucascade></cmd>
      </step>
    </steps>
  </taskbody>
</task>
```

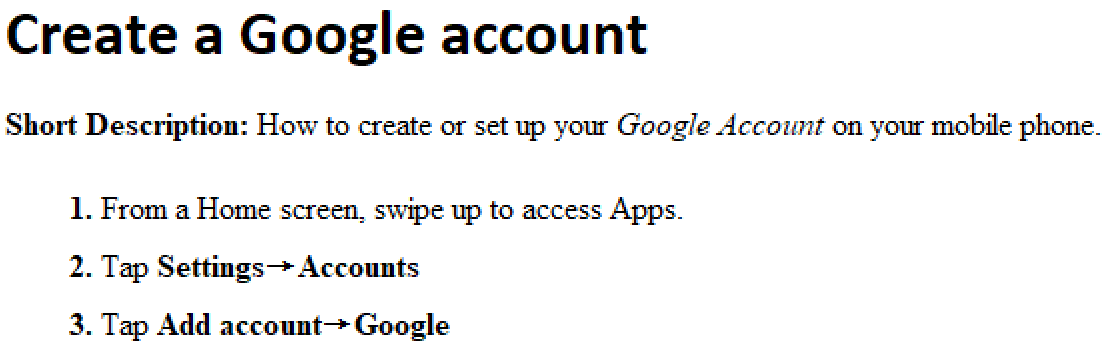
Side by side – visual mode



Create a Google account

How to create or set up your **Google Account** on your mobile phone.

1. From a Home screen, swipe up to access Apps.
2. Tap **Settings** > **Accounts**
3. Tap **Add account** > **Google**.



Create a Google account

Short Description: How to create or set up your *Google Account* on your mobile phone.

1. From a Home screen, swipe up to access Apps.
2. Tap **Settings**→**Accounts**
3. Tap **Add account**→**Google**

Scenario

- Main documentation project written in DITA
- Contributors (devs) sending content in Markdown

Will they be foes?

https://pixabay.com/photos/cat-dog-pets-curious-play-friends-793276/

Will they be friends?

https://pixabay.com/photos/friends-cat-and-dog-cats-and-dogs-1149841/

[1] Convert MD to DITA

- MD => HTML => DITA
  - https://pandoc.org/
  - http://dita-ot.sourceforge.net/1.5.2/readme/DITA-h2d-ant.html#h2d-ant
- MD => DITA
  - DITA-OT plugin developed by Jarno Elovirta

    https://github.com/jelovirt/dita-ot-markdown
  - Oxygen Batch Converter plugin

    https://github.com/oxygenxml/oxygen-resources-converter

DEMO TIME

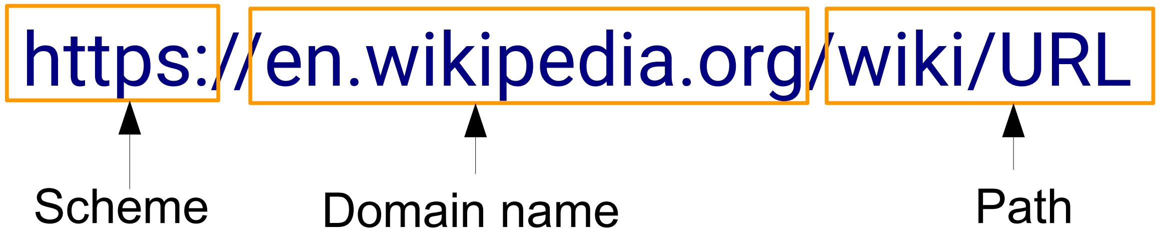
- From the command line

```
cd dita-meets-markdown
dita-ot-3.3.4\bin\dita.bat -i demo-files\conversion\sample.ditamap -format dita -o out
```

- From Oxygen

[2] Keep MD and use it in DITA

- Dynamic conversion (custom URL)
  - https://en.wikipedia.org/wiki/URL



  - <topicref href="md2dita:/topic.md" format=”dita"/>
  - https://github.com/oxygenxml/dita-glass

[2] Keep MD and use it in DITA

- Dynamic conversion (custom URL)
  - <topicref href="md2dita:/topic.md" format=”dita"/>
  - https://github.com/oxygenxml/dita-glass
- Refer MD files directly in your map
  - <topicref href="tasks/changingtheoil.md" format="markdown"/>
- Seamless publishing

DEMO TIME

- Using specific DITA concepts in MD
  - Metadata
  - Specialization types
  - Titles and document structure
  - Image and Key references

Syntax reference:
https://github.com/jelovirt/dita-ot-markdown/wiki/Syntax-reference

What is Lightweight DITA?

