Documentation Quality Assurance with ISO Schematron

Octavian Nadolu, Syncro Soft
octavian_nadolu@oxygenxml.com
@OctavianNadolu

syncro soft

®
XML Editor

About me

Software Architect at Syncro Soft

octavian.nadolu@oxygenxml.com

- 15+ years of XML technology experience
- Contributor for various XML-related open source projects
- Editor of Schematron QuickFix specification developed by a W3C community group

schematron
Structured
editing
XML
review
XQuery
Publish
PDF
DTD DocBook
oXygen
authoring
XML Editor
XSD SCH XSD
XPR RNC FO
frameworks
Profiling
WSDL
styles
visual
WebHelp
DITA
TEI
XSL
PHP
Ant
Js

JS
KML
XSLT
SVN
JSON
SVG
IDREFS
WebDAV
Single
Sourc
Datab
XHTML
Cha
Col

Agenda

- Quality assurance (QA)

- ISO Schematron

- Schematron rules for a documentation project

- Types of quality assurance rules

- Generate a report
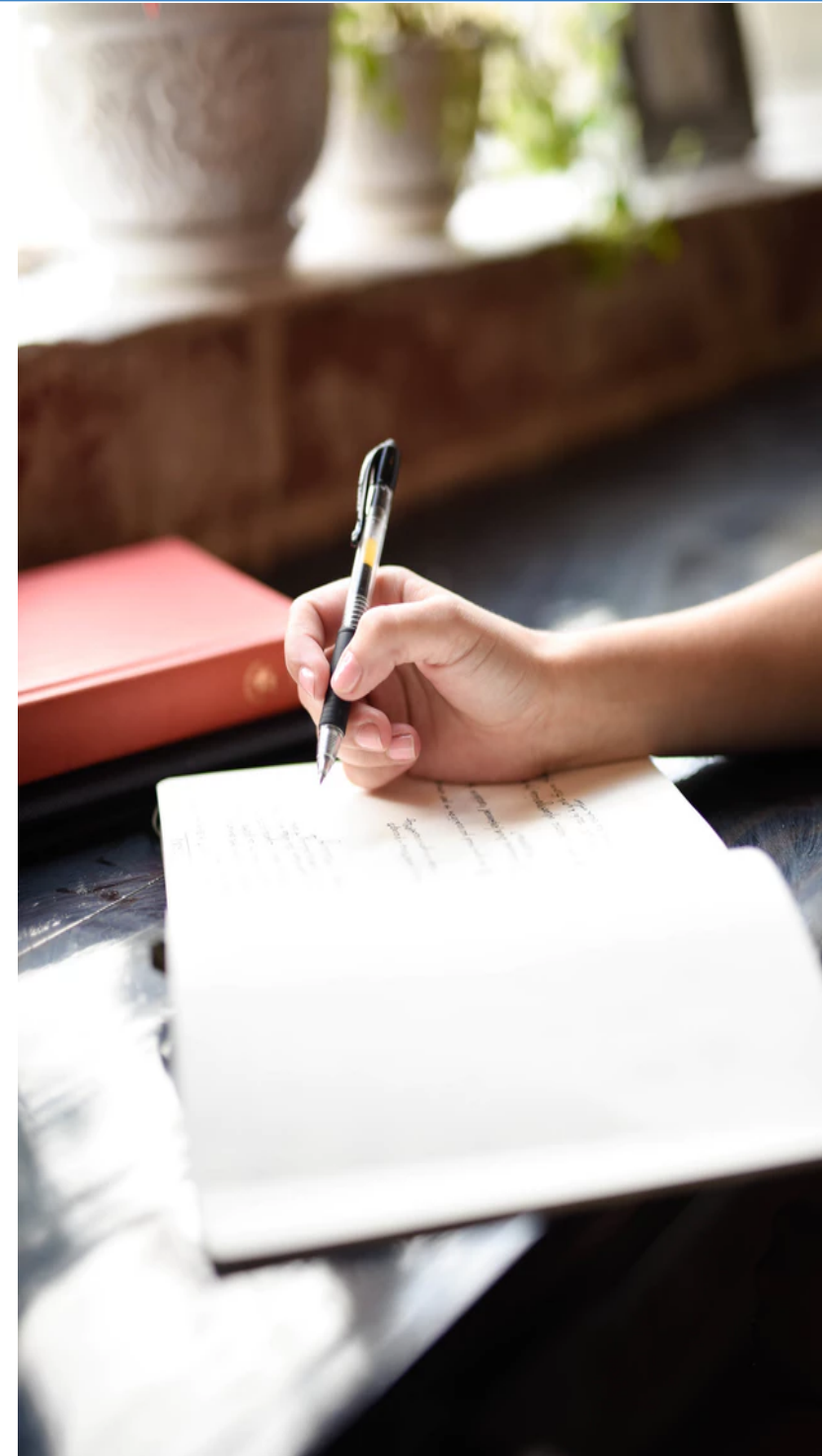
Quality Assurance

A way of preventing mistakes and defects in manufactured products

What is Schematron?

