Enforcing Business Rules
Automatic Fixes

Understanding and Developing Schematron Quick Fixes

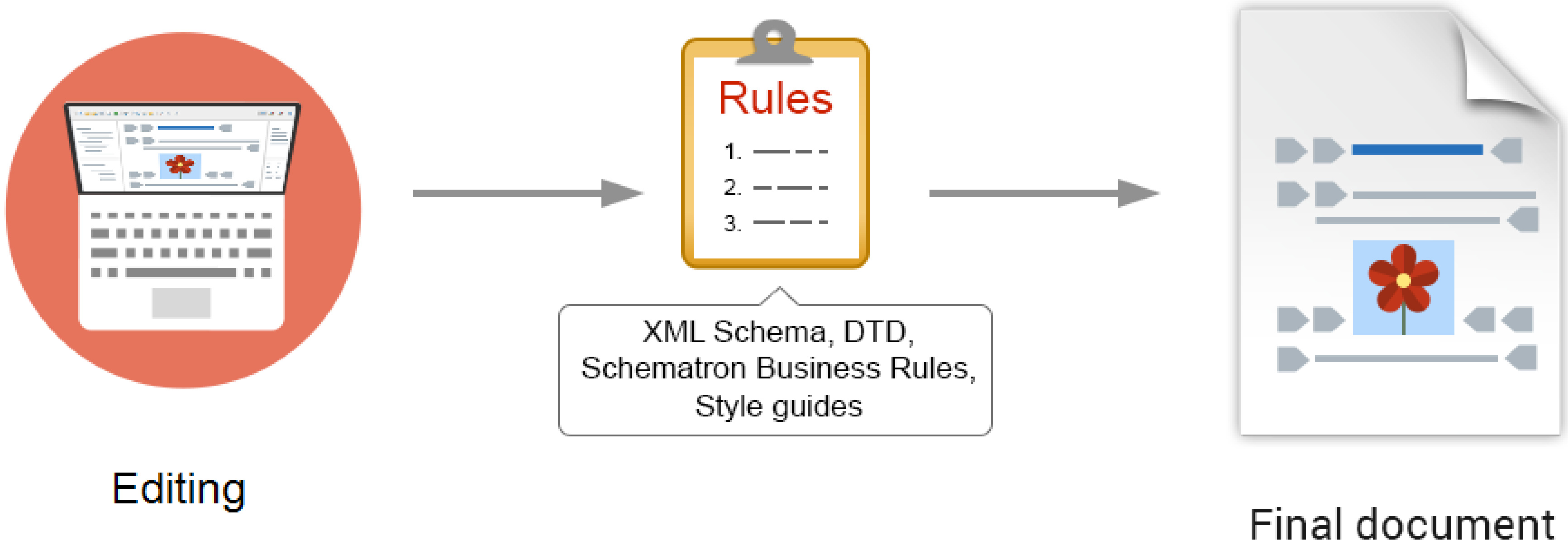
Octavian Nadolu

octavian_nadolu@oxygenxml.com
@OctavianNadolu

Overview

- Using Schematron to automatically check business rules
- Fix proposals for business rules using Schematron Quick Fix language (SQF)
- How SQF integrates with Schematron
- Introduction to SQF through examples
- SQF use-cases (error correction, refactoring, automatic tagging, etc.)

