Docs as Code

Documentation Management Inspired by Software Development

Alex Jitianu

alex_jitianu@oxygenxml.com

@AlexJitianu

Agenda

- What is *Docs as Code*
- Choosing a text markup language
- Version Control
- Continuous integration and delivery
- Collaboration and review
- *A working Docs as Code setup*

1. Docs as Code

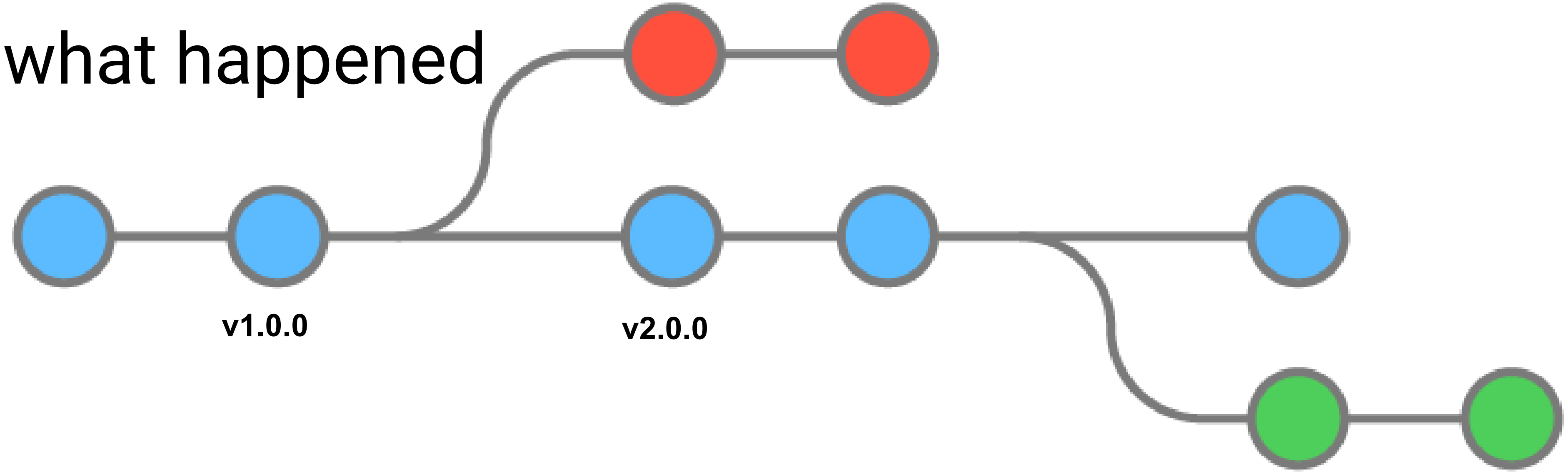
- Refers to a philosophy that you should be writing documentation with the same **tools** and **workflows** as code:
  - Version Control
  - Collaboration and Review Process
  - Automated Tests, Builds, Delivery
  - Issue Trackers
  - Plain Text Markup

2. Choosing a Suitable Text Markup

- Markdown
- Asciidoc
- ***XML***
  - ***Docbook***
  - ***DITA***

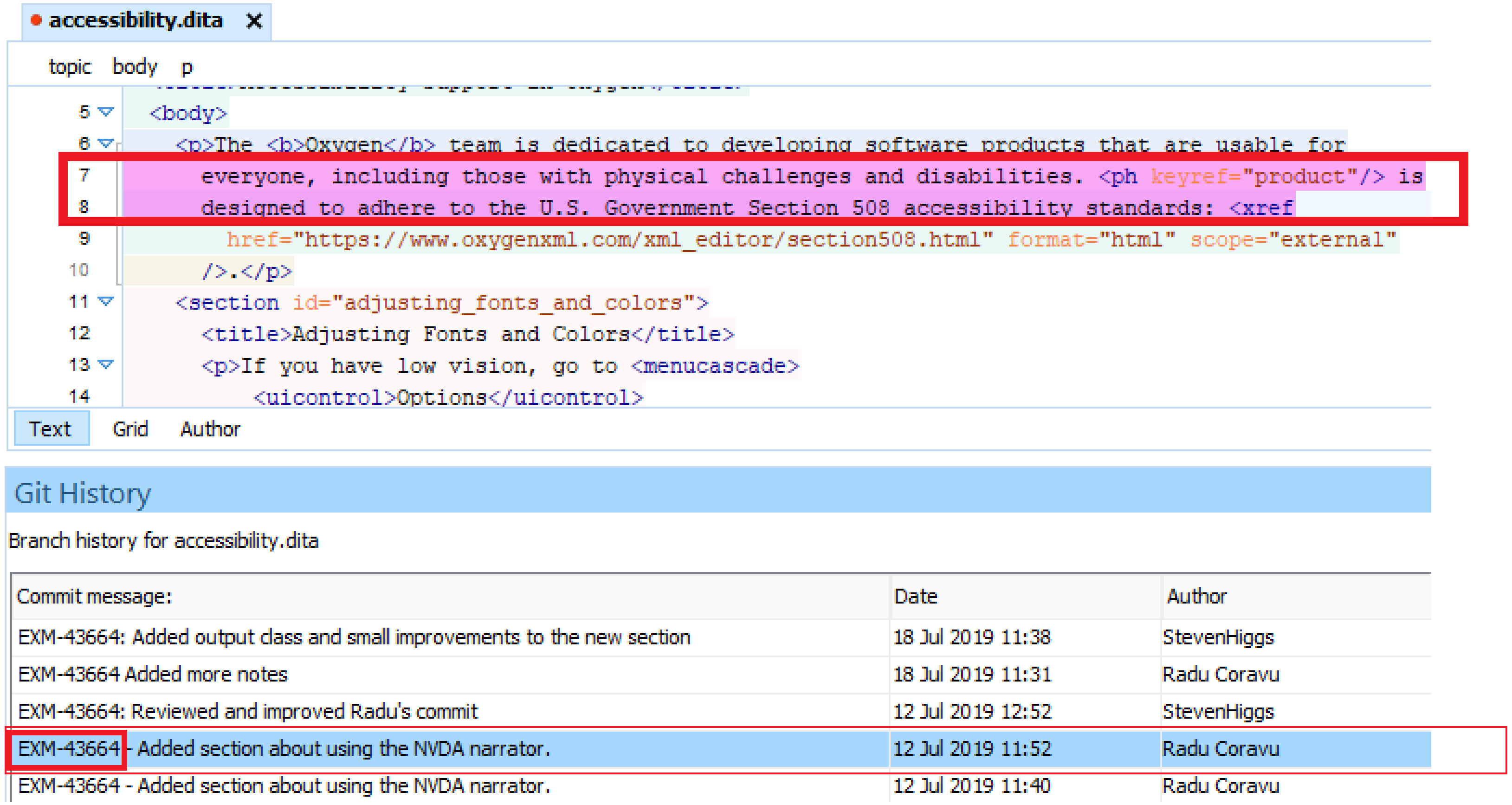
3. Why should I use a Version Control?

- Storing versions
  - *The basics of version control is the ability to save changes made to files, while retaining the changes from all previous versions.*
- Collaboration and review
- Understand what happened



Why should I use a Version Control?

- Understand what happened



```
  <body>
    <p>The <b>Oxygen</b> team is dedicated to developing software products that are usable for
      everyone, including those with physical challenges and disabilities. <ph keyref="product"/> is
      designed to adhere to the U.S. Government Section 508 accessibility standards: <xref
        href="https://www.oxygenxml.com/xml_editor/section508.html" format="html" scope="external"
      />.</p>
    <section id="adjusting_fonts_and_colors">
      <title>Adjusting Fonts and Colors</title>
      <p>If you have low vision, go to <menucascade>
          <uicontrol>Options</uicontrol>
```

