

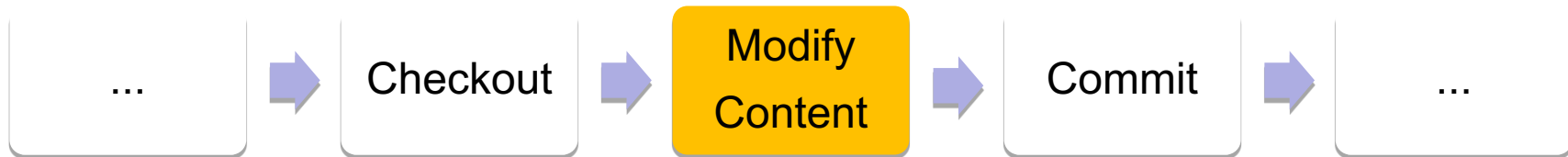
# Integrate Collaboration into your Workflow



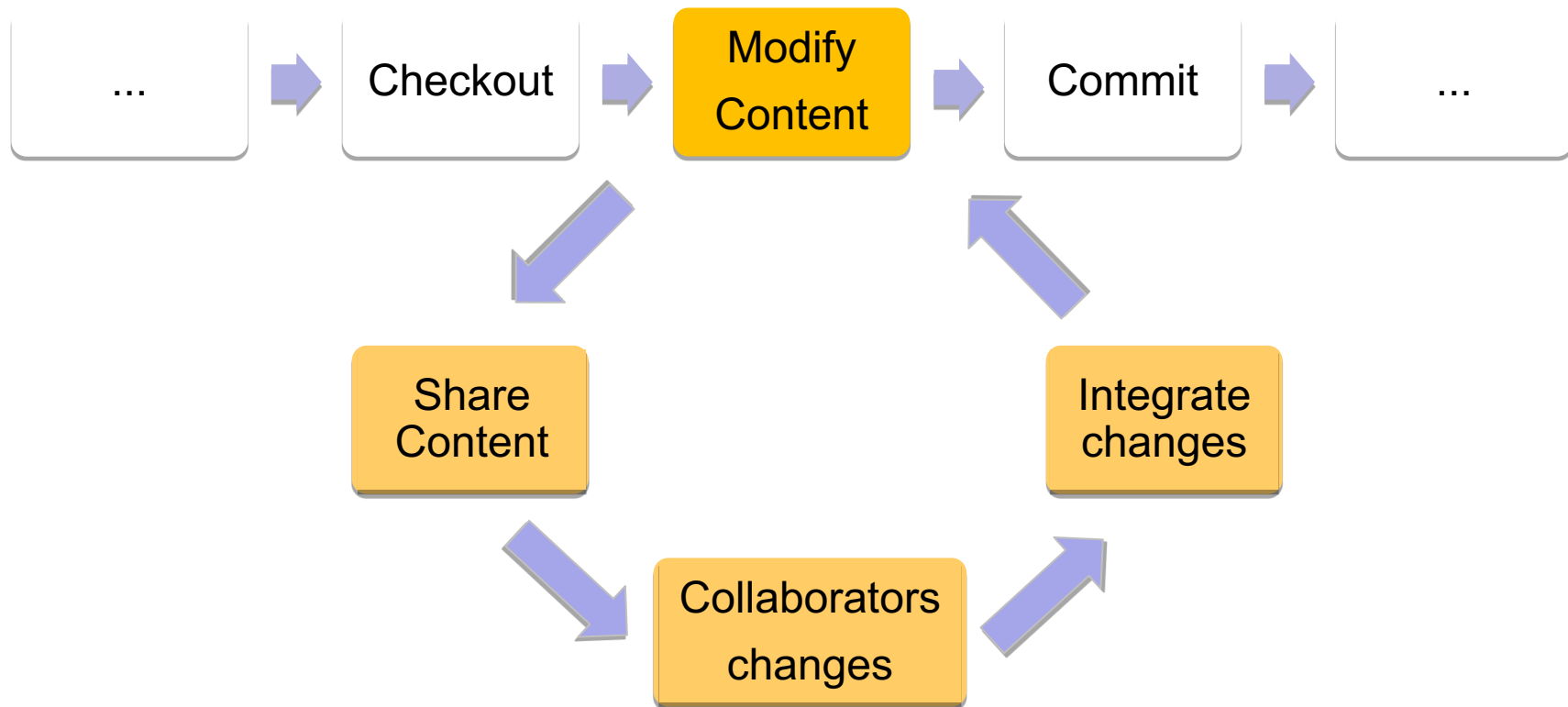
George Bina  
[george@oxygenxml.com](mailto:george@oxygenxml.com)  
@georgebina



# Existing workflow



# Integrating collaboration workflow



# oxygen Content Fusion

Collaboration on XML content using a browser!  
(DITA, DocBook, XHTML, your own XML)

# Why oXygen Content Fusion?

You need to **collaborate**

You have **access to specific tools**

You work within a specific **workflow**

You have **access** to a repository

You want to **approve changes** before reaching the repository

You need a **less formal** way to receive feedback

# It cannot be easier to collaborate!

**oxygen Content Fusion** allows you to:

1. **Create** a review/collaboration task from XML files
2. **Share** task URL with your contributors

**Contributors need only a web browser!**

- a) Follow link
  - b) Make changes
  - c) Mark as done
3. **Get changes** back into the original files

# Tools



oXygen XML Editor or oXygen XML Author  
Content Fusion Connector plugin  
oXygen Content Fusion server

# Process

1. **Create** review/contribute task
2. **Share** it with reviewers/contributors
3. **Integrate** changes



# Let's see how it works

## Examples:

1. Create a simple review task with a few topics
2. Provide a place for contributors to enter content
3. Restrict editing to specific content

# Supported functionality

One or multiple XML files

DITA Map to provide the editing context for DITA

Multiple reviewers/contributors

Automatic locking

Change tracking

Visual XML-aware 3-way merge with automatic merge support for pseudo-conflicts

Task messages and email notifications

# Enable/optimize workflows

# Distributed services

- Storage
- Versioning
- Publishing
- Communication
- Continuous Integration

# Distributed services

- Storage
- Versioning
- Publishing
- Communication
- Continuous Integration
- XML Authoring?