Enforcing Editing Rules

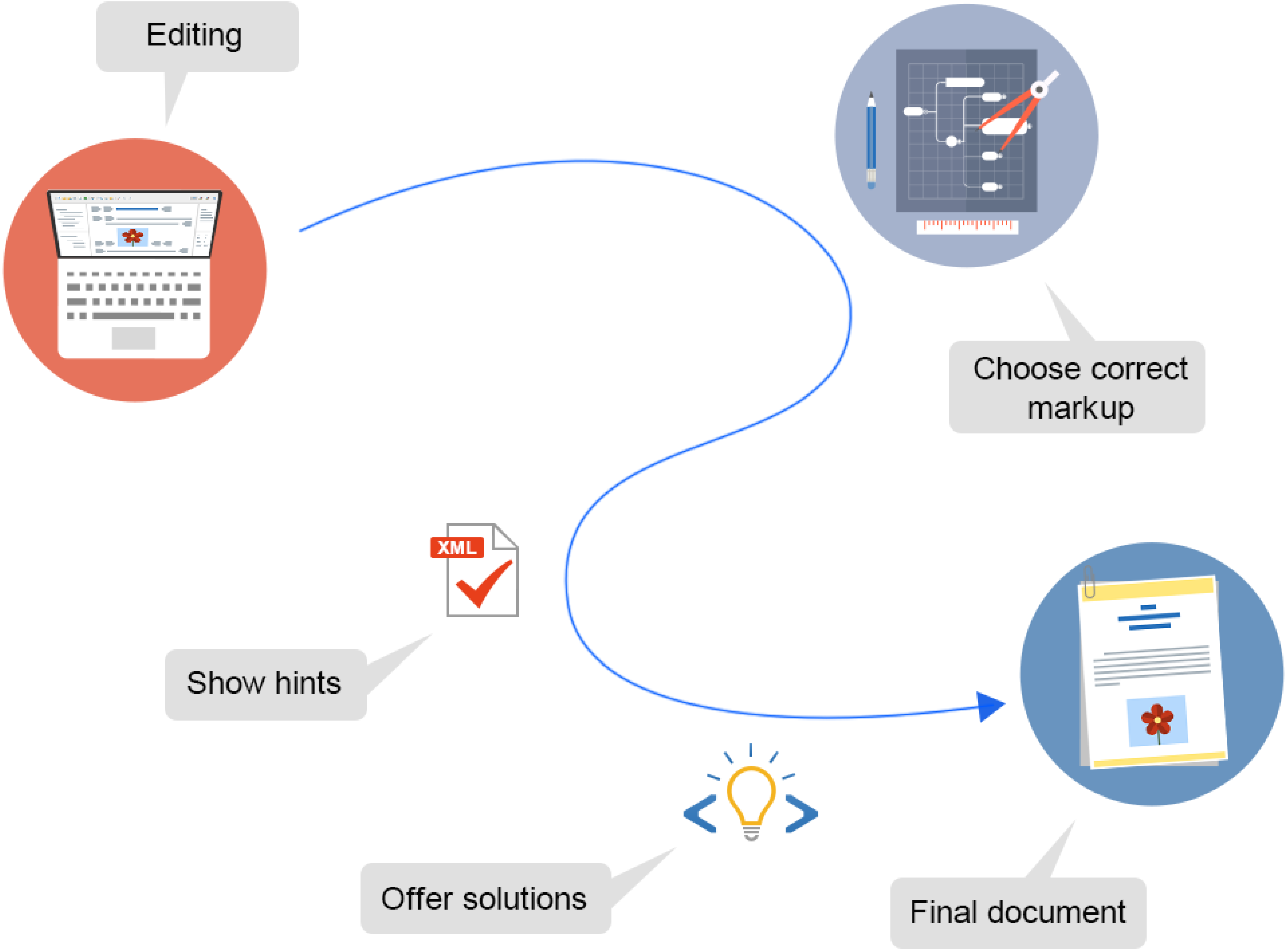


XML Schema, DTD,
Schematron Business Rules,
Style guides

Editing

Final document

Guiding the User



Editing

Choose correct markup

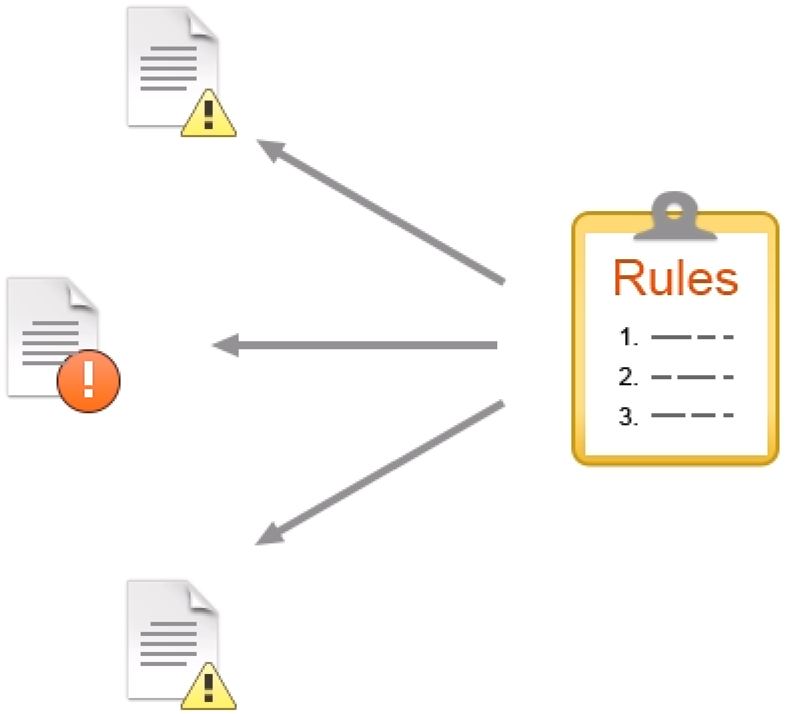
Show hints

Offer solutions

Final document

Business Rules

Rules for your documents that cannot be imposed by the schema



Examples:

- Titles should have content
- Consecutive lists are not allowed
- IDs must follow a certain pattern
- Consecutive notes of a same type
- Too many entries in a table row
- Sections must have IDs
- Titles are too long
- ...

Business Rule Types

Various types of business rules:

