

New in oXygen XML Editor 17 - XML Quick Fixes -



Octavian Nadolu



George Bina

Useful Information

- Yes, we are recording this webinar.
- Please ask questions anytime during the presentation in the Questions panel.
- Use [#oxygenXML](#) for webinar-related tweets.

Overview

- Generic Quick Fix support in oXygen
- Quick fix support for XML validated against XSD
- Quick fix support for XML validated against Relax NG
- Schematron Quick Fix language
- Developing custom quick fixes in Schematron

Quick Fixes

- It is important to have documents without errors
- Not all users know how to fix the errors
- Suggestions to fix errors
- Similar to spell check proposals



Proposals to Solve Errors

- Analyze the errors and propose valid fixes
- Some errors can be fixed automatically
 - Required element missing
 - Attribute not allowed
- Write your own quick fixes



<oxygen/> Quick Fixes Support

 For XSLT documents

 For Ant Build documents

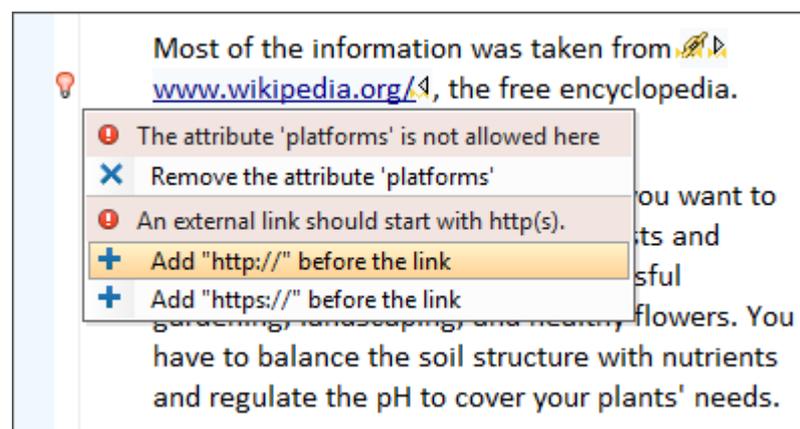
http://www.oxygenxml.com/events/2014/webinar_xslt_quick_fixes.html

 For XML documents

- validation against XML Schema
- validation against Relax NG
- validation against Schematron

Using Quick Fixes

- Placing the caret on error highlight will display a red bulb button
- Show fix proposals:
 - Click the red bulb
 - Invoke the quick fix menu with Alt + 1



Apply Quick Fix

- Just select the proposal
- Available in both Text and Author page
- Reconfigure on document modification
- Undo modification

Horizontal Span	a3
<i>f1</i>	The value of the attribute 'align' is invalid
b1	Set 'align' attribute value to 'center'
c1	Set 'align' attribute value to 'char'
d1	Set 'align' attribute value to 'justify'
	Set 'align' attribute value to 'left'
	Set 'align' attribute value to 'right'

Docbook 5 also supports the **HTML** tables:



Horizontal Span	a3
<i>f1</i>	<i>f2</i>
b1	b2
c1	
d1	Spans Both directions

Docbook 5 also supports the **HTML** tables:

XML Quick Fixes

Solve errors that appear in an XML document

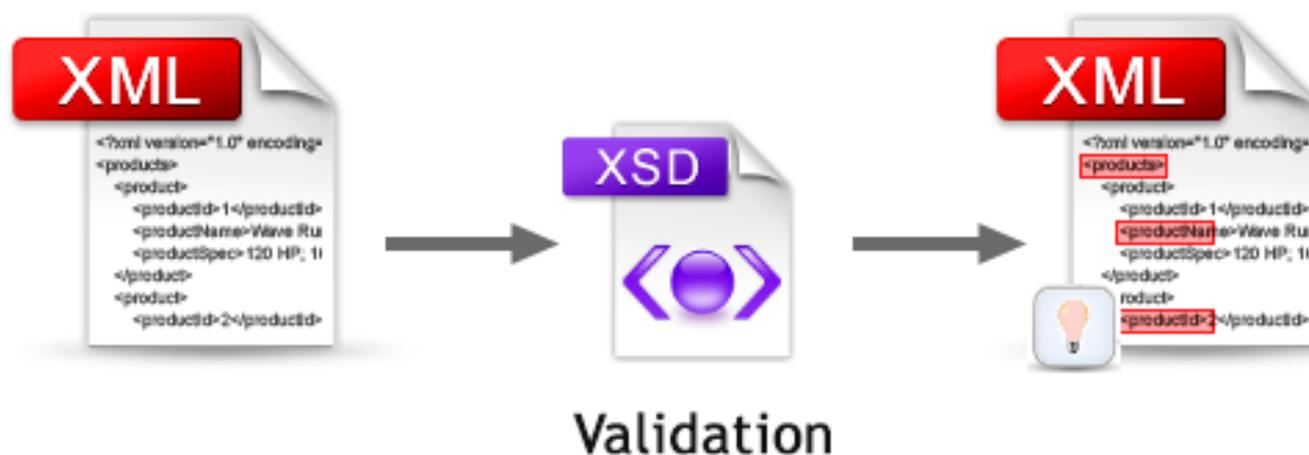
- Fixes generated automatically
 - validation against XML Schema
 - validation against Relax NG
- Fixes provided by the developer
 - validation against Schematron



