Documentation in Research Software

RSE summer school 2024

Jessica Mitchell
j.mitchell@fz-juelich.de

Raise your hand if you agree

Raise your hand if you agree

I have struggled to understand or use another person’s software because of poor documentation

I feel confident that my software's documentation is clear enough for new users to get started quickly

Documentation is considered a priority in my current team or project

Good documentation is as important as the code itself

Clear documentation can make the difference between a project being reused by others or forgotten.

Raise your hand if you agree

I have struggled to understand another researcher’s software because of poor documentation

I feel confident that my software's documentation is clear enough for new users to get started quickly

Documentation is considered a priority in my current team or project

Good documentation is as important as the code itself

Clear documentation can make the difference between a project being reused by others or forgotten.

Raise your hand if you agree

I have struggled to understand another researcher's software because of poor documentation

I feel confident that my software's documentation is clear enough for new users to get started quickly

Documentation is considered a priority in my current team or project

Good documentation is as important as the code itself

Clear documentation can make the difference between a project being reused by others or forgotten.

Raise your hand if you agree

I have struggled to understand another researcher’s software because of poor documentation

I feel confident that my software's documentation is clear enough for new users to get started quickly

Documentation is considered a priority in my current team or project

Good documentation is as important as the code itself

Outline

- Documentation principles
- Documentation starting point
- Docs as code
- Hands on

Break

- Content creation strategies
- Documentation Generators/Deployment
- Hands-on

Hedgedoc - link to documentation resources

- Templates, links, and examples

https://rse-summer-school-documentation.readthedocs.io

https://github.com/jessica-mitchell/RSE_summer_school_documentation/

Why write software documentation?

Why write software documentation?

- Understand your code in 6 months
- Get people to use your code
    - They don't know how your project meets their needs.
    - They can't find how to install your code.
    - They can't see how to use your code.
- Increase contributions to your code
- Improve your code
- Improve your technical writing

Docs or it didn't happen!

#writethedocs

Why write software documentation?

- Understand your code in 6 months
- Get people to use your code
  - They don't know how your project meets their needs.
  - They can't find how to install your code.
  - They can't see how to use your code.
- Increase contributions to your code
- Improve your code
- Improve your technical writing

Why write software documentation?

- Understand your code in 6 months
- Get people to use your code
  - They don't know how your project meets their needs.
  - They can't find how to install your code.
  - They can't see how to use your code.
- Increase contributions to your code
- Improve your code
- Improve your technical writing

Why write software documentation?

- Understand your code in 6 months
- Get people to use your code
  - They don't know how your project meets their needs.
  - They can't find how to install your code.
  - They can't see how to use your code.
- Increase contributions to your code
- Improve your code
- Improve your technical writing

Docs or it didn't happen!
#writethedocs

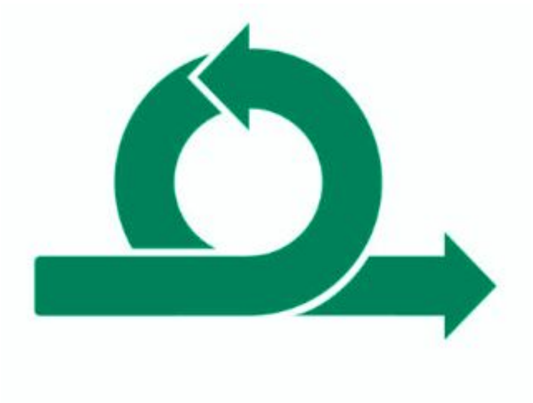
Why write software documentation?

- Understand your code in 6 months
- Get people to use your code
  - They don't know how your project meets their needs.
  - They can't find how to install your code.
  - They can't see how to use your code.
- Increase contributions to your code
- Improve your code
- Improve your technical writing

Docs or it didn't happen!
#writethedocs

Documentation - ‘it’s on my to do list’

Documentation needs to be evolving with the code



Documentation principles

Documentation principles

Documentation sources should be

- Nearby(or in) the source code
- Unique (single source of truth)

Content should be

- ARID (Accept (some) Repetition In Documentation)
- Skimmable
- Exemplary
- Consistent
- Current

Web Pages should be

- Discoverable
- Addressable
- Cumulative
- Complete
- Beautiful

Documentation should be

- Precursory
- Participatory

Documentation principles

Documentation sources should be

- Nearby the source code
- Unique (single source of truth)