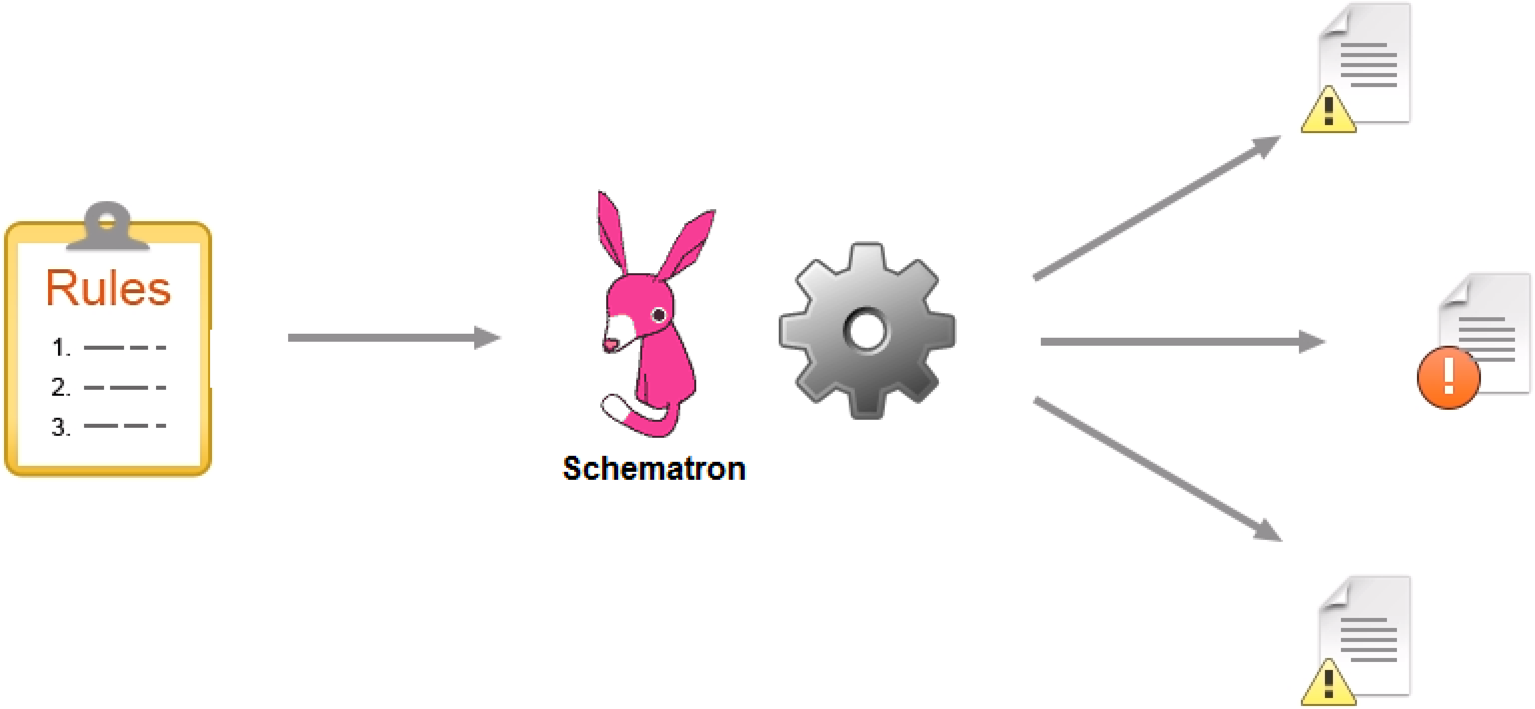
- Simple style rules
  - Styling is not allowed in titles
  - Semicolon is not allowed at the end of a list item
  - Text in the link and the value of the @href are the same
- Editing consistency rules
  - Topic ID must be equal to file name
  - All sections should have an @id
  - Consecutive lists are not allowed
- Structure rules
  - Missing cells in a table
  - Too many nested lists
  - List contains only one item
- Output related rules
  - Lines in codeblocks should not exceed 80 characters.

Business Rules Check

- Hard for your team members to remember all business rules

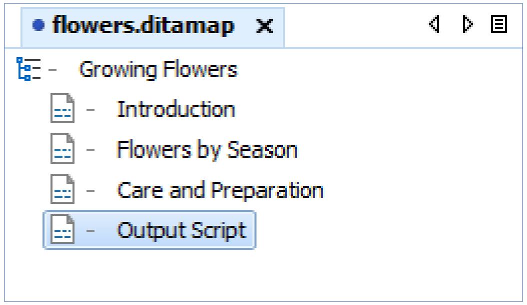
Business Rules Automatic Check

Implement automatic checks for business rules using Schematron



Example

- Implement automatic business rules check in a simple DITA project



flowers.ditamap

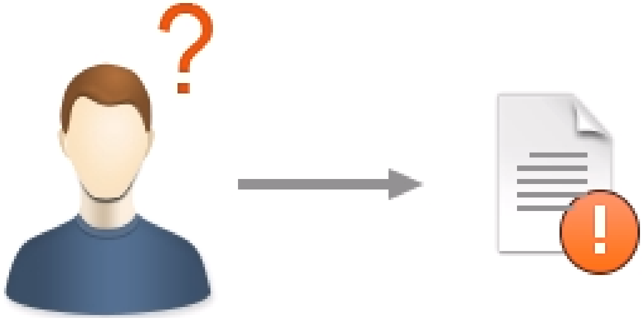
- Growing Flowers
  - Introduction
  - Flowers by Season
  - Care and Preparation
  - Output Script

Conclusions

- Use Schematron to check the business rules
- Schematron is a ISO standard, can be used in any other application
- Each project can have its own custom checks
- Extend the framework and add custom rules

