

Schematron QuickFix

Nico Kutscherauer

contact@schematron-quickfix.com
[@nkutsche](#)

Octavian Nadolu

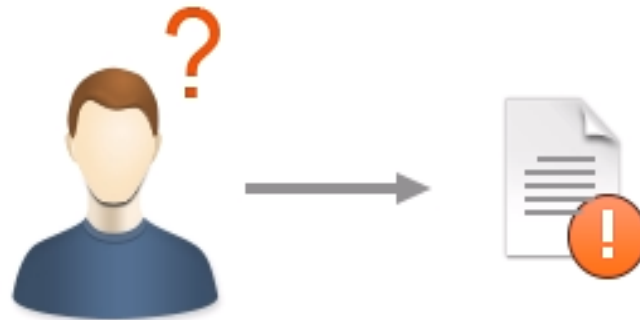
octavian_nadolu@oxygenxml.com
[@OctavianNadolu](#)

SQF

The logo for Oxygen XML Editor, featuring the word "oxygen" in blue with a large orange "X" and the text "XML Editor" below it.

Error Fixes

- Error fixing has always been a challenge
- Solutions offered by IDEs



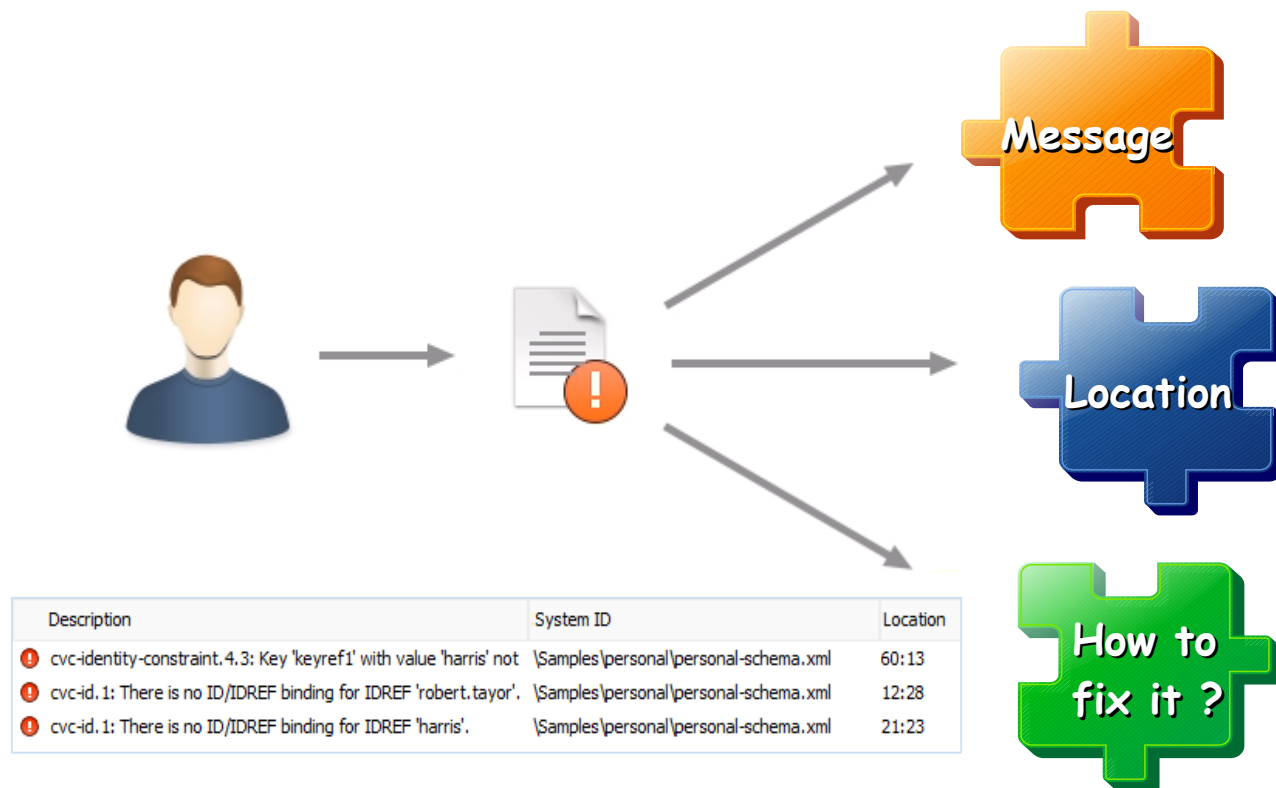
Fix Proposals

- Using fix proposals to solve errors:
 - Better understanding of the problem
 - Less (no) mistakes
 - Saves time (money)



XML Validation Errors

- From the user perspective the validation error is composed by three things



Fixing XML Validation Errors



- Predefined Errors - defined in the validation engine
 - Fixes generated from the validation engine
 - Fixes based on the message (error code) and location
- Custom Errors - defined by the user
 - Difficult to generate fixes based on message and location
 - A language to create fixes is more appropriate

