

Enforce Style Guide Rules

Octavian Nadolu

octavian_nadolu@oxygenxml.com
@OctavianNadolu

Overview

- **Technical documentation projects**
- Achieve accurate, consistent documentation that conforms to your project guidelines
- **How to define style guide rules for your documentation projects**



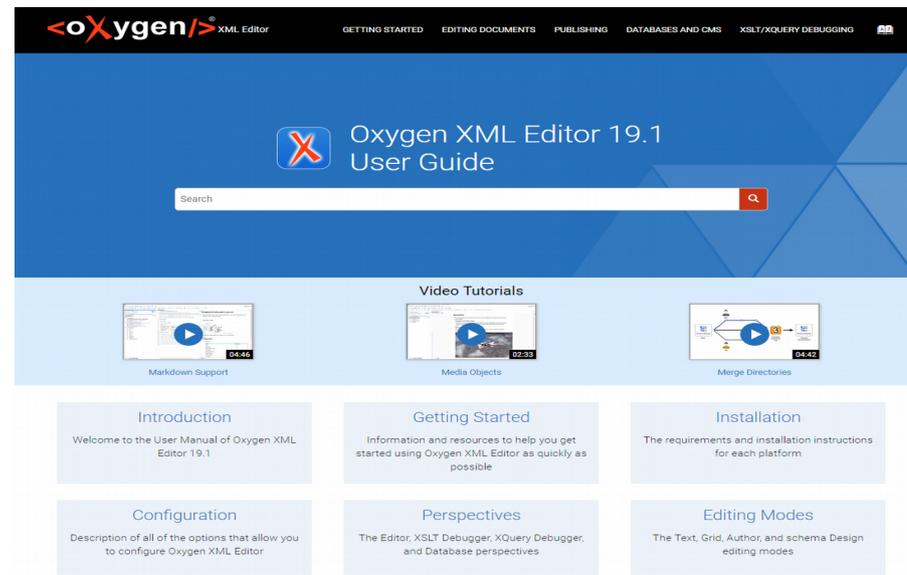
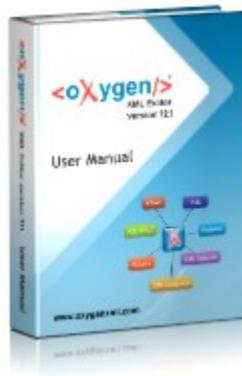
Technical Documentation

- User or system manuals
- Help documentation
- Training material
- Work instructions

Can be defined as information about a product.