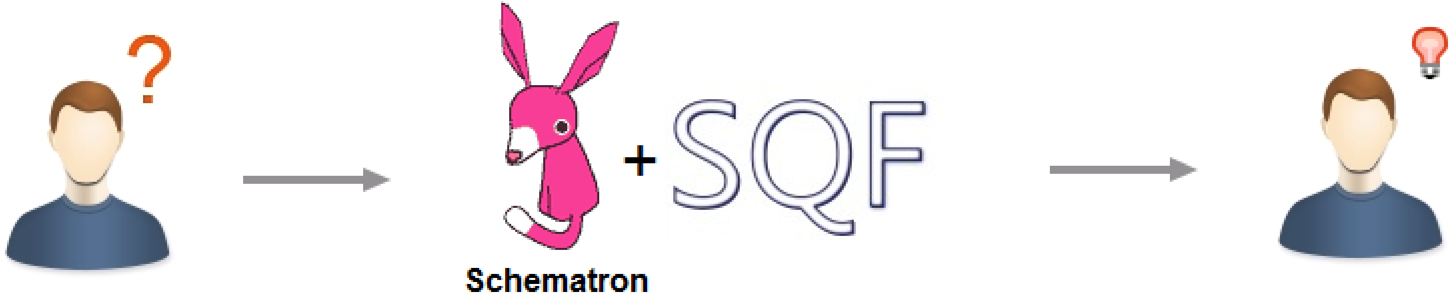
Business Rules Fix Proposals

- Business rule messages are not always enough for the user to find a solution
- Solutions to automatically apply business rule constraints



Business Rule Fixes

Implement fix proposals using Schematron QuickFix (SQF) language



Schematron QuickFix Proposals

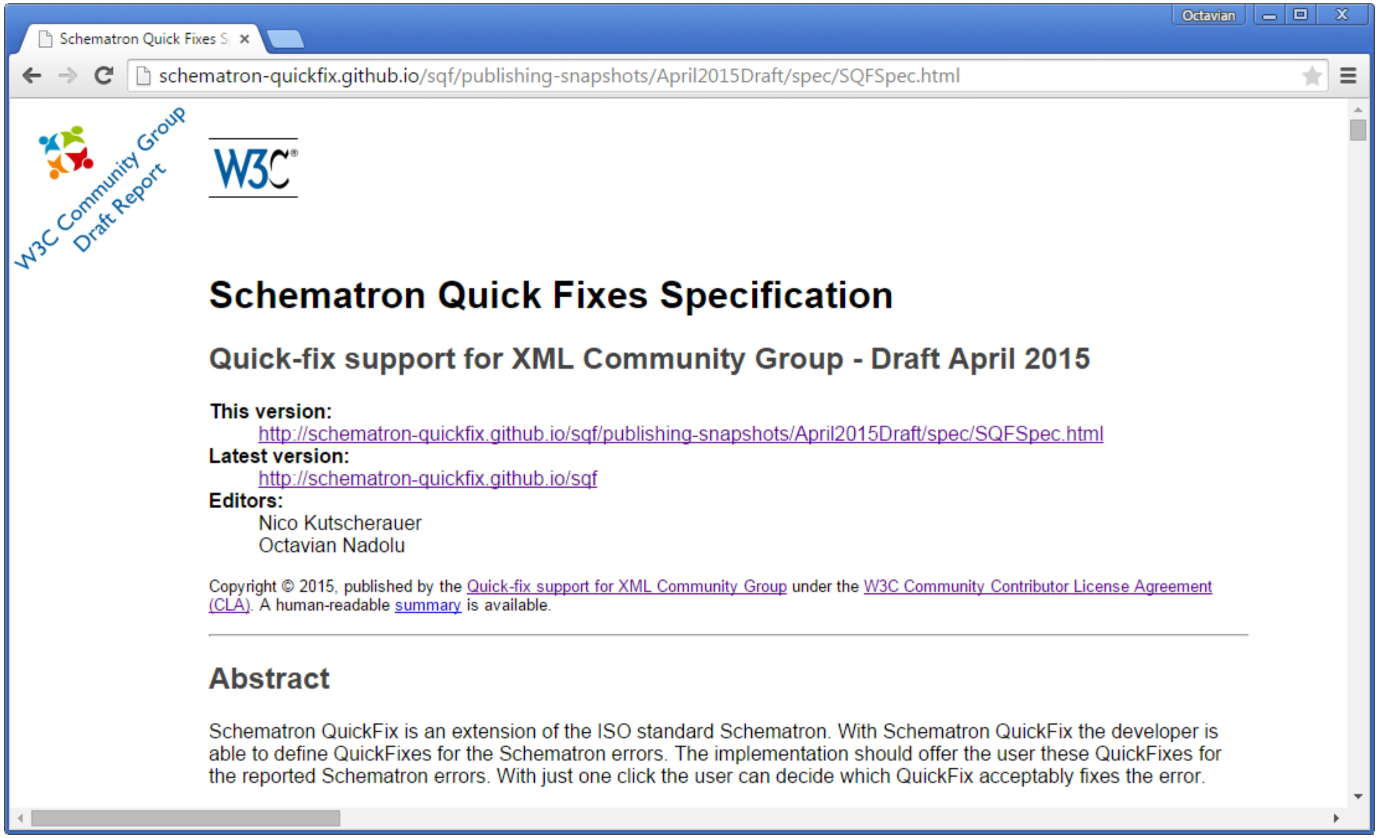
- User-defined fixes for Schematron errors
- Schematron QuickFix (SQF) language
  - Extension of the Schematron language
  - SQF initiated by Nico Kutscherauer