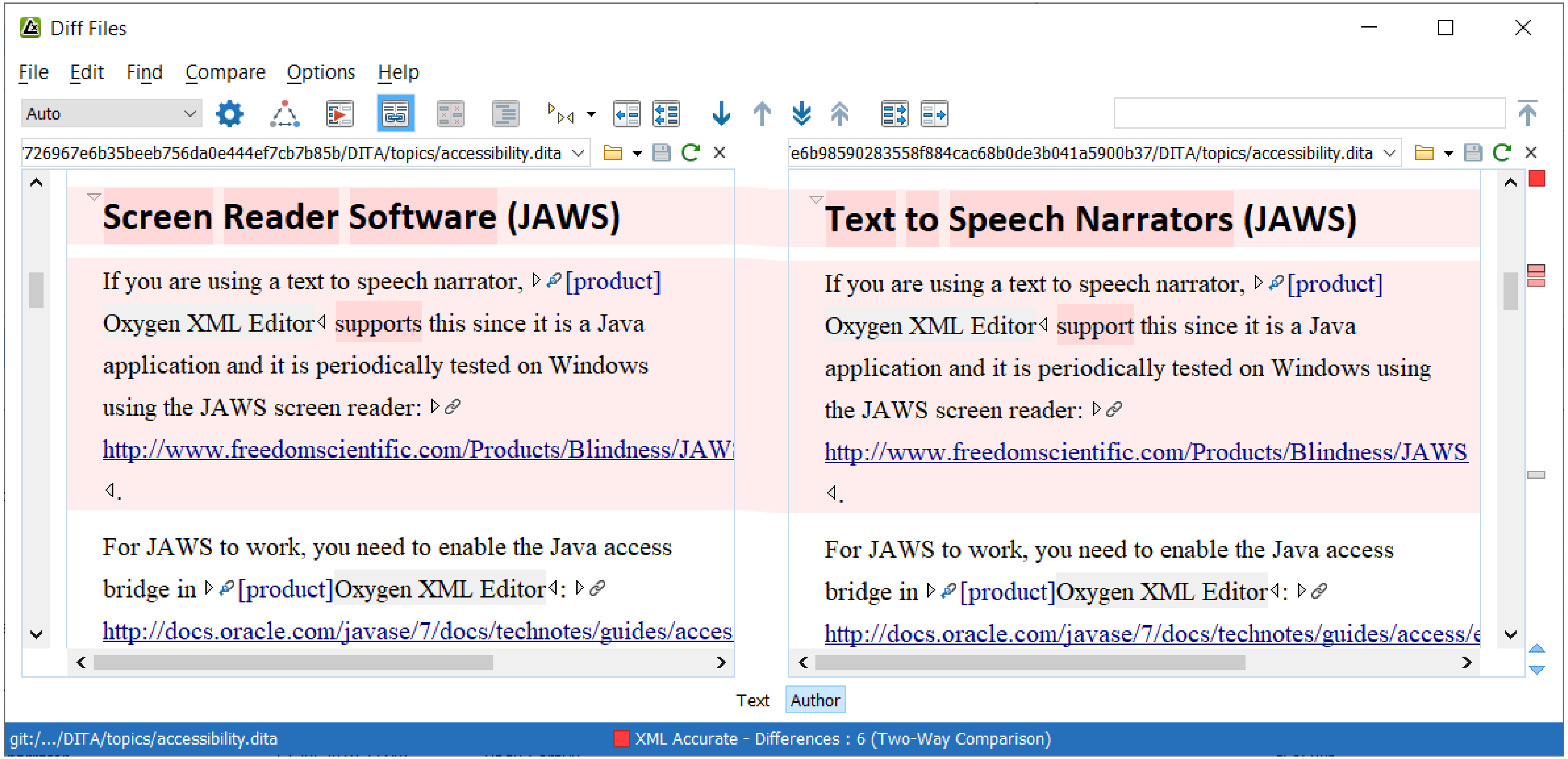
Git History

Branch history for accessibility.dita

Commit message:	Date	Author
EXM-43664: Added output class and small improvements to the new section	18 Jul 2019 11:38	StevenHiggs
EXM-43664 Added more notes	18 Jul 2019 11:31	Radu Coravu
EXM-43664: Reviewed and improved Radu's commit	12 Jul 2019 12:52	StevenHiggs
EXM-43664 - Added section about using the NVDA narrator.	12 Jul 2019 11:52	Radu Coravu
EXM-43664 - Added section about using the NVDA narrator.	12 Jul 2019 11:40	Radu Coravu

Why should I use a Version Control?

- Understand what happened



726967e6b35beeb756da0e444ef7cb7b85b/DITA/topics/accessibility.dita

e6b98590283558f884cac68b0de3b041a5900b37/DITA/topics/accessibility.dita

Screen Reader Software (JAWS)

If you are using a text to speech narrator, [product] Oxygen XML Editor supports this since it is a Java application and it is periodically tested on Windows using the JAWS screen reader: http://www.freedomscientific.com/Products/Blindness/JAW .

For JAWS to work, you need to enable the Java access bridge in [product]Oxygen XML Editor: http://docs.oracle.com/javase/7/docs/technotes/guides/acces

Text to Speech Narrators (JAWS)

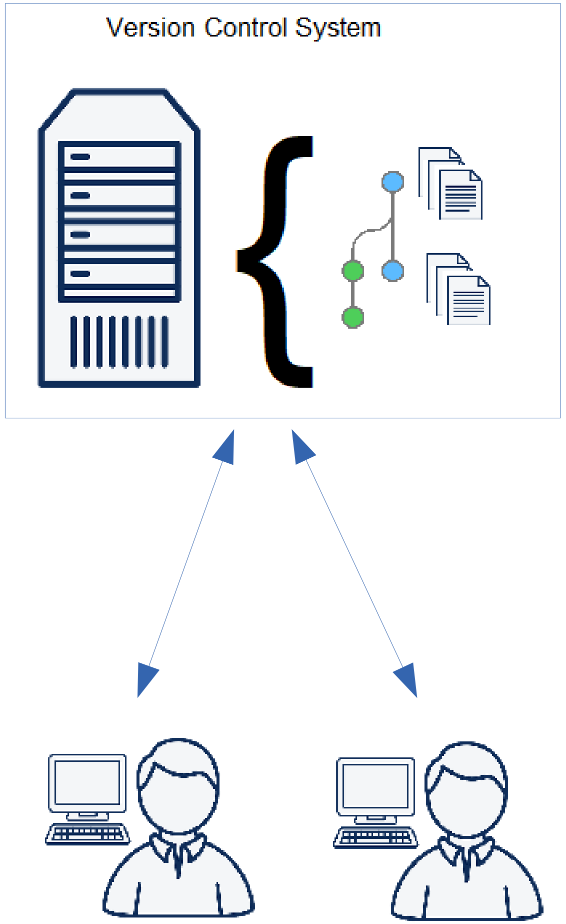
If you are using a text to speech narrator, [product] Oxygen XML Editor support this since it is a Java application and it is periodically tested on Windows using the JAWS screen reader: http://www.freedomscientific.com/Products/Blindness/JAWS .

For JAWS to work, you need to enable the Java access bridge in [product]Oxygen XML Editor: http://docs.oracle.com/javase/7/docs/technotes/guides/access/e

git:/.../DITA/topics/accessibility.dita

XML Accurate - Differences : 6 (Two-Way Comparison)

Why should I use a Version Control?



Which Version Control System?

- Git
- Mercurial
- Subversion

In the Cloud or on Premise

- On premise Git repositories
  - *GitLab Community Edition (CE) is an **open source** end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more. Self-host GitLab CE on your own servers, in a container, or on a cloud provider.*
- Web-based Git repositories
  - GitLab [https://about.gitlab.com/pricing]
  - GitHub [https://github.com/pricing]
  - Bitbucket [https://bitbucket.org/product/pricing]

4. Continuous Integration (CI)

- Changes are validated as soon as they are committed, by creating a build and running *automated tests*.
  - *Helps avoid "integration hell" where the software works on individual developers' machines, but it fails when all developers combine (or "integrate") their code*
- Puts a great emphasis on *testing automation.*

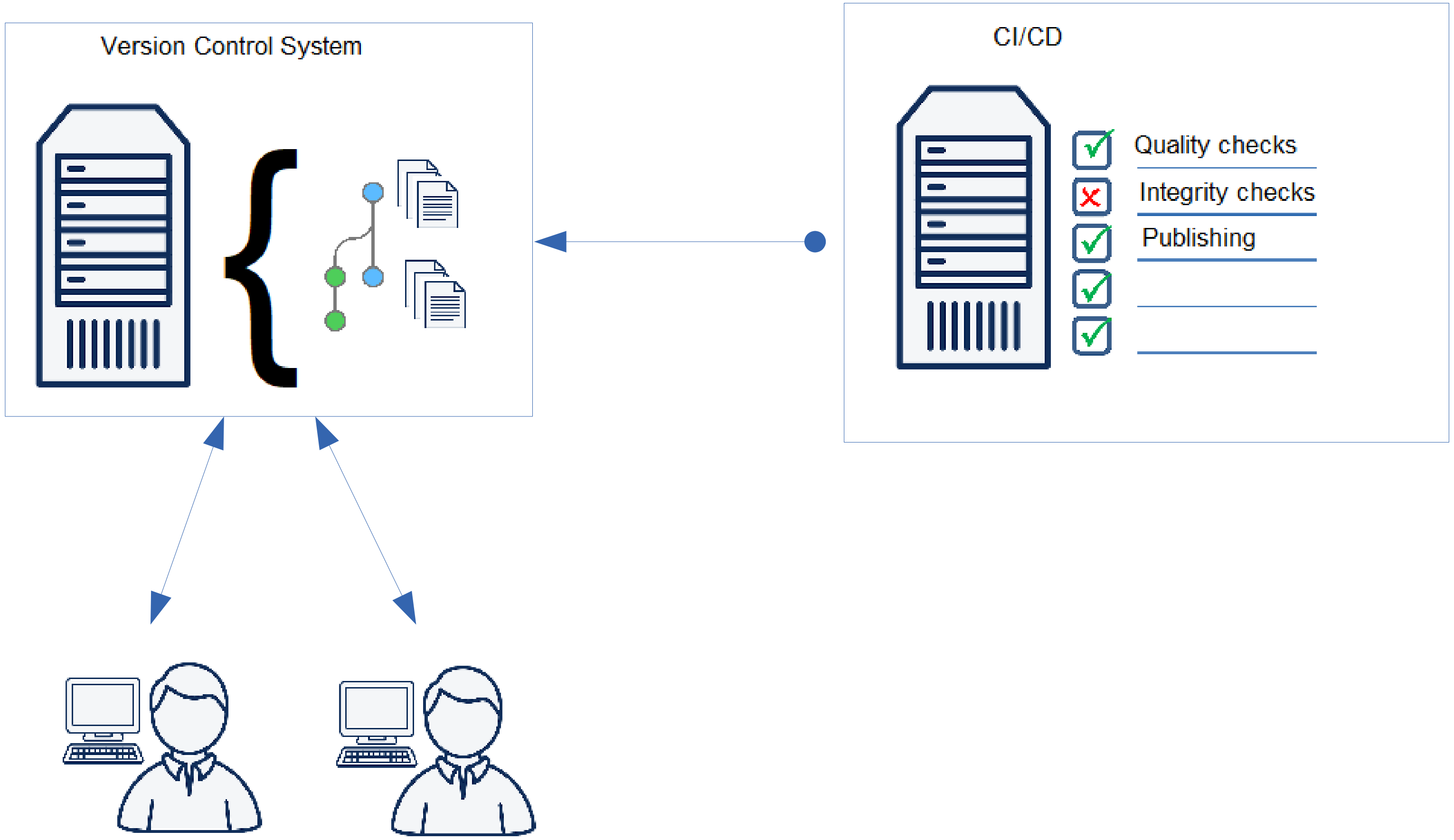
What can we automate for a documentation project?