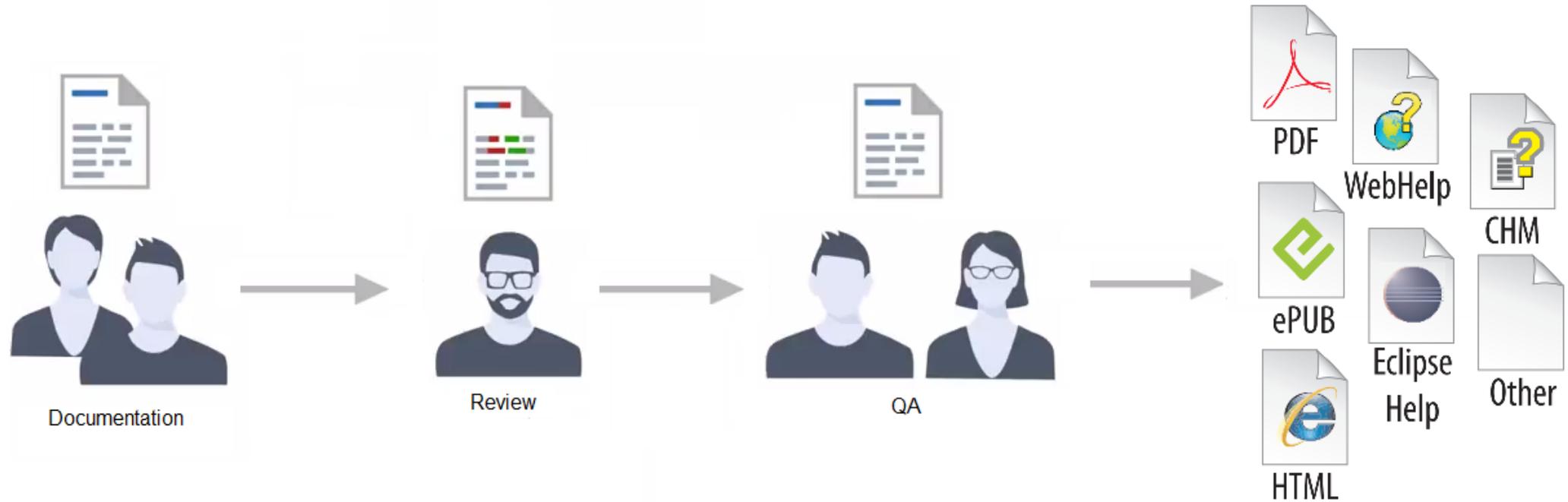
Example

- <oxygen/> User Manual
- DITA project  <dita>
- About 2600 topics



The screenshot shows the Oxygen XML Editor 19.1 User Guide website. The header includes the Oxygen logo and navigation links: GETTING STARTED, EDITING DOCUMENTS, PUBLISHING, DATABASES AND CMS, and XSLT/XQUERY DEBUGGING. The main content area features the title "Oxygen XML Editor 19.1 User Guide" and a search bar. Below the search bar, there are three video tutorial thumbnails: "Markdown Support" (04:46), "Media Objects" (02:29), and "Merge Directories" (04:42). At the bottom, there are six topic cards: "Introduction" (Welcome to the User Manual of Oxygen XML Editor 19.1), "Getting Started" (Information and resources to help you get started using Oxygen XML Editor as quickly as possible), "Installation" (The requirements and installation instructions for each platform), "Configuration" (Description of all of the options that allow you to configure Oxygen XML Editor), "Perspectives" (The Editor, XSLT Debugger, XQuery Debugger, and Database perspectives), and "Editing Modes" (The Text, Grid, Author, and schema Design editing modes).