SQF

www.schematron-quickfix.com

github.com/schematron-quickfix/sqf

Schematron Quick Fixes Spec


Schematron Quick Fixes S ×
Octavian
schematron-quickfix.github.io/sqf/publishing-snapshots/April2015Draft/spec/SQFSpec.html
W3C Community Group Draft Report
W3C®

Schematron Quick Fixes Specification

Quick-fix support for XML Community Group - Draft April 2015

This version:
http://schematron-quickfix.github.io/sqf/publishing-snapshots/April2015Draft/spec/SQFSpec.html
Latest version:
http://schematron-quickfix.github.io/sqf
Editors:
Nico Kutscherauer
Octavian Nadolu

Abstract

Schematron QuickFix is an extension of the ISO standard Schematron. With Schematron QuickFix the developer is able to define QuickFixes for the Schematron errors. The implementation should offer the user these QuickFixes for the reported Schematron errors. With just one click the user can decide which QuickFix acceptably fixes the error.

www.w3.org/community/quickfix

schematron-quickfix.github.io/sqf

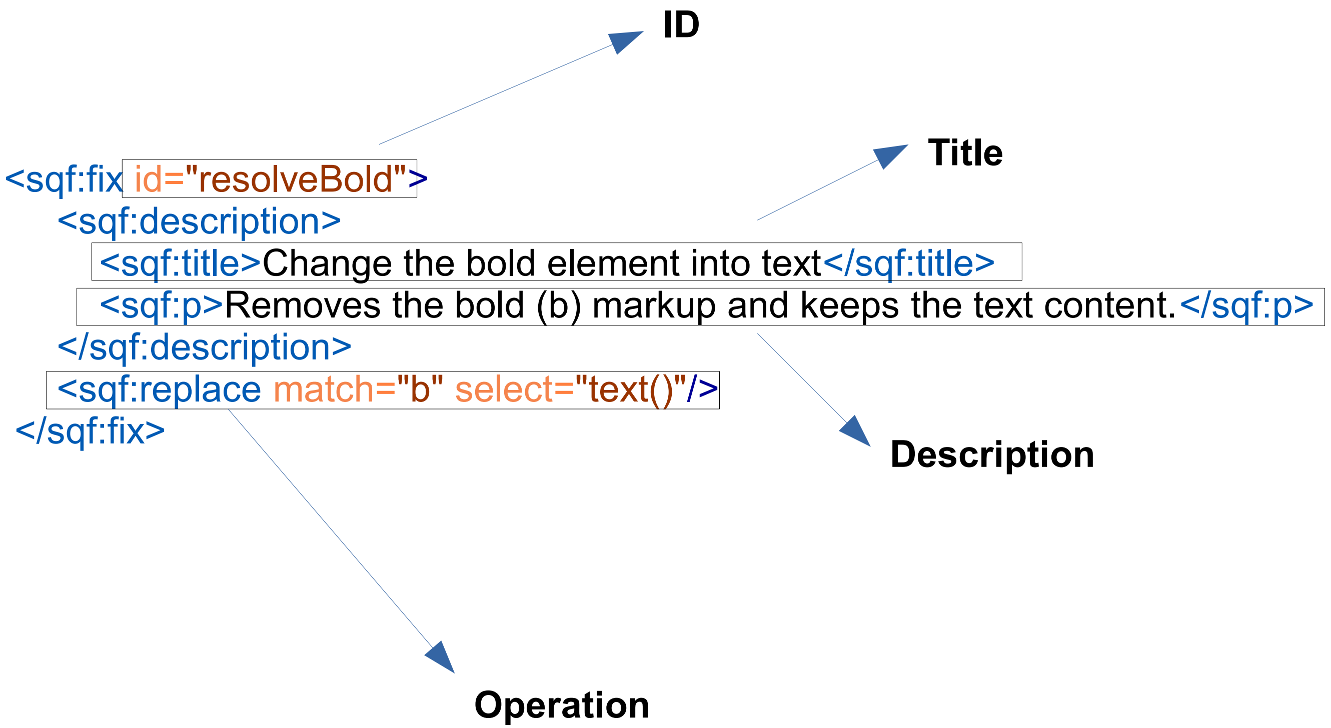
SQF Extension of Schematron

- Added as Schematron annotations
- Associate fixes with assert and report elements

```
<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:fix id="resolveBold">
    <sqf:description>
      <sqf:title>Change the bold element into text</sqf:title>
      <sqf:p>Removes the bold (b) markup and keeps the text content.</sqf:p>
    </sqf:description>
    <sqf:replace match="b" select="text()"/>
</sqf:fix>
```

SQF Operations

The following 4 types of operations are supported:

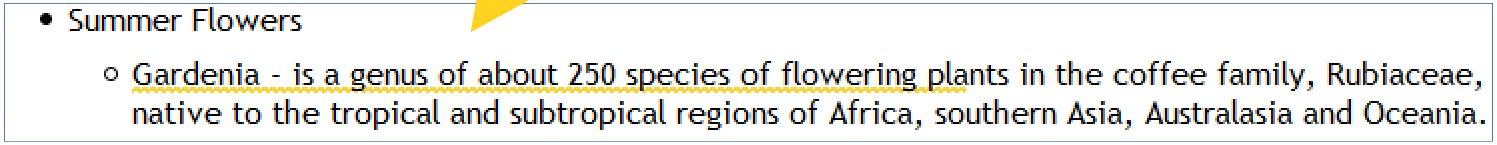
- <sqf:add> - To add a new node or fragment in the document
- <sqf:delete> - To remove a node from the document
- <sqf:replace> - To replace a node with another node or fragment
- <sqf:stringReplace> - To replace text content with other text or a fragment

Introduction to SQF through examples

1. SQF “add” operation

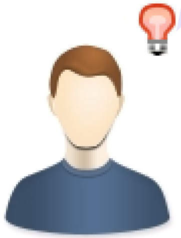
- Example of using the “add” operation: add new list item in a list

List contains only one item



- Summer Flowers
    - Gardenia - is a genus of about 250 species of flowering plants in the coffee family, Rubiaceae, native to the tropical and subtropical regions of Africa, southern Asia, Australasia and Oceania.

Add new list item



1. SQF “add” operation

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

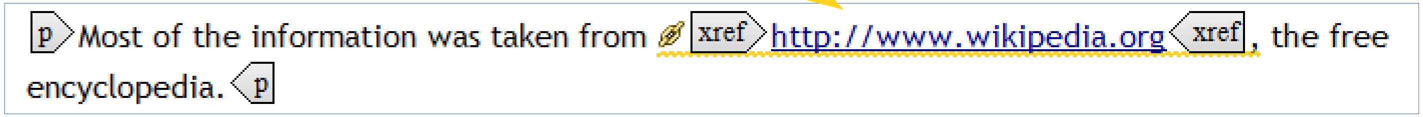
```
<rule context="ul">
    <assert test="count(li) > 1" sqf:fix="addListItem">A list must have more than one item.</assert>

    <sqf:fix id="addListItem">
        <sqf:description>
            <sqf:title>Add new list item</sqf:title>
        </sqf:description>
        <sqf:add node-type="element" target="li" position="last-child"/>
    </sqf:fix>
</rule>
```

2. SQF “delete” operation

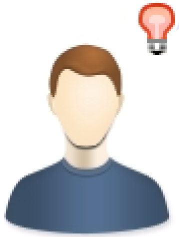
- Example of using the “delete” operation: remove redundant link text

Text in the link and the value of the @href are the same



Most of the information was taken from http://www.wikipedia.org, the free encyclopedia.

Remove redundant link text



2. SQF “delete” operation

- <sqf:delete> element specifies the nodes for the deletion

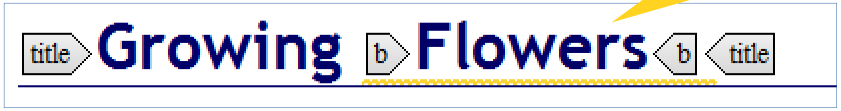
```
<rule context="xref">
    <report test="@href = text()" sqf:fix="removeText">
        Link text is same as @href attribute value. Please remove.</report>

    <sqf:fix id="removeText">
        <sqf:description>
            <sqf:title>Remove redundant link text</sqf:title>
        </sqf:description>
        <sqf:delete match="text()"/>
    </sqf:fix>
</rule>
```

3. SQF “replace” operation

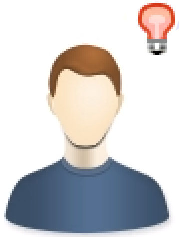
- Example of using the “replace” operation: replace bold element with text

Styling is not allowed in titles



title Growing b Flowers b title

Change the bold into text



3. SQF “replace” operation

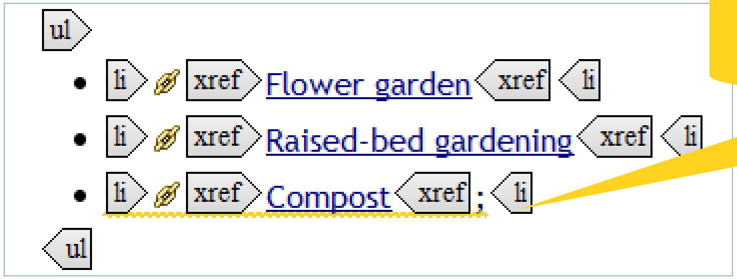
- <sqf:replace> element specifies the nodes to be replaced and the replacing content