# XML Authoring as a Service

Access XML Authoring by following a URL

<http://server/service?url=file/to/edit>

# Technical support to documentation

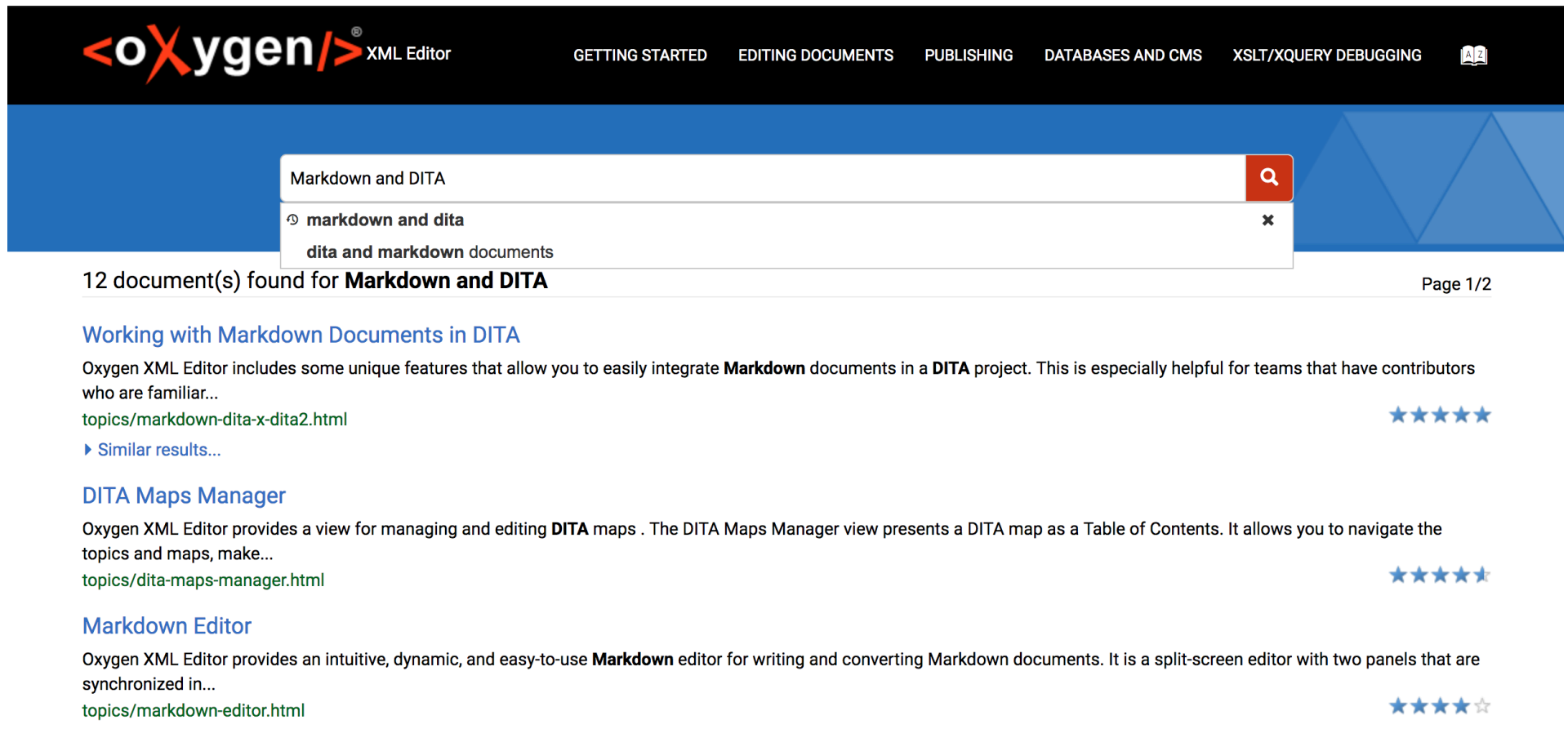
Create a channel between technical support team and documentation team to enable **continuous improvement of documentation**

# Technical support to documentation

Provide a URL the technical support engineer can follow to suggest changes to documentation!



# Identify documentation topic



The screenshot shows the Oxygen XML Editor interface with a search bar containing "Markdown and DITA". A dropdown menu shows search suggestions: "markdown and dita" and "dita and markdown documents". Below the search bar, it indicates "12 document(s) found for **Markdown and DITA**". The results list includes:

- Working with Markdown Documents in DITA**  
Oxygen XML Editor includes some unique features that allow you to easily integrate **Markdown** documents in a **DITA** project. This is especially helpful for teams that have contributors who are familiar...  
[topics/markdown-dita-x-dita2.html](#) ★★★★★  
▶ Similar results...
- DITA Maps Manager**  
Oxygen XML Editor provides a view for managing and editing **DITA** maps . The DITA Maps Manager view presents a DITA map as a Table of Contents. It allows you to navigate the topics and maps, make...  
[topics/dita-maps-manager.html](#) ★★★★★
- Markdown Editor**  
Oxygen XML Editor provides an intuitive, dynamic, and easy-to-use **Markdown** editor for writing and converting Markdown documents. It is a split-screen editor with two panels that are synchronized in...  
[topics/markdown-editor.html](#) ★★★★★

# See if it responds to user issue

Home / DITA Authoring and Publishing / Working with Markdown Documents in DITA



## Working with Markdown Documents in DITA

History | Edit

Oxygen XML Editor includes some unique features that allow you to easily integrate **Markdown** documents in a **DITA** project. This is especially helpful for teams that have contributors who are familiar with the **Markdown** syntax, but they want their output to be generated from **DITA** projects. The integration between the **Markdown** editor and **DITA** includes actions to export or convert **Markdown** documents to **DITA** topics and the **DITA** tab in the *Preview* pane provides a visualization of how the topic will look after conversion.

### ▼ Export **Markdown** as a **DITA** Topic

The **Markdown** editor includes an option to quickly convert the current **Markdown** document into a **DITA** topic. The **Export as DITA Topic** action is available in the contextual menu of the left-side text editor and the right-side *Preview* pane when the **DITA** tab selected.

The conversion creates a new XML file that is defined as a **DITA** topic and opens it in the **Text** editing mode. You can then work with the document as you would with any other **DITA** topic, although you may need to manually correct some issues where the parser could not properly map **Markdown** syntax to **DITA** markup.

### ▼ Working with **Markdown** Documents in the **DITA** Maps Manager

Oxygen XML Editor has some specialized features that allow you to integrate **Markdown** documents directly into your **DITA** project using the **DITA Maps Manager**. The following features are available for **Markdown** documents in the **DITA Maps Manager** view:

- **Insert Reference to **Markdown** Document** - You can use the **New**, **Reference**, and **Reference to the currently edited file** actions from the **Append Child**, **Insert Before**, or **Insert After** submenu when invoking the contextual menu in the **DITA Maps Manager** to insert a reference to a **Markdown** document at the selected location in the map.

### DITA Authoring and Publishing

Working with DITA Maps

Working with DITA Topics

**Working with Markdown Documents in DITA**

Working with Keys

Reusing DITA Content

Linking in DITA

Publishing DITA Output

DITA Profiling / Conditional Text

DITA Open Toolkit Support

DITA Specialization Support

Master Files Support in DITA

Metadata

Migrating MS Office Documents to DITA

DITA 1.3 Support

# Use the edit link to suggest changes

The screenshot shows the Oxygen XML Web Author interface. The top bar includes the Oxygen logo, the text 'XML Web Author', navigation icons, the file name 'markdown-dita.dita', the user name 'George Bina', and a menu icon. Below the top bar is a toolbar with two sections: 'Review' (containing icons for adding, deleting, and commenting) and 'DITA' (containing icons for document operations like bold, italic, underline, link, image, etc.). The main content area displays a document titled 'topic title' with the following structure:

## Working with Markdown Documents in DITA

- Metadata**
  - Keywords:**
    - (Markdown documents (DITA) )
    - (DITA and Markdown documents)

[product]Oxygen XML Editor includes some unique features that allow you to easily integrate Markdown documents in a DITA project. This is especially helpful for teams that have contributors who are familiar with the Markdown syntax, but they want their output to be generated from DITA projects. The integration between the Markdown editor and DITA includes actions to export or convert Markdown documents to DITA topics and the **DITA** tab in the *Preview* pane provides a visualization of how the topic will look after conversion.

- Export Markdown as a DITA Topic**

The Markdown editor includes an option to quickly convert the current Markdown document into a DITA topic. The **Export as DITA Topic** action is available in the contextual menu of the left-side text editor and the right-side *Preview*

The right sidebar contains a 'Review' panel with a filter input, an information icon, and instructions: 'To add a comment, select some content and press [comment icon] from the toolbar.' Below the instructions are 'Attributes' and 'Validation' sections, both with dropdown arrows. The 'Validation' section shows a green checkmark.


# Developers to review documentation

Enable developers to review the incremental documentation related to the issue they resolve

# Developers to review documentation

Provide a URL the developer can follow to get access to the modified documentation!


# Automatic message with links to modified topics


 oXygen / EXM-38653 337 of 36,719 ▲▼  
Return to Search


## Suggest only maps as possible DITA master files


Edit Comment Assign More ▼ Reopen for documentation Reopen Issue Export ▼

```
ro.sync.azcheck.ui.SpellControl
```

▼  Sorin Carbutaru added a comment - 23/Feb/17 1:52 PM  
Eliminat ciclul de clase.


▼  Dorina Jitianu [syncro] added a comment - 24/Feb/17 10:13 AM ↻ ✎ 🗑  
Detectarea de master file-uri de DITA va contine numai map-uri.

▼  Steven Higgs added a comment - 24/Feb/17 11:28 AM  
I added a note in "detecting-master-files.dita" that explains that for DITA project, only DITA maps are detected as possible master files.

▼  Web Author Review Bot added a comment - 25/Feb/17 2:41 AM  
Files to be reviewed:

- [DITA/topics/detecting-master-files.dita](#)

A diff of the changes can be seen on [GitHub](#)

▼  Sorin Carbutaru added a comment - 10/Mar/17 12:12 PM  
Reviewed the documentation.