Documentation Process

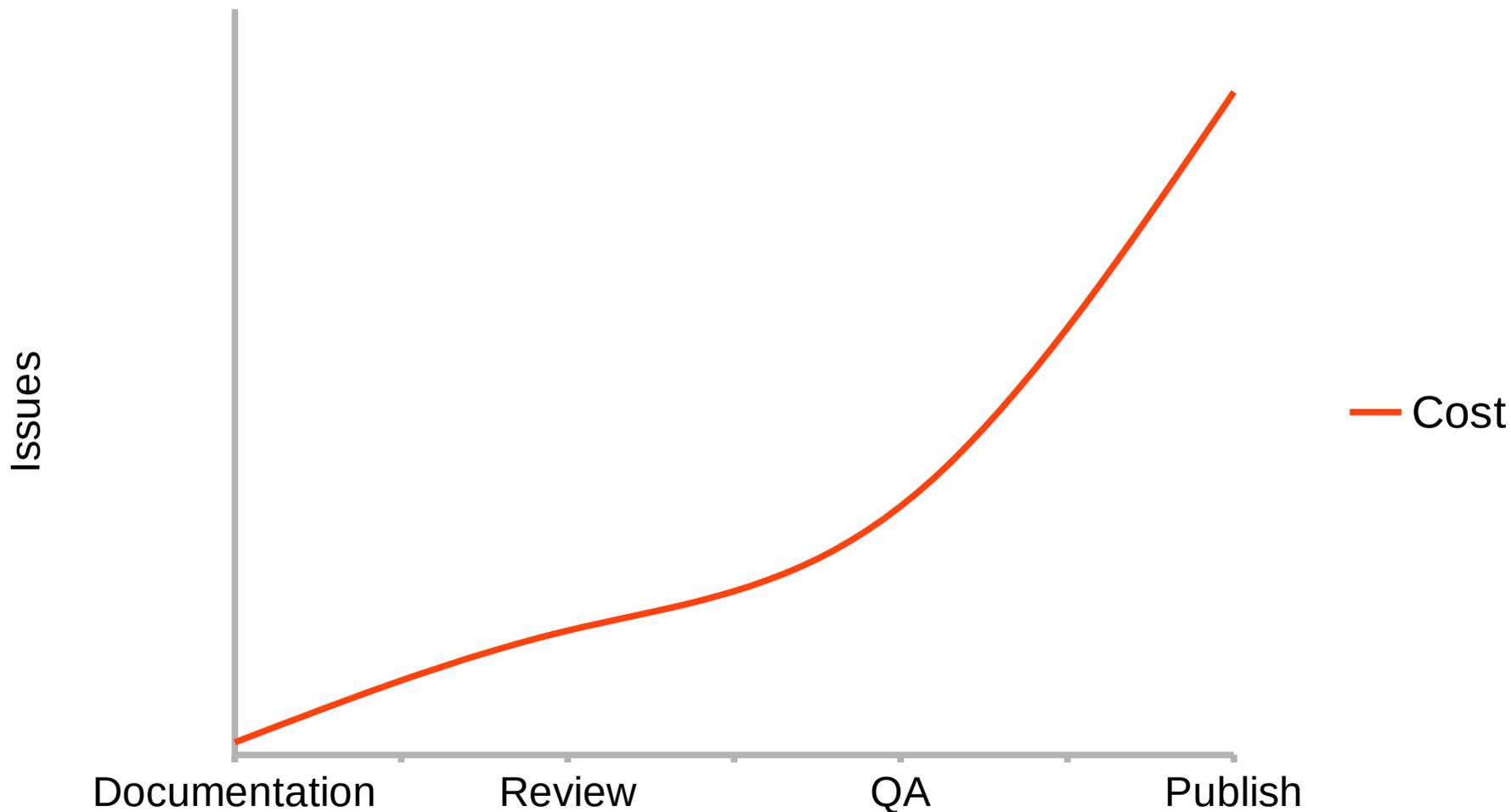


Challenges

- Maintaining consistency for all documents
- Impose a structure and a set of styling rules
- Avoid known issues in documentation output



Cost-Efficiency



Solutions

- Prevent issues by defining rules
- Detect the issues at the earliest editing stage
- Propose consistent solutions to solve the issues



Impose Structure

topic

title

Growing Flowers

title

body

section

title

Introduction

title

p With just a little bit of care and preparation, any flower garden can be a vibrantly colored environment. Flowers can be selected for specific blooming seasons, colors and shapes. Both annual and perennial flower gardens can be planted depending on climate and specific needs. p

p Read more about growing flowers here: p

ul

- li Flower garden; li
- li Raised-bed gardening; li

ul

section

body

topic

Document Structure Rules

- Impose a document structure using a schema language:
 - DTD
 - XML Schema
 - Relax NG
 - NVDL
- Oxygen supports all schema languages

Content Completion and Annotations

The screenshot displays a DITA document structure with annotations for content completion and styling. The document is organized as follows:

- topic
- title **Growing Flowers** title
- body
 - section
 - title **Introduction** title
 - p With just a **little** bit of care and preparation, any flower garden can be a vibrantly colored one during the blooming seasons, and many gardens can be planned.
 - p Read more about [gardening](#).
 - ul
 - li Flower garden; li
 - li Raised-bed gardening; li

A context menu is open over the word "little", showing a list of elements to surround it with:

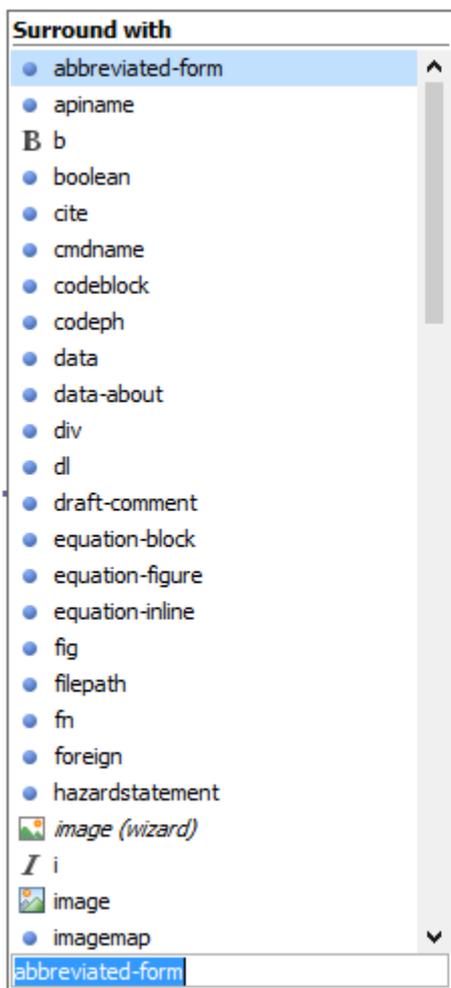
- abbreviated-form
- apiname
- B b**
- boolean
- cite
- cmdname

The selected option is **B b**. A tooltip for the `` element is also visible, providing the following information:

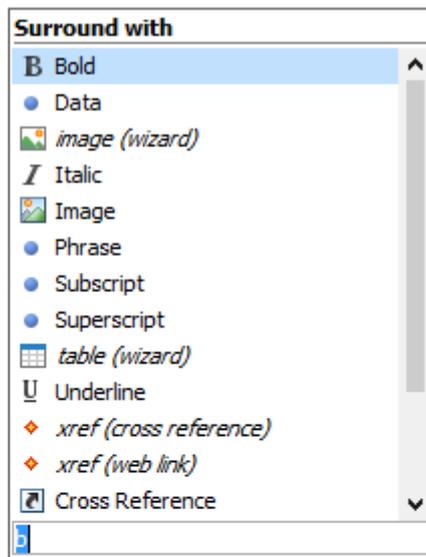
- Category: Highlighting elements
- [DITA Style Guide](#)
- [DITA Specs](#)

At the bottom of the document structure, the following elements are listed: ul, section, body, topic.