Content should be

- ARID (Accept (some) Repetition In Documentation)
- Skimmable
- Exemplary
- Consistent
- Current

Web Pages should be

- Discoverable
- Addressable
- Cumulative
- Complete
- Beautiful

Documentation should be

- Precursory
- Participatory

Documentation principles

Documentation sources should be

- Nearby the source code
- Unique (single source of truth)

Content should be

- ARID (Accept (some) Repetition In Documentation)
- Skimmable
- Exemplary
- Consistent
- Current

Web Pages should be

- Discoverable
- Addressable
- Cumulative
- Complete
- Beautiful

Documentation should be

- Precursory
- Participatory

Documentation principles

Documentation sources should be

- Nearby the source code
- Unique (single source of truth)

Content should be

- ARID (Accept (some) Repetition In Documentation)
- Skimmable
- Exemplary
- Consistent
- Current

Web Pages should be

- Discoverable
- Addressable
- Cumulative
- Complete
- Beautiful

Documentation as as whole should be

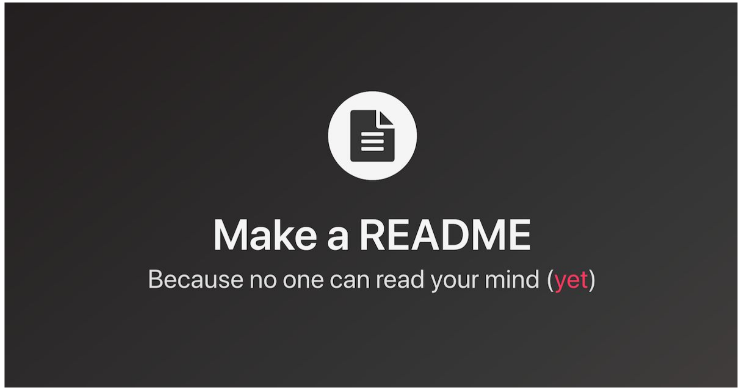
- Precursory
- Participatory

Starting point - the essential documentation

- Consider both developer-facing and user-facing documentation

User-facing documentation enables users to understand and use the software

It *starts* with a good README



Make a README

Because no one can read your mind (yet)

What makes a good README?

- Description of the software : who is it for? What are the key features?
- Installation steps (including prerequisites!)
- Example use case
- How to contribute: Create an issue or pull request
- How to get help (contact info, mailing-list, forum . . .)
- How to cite: What version of software did they use? Is there an official software publication?
- License

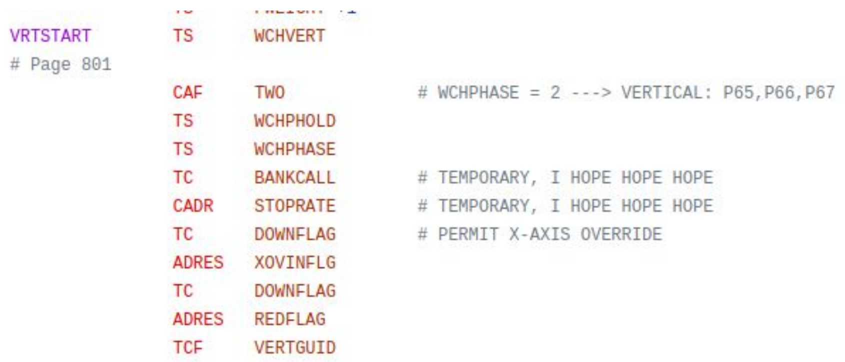
What makes a good README?

- Description of the software : who is it for? What are the key features?
- Installation steps (including prerequisites!)
- Example use case
- How to contribute: Create an issue or pull request
- How to get help: contact info, mailing-list, forum etc
- How to cite: What version of software did they use? Is there an official software publication?
- License

Developer-facing documentation enables continued long-term development

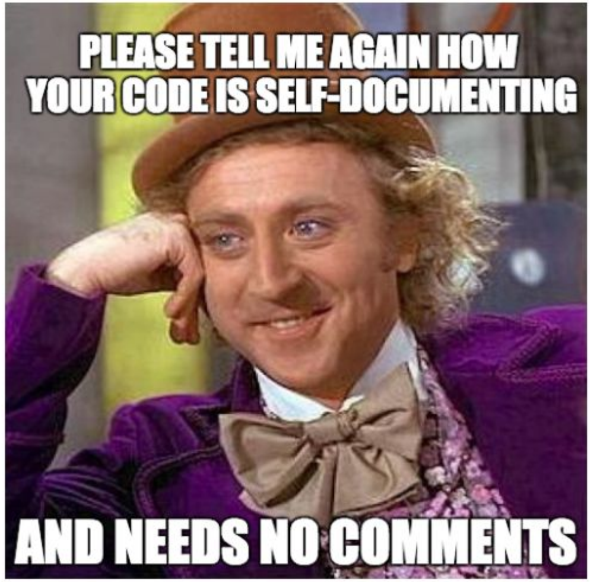
Developer-facing documentation enables continued long-term development