Comment

# Follow link to review

oxygen XML Web Author | detecting-master-files.dita | George Bina

Review

DITA

topic title

## Detecting Master Files

**Metadata**

**Keywords:**

- (Master Files (Detecting))
- (Projects (Master files (Detecting)))
- (Detect Master Files)
- (Detect Master Files from Project)

[product]Oxygen XML Editor allows you to detect the *master files* using the **Detect Master Files** option. This action applies to the folders you select in the project.

To detect *master files* over the entire project, do one of the following:

- Right-click the root of the project and select **Detect Master Files** (if product is 'author' or 'developer' or 'editor') **Detect Master Files from Project** (if product is 'authorEclipse' or 'developerEclipse' or 'editorEclipse').
- Use the **Detect Master Files from Project** option, available in the contextual menu of the *Master Files* folder.

Both of these options display the **Detect Master Files** wizard. In the first panel you can select the type of *master files* you want [product]Oxygen XML Editor to detect. In the subsequent panel the detected *master files* are presented in a tree-like fashion. (if product is 'developer' or 'developerEclipse' or 'editor' or 'editorEclipse') The detected *master files* are presented in a tree-like fashion. (if product is 'author' or 'authorEclipse') The resources are grouped into three categories:

Review

Filter

To add a comment, select some content and press [comment icon] from the toolbar.

Attributes

Validation ✓

# Follow link to see changes

[Code](#) [Issues 1](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)

## EXM-38653: Added a note for DITA projects

[Browse files](#)

dev oxygen19.0

StevenHiggs committed on 24 Feb

1 parent 0b0b5c4 commit 45c7c29ed66383b6a4c38d94d88d91ed4c1e468d

Showing 1 changed file with 3 additions and 1 deletion.

[Unified](#) [Split](#)

4 DITA/topics/detecting-master-files.dita

[View](#)

```
@@ -32,7 +32,9 @@
32     presented in a tree-like fashion. </ph>The resources are grouped into three
    categories: <ul>
33     <li><b>Possible <term>master files</term></b> - The files presented on the first
    level in
34     this category are not imported or included from other files. These files are most
    likely
35     - to be set as <term>master files</term>.</li>
36     <li><b>Cycles</b> - The files that are presented on the first level have circular
37     dependencies between them. Any of the files presented on the first level of a cycle
    is a
38     possible <term>master file</term>. </li>
+
+
+
35     + to be set as <term>master files</term>.<note>For DITA projects, only <xref
36     keyref="glossentry_ditamap"><term>DITA Maps</term></xref> are reported as
    possible
37     <term>master files</term>.</note></li>
38     <li><b>Cycles</b> - The files that are presented on the first level have circular
39     dependencies between them. Any of the files presented on the first level of a cycle
    is a
40     possible <term>master file</term>. </li>
```



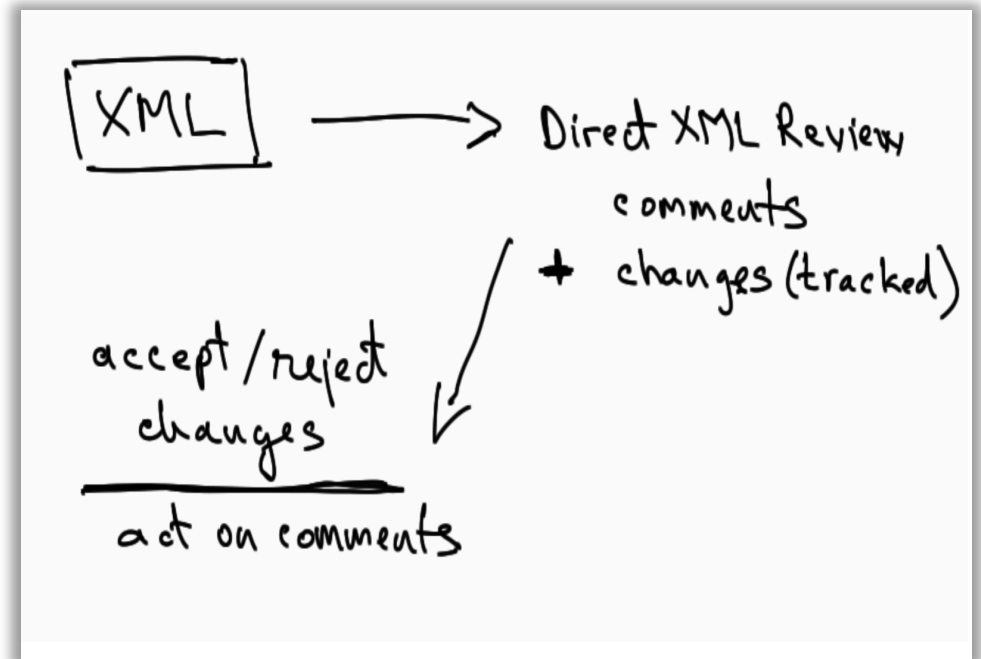
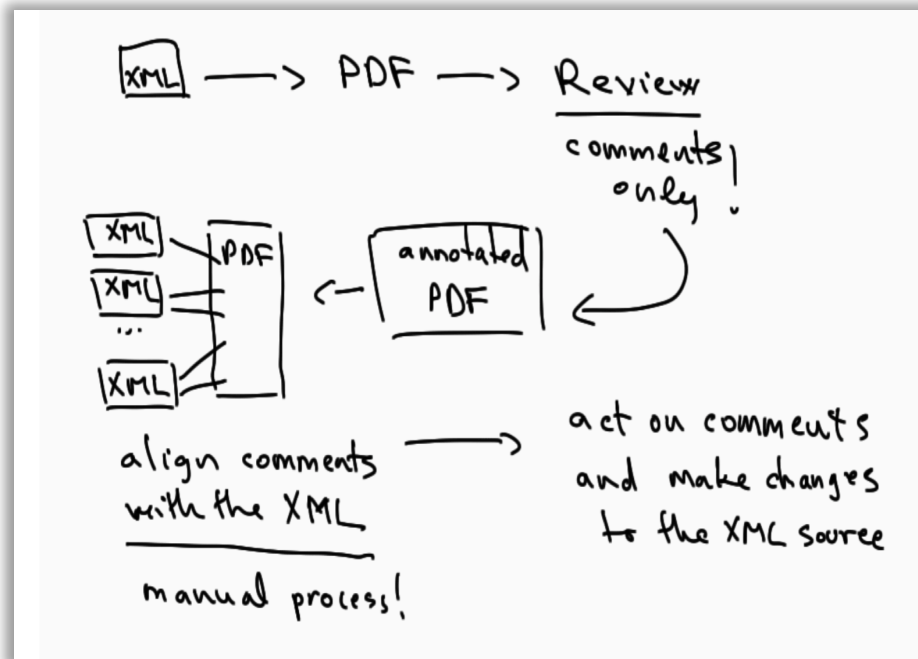
# Immediate Review