Limit the Number of Proposals



~90 proposals



~15 proposals

- **Configure** the proposals for elements and attributes
- Lightweight DITA `<ditā>`
- **Controlled Values & Markup Options**

Conclusions

- ✓ Impose a structure
- ✓ Suggestions to insert valid markup
- ✓ Annotations to help you understand what the markup represents
- ✓ Links to additional information and style guides
- ✓ Powerful schema-aware support

Impose Business Rules

Rules for your documents that cannot be imposed by the schema

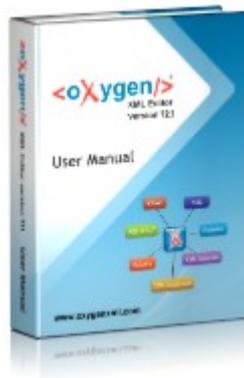


Examples:

- 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
- ...

Example

- For <oxygen/> User Manual we defined more than 200 rules.



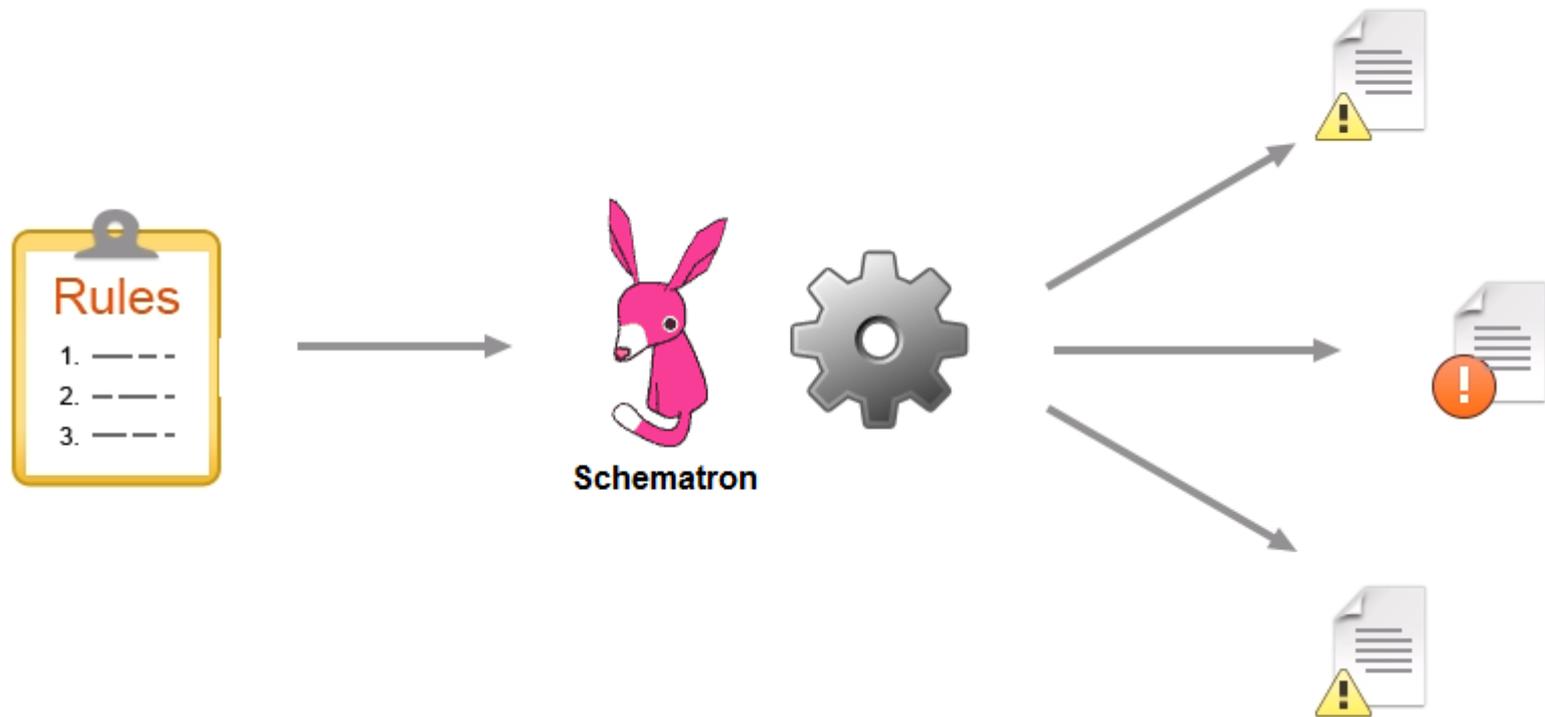
Rules Check Challenges

- Documentation team members needs to **remember** all rules
- Determine if a rule is **violated**
- Provide also **consistent solutions** to correct the problems