XML Validated with XSD

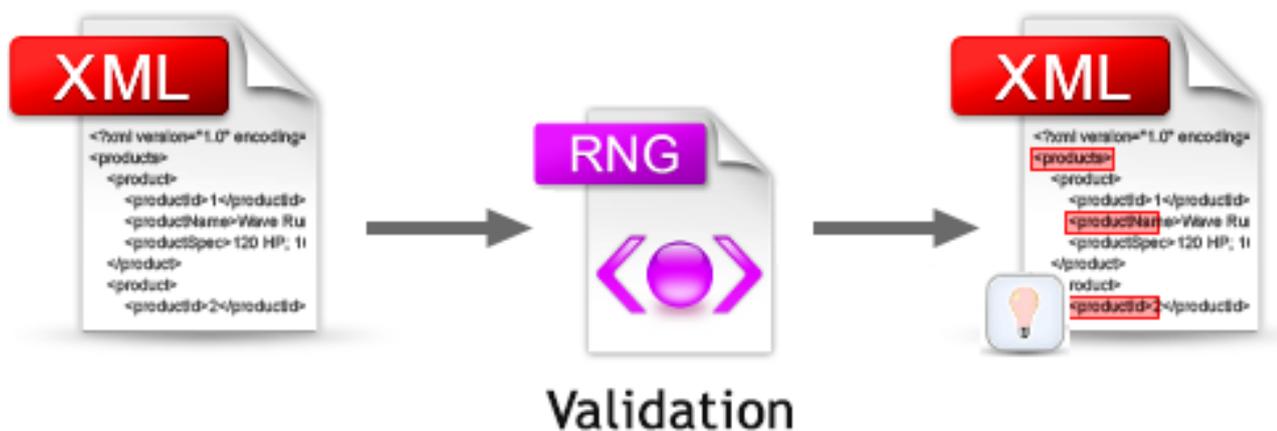
- Quick fixes for common errors
 - A required attribute is missing
 - An element is invalid in the current context
 - References to an invalid ID

...



XML Validated with Relax NG

- Quick fixes for common validation errors
 -  A required attribute is missing
 -  An element is invalid in the current context
 -  References to an invalid ID
- Both RNG and RNC schemas are supported

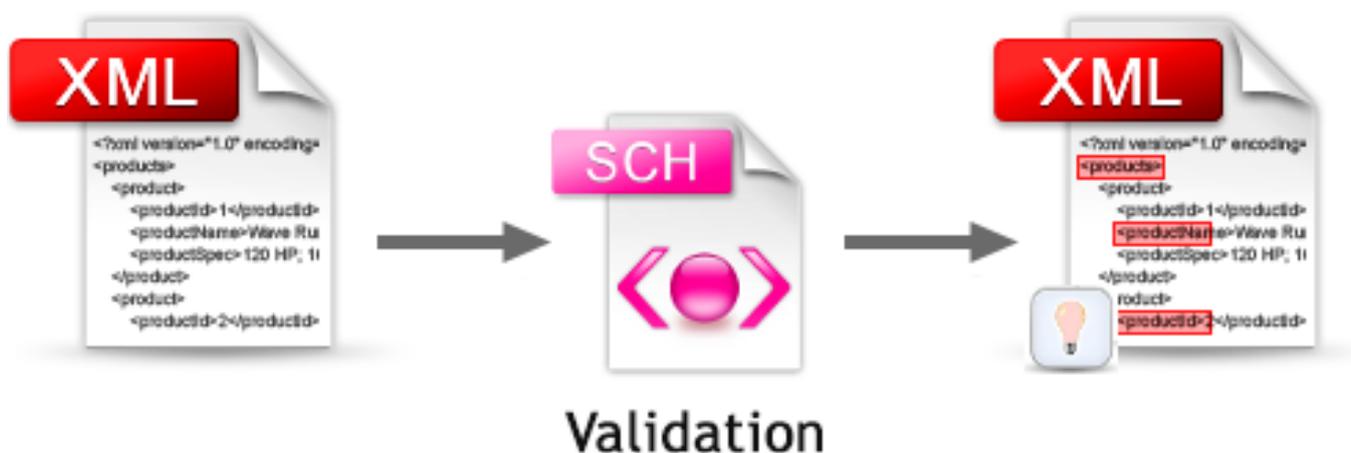


Benefits

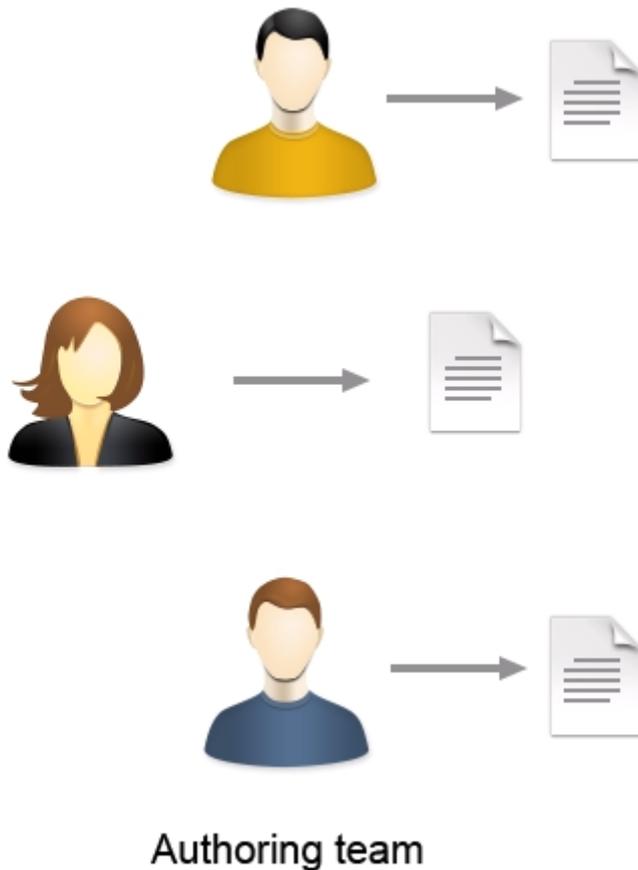
- Fixes generated automatically for common errors
- A better understanding of the problem
- Solve errors with just one click