Review directly the source

Review a published format

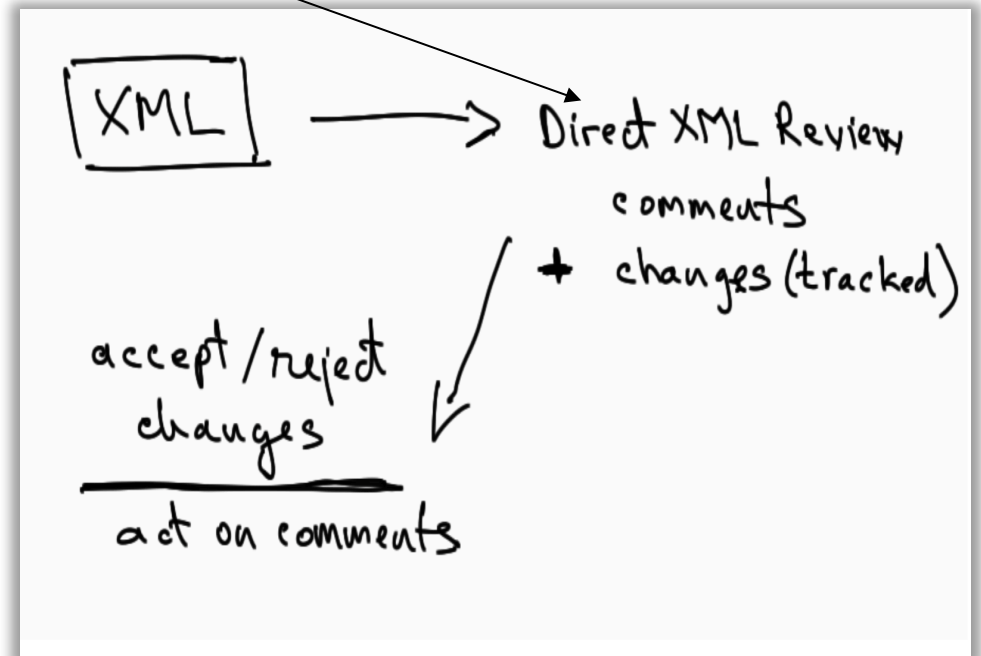
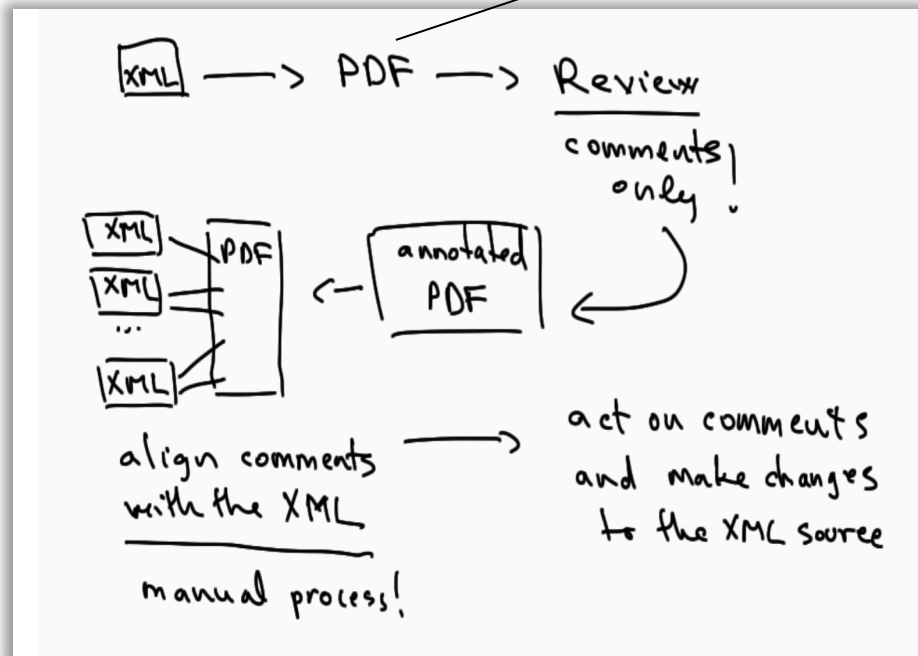
- PDF
- HTML
- Etc.

# Classic vs Direct XML review

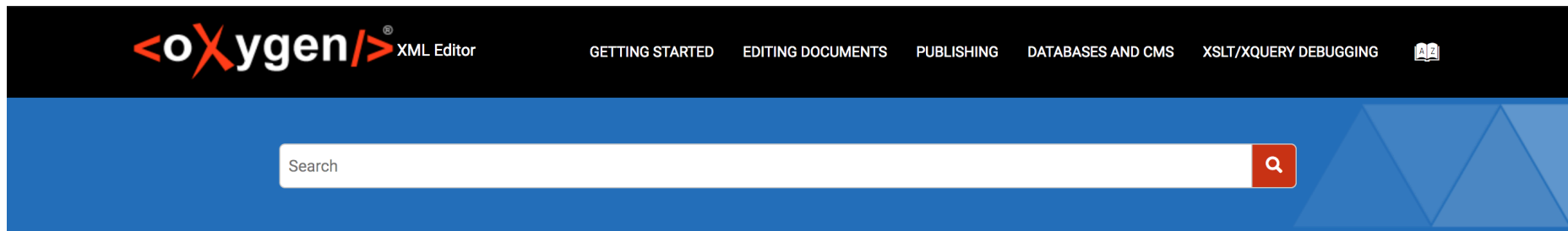


# Immediate Review

Provide a URL to edit the source directly from the published format!



# HTML-based output



Home / Working with Archives




## Working with Archives

History | Edit

Oxygen XML Editor includes a useful [Archive Browser view](#) that offers the means to work with files directly from various types of archives (for example, opening and saving files directly in archives, or browsing and modifying archive structures). The archive support is available for all ZIP-type archives, including:

- ZIP archives
- EPUB books
- [JAR archives](#)
- Office Open XML (OOXML) files
- Open Document Format (ODF) files
- [IDML files](#)

You can transform, validate, and perform many other operations on files directly from an archive. For instance, you can transform, or validate files directly from OOXML or ODF packages, and the structure and content of the ZIP archives can be opened, edited, and saved, similar to any other ZIP archive browsing tool. Also, when browsing for a URL in various dialog boxes, you can use the  **Browse for archived file** action to browse and select files from a particular archive.

### Working with Archives

[Browsing Archives](#)

[Working with Archive Files](#)

[Browsing Archives](#)

# Immediate access to source









The screenshot shows the Oxygen XML Editor interface. At the top, the title bar reads "oxygen XML Web Author" and "working-with-archives.dita". The user "George Bina" is logged in. Below the title bar are two toolbars: "Review" and "DITA". The main content area displays a document titled "Working with Archives" with a "Metadata" section containing "Keywords: (Archives)". The text below discusses the "Archive Browser view" and lists supported archive types: ZIP archives, EPUB books, JAR archives, Office Open XML (OOXML) files, Open Document Format (ODF) files, and IDML files. A "Browse for archived file" action is also mentioned. On the right, a "Review" sidebar is open, showing a "Filter" field and instructions: "To add a comment, select some content and press [comment icon] from the toolbar." Below the instructions are "Attributes" and "Validation" sections.

# PDF-based output

## Saving Documents

[Edit online](#)

You can save the document you are editing with one of the following actions:

- **File >  Save.**
- ** Save toolbar button** - If the document was not yet saved, it displays the **Save As** dialog box.
- **File > Save As** - Displays the **Save As** dialog box, used either to name and save an open document to a file or to save an existing file with a new name.
- **File > Save To URL** - Displays a **Save to URL** dialog box that can be used to save a file identified by its URL (defined by a protocol, host, resource path, and an optional port). Use the drop-down action list to choose one of the available save actions:
  - ** Browse for local file** - Opens a local file browser dialog box allowing you to save the document locally.
  - ** Browse for remote file** - Displays a **Save to URL** dialog box that allows you to save the document to a remote location (accessible through FTP, SFTP or WebDAV).
  - ** Browse for archived file** - Displays the *Archive Browser* that allows you to save the document inside an archive.
  - ** Browse Data Source Explorer** - Opens a *Data Source Explorer* that allows you to browse the data sources defined in the *Data Sources preferences page*.
    - ** Tip:** You can get to the **Data Sources** preferences page, using the **Configure Database Sources** shortcut from the **Save to URL** dialog box.
  - ** Search for file** - Displays the *Open/Find Resource dialog box*.
- **File > Save All** - Saves all open documents. If any document does not have a file, displays the **Save As** dialog box.