A natural language for making assertions in documents

Schematron is an ISO Standard

Schematron is an ISO standard adopted by hundreds of major projects in the past decade

Schematron History

Schematron was invented by Rick Jelliffe

- Schematron 1.0 (1999), 1.3 (2000), 1.5 (2001)
- ISO Schematron (2006, 2010, 2016)

Why Schematron?

You can express constraints in a way that you cannot perform with other schemas (like XSD, RNG, or DTD).

- XSD, RNG, and DTD schemas define structural aspects and data types of the XML documents

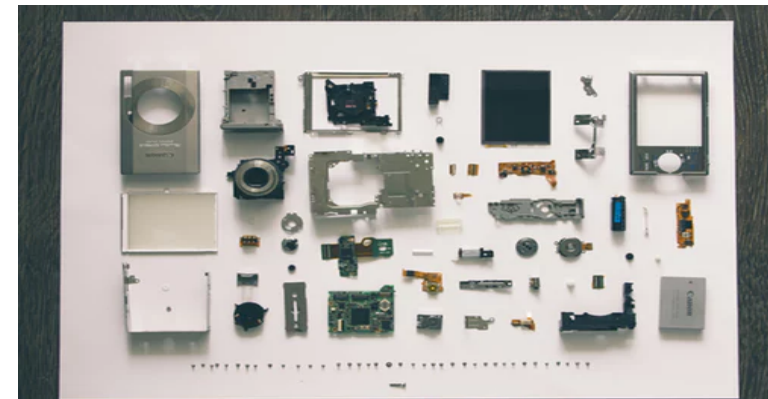- Schematron allows you to create custom rules specific to your project
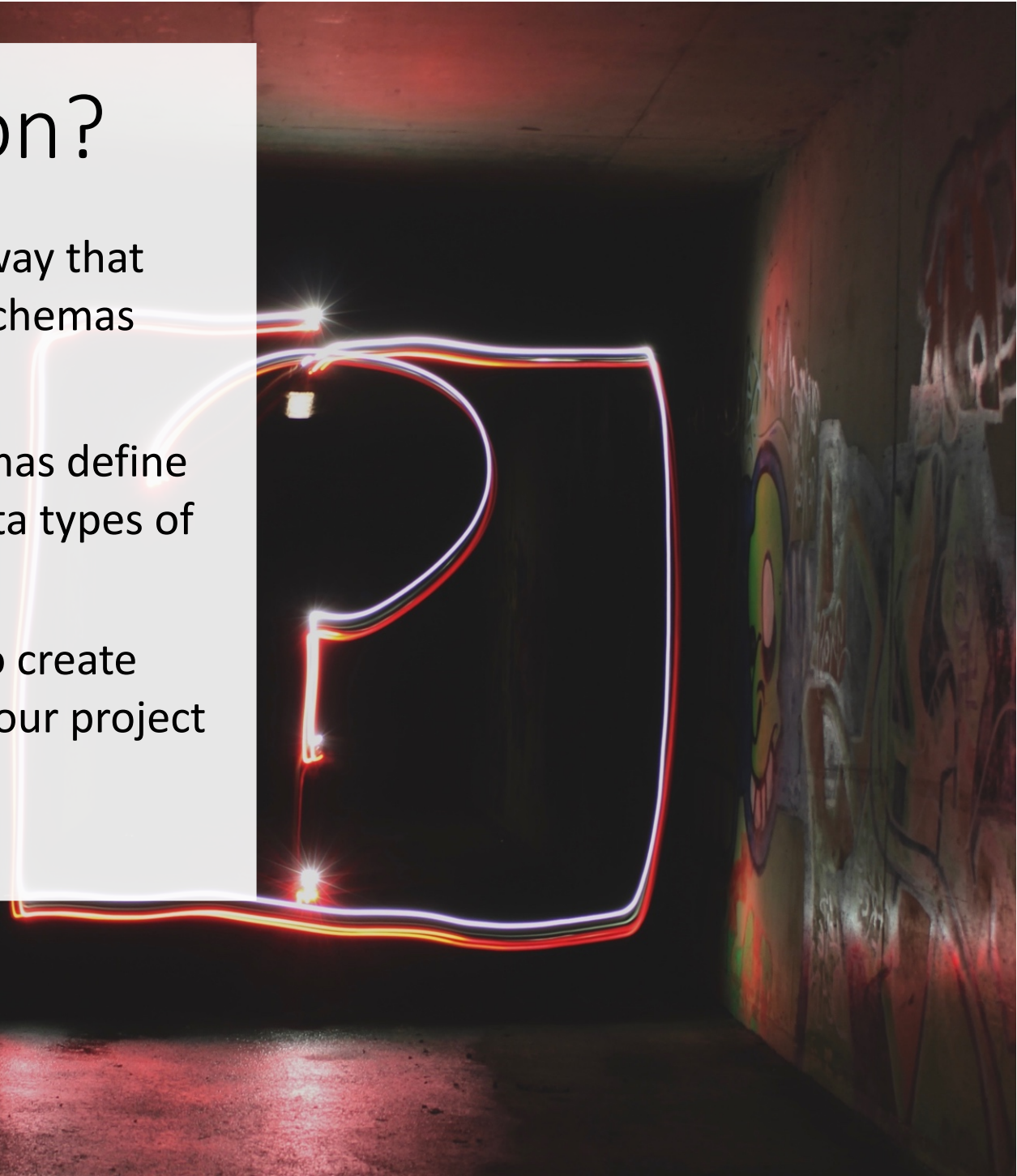
Schematron Usage