XML Validated with Schematron

- Define custom errors in Schematron
- Create custom fixes with SQF

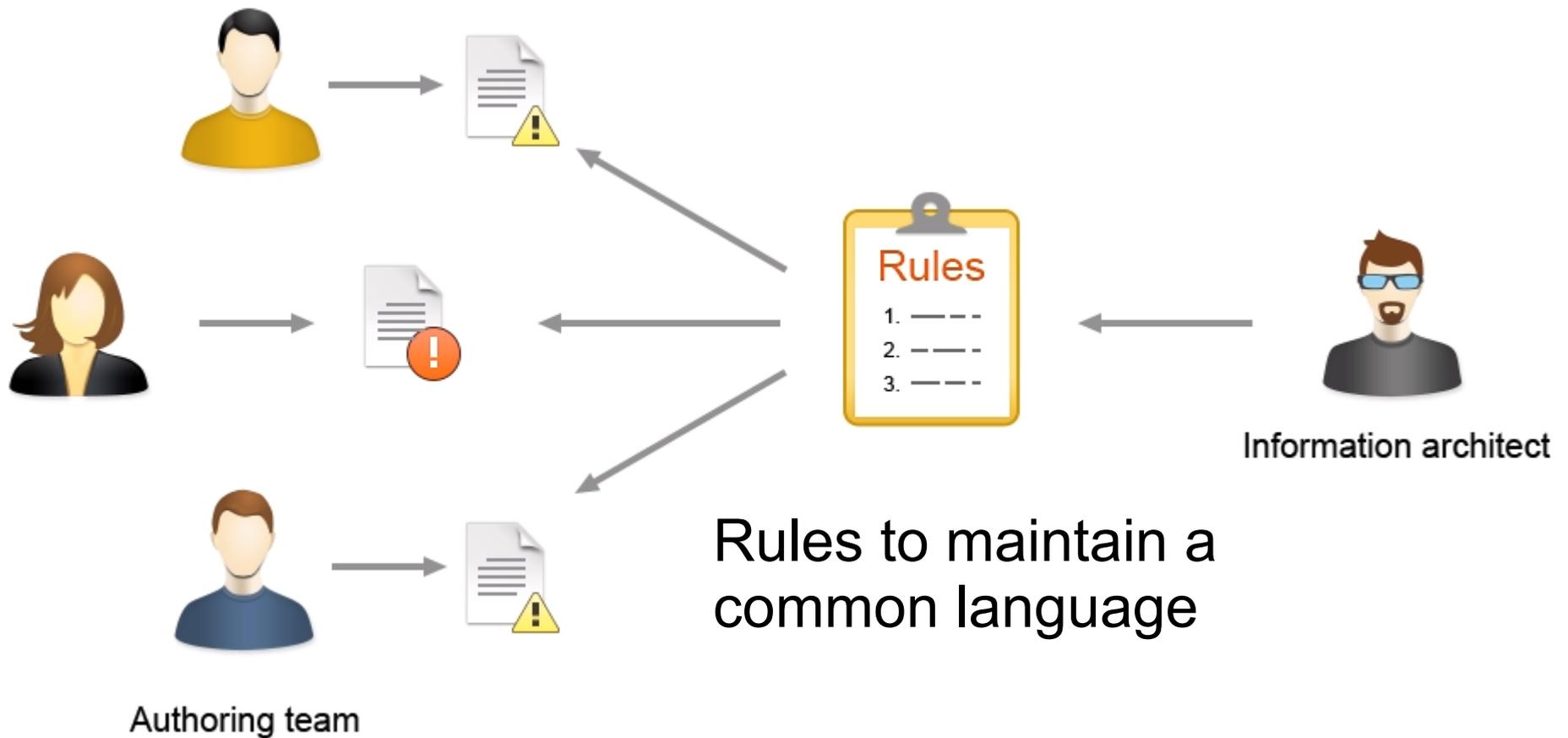


Example



A group of technical writers works on a documentation project

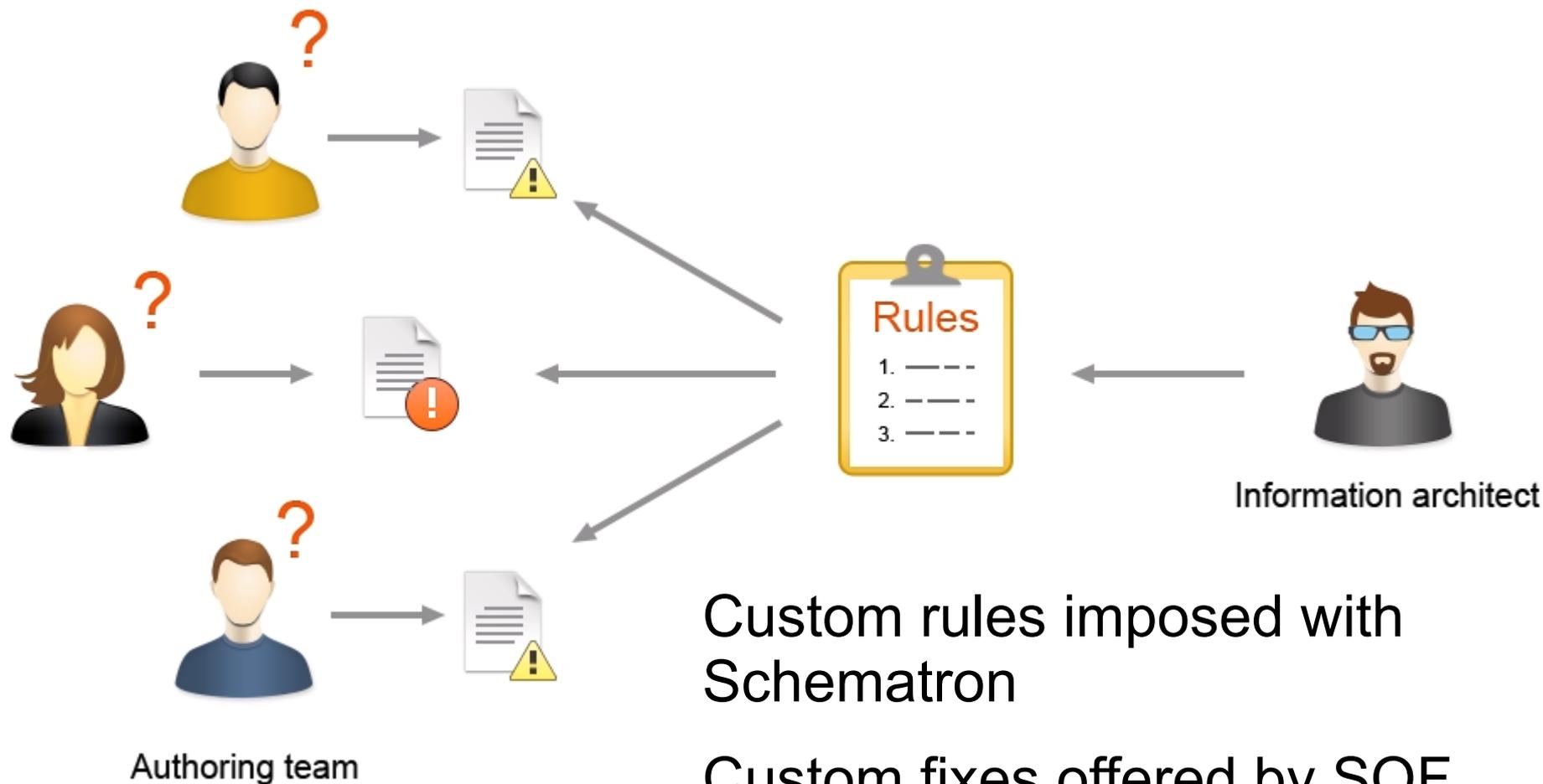
Rules



Business Rules Examples

- ⚠ Bold element is not allowed in title
- ❗ An external link requires the @format attribute
- ⚠ Semicolon is not allowed after list item
- ❗ Each row must have the same number of cells
- ❗ Ordered lists are not allowed

Schematron Rules