# Immediate access to source

The screenshot shows the Oxygen XML Editor interface. The top bar includes the Oxygen logo, 'XML Web Author', navigation icons, the file name 'saving-documents.dita', and the user 'George Bina'. Below the top bar are two toolbars: 'Review' and 'DITA'. The main content area displays a document titled 'Saving Documents' with a 'Metadata' section containing 'Keywords: (Save file)'. A list of actions for saving the document is provided, including 'File→Save', 'Save toolbar button', 'File→Save As', and 'File→Save To URL'. The 'Review' sidebar on the right contains a 'Filter' field, a message 'To add a comment, select some content and press [comment icon] from the toolbar.', and sections for 'Attributes' and 'Validation' (with a green checkmark).

topic title

## Saving Documents

**Metadata**

**Keywords:**  
(Save file)

You can save the document you are editing with one of the following actions:

- **File→Save.**
- **Save toolbar button** - If the document was not yet saved, it displays the **Save As** dialog box. (if product is 'editor' or 'author' or 'developer')
- **File→Save As** - Displays the **Save As** dialog box, used either to name and save an open document to a file or to save an existing file with a new name.
- **File→Save To URL** - Displays a **Save to URL** dialog box that can be used to save a file identified by its URL (defined by a protocol, host, resource path, and an optional port). Use the drop-down action list to choose one of the available save actions:
  - **Browse for local file** - Opens a local file browser dialog box allowing you to save the document locally.
  - **Browse for remote file** - Displays a **Save to URL** dialog box that allows you to save the document to a remote location (accessible through FTP, SFTP or WebDAV).
  - **Browse for archived file** - Displays the **Archive Browser** that allows you to save the document inside an

**Review**

Filter

To add a comment, select some content and press [comment icon] from the toolbar.

**Attributes**

**Validation** ✓

# Crowd-sourcing docs

Enable anyone to contribute to your project documentation!



# DITA-OT documentation

## Services used by the DITA-OT documentation

- **GitHub**
  - storage
  - versioning
  - access control
- **GitHub Pages**
  - web publishing
- **Travis**
  - automate publishing
- **Slack**
  - project communication

# Click on website to edit its content



- DITA Open Toolkit 2.3
- Release Notes
- Getting Started
- User Guide
- Parameter Reference
- Developer Reference
- Contributing
- Who we are

## DITA Open Toolkit 2.3

The *DITA Open Toolkit*, or *DITA-OT* for short, is a set of Java-based, open-source tools that provide processing for XML content authored in the *Darwin Information Typing Architecture*.

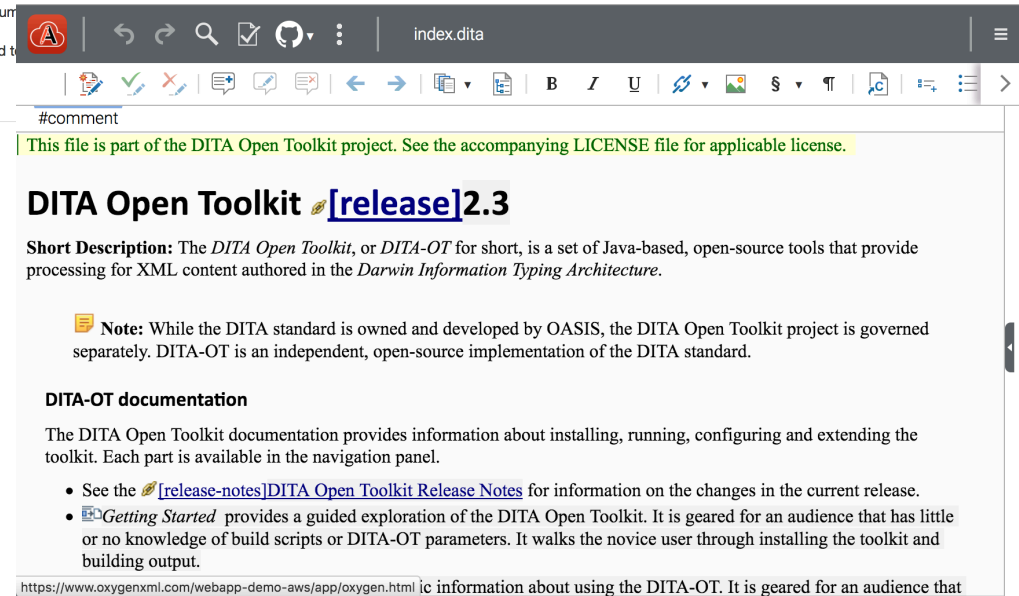
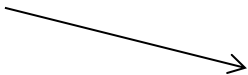
**Note:** While the DITA standard is owned and developed by OASIS, the DITA Open Toolkit project is governed separately. DITA-OT is an independent, open-source implementation of the DITA standard.

### DITA-OT documentation

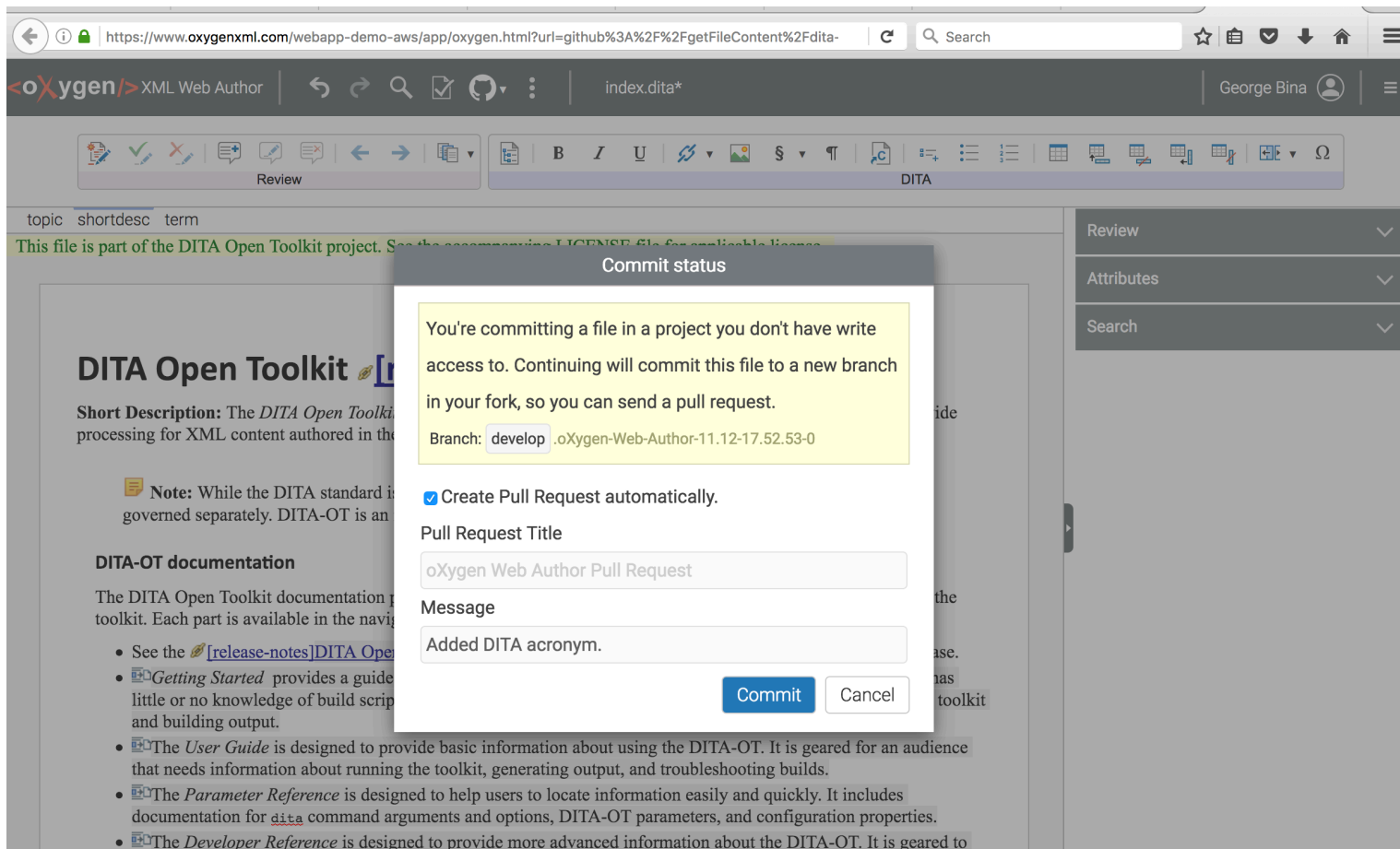
The DITA Open Toolkit documentation provides information about installing, running, configuring and extending the toolkit. Each part is available in the navigation panel.