- Verify data inter-dependencies

- Check data cardinality

- Perform algorithmic checks

Used in Multiple Domains

- Financial

- Insurance

- Government

- Technical publishing

Schematron is Very Simple

There are only 6 basic elements:

assert

report

rule

pattern

schema

ns

Example of Rules

- **Titles** must be uppercase

- **Short descriptions** should not exceed 50 characters

- Avoid having **empty elements**

- Avoid having a semi-colon at the end of a **list item**

- **Lists** should contain more than one item

Oxygen User Guide

Quality assurance rules used in Oxygen user guide project

Schematron Validation Result

Type of Problems

- Different type of problems

  – Code smell

  – Bug

  – Vulnerability

- Different severity

  – Fatal

  – Error

  – Warn

  – Info

Schematron Messages Severity

Severity level can be set in the value of the @role attribute from an assert or report element (possible values: fatal, error, warn, info)

```
<sch:rule context="related-links/linklist">
  <sch:assert test="title" role="warn">
    The linklist should have a title</sch:assert>
</sch:rule>
```

Validation Result in Editing Framework

Validation Result as HTML

Export validation result in an HTML file

Severity	Description	File	Location
error	The indexterm element should be in a prolog.	C:\samples.dita	Start line 71:22
error	Link text is same as @href attribute value. Please remove.	C:\samples.dita	Start line 32:84
error	Image without a reference.	C:\samples.dita	Start line 37:47
warning	Two consecutive unordered lists. You can probably merge them into one.	C:\samples.dita	Start line 43:33
warning	Two consecutive ordered lists. You can probably merge them into one.	C:\samples.dita	Start line 57:37
warning	Lines (2 - 124,) in codeblocks should not exceed 90 characters.	C:\samples.dita	Start line 135:50
warning	Possible XML Codeblock without @outputclass set to it.	C:\samples.dita	Start line 78:22
warning	Try to avoid inserting two consecutive notes with the same type.	C:\samples.dita	Start line 86:32
error	Empty element. Most DITA elements should not be empty.	C:\samples.dita	Start line 12:11
warning	All sections should have an @id attribute	C:\samples.dita	Start line 93:16
warning	The title is too long (78 chars). It should be less than 75 characters.	C:\samples.dita	Start line 100:18
warning	"menucascade" should contain more than one "uicontrol"	C:\samples.dita	Start line 106:33
warning	The figure should have a title.	C:\samples.dita	Start line 113:17
warning	The section should have a title.	C:\samples.dita	Start line 119:35
error	The text in a section element should be in a paragraph.	C:\samples.dita	Start line 119:35
error	Should use product key instead!	C:\samples.dita	Start line 130:12

Validation Result as SVRL

- As an XML file describing the validation errors, normally in Schematron Validation Reporting Language (SVRL)

```
<svrl:failed-assert test="matches(@href, '^http(s?)://')" role="warn"
                    location="/topic[1]/body[1]/section[1]/p[3]/xref[1]">
    <svrl:text>An external link should start with http(s).</svrl:text>
</svrl:failed-assert>
```

Generate HTML from SVRL

HTML Report – DE GRUYTER

Integrating Schematron in the Development Process

Validate XML with Schematron

- Associate Schematron in the XML file

```
<?xml-model href="books.sch" type="application/xml"
            schematypens="http://purl.oclc.org/dsdl/schematron"?>
```

- Use tool-specific association options

  – Associate Schematron file with a set of XML files (all files with a specific namespace, or from a directory)