- Quality checks
  - Business rules (Schematron, Vale)
  - Integrity checks (Validate and Check for Completeness)
- Reuse metrics
- Publishing pipelines
- Translation packages

4. Continuous Delivery (CD)

- The goal of CD is to make sure the software is always ready to go to production
  - You have automated your release process
  - You can deploy your application at any point of time by clicking on a button.

CI/CD


Version Control System
CI/CD
Quality checks
Integrity checks
Publishing

CI/CD Platforms/Servers

- Jenkins
- GitLab CI/CD
- Travis CI
- Netlify

On Premise

Cloud-based

5. Collaboration

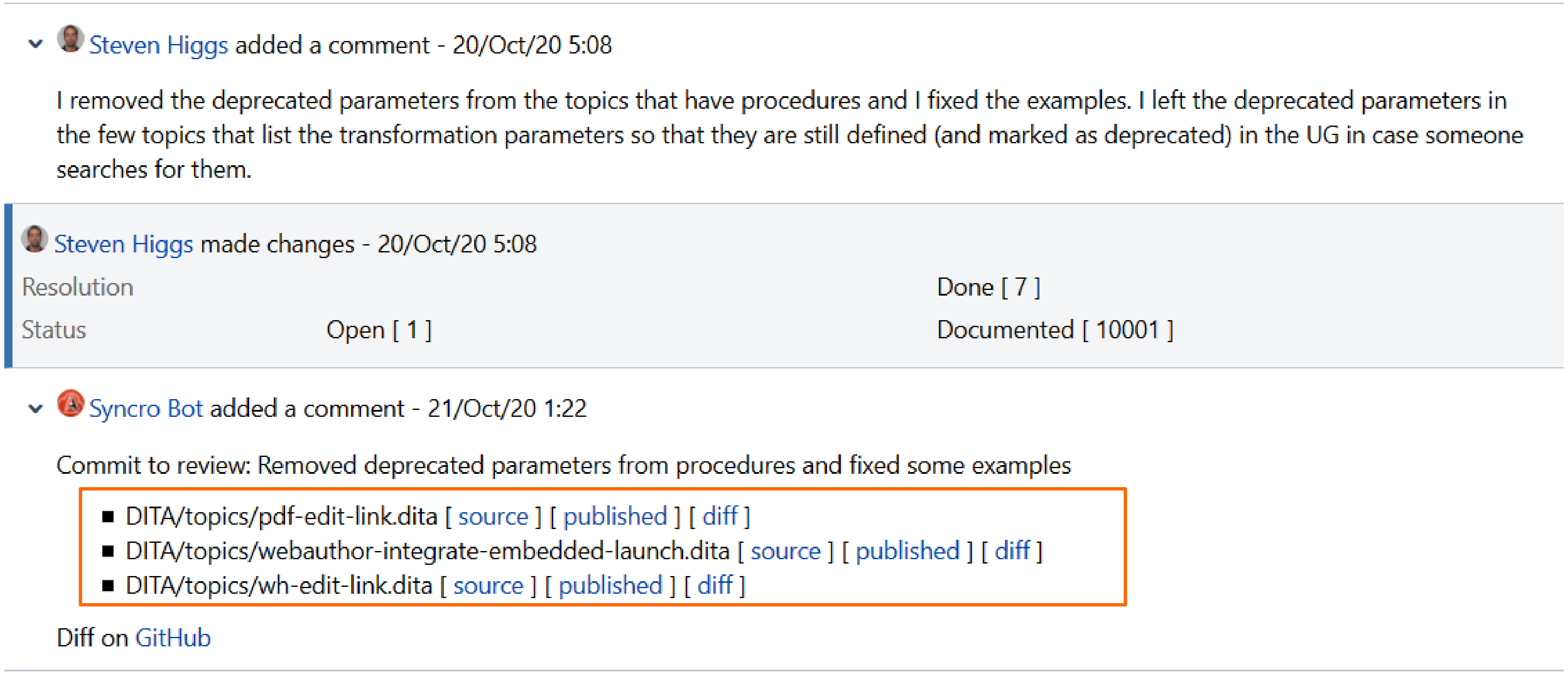
- Using version control (Git)
- GitHub pull requests
  - https://github.com/features/code-review/
  - https://github.com/oxygenxml/blog/pull/25
- Using dedicated solutions (Oxygen Content Fusion, Oxygen XML Web Author)

Issue Tracking

- GitHub Issue Tracker
  - https://github.com/oxygenxml/userguide/issues
- GitLab Issue Tracker
  - https://gitlab.com/jitianualex83/my-test-project/issues
- Atlassian Jira
  - https://www.atlassian.com/ro/software/jira

Issue Tracking

- One issue to track software and doc changes (*software documentation)*



Steven Higgs added a comment - 20/Oct/20 5:08

I removed the deprecated parameters from the topics that have procedures and I fixed the examples. I left the deprecated parameters in the few topics that list the transformation parameters so that they are still defined (and marked as deprecated) in the UG in case someone searches for them.

Steven Higgs made changes - 20/Oct/20 5:08

Resolution		Done [7]
Status	Open [1]	Documented [10001]

Syncro Bot added a comment - 21/Oct/20 1:22

Commit to review: Removed deprecated parameters from procedures and fixed some examples

- DITA/topics/pdf-edit-link.dita [source] [published] [diff]
- DITA/topics/webauthor-integrate-embedded-launch.dita [source] [published] [diff]
- DITA/topics/wh-edit-link.dita [source] [published] [diff]

Diff on GitHub

Editing tools

- Any text processor
- GitLab, GitHub built-in text editors
- Commercial XML editors
  - Oxygen Web Author
  - Oxygen XML Editor

6. Docs as Code setup

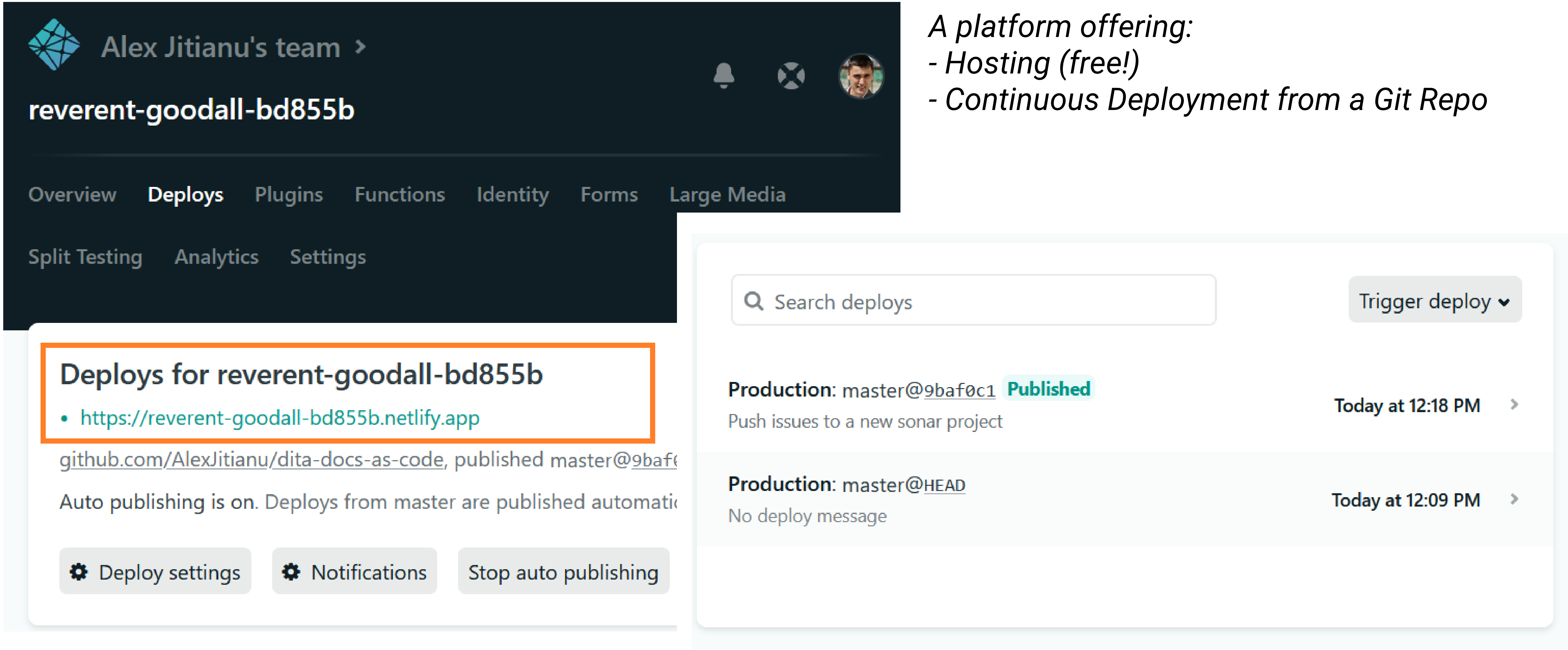
- Text Markup: *DITA + Markdown*
- Version control: *GitHub*
- Issues tracker: *GitHub*
- CI/CD : *Netlify + SonarCloud*
- Collaboration: *Oxygen Web Author* [links in published output]

What can we automate for a documentation project?