Code comments



```
VRTSTART        TS      WCHVERT
# Page 801
                CAF     TWO             # WCHPHASE = 2 ---> VERTICAL: P65,P66,P67
                TS      WCHPHOLD
                TS      WCHPHASE
                TC      BANKCALL        # TEMPORARY, I HOPE HOPE HOPE
                CADR    STOPRATE        # TEMPORARY, I HOPE HOPE HOPE
                TC      DOWNFLAG        # PERMIT X-AXIS OVERRIDE
                ADRES   XOVINFLG
                TC      DOWNFLAG
                ADRES   REDFLAG
                TCF     VERTGUID
```

Code from Apollo 11:
LUNAR_LANDING_GUIDANCE_EQUATIONS.agc



Developer-facing documentation enables continued long-term development

The details	How to get things done	The big picture
Code comments	Workflows/ Guidelines	Architecture/Road maps

Developer-facing documentation enables continued long-term development

Contributor Guidelines

- What are your expectations?
- What do you need help with?

Reviewer Guidelines

- How should reviewers communicate
- Checklist for reviewers

Style guide

- Define language and styles (e.g., American vs British; markup, bibliography style)
- Set specific rules for spelling, markup, using links
- Keep it light weight, and use pre-existing style guides for most things

Developer-facing documentation enables continued long-term development

Contributor Guidelines

- What are your expectations?
- What do you need help with?

Reviewer Guidelines

- How should reviewers communicate
- Checklist for reviewers

Style guide

- Define language and styles (e.g., American vs British; markup, bibliography style)
- Set specific rules for spelling, markup, using links
- Keep it light weight, and use pre-existing style guides for most things

Developer-facing documentation enables continued long-term development

Contributor Guidelines

- What are your expectations?
- What do you need help with?

Reviewer Guidelines

- How should reviewers communicate
- Checklist for reviewers

Style guide

- Define language and styles (e.g., American vs British; markup, bibliography style)
- Set specific rules for spelling, markup, using links
- Keep it light weight and use pre-existing style guides for most things

Docs as code

A **philosophy** that you should be writing documentation with the same tools as code

Docs as code

A **philosophy** that you should be writing documentation with the same tools as code

- Docs in same repo as code
- Issue Trackers
- Version Control (Git)
- Plain Text Markup (Markdown, reStructuredText, Asciidoc)
- Code/Doc Reviews
- Automated Tests

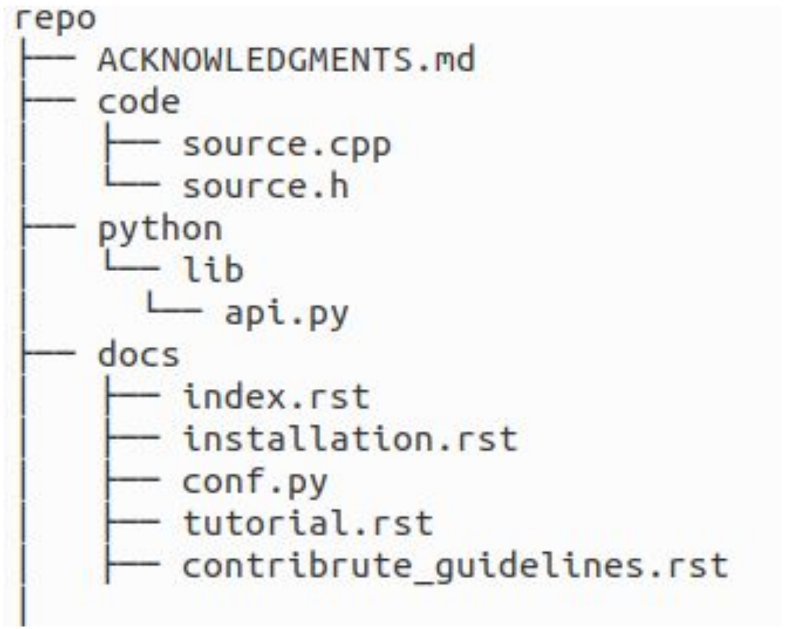
When to use *docs as code*

- Any size project can implement these concepts

- Contributors are familiar with (or willing to learn)
  - scripting languages or plain text formats and
  - version control systems