- LwDITA is a proposed standard for expressing *simplified DITA documents* in *XML, HTML5, and Markdown*.
- The core goals of LwDITA:
  - Provide a simpler DITA experience
  - Provide mappings between *XML, HTML5, and Markdown* that enable individuals to:
    - Author content in the format of their choice
    - Easily exchange and publish content whose source exists in these different markup languages

What is Lightweight DITA?

- LwDITA is a proposed standard for expressing *simplified DITA documents* in *XML, HTML5, and Markdown*.
- The core goals of LwDITA:
  - Provide a simpler DITA experience
  - Provide mappings between *XML, HTML5, and Markdown* that enable individuals to:
    - Author content in the format of their choice
    - Easily exchange and publish content whose source exists in these different markup languages

[2] Keep MD and use it in DITA

- Dynamic conversion (custom URL)
  - <topicref href="md2dita:/topic.md" format=”dita"/>
  - https://github.com/oxygenxml/dita-glass
- Refer MD files directly in your map
  - <topicref href="tasks/changingtheoil.md" format="markdown"/>
  - <topicref href="tasks/changingtheoil.md" **format="mdita"**/>
- Seamless publishing

DEMO TIME

- Working on LWDITA topics
  - YAML header
  - Short descriptions
  - Reuse
    - Content references
    - Key reference
    - Linking
    - Variable text

Advantages

- Single sourcing across DITA and Markdown
- Collaboration on Markdown source
- Use DITA features and publishing options with Markdown
- Use Markdown publishing options
  - https://github.com/jelovirt/dita-ot-markdown#generating-markdown-output
  - Make use of publishing platforms like Jekyll, Vuepress, Mkdocs etc.
    - https://nostalgic-hamilton-b6dae0.netlify.com/

Disadvantages

- Markdown lacks semantics
  - https://github.com/IBM-Cloud/docs-services/tree/staging#using-the-copyright-and-last-updated-header-required
  - https://raw.githubusercontent.com/IBM-Cloud/docs-services/staging/getting_started_template/servicename_task.md

Disadvantages

- Markdown lacks semantics
- Consistency challenges (not a standard, can receive different flavors)
    - https://github.com/IBM-Cloud/docs-services/tree/staging#using-the-copyright-and-last-updated-header-required
    - https://raw.githubusercontent.com/IBM-Cloud/docs-services/staging/getting_started_template/servicename_task.md

Disadvantages

- Markdown lacks semantics
- Consistency challenges (not a standard, can receive different flavors)
- Markdown language restrictions
- No reuse mechanisms (unless you go with LwDITA)

Disadvantages

- Review/Collaboration tracking challenges
  - *When done on the Markdown source*
    - No support for tracking changes
    - Can be difficult to visualize changes (diffs needed)
  - *When done on converted content, like PDF*
    - *Writer* Generates PDF => *Devs* review on PDF => *Writer* incorporates Review
    - Extra overhead to incorporate review into source

Markdown Consistency Challenges

- What can we do?

Vale

- A syntax-aware linter for prose built with speed and extensibility in mind. https://errata-ai.github.io/vale/
- Supports plain text, markup (**Markdown**, reStructuredText, AsciiDoc, and HTML)
- YAML-based extension system

Vale

- Support the following types of rules/styles:
  - Existance
  - Substitution
  - Occurrence
  - Repetition
  - ……….

```
extends: existence
message: "Don’t use end punctuation in headings."
link:
https://docs.microsoft.com/en-us/style-guide/punctuation/periods
nonword: true
level: warning
scope: heading
tokens:
  - '[a-z0-9][.?!](?:\s|$)'
```

https://errata-ai.github.io/vale/styles/#creating-a-style

https://github.com/errata-ai/Microsoft/blob/master/Microsoft/HeadingPunctuation.yml

Markdown Consistency Challenges

- XML/DITA has Schematron
- It does structure checks too

```
<sch:pattern>
    <sch:rule context="topic">
        <sch:assert test="shortdesc">Please add a short description.</sch:assert>
    </sch:rule>
</sch:pattern>
```

schematron
A language for making assertions about patterns found in XML documents

Schematron for Markdown

- Markdown syntax maps to a subset of HTML tags
- Apply Schematron on the HTML with back-mapping support

Why do Devs tend to use Markdown?