- Quality checks
  - Business rules (Schematron, Vale)
  - Integrity checks (Validate and Check for Completeness)
- Reuse metrics
- Publishing pipelines
- Translation packages (DITA Translation Package Builder)

Netlify

A platform offering:
- Hosting (free!)
- Continuous Deployment from a Git Repo



Alex Jitianu's team ›

reverent-goodall-bd855b

Overview Deploys Plugins Functions Identity Forms Large Media Split Testing Analytics Settings

Deploys for reverent-goodall-bd855b

- https://reverent-goodall-bd855b.netlify.app

github.com/AlexJitianu/dita-docs-as-code, published master@9baf

Auto publishing is on. Deploys from master are published automati

⚙ Deploy settings ⚙ Notifications Stop auto publishing

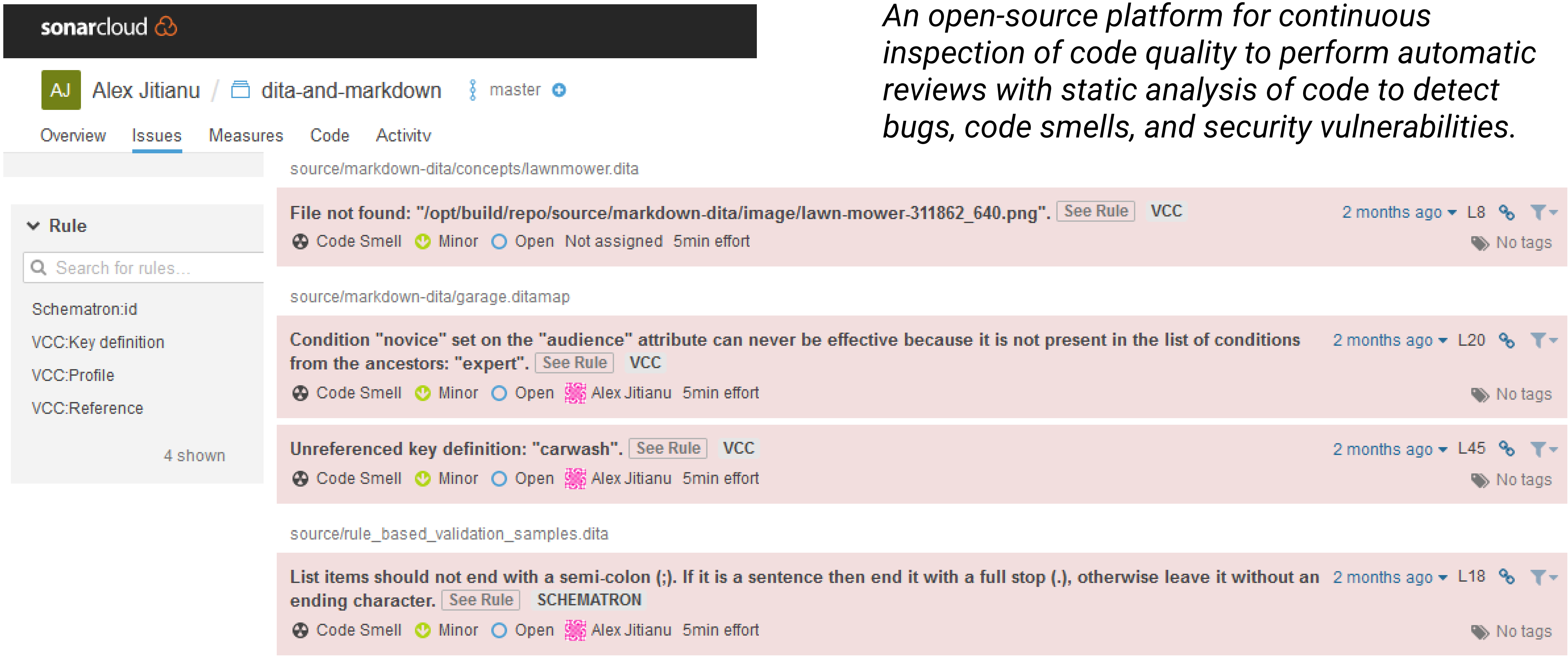
Search deploys Trigger deploy ⌄

Production: master@9baf0c1 **Published** Today at 12:18 PM ›
Push issues to a new sonar project

Production: master@HEAD Today at 12:09 PM ›
No deploy message

SonarCloud

An open-source platform for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.



sonarcloud

AJ Alex Jitianu / dita-and-markdown master

Overview Issues Measures Code Activity

Rule

Search for rules...

Schematron:id
VCC:Key definition
VCC:Profile
VCC:Reference

4 shown

source/markdown-dita/concepts/lawnmower.dita

File not found: "/opt/build/repo/source/markdown-dita/image/lawn-mower-311862_640.png". See Rule VCC — 2 months ago L8
Code Smell Minor Open Not assigned 5min effort — No tags

source/markdown-dita/garage.ditamap

Condition "novice" set on the "audience" attribute can never be effective because it is not present in the list of conditions from the ancestors: "expert". See Rule VCC — 2 months ago L20
Code Smell Minor Open Alex Jitianu 5min effort — No tags

Unreferenced key definition: "carwash". See Rule VCC — 2 months ago L45
Code Smell Minor Open Alex Jitianu 5min effort — No tags

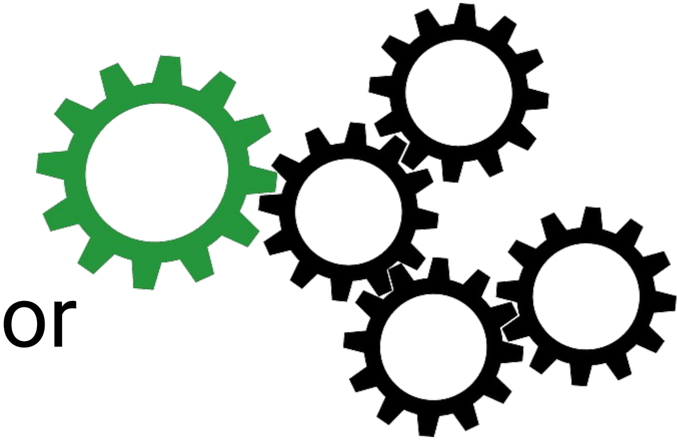
source/rule_based_validation_samples.dita

List items should not end with a semi-colon (;). If it is a sentence then end it with a full stop (.), otherwise leave it without an ending character. See Rule SCHEMATRON — 2 months ago L18
Code Smell Minor Open Alex Jitianu 5min effort — No tags

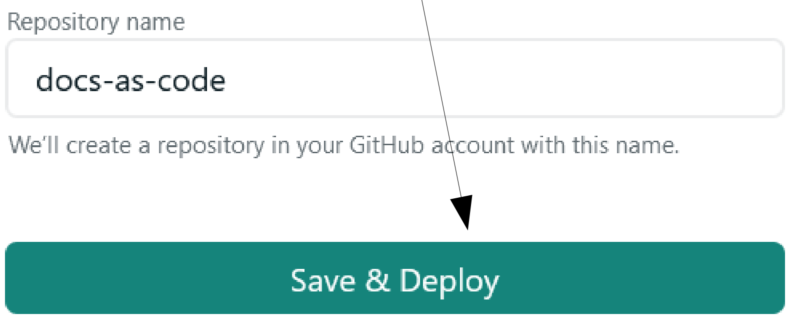
What can we automate for a documentation project?

- Quality checks
  - Business rules (Schematron, Vale)
  - Integrity checks (Validate and Check for Completeness)
- Reuse metrics
- Publishing pipelines
- Translation packages (DITA Translation Package Builder)

Hands on Time



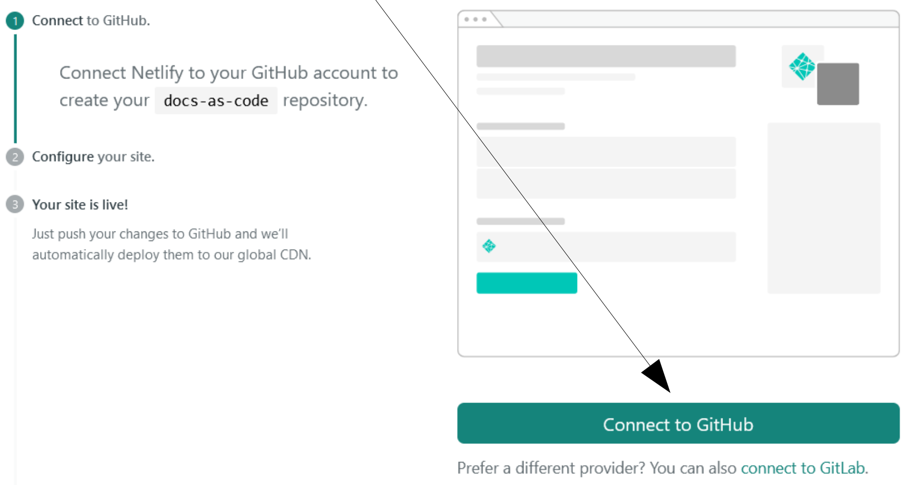
- Deploy to Netlify our *docs-as-code* solution for DITA and Markdown
  - Go to: https://github.com/AlexJitianu/docs-as-code/
  - Click the button Deploy to netlify
  - In the launched wizard, click *Connect to Github*
  - Click *Save & Deploy* to finish



Repository name

docs-as-code

We'll create a repository in your GitHub account with this name.