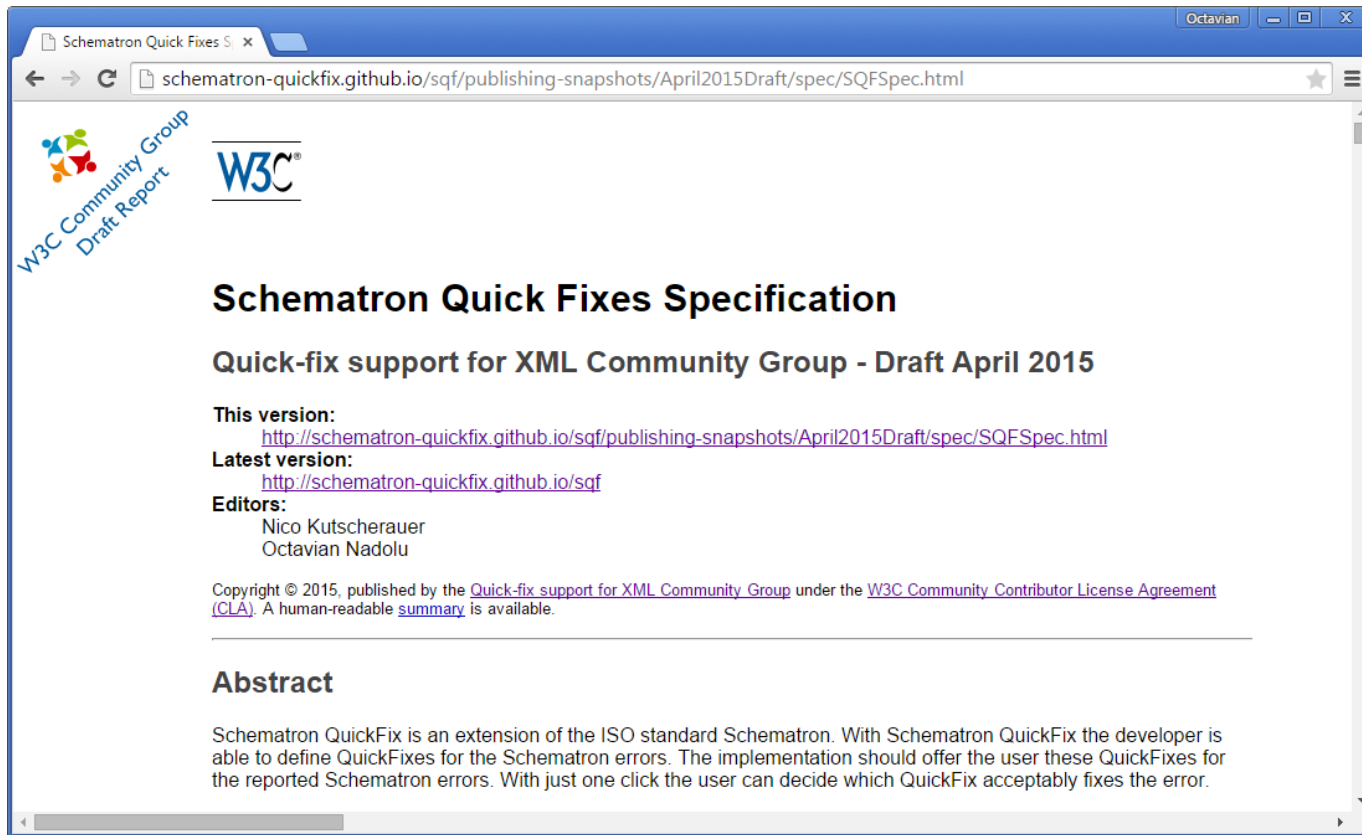
Schematron Fix Proposals

- User defined fixes for Schematron errors
- Schematron QuickFix (SQF) language
 - Extends the Schematron language
 - SQF initiated by Nico Kutscherauer

The letters 'SQF' are rendered in a large, blue, outlined font with a slight 3D effect.

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

Schematron Quick Fixes Spec



www.w3.org/community/quickfix



schematron-quickfix.github.io/sqf

SQF Extension of the Schematron

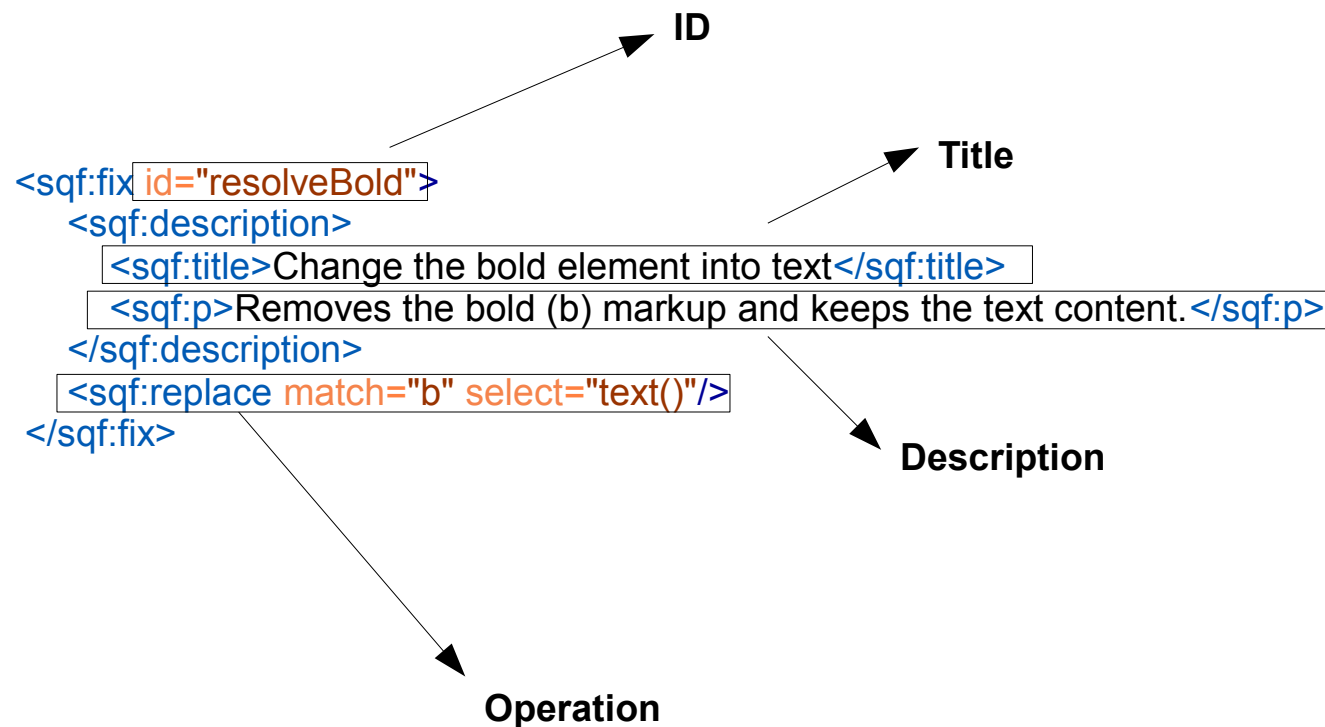
- Associated with assert and report elements
- Added as Schematron annotations

```
<rule context="html">  
  <report test="//comment()" sqf:fix="removeComments">  
    Comments are not allowed in document.</report>
```

```
<sqf:fix id="removeComments" role="delete">  
  <sqf:description>  
    <sqf:title>Remove all comments</sqf:title>  
    <sqf:p>Remove all comment nodes from the current document</sqf:p>  
  </sqf:description>  
  <sqf:delete match="//comment()"/>  
</sqf:fix>
```

```
</rule>
```