Custom rules imposed with Schematron

Custom fixes offered by SQF

Custom Fixes Examples

- ⚠ Bold element is not allowed in title
 - ✖ Remove bold styling
- ❗ An external link requires the @format attribute
 - ✚ Add @format attribute
- ⚠ Semicolon is not allowed after list item
 - ✖ Remove semicolon
 - 🔗 Replace semicolon with full stop
- ❗ Each row must have the same number of cells
 - ✚ Add missing cells
- ❗ Ordered lists are not allowed
 - 🔄 Convert ordered list to unordered one

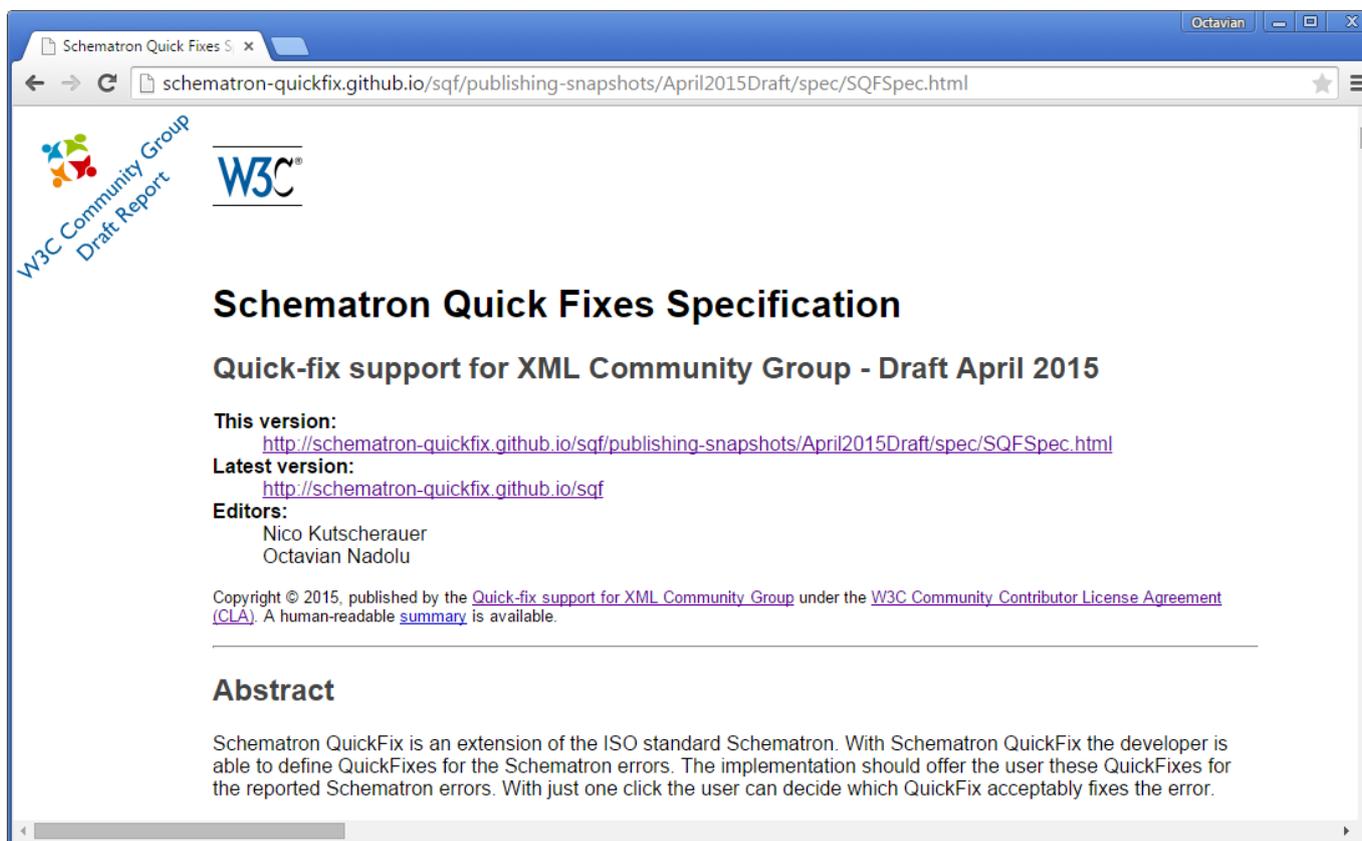
Schematron Quick Fixes (SQF)