Docs in the repo

- You need a documentation folder nearby the source (within repository/source files)



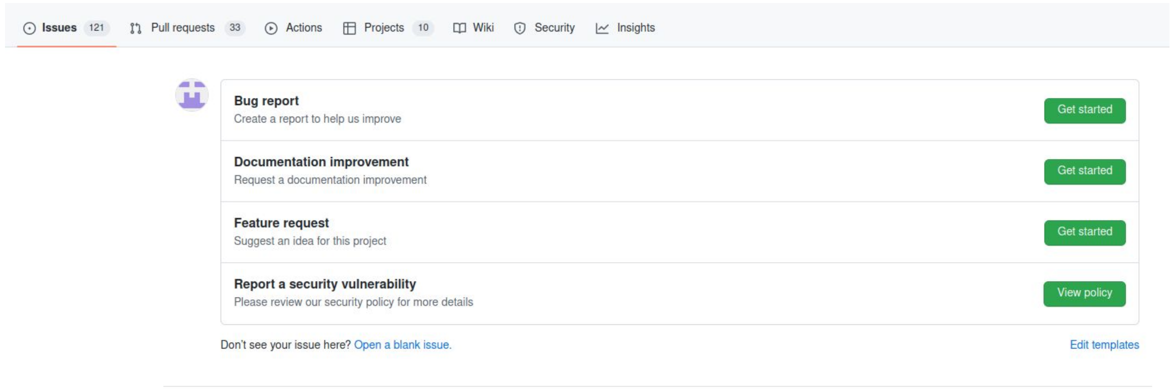
```
repo
├── ACKNOWLEDGMENTS.md
├── code
│   ├── source.cpp
│   └── source.h
├── python
│   └── lib
│       └── api.py
├── docs
│   ├── index.rst
│   ├── installation.rst
│   ├── conf.py
│   ├── tutorial.rst
│   ├── contribrute_guidelines.rst
│
```

Issue trackers

- Track issues for documentation the same way you track issues with code

Issue trackers

- Track issues for documentation the same way you track issues with code
  - **Define issue templates**
  - Use labels / tags / projects



Bug report
Create a report to help us improve
Get started

Documentation improvement
Request a documentation improvement
Get started

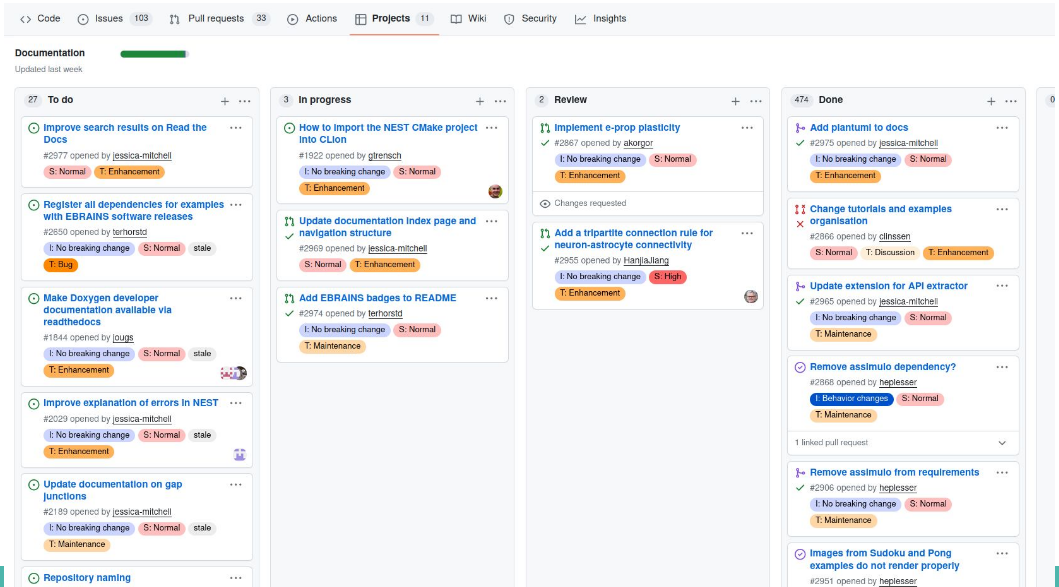
Feature request
Suggest an idea for this project
Get started