1 Connect to GitHub.

Connect Netlify to your GitHub account to create your `docs-as-code` repository.

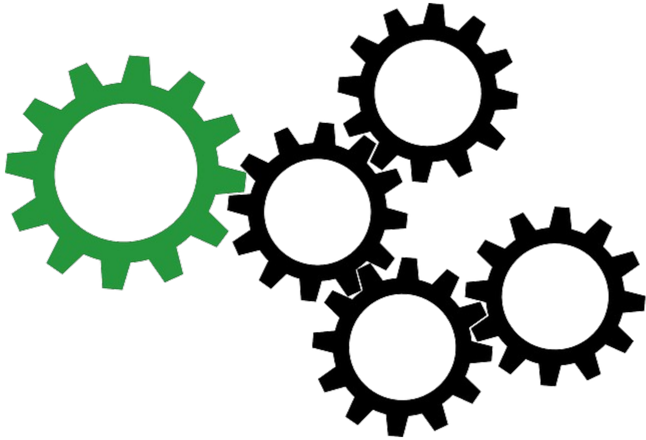
2 Configure your site.

3 Your site is live!

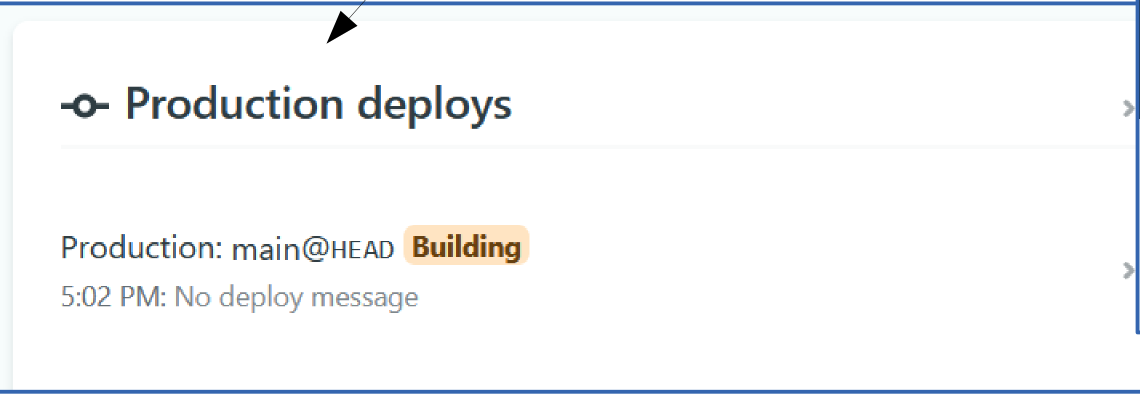
Just push your changes to GitHub and we'll automatically deploy them to our global CDN.

Prefer a different provider? You can also connect to GitLab.

Hands on Time Result

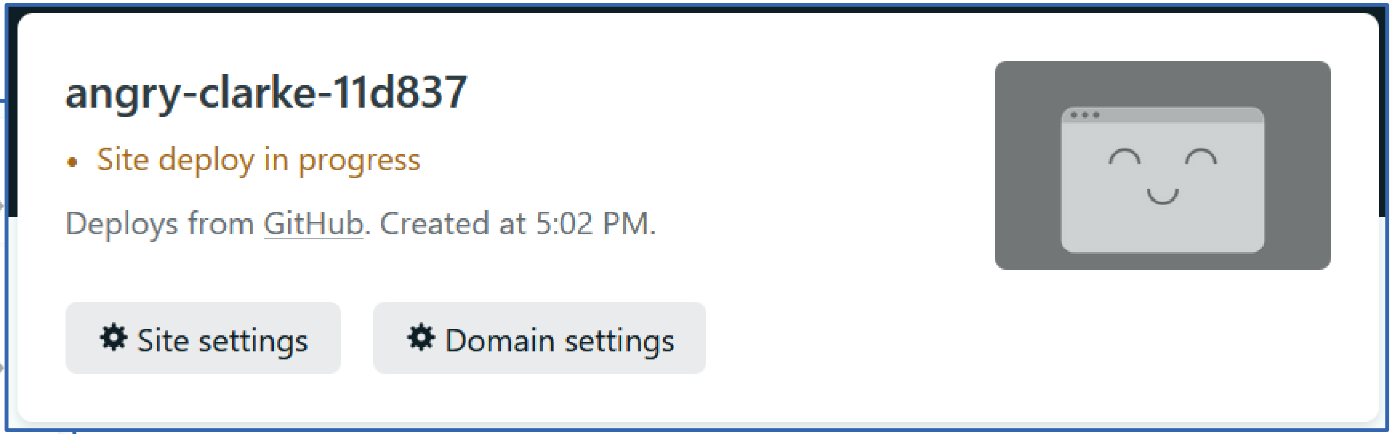


- You now have a Netlify Site accessible from *https://app.netlify.com/teams/{github-account-name}/sites*
- A first deploy is in progress



Production deploys

Production: main@HEAD Building
5:02 PM: No deploy message

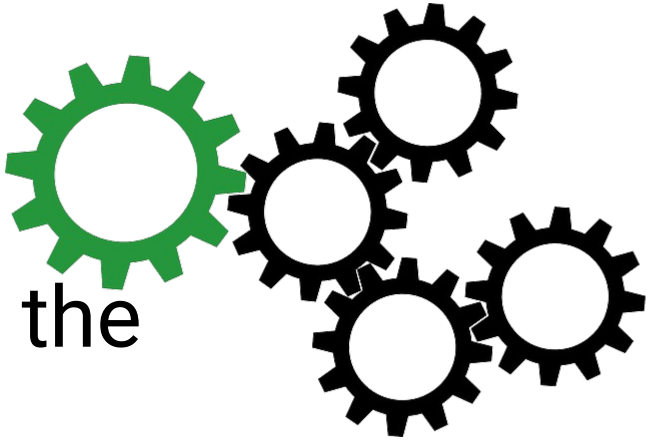


angry-clarke-11d837

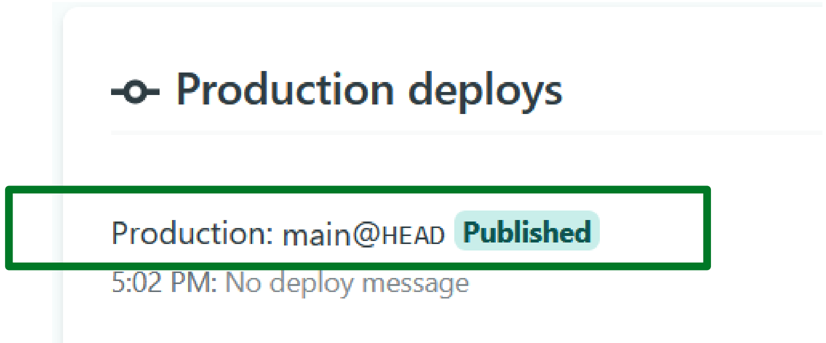
• Site deploy in progress

Deploys from GitHub. Created at 5:02 PM.

Hands on Time
Result



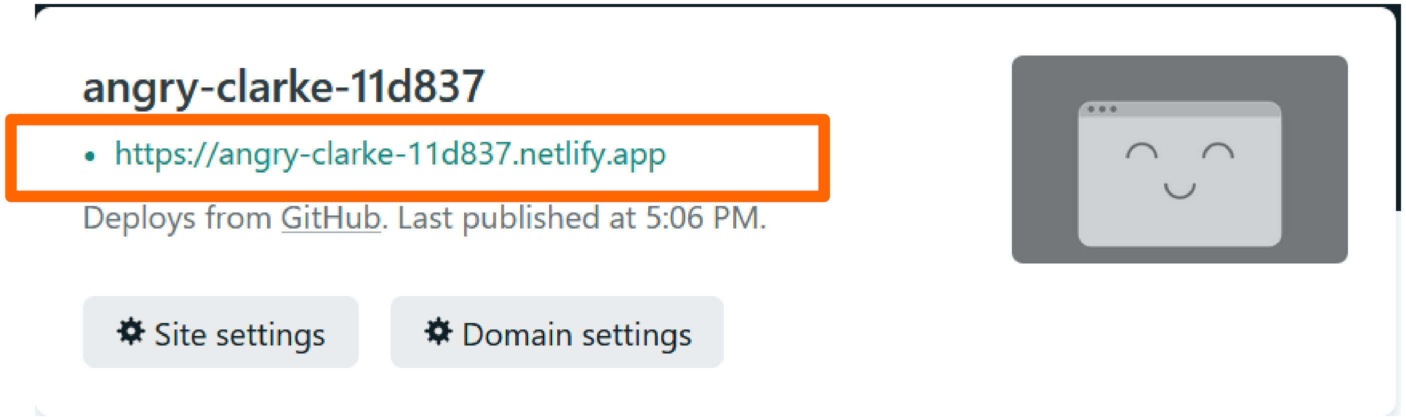
- After the deploy is Published you can inspect the results of the build