- Extends the Schematron language
- SQF initiated by Nico Kutscherauer/data2type
- Open source project

The letters "SQF" in a large, blue, outlined font.

www.schematron-quickfix.com
github.com/schematron-quickfix/sqf

Schematron Quick Fixes Spec



www.w3.org/community/quickfix



schematron-quickfix.github.io/sqf

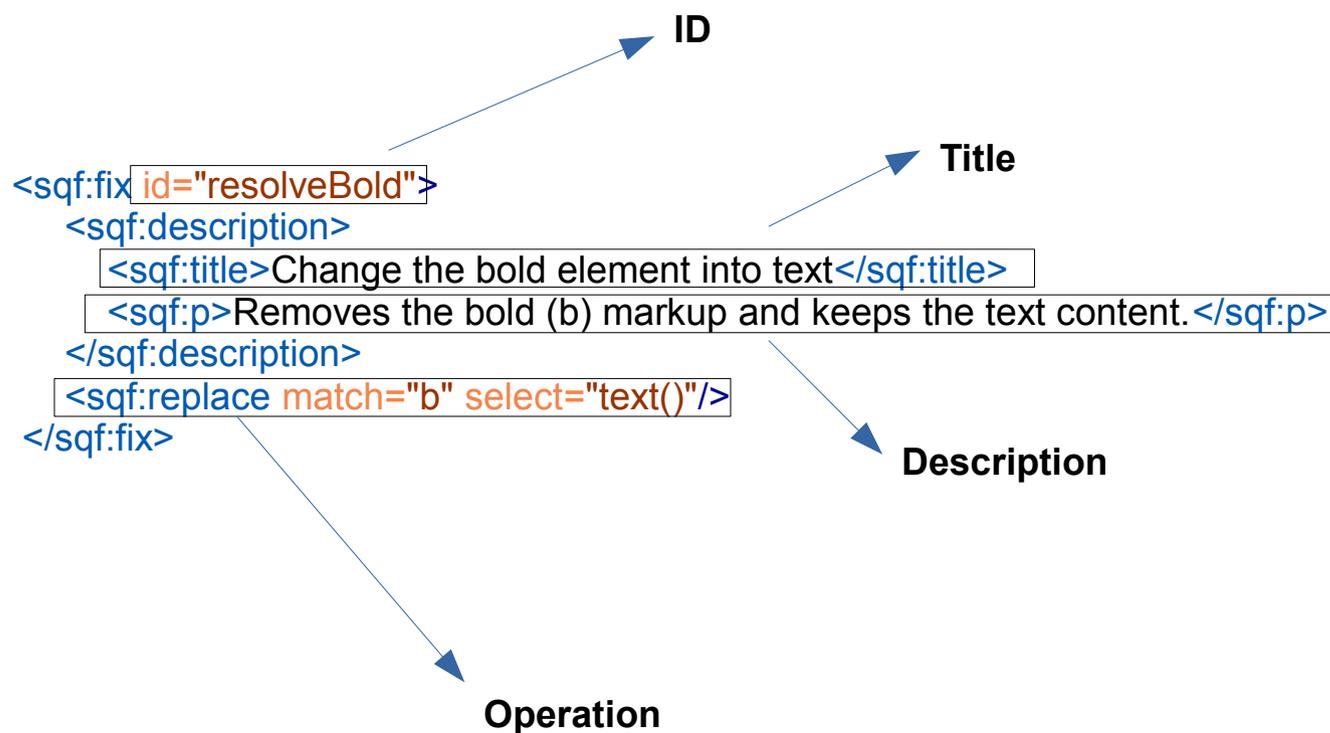
SQF in Schematron

- Associated with **assert** and **report** elements

```
<sch:rule context="title">  
  <sch:report test="exists(b)" sqf:fix="resolveBold">  
    Bold element is not allowed in title.</sch:report>  
  .....  
</rule>
```

```
<sch:rule context="xref">  
  <sch:assert test="matches(@href, '^http(s?)://')" sqf:fix="addHttp addHttps">  
    An external link should start with http(s).</sch:assert>  
  .....  
</sch:rule>
```