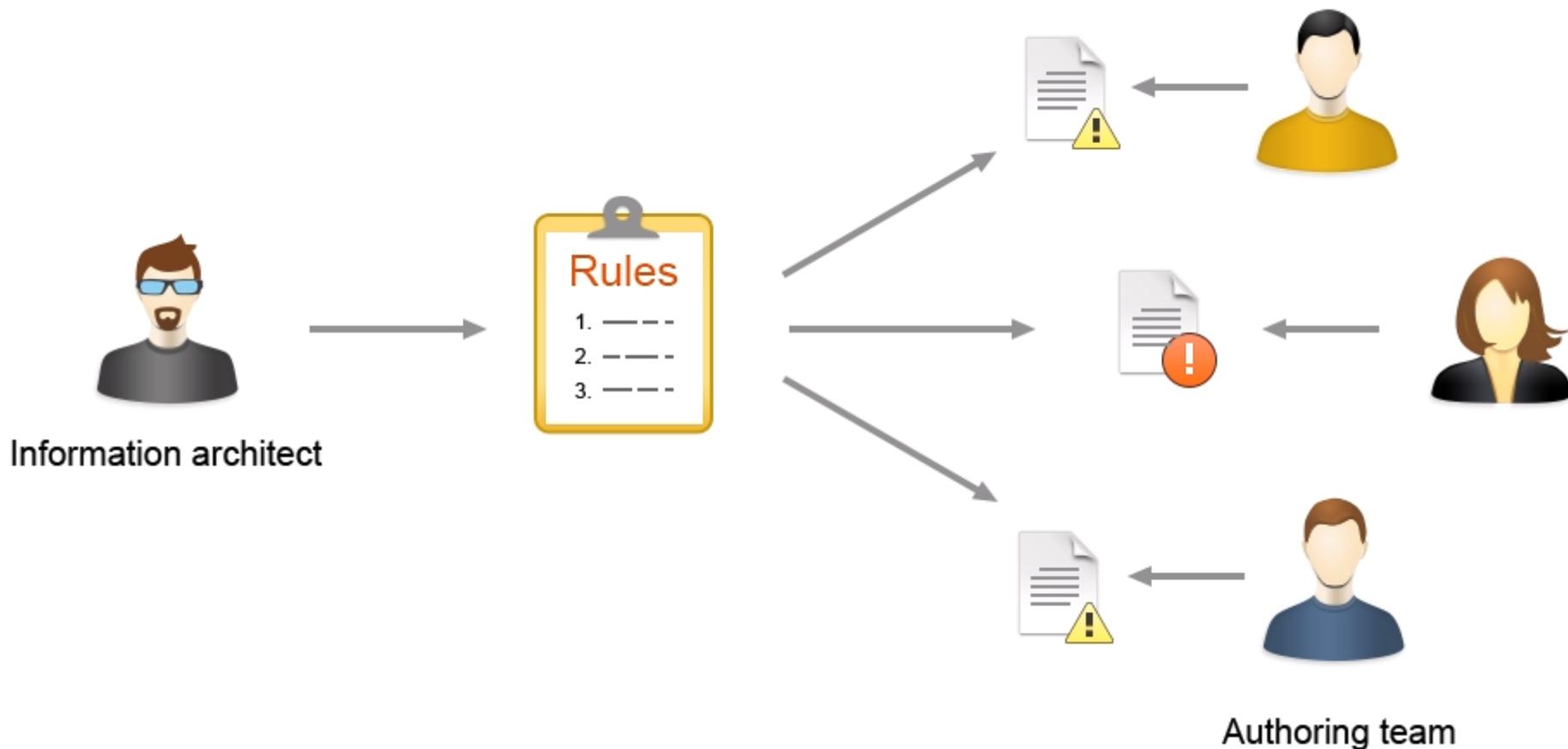


Automatic Check Rules

Implement automatic checks for rules using Schematron



Rules for All Documents and Authors



Example

- Create a simple documentation project
- Apply a set of business rules for all the documents