Production deploys

Production: main@HEAD Published

5:02 PM: No deploy message



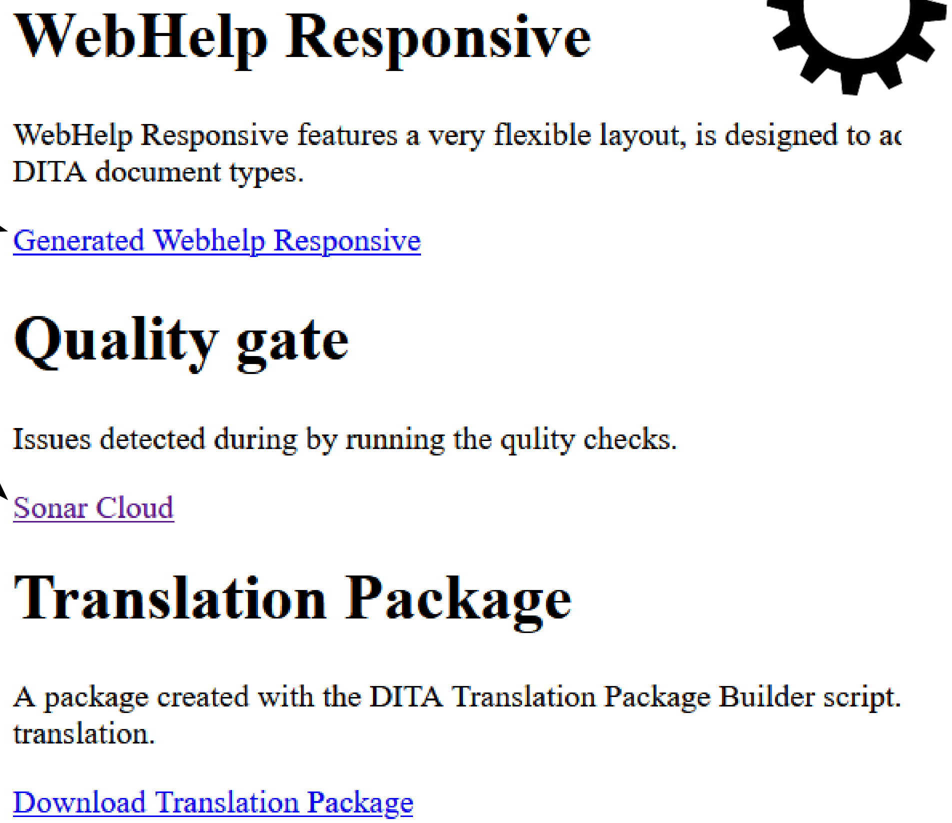
angry-clarke-11d837

- https://angry-clarke-11d837.netlify.app

Deploys from GitHub. Last published at 5:06 PM.

Hands on Time Result

- Results of the build
  - Our published content
  - The Quality Checks results
  - A package with the modified files that must be translated



WebHelp Responsive

WebHelp Responsive features a very flexible layout, is designed to ac
DITA document types.

Generated Webhelp Responsive

Quality gate

Issues detected during by running the qulity checks.

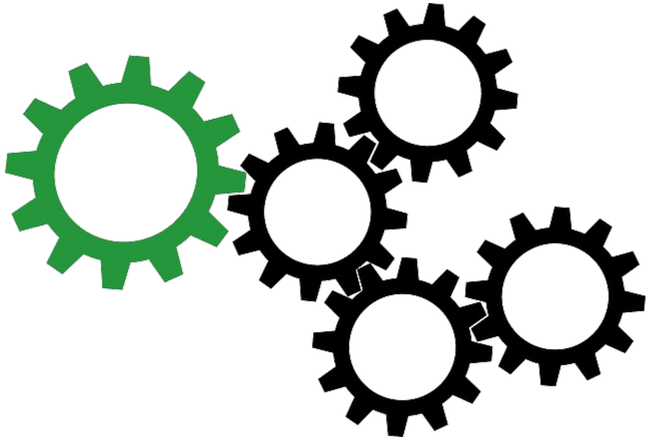
Sonar Cloud

Translation Package

A package created with the DITA Translation Package Builder script.
translation.

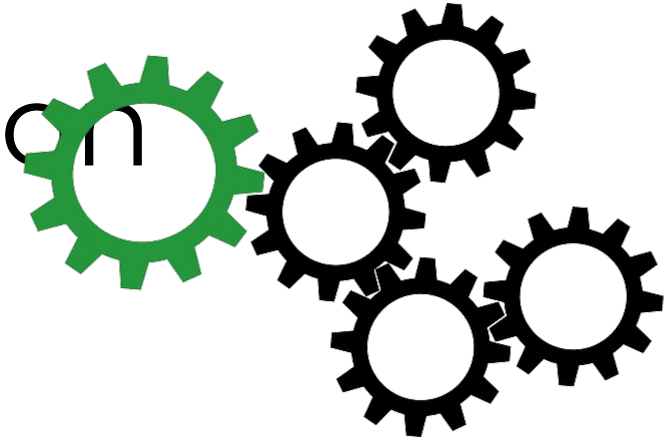
Download Translation Package

Hands on Time Sonarcloud

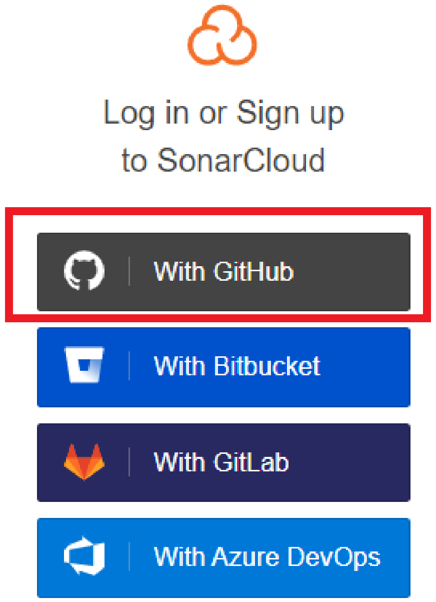


- Right now Netlify is deploying the QA results to my SonarCloud accout: https://sonarcloud.io/project/issues?id=docs-as-code&resolved=false
- We will create a new SonarCloud account and upload to it instead

SonarCloud Configuration

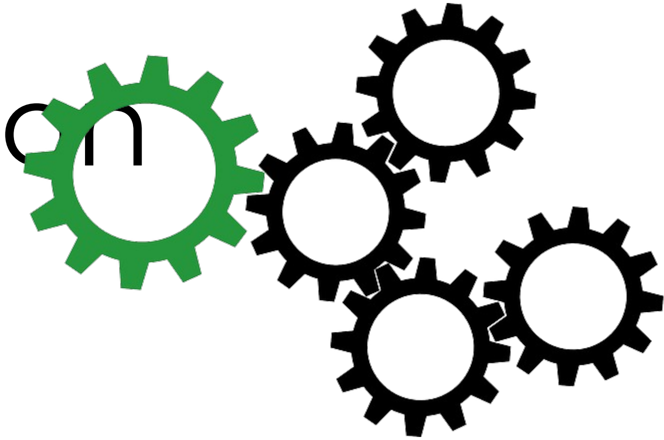


- Setup SonarCloud
  - Create account
    - Go to https://sonarcloud.io/about
    - On the top right corner, click *Log In* Log in
    - Log in with your GitHub account

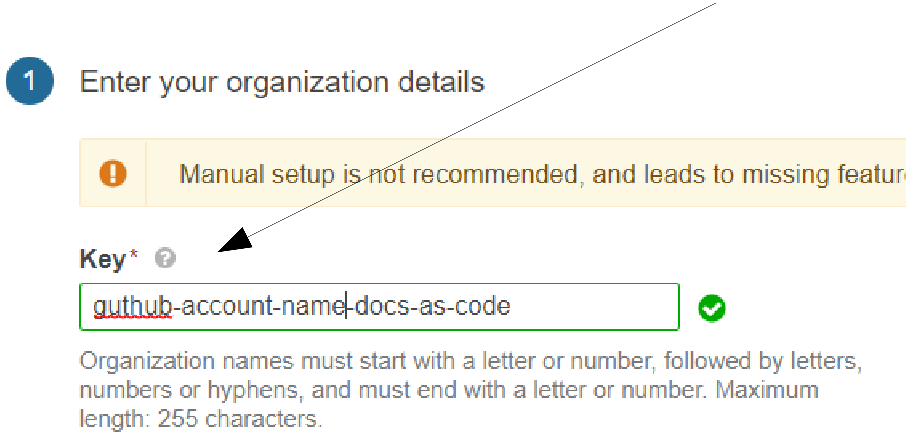


Log in or Sign up
to SonarCloud

SonarCloud Configuration



- Setup SonarCloud
  - Create organization
    - Go to https://sonarcloud.io/account/organizations
    - Click *Create* Create
    - Click create an organization manually
    - Give a unique organization name