- It has a low learning curve
- You do things fast
- Don't have time to learn another language
- They don’t need any additional tool installed on their system

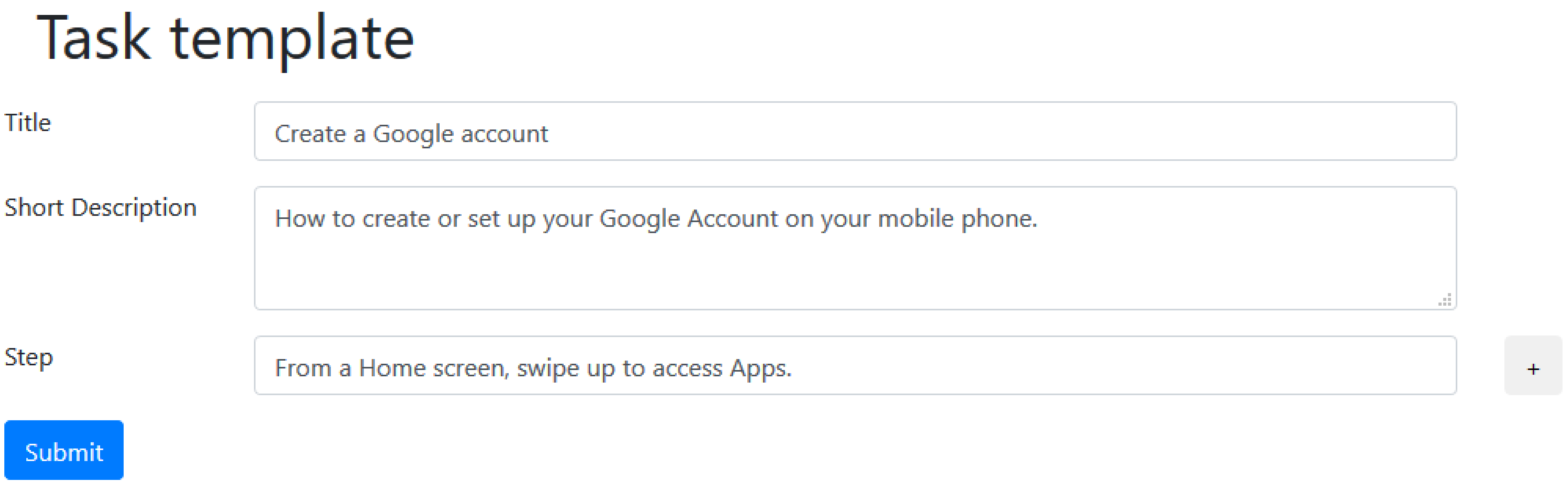
[3] Could Devs write DITA?

- Learn it as you use it through Markdown2DITA controlled conversions. Like a *“Learning assistant”*.
  - Powered by Schematron Quick Fixes
  - https://github.com/oxygenxml/ditaMark

[3] Could Devs write DITA?

- Learn it as you use it through Markdown2DITA controlled conversions. Like a *“Learning assistant”*.