Simple Style Rules

Styling is not allowed in titles

topic
title **Growing** b **Flowers** b title

body section

title Introduction title

p With just a little bit of care and preparation, any flower garden can be a vibrantly colored environment. Flowers can be selected for specific blooming seasons, colors and shapes. Both annual and perennial flower gardens can be planted depending on climate and specific needs. p

p Read more about growing flowers here: p

ul

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

Semicolon is not allowed at the end of a list item

ul

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

section body topic

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

Editing Consistency Rules

Topic ID must be equal to file name

topic id="id_kvj_bwn_fw"

title **Growing Flowers** title

body section title **Introduction** title

p With just a little bit of care and preparation, an **@id** ment.
Flowers can be selected for specific blooming seasons, colors and shapes. Both annual and perennial flower gardens can be planted depending on climate and specific needs. p section

section title **Flowers by Season** title

p The various climatic changes that occur in cyclic pattern are termed as 'Seasons'. There are four general seasons occurring on Earth - Spring, Summer, Autumn and Winter. p

ul

- li Spring Time, the time of growth and renewal of new plant and animal life. Spring comes at different times in the North and South Hemispheres. Spring time in the Northern hemisphere is between March and May, while in the Southern hemisphere it is between September - November in the Southern hemisphere. li

Consecutive lists are not allowed

ul

ul

- li Summer is the time of hot and warm weather. Floral growth is the best in the summer season. The Northern hemisphere experiences summer during June, July, August, while in Southern hemisphere during December - February. li

ul section body topic

Structure Rules

Growing Flowers

Care and Preparation

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

Missing cells in a table

Flowers by Season

- Spring Flowers
 - Iris - is a genus of between 200-300 species of flowering plants with showy flowers.
 - ◻ Irises are extensively grown as ornamental plants in home and botanical gardens.
 - ◻ Some rhizomes are traded as orris root and are used in perfume and medicine, though more common in ancient times than today.
 - Snowdrop - is the common name for a small genus of about 20 species in the family Amaryllidaceae.
- 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.

Too many nested lists

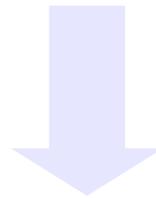
List contains only one item

Output Related Rules

Sample

On Linux, the Java command should look like this:

```
▶ java -Xmx600m -XX:MaxPermSize=256m "-Dcom.oxygenxml.editor.plugins.dir=$OXYGEN_HOME/plugins"
▶ java -Xmx600m -XX:MaxPermSize=256m \
  "-Dcom.oxygenxml.editor.plugins.dir=$AUTHOR_HOME/plugins"
▶ java -Xmx600m -XX:MaxPermSize=256m \
  "-Dcom.oxygenxml.editor.plugins.dir=$DEVELOPER_HOME/plugins"
```



Sample

On Linux, the Java command should look like this:

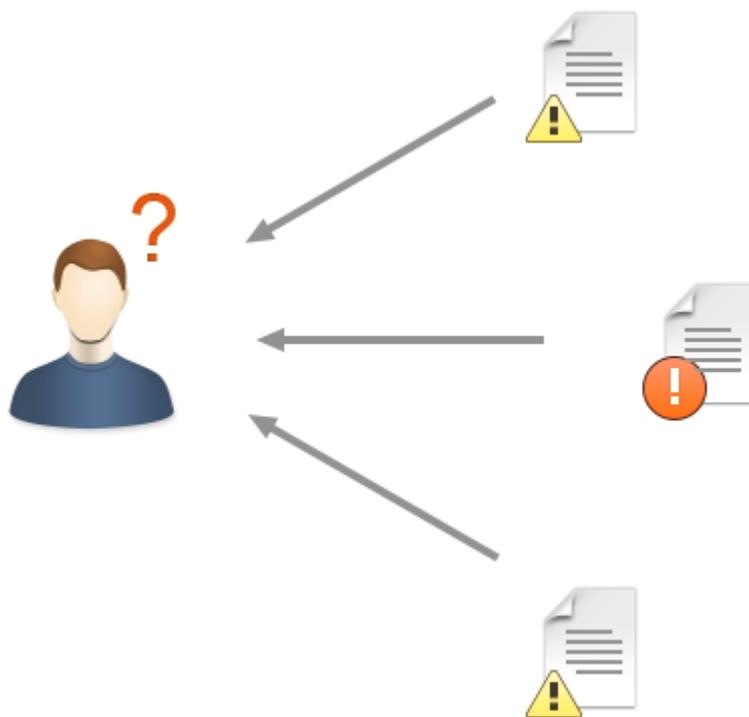
```
java -Xmx600m -XX:MaxPermSize=256m "-Dcom.oxygenxml.editor.plugins.dir=
$OXYGEN_HOME/plugins"
java -Xmx600m -XX:MaxPermSize=256m \
  "-Dcom.oxygenxml.editor.plugins.dir=$AUTHOR_HOME/plugins"
java -Xmx600m -XX:MaxPermSize=256m \
  "-Dcom.oxygenxml.editor.plugins.dir=$DEVELOPER_HOME/plugins"
```