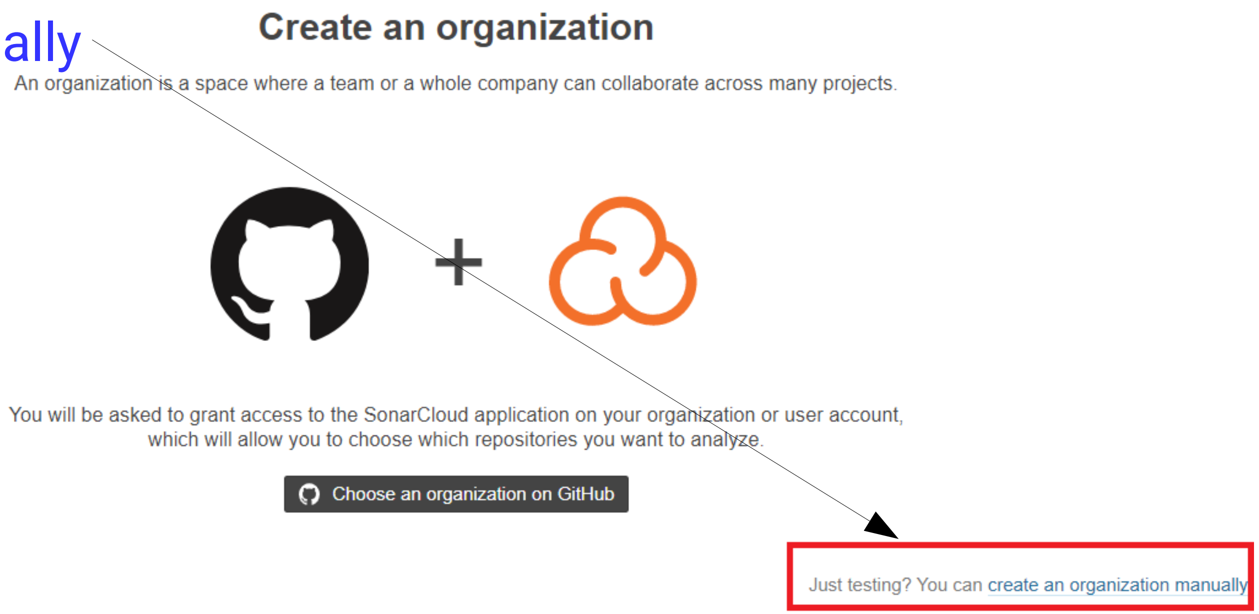
1 Enter your organization details

Manual setup is not recommended, and leads to missing featur

Key*

github-account-name-docs-as-code ✓

Organization names must start with a letter or number, followed by letters, numbers or hyphens, and must end with a letter or number. Maximum length: 255 characters.



Create an organization

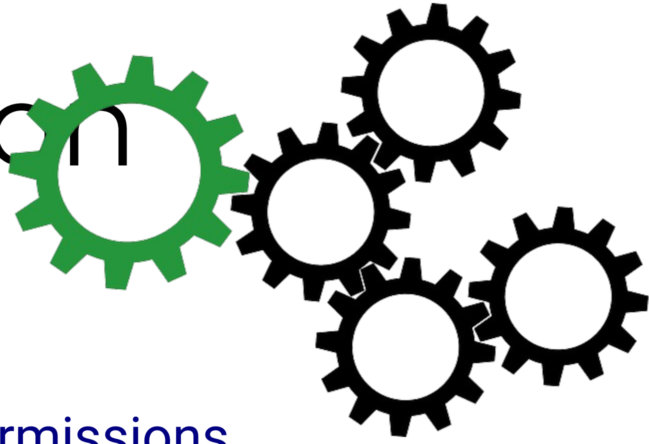
An organization is a space where a team or a whole company can collaborate across many projects.

You will be asked to grant access to the SonarCloud application on your organization or user account, which will allow you to choose which repositories you want to analyze.

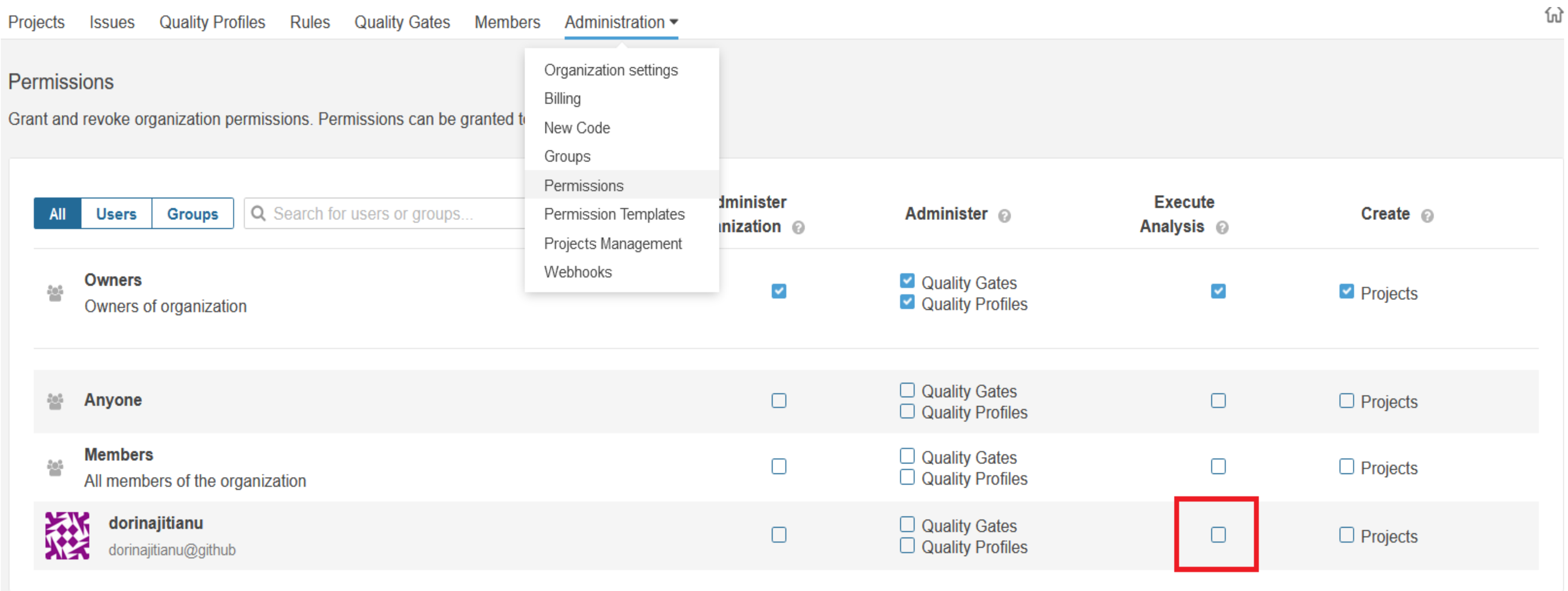
Choose an organization on GitHub

Just testing? You can create an organization manually

SonarCloud Configuration



- Set permissions (*Execute Analysis*)
  - https://sonarcloud.io/organizations/{organization-name}/permissions

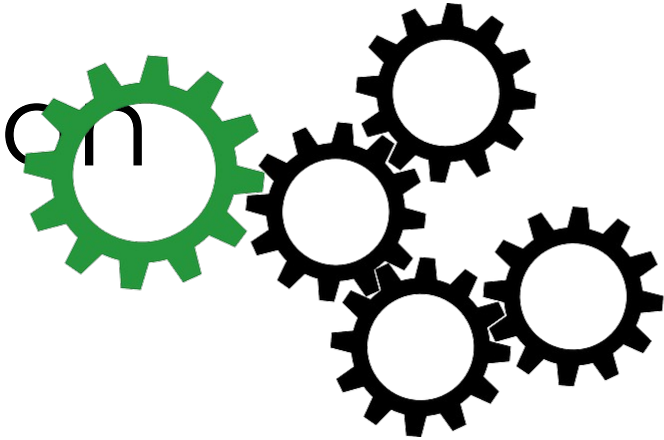


Permissions

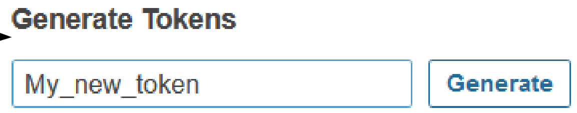
Grant and revoke organization permissions. Permissions can be granted t

	dminister nization	Administer	Execute Analysis	Create
Owners Owners of organization	☑	☑ Quality Gates ☑ Quality Profiles	☑	☑ Projects
Anyone	□	□ Quality Gates □ Quality Profiles	□	□ Projects
Members All members of the organization	□	□ Quality Gates □ Quality Profiles	□	□ Projects
dorinajitianu dorinajitianu@github	□	□ Quality Gates □ Quality Profiles	□	□ Projects

SonarCloud Configuration

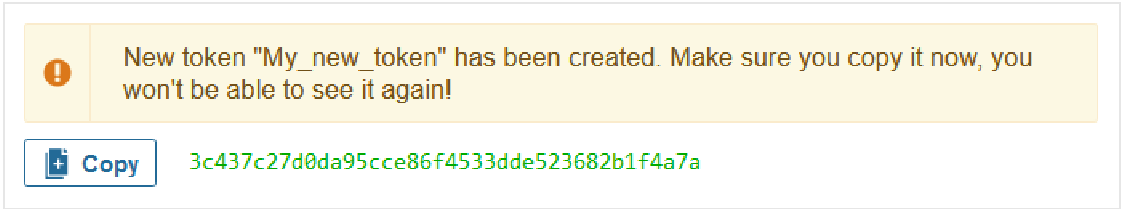


- Generate token
  - Go to https://sonarcloud.io/account/security/
  - Enter a name and click generate
  - Copy the token and keep it in a text editor for the next step