Schematron Quick Fix



Schematron Quick Fix

- **ID** - an unique ID, used to refer the fix
- **Title** - represents the name of the quick fix
- **Description** - details about the fix
- **Operations** - 4 basic operations
 - **Add** – adds one or more nodes
 - **Delete** – deletes the matched nodes
 - **Replace** – replace a node by another one
 - **StringReplace** – replace sub-strings of text nodes

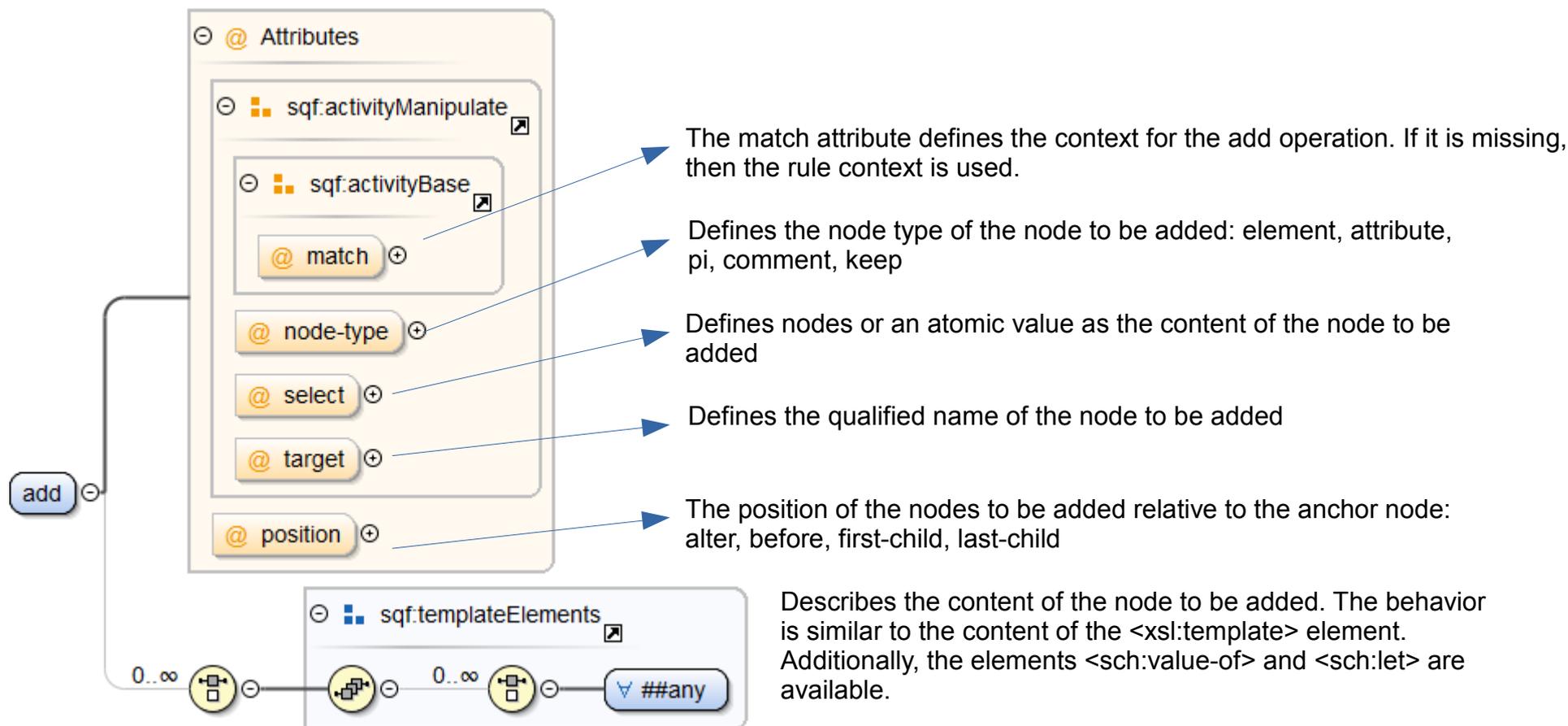
SQF “add” operation

- `<sqf:add>` element allows you to add one or more nodes to the XML instance

```
<sch:rule context="xref">  
  <sch:assert test="@format" sqf:fix="addFormat">  
    An external link requires the @format attribute.</sch:assert>
```

```
  <sqf:fix id="addFormat">  
    <sqf:description>  
      <sqf:title>Add @format attribute</sqf:title>  
    </sqf:description>  
    <sqf:add node-type="attribute" target="format"/>  
  </sqf:fix>  
</sch:rule>
```

SQF “add” operation



SQF “replace” operation

- `<sqf:replace>` element specify the nodes to be replaced and the replacing content

```
<sch:rule context="ol">
```

```
  <sch:assert test="false()" sqf:fix="convertOLinUL"> Ordered lists are not  
  allowed, use unordered lists instead.</sch:assert>
```

```
  <sqf:fix id="convertOLinUL">
```

```
    <sqf:description>
```

```
      <sqf:title>Convert ordered list to unordered list</sqf:title>
```