PDF

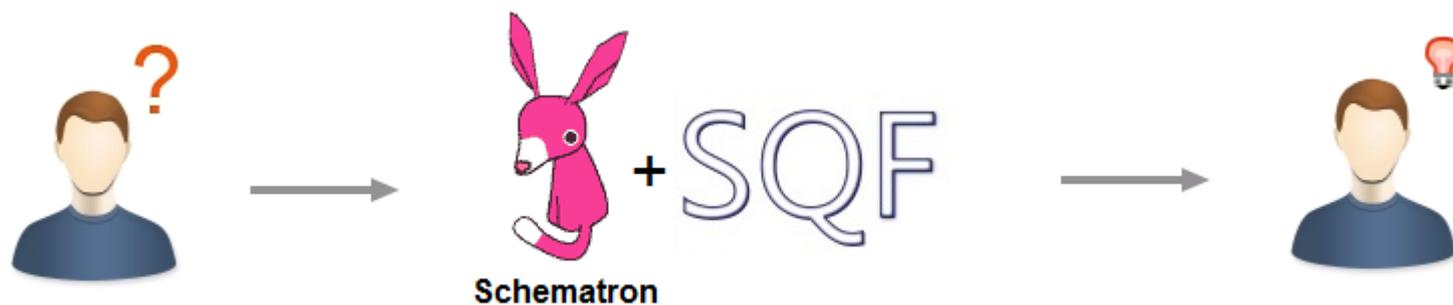
Consistent Solutions to Correct Problems