- See the [DITA Open Toolkit 2.3.3 Release Notes](#) for information on the changes in the current release.
- *Getting Started* provides a guided exploration of the DITA Open Toolkit. It is geared for an audience that has little or no knowledge of build scripts or DITA-OT parameters. It walks the novice user through installing the toolkit and building output.
- The *User Guide* is designed to provide basic information about using the DITA-OT. It is geared for an audience that needs information about running the toolkit, generating output, and troubleshooting builds.
- The *Parameter Reference* is designed to help users to locate information easily and quickly. It includes documentation on command arguments and options, DITA-OT parameters, and configuration properties.
- The *Developer Reference* is designed to provide more advanced information about the DITA-OT. It is geared for an audience that needs information about the DITA-OT architecture, extending the DITA-OT, and creating DITA-OT plug-ins.

DITA Open Toolkit is released under the Apache License, Version 2.0. [Edit this page.](#)



# Save to send a request for change



# XML-based editable websites

<http://xproc.org>

- Based on DocBook as source format
- DocBook source available on GitHub

<http://georgebina.github.io/ghd-wiki>

- Based on DITA, Lightweight DITA and Markdown
- Source available on GitHub

# xproc.org website

## XML Processing

Home Implementations Library Extensions

Overview | Background | FAQ | Contact us | «Edit this page on GitHub»

### Welcome to XProc.org

XProc.org is a website about *XProc: An XML Pipeline Language*, a specification under development at the [W3C](#), and its use, implementation, and extensions.

### What is it?

XProc is a language designed for describing operations to be performed on XML documents. The official specification for XProc is [XProc: An XML Pipeline Language](#). It is a [Recommendation](#).

The XProc specification was produced by the [XML Processing Model Working Group](#). The XProc WG operated in the public, anyone is free to read [the archives](#) of its mailing list.

While the WG was in operation, public comments on the specification could be sent to the [processing model comments](#) mailing list. [The archives](#) of the comments list are also open to the public. Now that the WG has been disbanded, readers are encouraged to submit comments to the community through the GitHub issue tracker.

### Why did you do it?

XProc is designed to address the common problem of how to compose XML processes. Many document processing scenarios involve some combination of XML technologies; canonical examples include XInclude, schema validation, and transformation.

Although it is possible to combine these technologies using general purpose tools such as [make](#) and [ant](#) (to name only two), these tools are not designed to deal specifically with the semantics of XML processing. As such they are often both more complicated and less useful than would be ideal

XProc has been designed specifically to allow authors to compose XML processes and share these compositions in a standard way.

# Access to the DocBook source

The screenshot shows the Oxygen XML Web Author interface. The top bar includes the Oxygen logo, "XML Web Author", navigation icons, the file name "index.xml", and the user name "George Bina". Below the top bar is a toolbar with two sections: "Review" (containing icons for undo, redo, search, and review) and "DocBook" (containing various XML editing icons). The main content area displays an article titled "Article: XProc.org". The article text includes an introduction, "Section 1: What is it?", and "Section 2: Why did you do it?". On the right side, there is a "Review" sidebar with a "Filter" input, a message "To add a comment, select some content and press [comment icon] from the toolbar.", and two expandable sections: "Attributes" and "Validation" (which has a green checkmark).

article info title

## Article: XProc.org

XProc.org is a website about *XProc: An XML Pipeline Language*, a specification under development at the [W3C](#), and its use, implementation, and extensions.

### Section 1: What is it?

XProc is a language designed for describing operations to be performed on XML documents. The official specification for XProc is *XProc: An XML Pipeline Language*. It is a [Recommendation](#).

The XProc specification was produced by the [XML Processing Model Working Group](#). The XProc WG operated in the public, anyone is free to read [the archives](#) of its mailing list.

While the WG was in operation, public comments on the specification could be sent to the [processing model comments](#) mailing list. [The archives](#) of the comments list are also open to the public. Now that the WG has been disbanded, readers are encouraged to submit comments to the community through the [GitHub](#) issue tracker.

### Section 2: Why did you do it?

XProc is designed to address the common problem of how to compose XML processes. Many document processing scenarios involve some combination of XML technologies; canonical examples include [XInclude](#), schema validation, and transformation.

Review

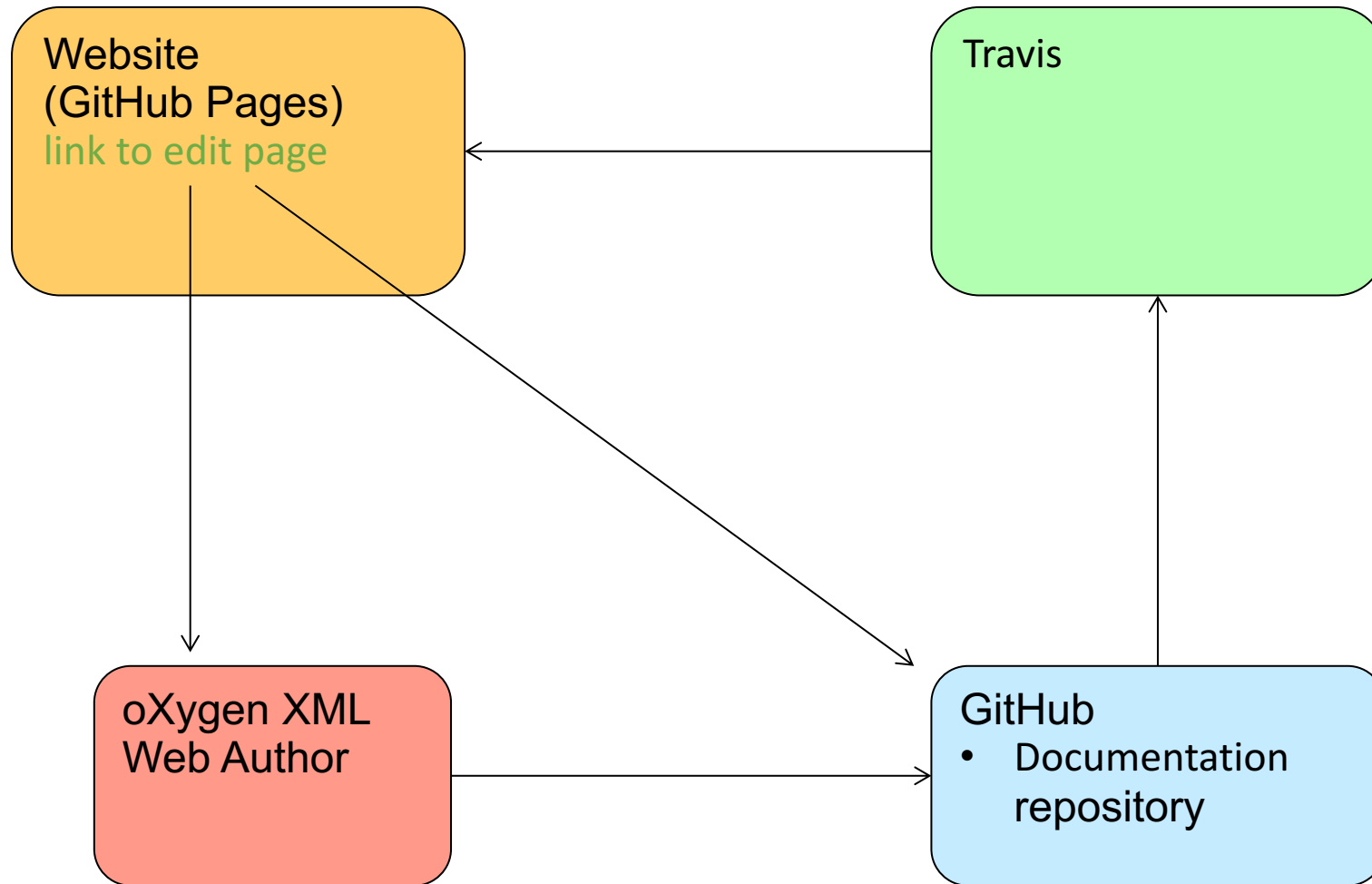
Filter

To add a comment, select some content and press from the toolbar.