Schematron QuickFix (SQF)



SQF Benefits

- ✓ Create custom quick fixes for errors

SQF Benefits

- Create custom quick fixes for errors
- ✓ Use the power of Schematron and XSLT

SQF Benefits

- Create custom quick fixes for errors
- Use the power of Schematron and XSLT
- ✓ Create refactoring actions using SQF

SQF Benefits

- Create custom quick fixes for errors
- Use the power of Schematron and XSLT
- Create refactoring actions using SQF
- ✓ Fix problems in external documents

SQF Benefits

- Create custom quick fixes for errors
- Use the power of Schematron and XSLT
- Create refactoring actions using SQF
- Fix problems in external documents
- ✓ Fixes for any XML documents

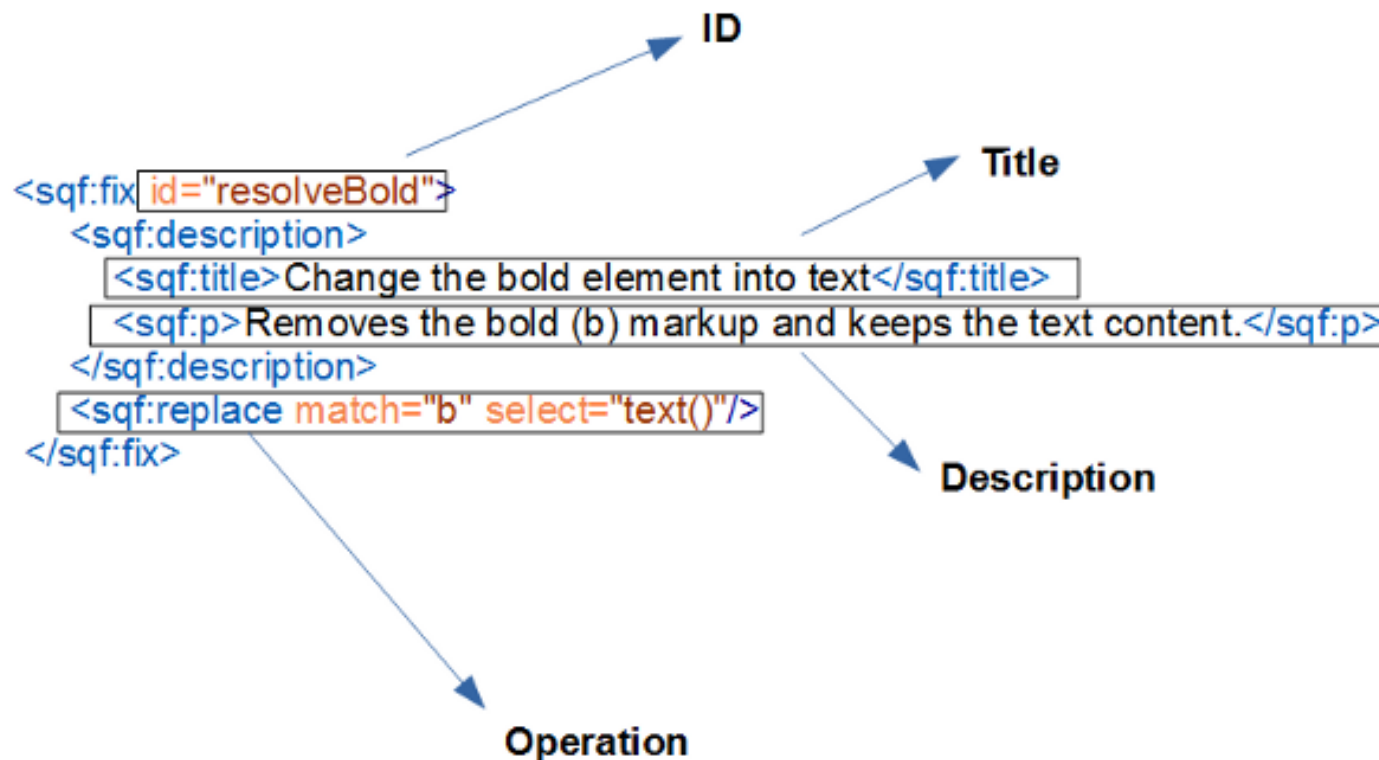
Language Overview

- Small language
 - Keep it easy, but extensible
 - No knowledge of XSLT is needed for simple cases
 - Integrate well-known standards for more complex things
- Own Namespace with prefix sqf:
<http://www.schematron-quickfix.com/validator/process>

Standard	Elements
XSLT 2.0	49
Schematron	21
SQF	15

QuickFix structure

- Reference
- title & description
- Activity Elements



Example 1

- Replace a node

```
<sch:rule context="b">
  <sch:report test="ancestor::b"
    sqf:fix="italic resolve">
    Bold in bold is not allowed.</sch:report>
  <sqf:fix id="italic">
    <sqf:description>
      <sqf:title>Change it to italic.</sqf:title>
    </sqf:description>
    <sqf:replace target="i" node-type="element" select="node()"/>
  </sqf:fix>
  <sqf:fix id="resolve">
    <!--...-->
    <sqf:replace select="node()"/>
  </sqf:fix>
</sch:rule>
```

Example 2

- Move a node – delete and add

```
<sch:rule context="title">
  <sch:report test="preceding-sibling::abstract" sqf:fix="abstract">