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

    <sqf:fix id="resolveBold">
        <sqf:description>
            <sqf:title>Change the bold into text</sqf:title>
        </sqf:description>
        <sqf:replace match="b" select="node()"/>
    </sqf:fix>
</rule>
```

4. SQF “stringReplace” operation

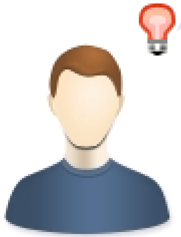
- Example of using the “stringReplace” operation: replace semicolon with full stop



- Flower garden
- Raised-bed gardening
- Compost;

Semicolon is not allowed at the end of a list item

Replace semicolon with full stop



4. SQF “stringReplace” operation

- <sqf:stringReplace> element defines the nodes that will replace the substrings

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

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

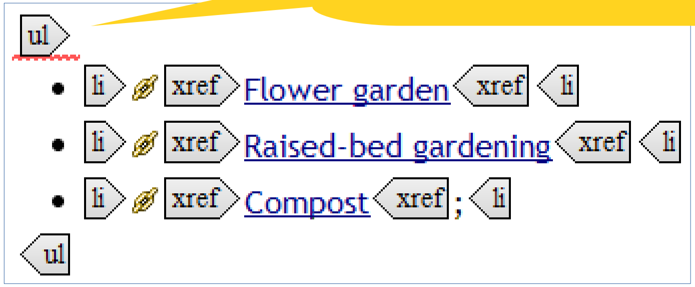
Conclusions

- Schematron QuickFix language is simple
- You can define custom fixes for your project
- Just 4 types of operations

User Entry

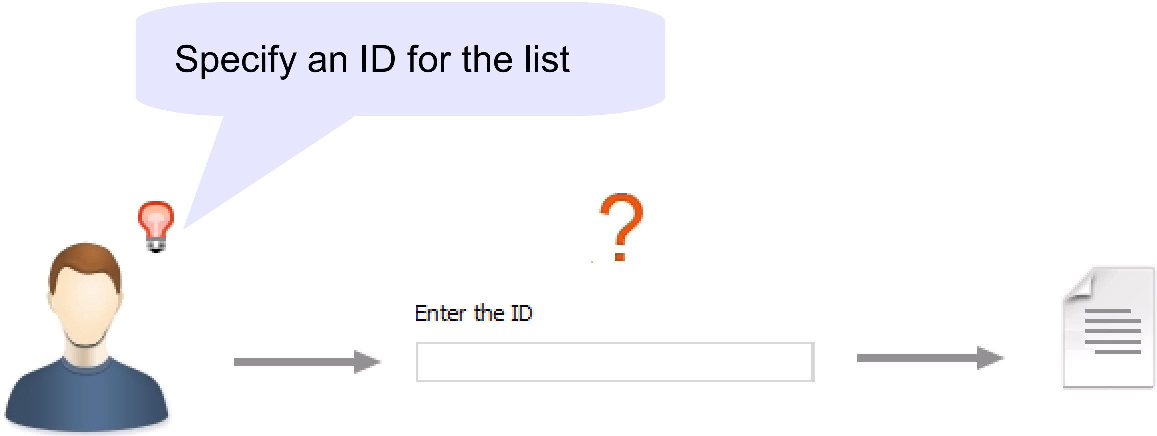
- Ask the user what value to insert

The list must have an ID



ul
- li xref Flower garden xref li
- li xref Raised-bed gardening xref li
- li xref Compost xref ; li

ul



User Entry

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

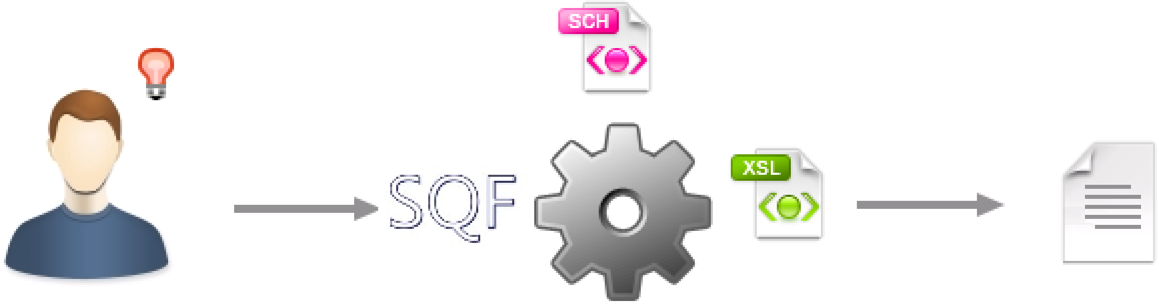
```
<rule context="ul">
  <assert test="@id" sqf:fix="addID">The list must have an ID</assert>

  <sqf:fix id="addID">
    <sqf:description><sqf:title>Specify an ID for the list</sqf:title></sqf:description>