Attributes

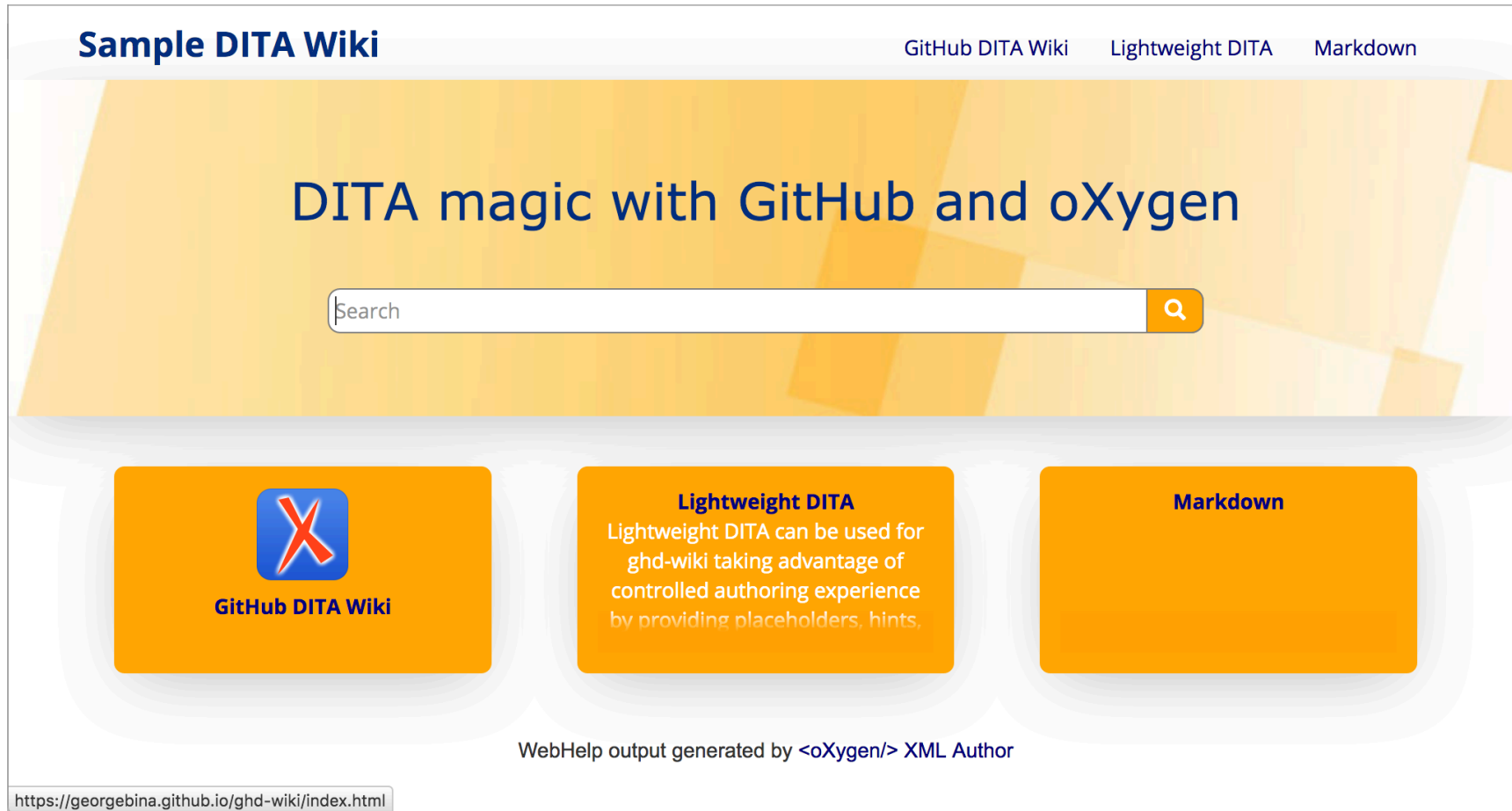
Validation

# DITA+Markdown based Wiki




# Website content


<https://georgebina.github.io/ghd-wiki/>

A screenshot of a web browser showing the "Sample DITA Wiki" page. The page has a blue header with the title "Sample DITA Wiki" and navigation links for "GitHub DITA Wiki", "Lightweight DITA", and "Markdown". The main content area has a yellow background with the text "DITA magic with GitHub and oXygen" and a search bar. Below this are three orange boxes: "GitHub DITA Wiki" with the oXygen logo, "Lightweight DITA" with a description, and "Markdown". At the bottom, it says "WebHelp output generated by <oXygen/> XML Author".

**Sample DITA Wiki**      GitHub DITA Wiki    Lightweight DITA    Markdown

## DITA magic with GitHub and oXygen

  
**GitHub DITA Wiki**

**Lightweight DITA**  
Lightweight DITA can be used for ghd-wiki taking advantage of controlled authoring experience by providing placeholders, hints.

**Markdown**

WebHelp output generated by <oXygen/> XML Author

<https://georgebina.github.io/ghd-wiki/index.html>



# Lightweight DITA topic

## Sample DITA Wiki

[GitHub DITA Wiki](#)

[Lightweight DITA](#)

[Markdown](#)



[Home](#) / [Lightweight DITA](#)



## Lightweight DITA

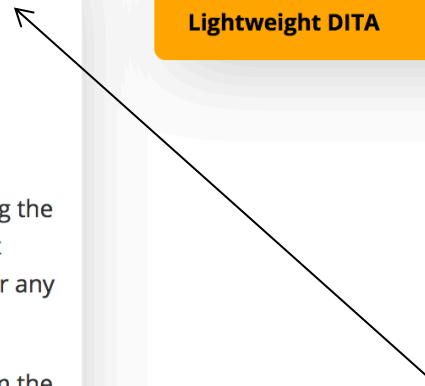
[History](#) | [Edit](#)

Lightweight DITA

Lightweight DITA can be used for **ghd-wiki** taking advantage of controlled authoring experience by providing placeholders, hints, inline actions and inline form controls.




Lightweight DITA is a great alternative to Markdown, having the same or less complexity degree while being very clearly specified. The user interface takes advantage of controlling the authoring experience through placeholders, hints, inline actions and form controls to edit attribute values in place, all these combined to provide a simple user interface suitable for any contributors.

It is very easy to create content in LW-DITA, especially if you can benefit of some help from the authoring tool. The **oxygen XML Web Author** is an example of such a tool, that provides special support for working with LW-DITA, support that can be extended further to match your actual needs.



History and Edit links

# Revisions history







This repository  [Pull requests](#) [Issues](#) [Gist](#)   


[georgebina](#) / [ghd-wiki](#) [Unwatch](#) 1 [Star](#) 0 [Fork](#) 5

[Code](#) [Issues](#) 0 [Pull requests](#) 0 [Projects](#) 0 [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)