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

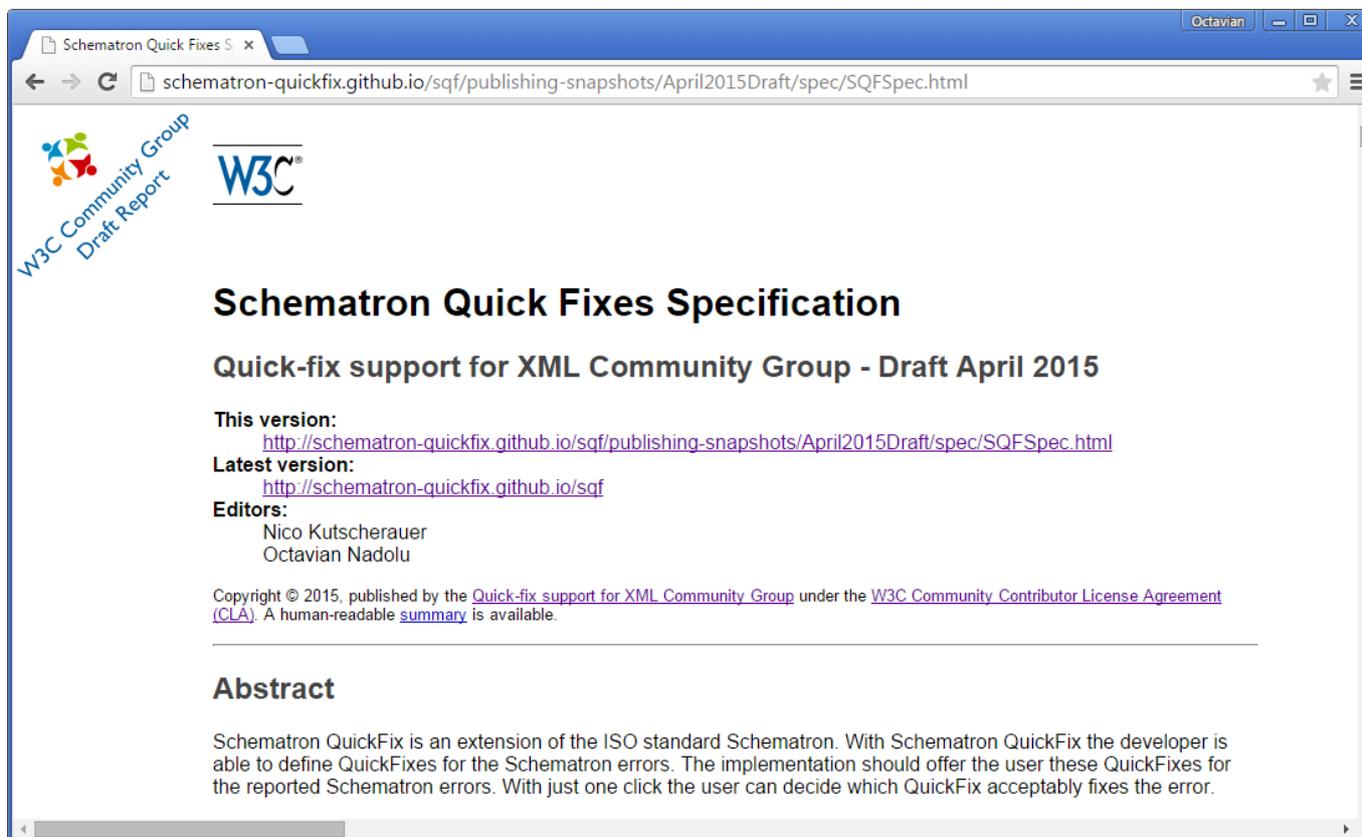


Business Rule Fixes

Implement fix proposals using Schematron QuickFix (SQF) language



Schematron Quick Fixes Spec



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

Copyright © 2015, published by the [Quick-fix support for XML Community Group](#) under the [W3C Community Contributor License Agreement \(CLA\)](#). A human-readable [summary](#) is available.

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

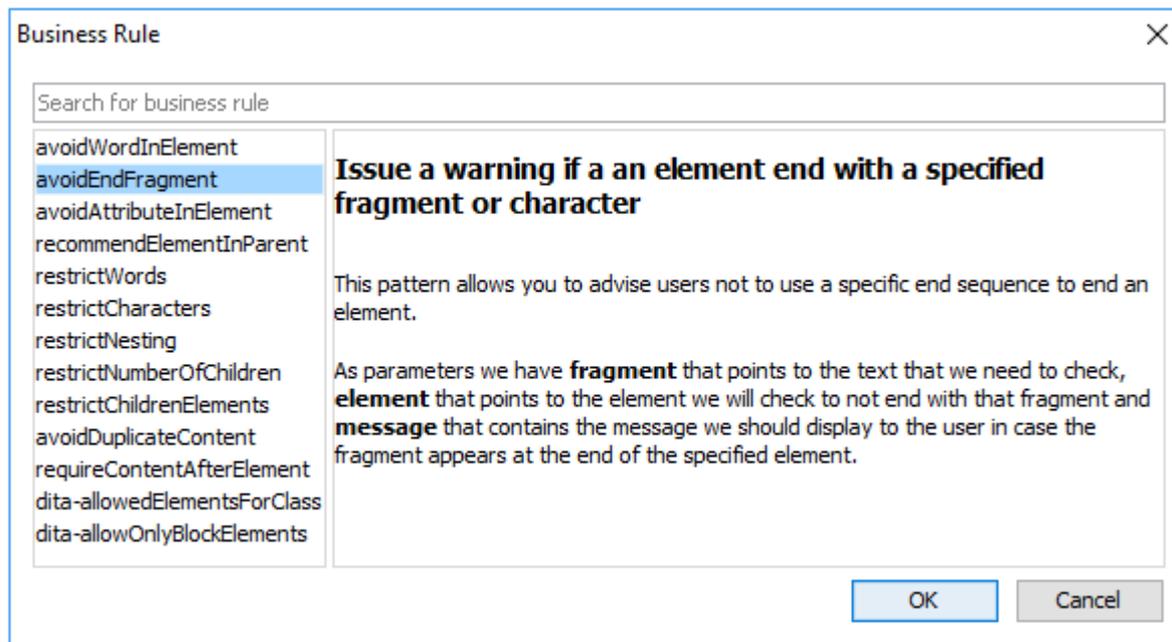


schematron-quickfix.github.io/sqf

Example

- Create fix proposals for the business rules
- Share the fix proposals with all team

Create Business Rules using DIM



Rule	avoidEndFragment
element	li
fragment	;
message	Avoid ; at the end of a list item

Dynamic Information Model (DIM)

- An implementation of an intelligent style guide
- Describes and enforces rules
- [Schematron for Information Architects](#)
presented by George Bina at XML Prague
- Open source project available on GitHub
<https://github.com/oxygenxml/dim>

Rules Check Advantages

- Detect the issues at editing time
- Prevent problems in the generated output
- All team follows same rules
- Consistent solutions to correct problems



Projects Using Business Rules



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

Business Rules Fix Proposals

Discover the technology behind the fix proposals for business rules:

-  Defining Schematron Quick Fixes
-  Understanding and Developing Schematron Quick Fixes
-  Schematron Quick Fixes Samples
-  Schematron Quick Fixes Specification

Thank you!

Questions?

`<oxygen/>` XML Editor

<http://www.oxygenxml.com>

octavian_nadolu@oxygenxml.com

[@OctavianNadolu](#)