  - Powered by Schematron Quick Fixes
  - https://github.com/oxygenxml/ditaMark
- Give Devs specialized editing enviroments (cloud based):
  - Specialized UI (e.g. HTML form) that generates consistent DITA
  - Specialized Web based DITA editors
    - *Oxygen XML Web Author*
    - *Xeditor*
    - *Fonto*

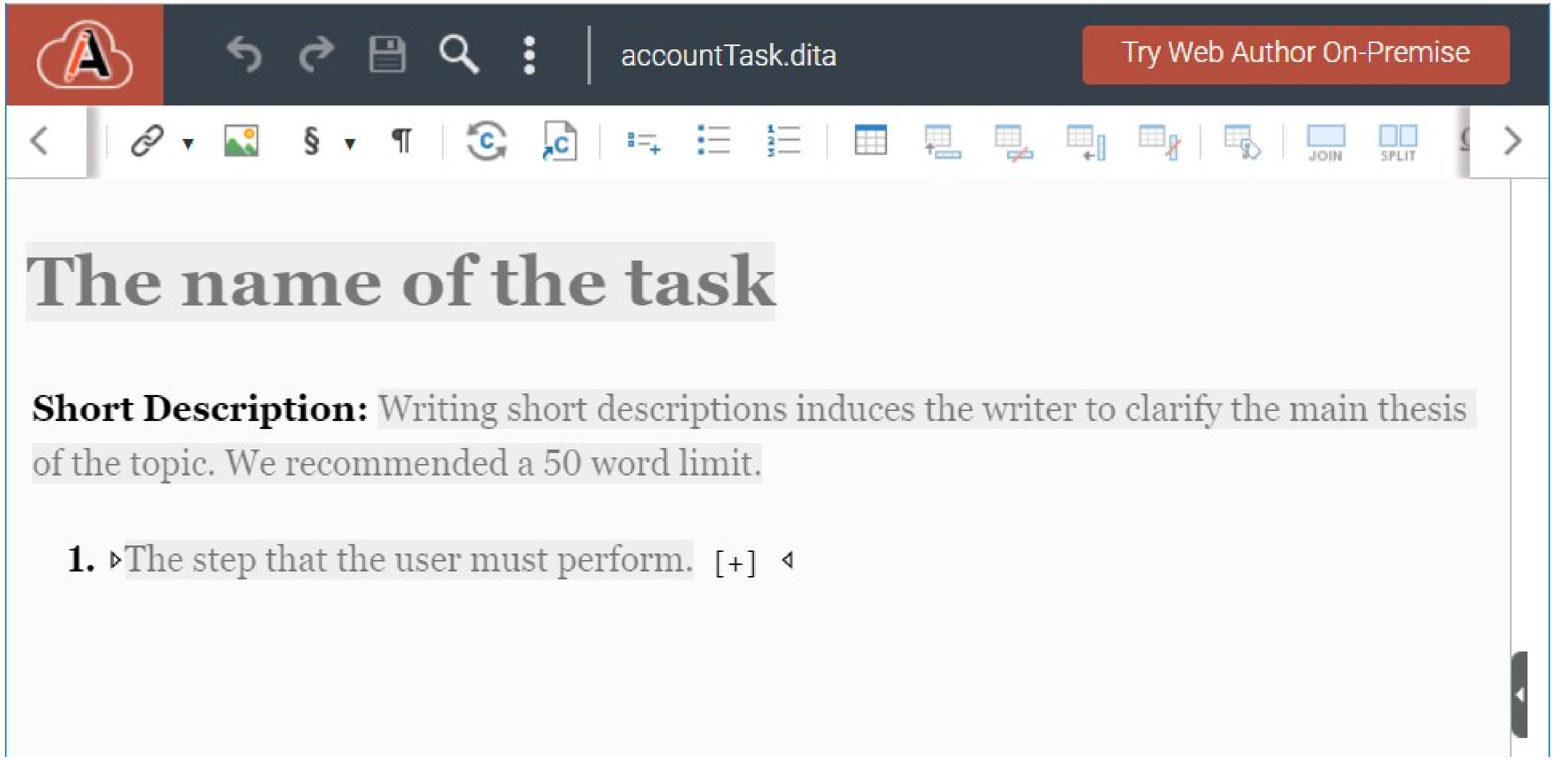
Could Devs write DITA?



Task template

Title	Create a Google account
Short Description	How to create or set up your Google Account on your mobile phone.
Step	From a Home screen, swipe up to access Apps.

Could Devs write DITA?



The name of the task

Short Description: Writing short descriptions induces the writer to clarify the main thesis of the topic. We recommended a 50 word limit.

1. The step that the user must perform. [+]

Which way to go?

- There is no one-size-fits-all solution
- Convert if it's a one time thing
- Keep them together, achieve single sourcing
- Consistency/Collaboration might require a switch to DITA

THANK YOU!

Any questions?

Alex Jitianu
alex_jitianu@oxygenxml.com
@AlexJitianu

tekom Jahrestagung 2019
STUTTGART, 12. – 14. NOVEMBER

tcworld conference 2019
STUTTGART, NOVEMBER 12 – 14

Your opinion is important to us! Please tell us what you thought of the lecture. We look forward to your feedback under

http://swd04.honestly.de

or scan the QR code

The feedback tool will be available even after the conference!