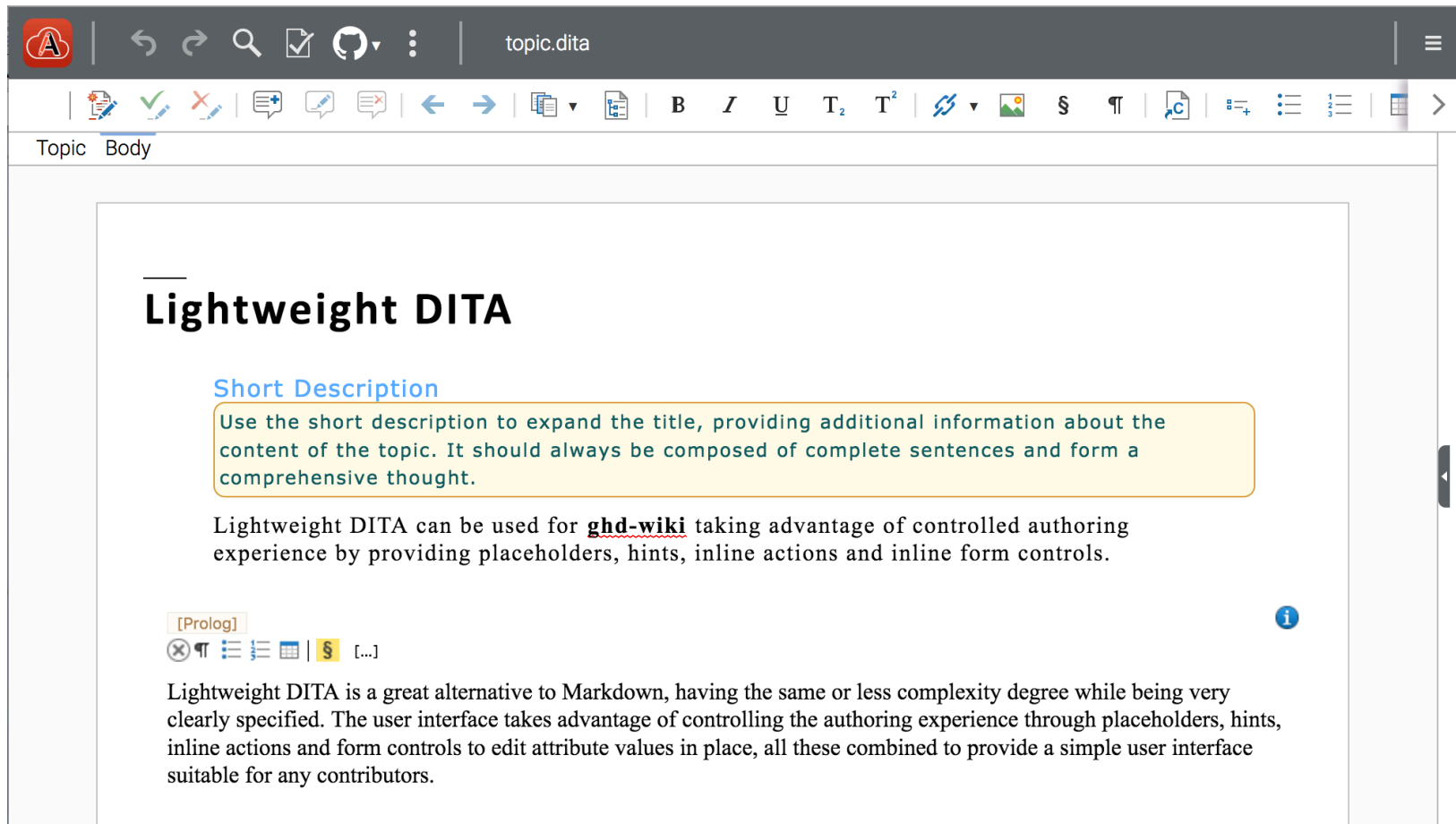
History for [ghd-wiki](#) / [wiki](#) / [topic.dita](#)

Commits on Oct 16, 2016

	<b>Committing with oXygen Web Author</b> georgebina committed 15 hours ago ✓	 <a href="#">34569e6</a> <a href="#">Code</a>
	<b>Make short description shorter.</b> georgebina committed 15 hours ago ✓	 <a href="#">2e7121b</a> <a href="#">Code</a>
	<b>Updated content.</b> georgebina committed 20 hours ago ✓	 <a href="#">060e34e</a> <a href="#">Code</a>

© 2016 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

# Controlled AX online editor

A screenshot of the Oxygen XML Editor interface. The top bar shows the file name "topic.dita" and various navigation icons. Below the top bar is a rich text editor toolbar with icons for undo, redo, search, and text formatting (bold, italic, underline, subscript, superscript, link, image, dollar sign, paragraph, list, table). The main editing area is titled "Topic Body" and contains the following content:

**Lightweight DITA**

**Short Description**

Use the short description to expand the title, providing additional information about the content of the topic. It should always be composed of complete sentences and form a comprehensive thought.

Lightweight DITA can be used for ghd-wiki taking advantage of controlled authoring experience by providing placeholders, hints, inline actions and inline form controls.



[Prolog]


⊗ ¶ ☰ ☷ | \$ [...] ⓘ


Lightweight DITA is a great alternative to Markdown, having the same or less complexity degree while being very clearly specified. The user interface takes advantage of controlling the authoring experience through placeholders, hints, inline actions and form controls to edit attribute values in place, all these combined to provide a simple user interface suitable for any contributors.

























# Travis project status

<https://travis-ci.org/georgebina/ghd-wiki/builds>

Travis CI  Blog Status Help George Bina 

georgebina / ghd-wiki  build passing

Current Branches Build History Pull Requests More options 

 master  Radu Coravu	Update README.md	 #73 passed  da8cf48	 46 sec  about 14 hours ago
 master  George Bina	Add link to project home on GitHub.	 #72 passed  8cd3646	 52 sec  about 14 hours ago
 master  George Bina	Committing with oxygen Web Author	 #71 passed  34569e6	 53 sec  about 15 hours ago
 master  George Bina	Make short description shorter.	 #70 passed  2e7121b	 1 min 8 sec  about 15 hours ago

# Take-aways

oXygen Content Fusion allows you to share your XML document with people you want to collaborate with and integrate their changes easily

oXygen Web Author can be used as a service to simplify existing workflows or to enable new ones, such as:

- Link support to documentation
- Get developers to review relevant documentation
- Simplify the review process

Any XML-based system can be a wiki-like system where anyone can immediately interact with the content

# Thank you

## Questions?



george@oxygenxml.com



@georgebina



<http://www.oxygenxml.com>

# What oXygen products solve

Development    Authoring    Publishing    Integration

Web Authoring

Collaboration and Review

# What oXygen products solve

Developer

Author

Development

Authoring

Publishing

Integration

Web Authoring

Collaboration and Review



# What oXygen products solve

Editor

Developer

Author

Development

Authoring

Publishing

Integration

Web Authoring

Collaboration and Review

# What oXygen products solve

Editor

Chemistry

Developer

Author

Webhelp

Development

Authoring

Publishing

Integration

Web Authoring

Collaboration and Review

# What oXygen products solve

Editor

Chemistry

Developer

Author

Webhelp

Scripting

Development

Authoring

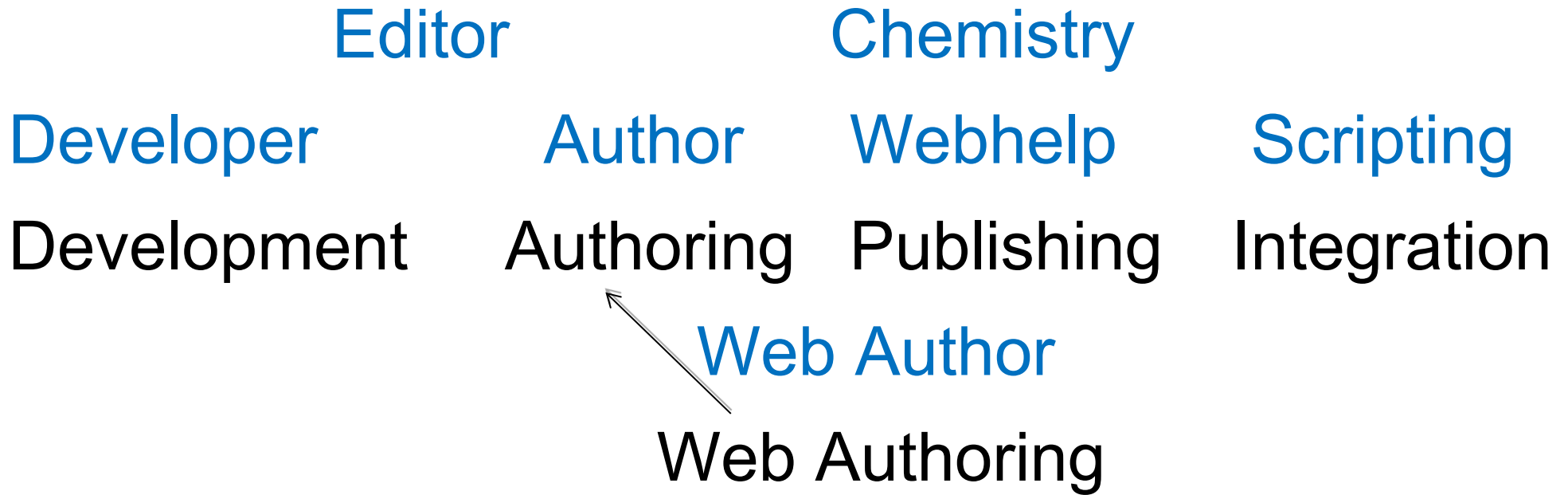
Publishing

Integration

Web Authoring

Collaboration and Review

# What oXygen products solve



Collaboration and Review

# What oXygen products solve