    The abstract element shouldn't be placed before the <sch:name/> element.
  </sch:report>
  <sqf:group id="abstract">
    <sqf:fix id="abstract.moveAbstract">
      <sqf:description>
        <sqf:title>Move the abstract element after the title</sqf:title>
      </sqf:description>
      <sqf:delete match="preceding-sibling::abstract"/>
      <sqf:add match="." select="preceding-sibling::abstract" position="after"/>
    </sqf:fix>
    <sqf:fix id="abstract.moveTitle"><!--...--></sqf:fix>
    <sqf:fix id="abstract.deleteAbstract"><!--...--></sqf:fix>
    <sqf:fix id="abstract.deleteTitle"><!--...--></sqf:fix>
  </sqf:group>
</sch:rule>
```

Example 3

- Replace sub strings

```
<sch:rule context="text()">
  <sch:report test="matches(., '____')"
    sqf:fix="form">
    More than three underscores in a row shouldn't be used.</sch:report>
  <sqf:fix id="form">
    <sqf:description>
      <sqf:title>Replace the missused characters by a form element.</sqf:title>
    </sqf:description>
    <sqf:stringReplace regex="____+">
      <form/>
    </sqf:stringReplace>
  </sqf:fix>
</sch:rule>
```

Example 4

- User Entry – parameter of the QuickFix

```
<sch:rule context="title">
  <sch:assert test="normalize-space(.) != ' ' " sqf:fix="title"
    >A title shouldn't be empty.</sch:assert>
  <sqf:fix id="title">
    <sqf:description>
      <sqf:title>Set a title</sqf:title>
      <sqf:p>This QuickFix will set a title by using a User Entry.</sqf:p>
    </sqf:description>
    <sqf:user-entry name="title" type="xs:string">
      <sqf:description>
        <sqf:title>Please enter the new title.</sqf:title>
      </sqf:description>
    </sqf:user-entry>
    <sqf:replace target="title" node-type="element"
      select="$title" />
    </sqf:fix>
  </sch:rule>
```