  – Associate Schematron with all XML files from a project

Run Schematron Validation

- From an XML editing framework

    - Check the XML files as you type

    - Run the Schematron validation on multiple files

    - Generate a report

Run Schematron Validation

- Using W3C's XProc pipeline language through its "validate-with-schematron" step
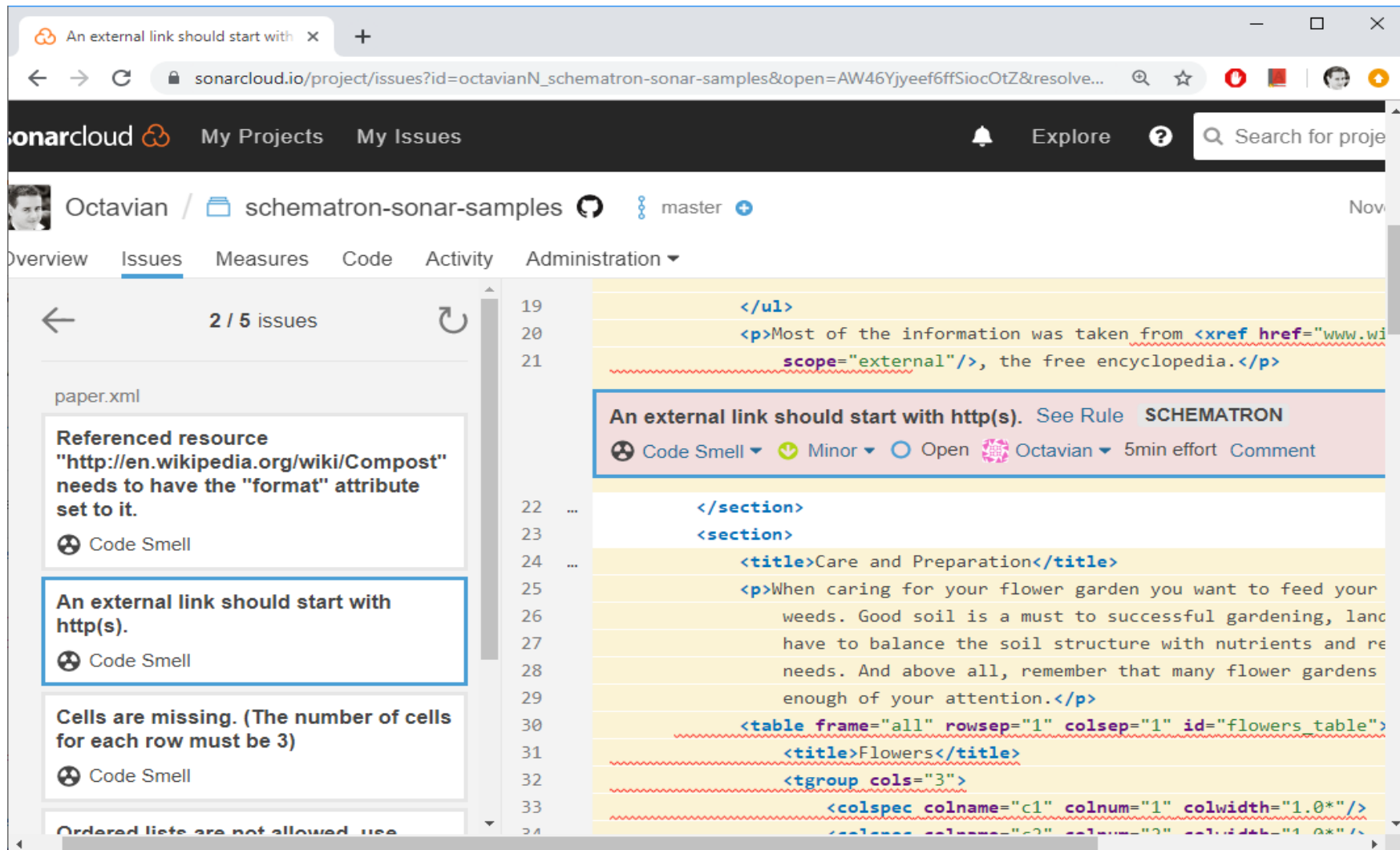
- Using Apache Ant, from bat/shell
  https://github.com/Schematron/ant-schematron

Integration server

- Use an integration server to run Schematron checks (such as Jenkins, Travis)

- Send a report to the user

Sonar Schematron

Conclusion

- Schematron can be used for documentation QA

- Problems with different severity

- Notification about the problem as you type

- Generate a QA report

Questions?

Octavian Nadolu
Software Architect at Syncro Soft

octavian.nadolu@oxygenxml.com
Twitter: @OctavianNadolu
LinkedIn: octaviannadolu