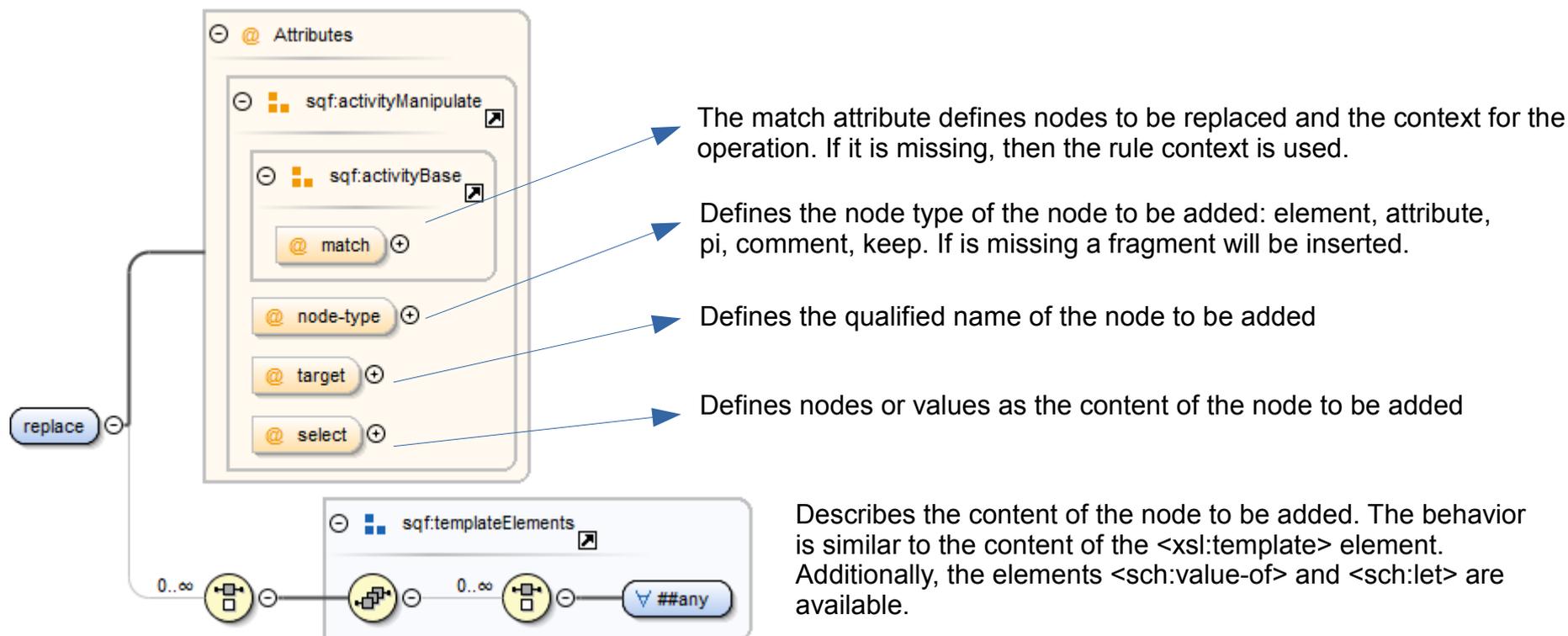
```
    </sqf:description>
```

```
    <sqf:replace target="ul" node-type="element" select="child::node()"/>
```

```
  </sqf:fix>
```

```
</sch:rule>
```

SQF “replace” operation



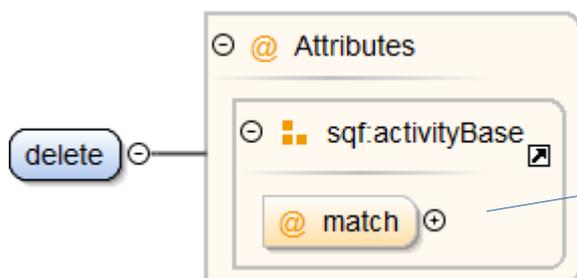
SQF “delete” operation

- `<sqf:delete>` element specify the nodes for the deletion

```
<sch:rule context="title">  
  <sch:report test="comment()" sqf:fix="removeComment">  
    Comments are not allowed in the title</sch:report>
```

```
<sqf:fix id="removeComment">  
  <sqf:description>  
    <sqf:title>Delete comments</sqf:title>  
  </sqf:description>  
  <sqf:delete match="comment()"/>  
</sqf:fix>  
</sch:rule>
```

SQF “delete” operation



The match attribute defines nodes to be deleted.

The default value is “.”. If the match attribute is not specified the context node will be deleted.