    <sqf:user-entry name="listID">
      <sqf:description>sqf:title>Enter the ID</sqf:title></sqf:description>
     </sqf:user-entry>
     <sqf:add node-type="attribute"  target="id" select="$listID"/>
  </sqf:fix>
</rule>
```

Complex Operations

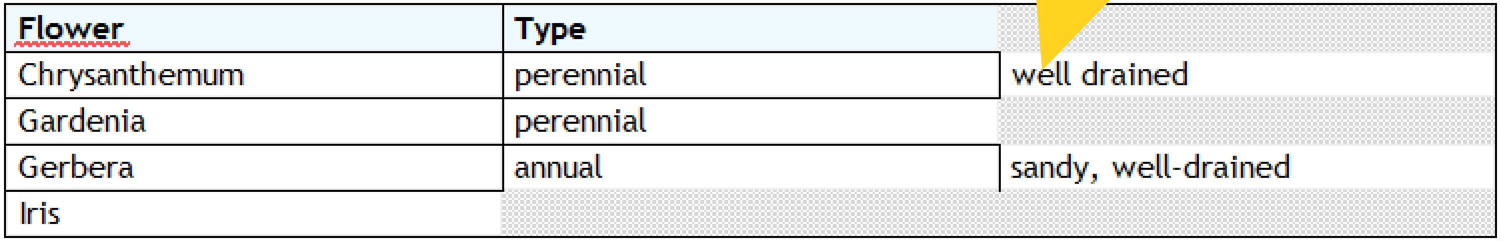
- Use SQF, Schematron and XSLT to create complex operations



Complex Operation Example 1

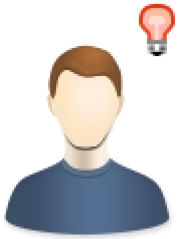
- Missing cells in a table

Cells are missing from table



Flower	Type	
Chrysanthemum	perennial	well drained
Gardenia	perennial	
Gerbera	annual	sandy, well-drained
Iris		

Add enough empty cells on each row



Complex Operation Example 1

- Add the missing cells from a table

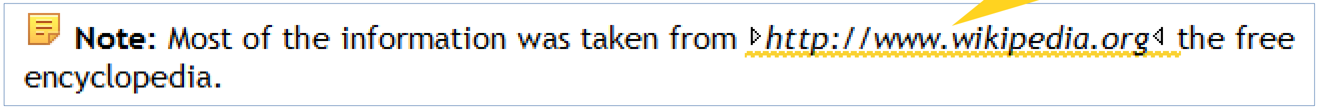
```
<sqf:fix id="addCells">
    <sqf:description>
        <sqf:title>Add enough empty cells on each row</sqf:title>
    </sqf:description>

    <sqf:add match="//row" position="last-child">
        <sch:let name="columnNo" value="count(entry)"/>
        <xsl:for-each select="1 to ($reqColumsNo - $columnNo)">
            <entry/>
        </xsl:for-each>
   </sqf:add>
</sqf:fix>
```

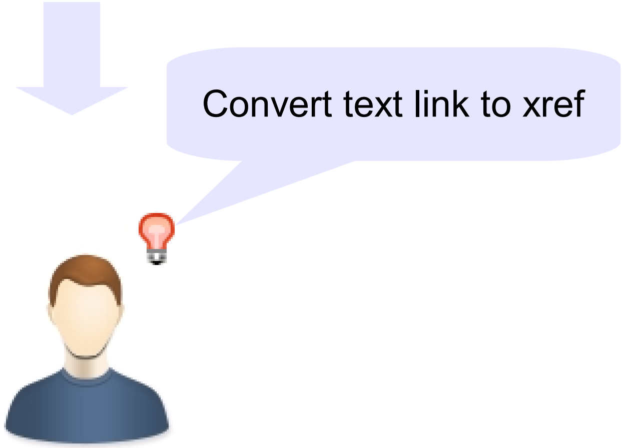
Complex Operation Example 2

- Automatic tagging

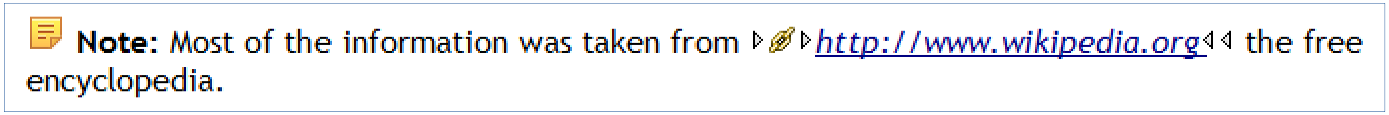
Link detected in the current element



Note: Most of the information was taken from *http://www.wikipedia.org* the free encyclopedia.



Convert text link to xref



Note: Most of the information was taken from *http://www.wikipedia.org* the free encyclopedia.

Complex Operation Example 2

- Convert text link to 'xref'

```
<sqf:fix id="convertToLink">
    <xsl:variable name="linkValue">
        <xsl:analyze-string select="." regex="(http|www)\S+">
            <xsl:matching-substring>
                <xsl:value-of select="regex-group(0)"/>
            </xsl:matching-substring>
        </xsl:analyze-string>
    </xsl:variable>

    <sqf:description>
        <sqf:title>Convert '<sch:value-of select="$linkValue"/>' text to xref</sqf:title>
    </sqf:description>
    <sqf:stringReplace regex="(http|www)\S+">
        <xref href="{$linkValue}" format="html"/>
    </sqf:stringReplace>
</sqf:fix>
```

Other SQF Elements

- **call-fix** – calls another fix within a fix
- **with-param** – refers to a parameter of the called fix
- **param** – defines a parameter for a fix
- **fixes** – global element that contains fixes
- **group** – defines a group of fixes that can be referenced
- **keep** – used to copy the selected nodes

schematron-quickfix.github.io/sqf

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 - a product developed by pagina GmbH used to facilitate EPUB production
- ART-DECOR - an open source tool suite that supports SDOs active in the healthcare industry
  Sample SQF embedded in XSD
- ATX custom framework – used by a major automotive manufacturer

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

- Business rules can be signaled automatically
- Fix proposals created using SQF, a simple and useful language
- SQF helps users to solve the problems in less time and with no errors

Thank you!

Questions?

<oXygen/> XML Editor

http://www.oxygenxml.com

octavian_nadolu@oxygenxml.com

@OctavianNadolu