Example 5

- Conditions – provide a QuickFix only if it makes sense

```
<sch:rule context="html:title">
  <sch:assert test="string-length(normalize-space(.)) gt 20 "
    sqf:fix="title">
    A title shouldn't be empty.</sch:assert>
    <sqf:fix id="title" use-when="//html:h1[1][string-length(.) le 20]">
      <sqf:description>
        <sqf:title>
          Set the title to "<sch:value-of select="//html:h1[1]"/>".
        </sqf:title>
      </sqf:description>
      <sqf:replace target="title" node-type="element">
        <sch:value-of select="//html:h1[1]"/>
      </sqf:replace>
    </sqf:fix>
  </sch:rule>
```

Example 6

- Generic QuickFixes – in the future

```
<sch:rule context="html:title">
  <sch:assert test="string-length(normalize-space(.)) gt 20 "
    sqf:fix="title">
    A title shouldn't be empty.</sch:assert>
    <sqf:fix id="title" use-for-each="//html:h1[string-length(.) le 20]">
      <sqf:description>
        <sqf:title>
          Set the title to "<sch:value-of select="$sqf:current"/>".
        </sqf:title>
      </sqf:description>
      <sqf:replace target="title" node-type="element">
        <sch:value-of select="$sqf:current"/>
      </sqf:replace>
    </sqf:fix>
  </sch:rule>
```

Example 7

- Generic QuickFixes – in the future

```
<sqf:fix id="title">
  <sqf:call-fix ref="createElementRowAsFirstChild">
    <sqf:with-param name="match" select="."/>
    <sqf:with-param name="el" select="col"/>
    <sqf:with-param name="count" select="count(max(../tr/count(td|th)))/>
  </sqf:call-fix>
</sqf:fix>
<sqf:fix id="createElementRowAsFirstChild">
  <sqf:param name="match" type="node()*/>
  <sqf:param name="el" type="xs:QName"/>
  <sqf:param name="count" type="xs:integer"/>
  <sqf:description>
    <sqf:title>Create a row of <sch:value-of select="$count"/>
      <sch:name path="$el"/> elements as a first child of the
      <sch:name path="$match"/> element(s).</sqf:title>
  </sqf:description>
  <sqf:add match="$match" position="first-child">
    <xsl:for-each select="1 to $count">
      <xsl:element name="{ $el }"/>
    </xsl:for-each>
  </sqf:add>
</sqf:fix>
```

SQF Implementations

- <oXygen/> XML Editor validation engine

<http://www.oxygenxml.com>

- Escali Schematron engine

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

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

Projects using SQF



Thieme - publishing company uses a custom framework to create and edit XML documents



parsX - it is a product developed by **pagina GmbH** used to facilitate EPUB production



ART-DECOR is open source tool suite that supports SDOs active in the healthcare space
Sample SQF embedded in XSD



ATX custom framework – used by a major automotive producer

Projects using SQF

- [Dynamic Information Model \(DIM\)](#) - an implementation of an intelligent style guide
- [Schematron for TEI](#) - collection of Schematron and SQF resources for TEI
- [<oXygen/> DITA framework](#) - built-in framework in [<oXygen/> XML Editor](#) for DITA documents
- [<oXygen/> XML userguide](#) - the public version of the [<oXygen/> User Manual](#)

Conclusions and Future Plans

- SQF is a simple and useful language
- Helps users to solve the problems in less time and with fewer (no) errors
- Update the SQF specification
- Publish the second draft of the Schematron QuickFix specification

Thank you!

Questions?

contact@schematron-quickfix.com
[@nkutsche](https://twitter.com/nkutsche)

octavian_nadolu@oxygenxml.com
[@OctavianNadolu](https://twitter.com/OctavianNadolu)