Generate Tokens

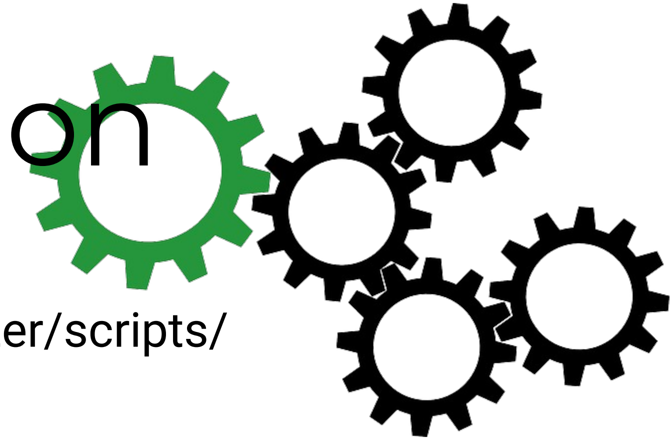
My_new_token



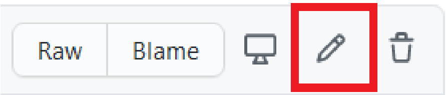
New token "My_new_token" has been created. Make sure you copy it now, you won't be able to see it again!

```
3c437c27d0da95cce86f4533dde523682b1f4a7a
```

SonarCloud Configuration



- Open https://github.com/{github-account-name}/docs-as-code/blob/master/scripts/sonar/**sonar.properties**
- Edit



- Replace these values with your owns

```
# These details need to be filled in on a per-project basis
sonar.organization={sonarcloud.organization.name}
sonar.login={sonarcloud.auth.token}
sonar.projectKey={unique.project.name}
```

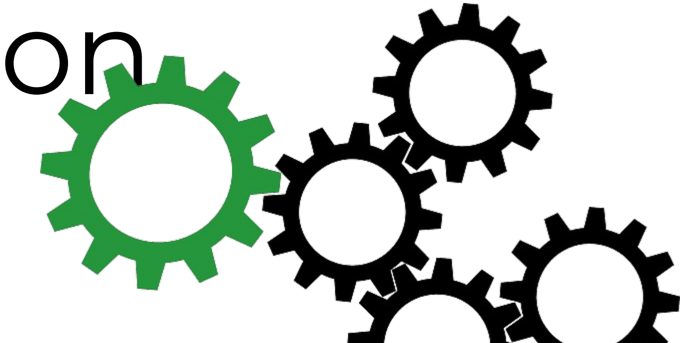
The name of the organization you've created

The value of the generated token

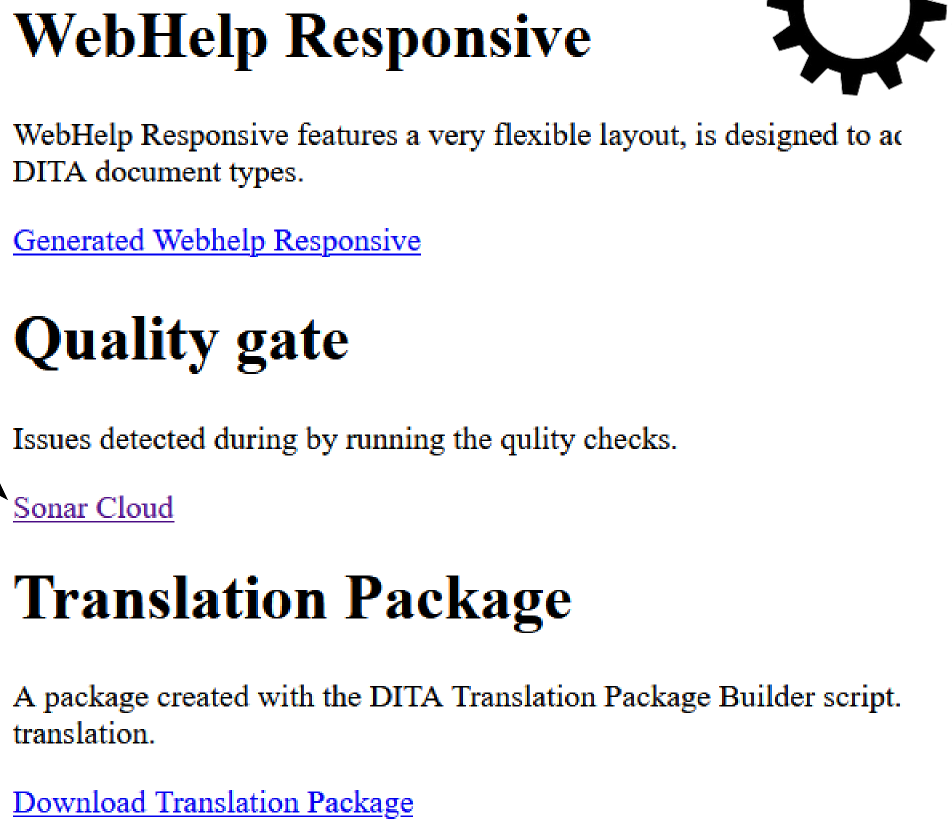
A unique project name. For example: {githut-account-name}-docs-as-code

- Click *Commit changes* on bottom of the page

SonarCloud Configuration Result



- The quality gate link will open your own SonarCloud project



WebHelp Responsive

WebHelp Responsive features a very flexible layout, is designed to ac
DITA document types.

Generated Webhelp Responsive

Quality gate

Issues detected during by running the qulity checks.

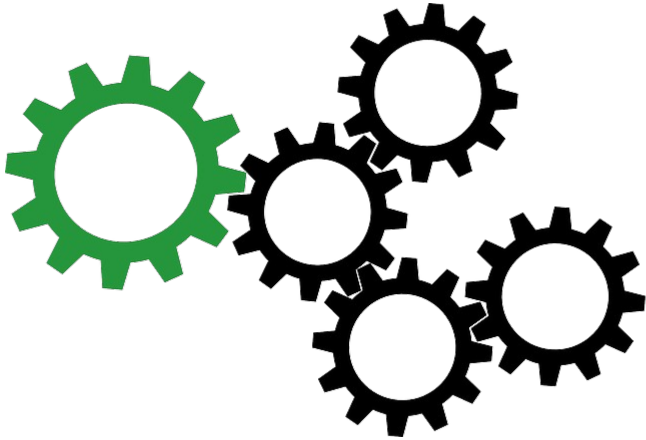
Sonar Cloud

Translation Package

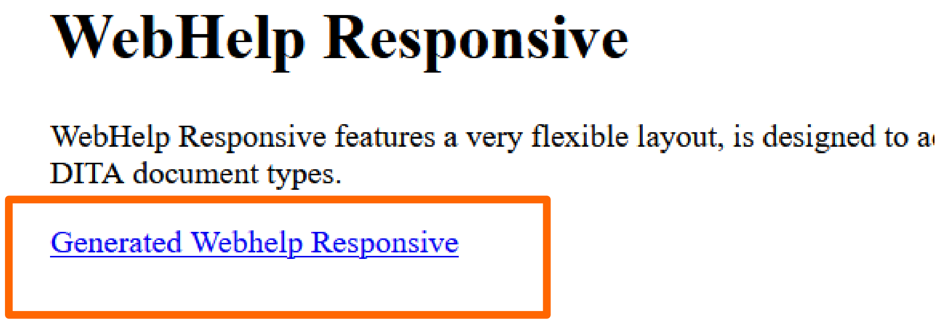
A package created with the DITA Translation Package Builder script.
translation.

Download Translation Package

Trigger a change



- Edit a topic and trigger the build
  - Either using the Git pull/push mechanism
  - Or though Oxygen XML Author powered Edit Links

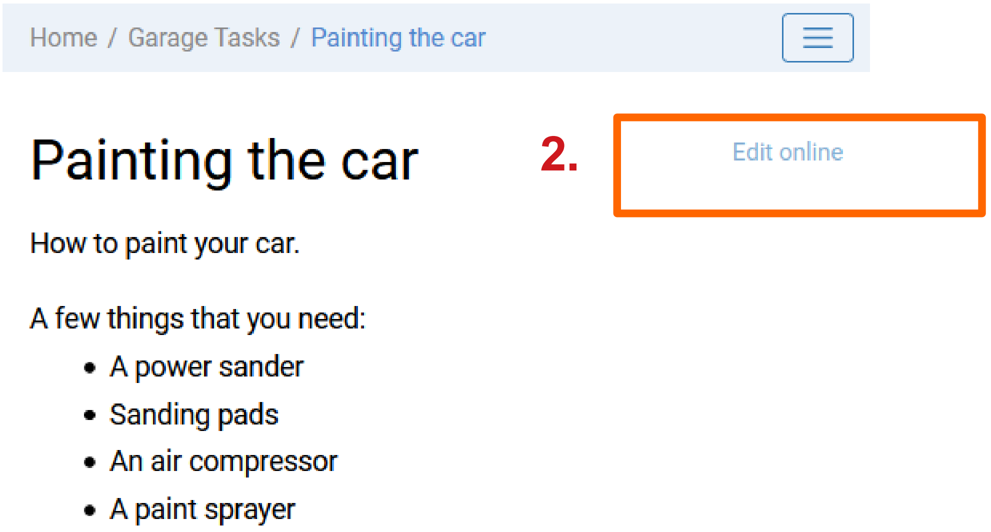


1.

WebHelp Responsive

WebHelp Responsive features a very flexible layout, is designed to a
DITA document types.

Generated Webhelp Responsive



2.

Home / Garage Tasks / Painting the car

Edit online

Painting the car

How to paint your car.

A few things that you need:

- A power sander
- Sanding pads
- An air compressor
- A paint sprayer

THANK YOU!

Alex Jitianu

alex_jitianu@oxygenxml.com

@AlexJitianu