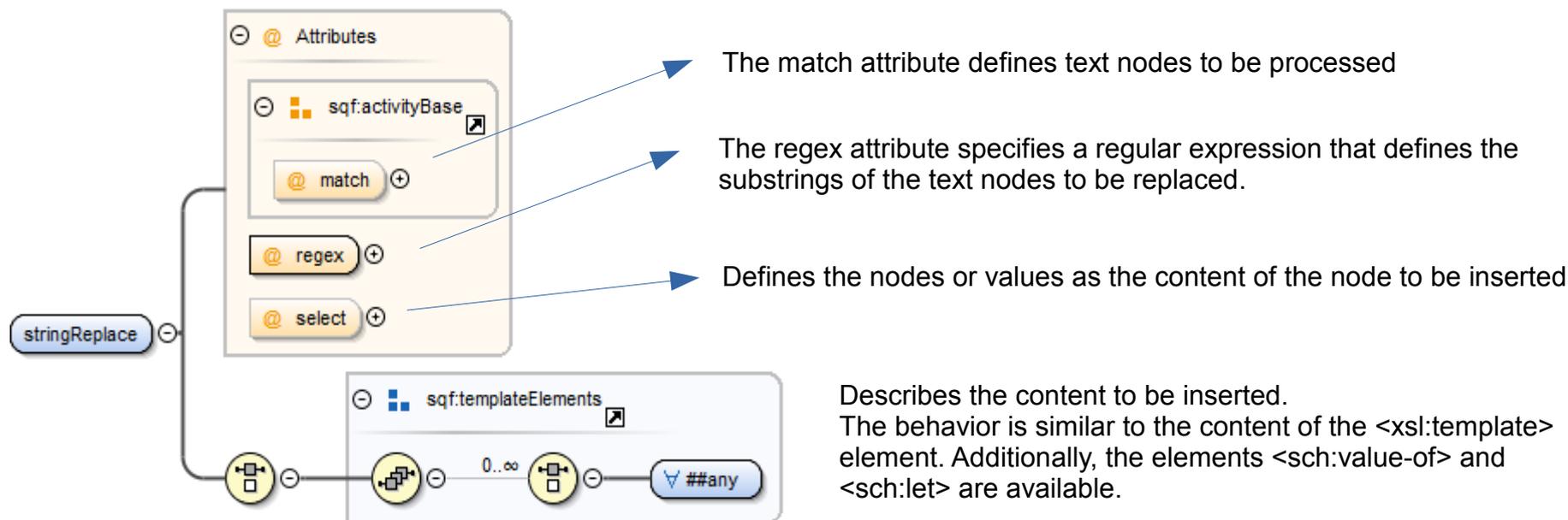
SQF “stringReplace” operation

- `<sqf:stringReplace>` element defines the text or nodes which will replace the matched substrings

```
<sch:rule context="li">
  <sch:report test="matches(text()[last()], ';$')" sqf:fix="replaceSemicolon"
    role="warn"> Semicolon is not allowed after list item.</sch:report>

  <sqf:fix id="replaceSemicolon">
    <sqf:description>
      <sqf:title>Replace semicolon with full stop</sqf:title>
    </sqf:description>
    <sqf:stringReplace match="text()" regex=";$">.</sqf:stringReplace>
  </sqf:fix>
</sch:rule>
```

SQF “stringReplace” operation



Formatting Inserted Content

- By default, oXygen performs line indenting
- Use the **@xml:space** attribute and set its value to **preserve** for full control of the generated fragment
- Whitespace is generated similar to XSLT processing:
 - No whitespace for element only content
 - Whitespace is significant in mixed content

Formatting Inserted Content

- `<xsl:text>` - format content and keep automatic indentation

```
<sqf:add position="last-child">
  <row>
    <entry>First column</entry>
    <entry>Second column</entry>
  </row>
</sqf:add>
```

```
<sqf:add position="last-child">
  <row><xsl:text>
</xsl:text>
  <entry>First column</entry><xsl:text>
</xsl:text>
  <entry>Second column</entry><xsl:text>
</xsl:text>
</row><xsl:text>
</xsl:text>
</sqf:add>
```

- `@xml:space` – preserve space

```
<sqf:add node-type="element" target="codeblock" xml:space="preserve">
/* a long sample program */
  Do forever
  Say "Hello, World"
End</sqf:add>
```

The Use-When Condition

- Used to make available a *quick fix* or a specific *operation* only if certain conditions are met
- The condition of the *use-when* attribute is an XPath expression

```
<sch:rule context="li">  
  <sch:report test="matches(text()[last()], ';$')" sqf:fix="replaceSemicolon">  
    Semicolon is not allowed after list item.</sch:report>
```

```
<sqf:fix id="replaceSemicolon" use-when="position() = last()">  
  <sqf:description>  
    <sqf:title>Replace semicolon with full stop</sqf:title>  
  </sqf:description>  
  <sqf:stringReplace regex=";$" match="text()">.</sqf:stringReplace>  
</sqf:fix>  
</sch:rule>
```

Additional SQF Elements

<sqf:call-fix> – calls another fix within a fix

<sqf:fixes> – global element that contains fixes

<sqf:group> – defines a group of fixes that can be referred

<sqf:keep> – used to copy the selected nodes

<sqf:user-entry>* – defines a value that must be set manually by the user

*not supported in <oxygen/> 17

 [schematron-quickfix.github.io/sqf](https://github.com/schematron-quickfix/sqf)

SQF Framework

- Support to edit SQF files
- Support to edit SQF embedded in Schematron
- Validation as you type
- Content completion
 - SQF elements and attributes
 - XPath functions
 - XSLT functions and variables
 - Quick Fix ids
- Search and Refactoring



SQF Quick Fixes benefits

- Create custom quick fixes for errors
- Use the power of Schematron and XSLT
- Custom solution for different type of documents
- Create refactoring actions using SQF

SQF for any XML

- Schematron can be applied on any XML
- Fixes can be provided for any XML
- SQF used in development
 - XSLT
 - XML Schema
 - Relax NG
 - ...

Projects using SQF

- <oxygen/> DITA framework
- <oxygen/> User Manual

<https://github.com/oxygenxml/userguide>

- DIM project

<https://github.com/oxygenxml/dim>

- TEI

<http://wiki.tei-c.org/index.php/Category:Schematron>

Related SQF Projects

- Escali Schematron engine

http://schematron-quickfix.com/escali_xsm.html

- Escali Schematron command line tool
- Oxygen plugin for invoking Escali Schematron

Thank you!

Questions?

<oxygen/> XML Editor
<http://www.oxygenxml.com>
octavian_nadolu@oxygenxml.com
[@OctavianNadolu](#)

Upcoming Events



June 3

[Refactoring XML Documents](#)



June 5

[oXygen Users Meetup London](#)



June 6 - 7

[XML London 2015](#)



June 10

[DITA Support](#)



June 17

[XML Authoring on the Web](#)

Recordings of the previous oXygen 17 webinars:

- [Overview of oXygen XML Editor 17](#)
- [UI Improvements](#)