Report a security vulnerability
Please review our security policy for more details
View policy

Don't see your issue here? Open a blank issue.
Edit templates

Issue trackers

- Track issues for documentation the same way you track issues with code
  - Define issue templates
  - **Use labels / tags / projects**



Documentation
Updated last week

27 To do

Improve search results on Read the Docs
#2977 opened by jessica-mitchell
S: Normal T: Enhancement

Register all dependencies for examples with EBRAINS software releases
#2650 opened by terhorstd
I: No breaking change S: Normal stale T: Bug

Make Doxygen developer documentation available via readthedocs
#1844 opened by jougs
I: No breaking change S: Normal stale T: Enhancement

Improve explanation of errors in NEST
#2029 opened by jessica-mitchell
I: No breaking change S: Normal stale T: Enhancement

Update documentation on gap junctions
#2189 opened by jessica-mitchell
I: No breaking change S: Normal stale T: Maintenance

Repository naming

3 In progress

How to import the NEST CMake project into CLion
#1922 opened by gtrensch
I: No breaking change S: Normal T: Enhancement

Update documentation index page and navigation structure
#2969 opened by jessica-mitchell
S: Normal T: Enhancement

Add EBRAINS badges to README
#2974 opened by terhorstd
I: No breaking change S: Normal T: Maintenance

2 Review

Implement e-prop plasticity
#2867 opened by akorgor
I: No breaking change S: Normal T: Enhancement
Changes requested

Add a tripartite connection rule for neuron-astrocyte connectivity
#2955 opened by HanjiaJiang
I: No breaking change S: High T: Enhancement

474 Done

Add plantuml to docs
#2975 opened by jessica-mitchell
I: No breaking change S: Normal T: Enhancement

Change tutorials and examples organisation
#2866 opened by clinssen
S: Normal T: Discussion T: Enhancement

Update extension for API extractor
#2965 opened by jessica-mitchell
I: No breaking change S: Normal T: Maintenance

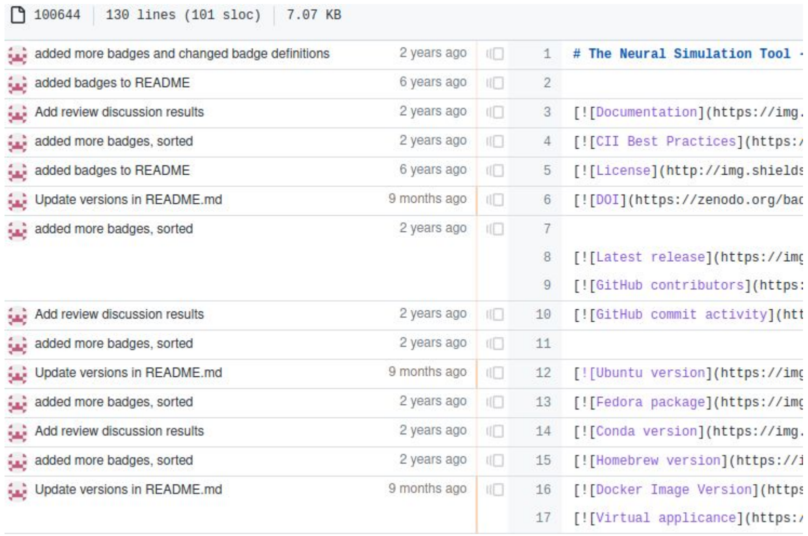
Remove assimulo dependency?
#2868 opened by heplesser
I: Behavior changes S: Normal T: Maintenance
1 linked pull request

Remove assimulo from requirements
#2906 opened by heplesser
I: No breaking change S: Normal T: Maintenance

Images from Sudoku and Pong examples do not render properly
#2951 opened by heplesser

Version control system (e.g., git)

- Tracking every change to the documentation with commits
- History of who did what, when, and (hopefully) why



100644 | 130 lines (101 sloc) | 7.07 KB

added more badges and changed badge definitions	2 years ago	`# The Neural Simulation Tool -`
added badges to README	6 years ago	
Add review discussion results	2 years ago	`[![Documentation](https://img.`
added more badges, sorted	2 years ago	`[![CII Best Practices](https:/`
added badges to README	6 years ago	`[![License](http://img.shields`
Update versions in README.md	9 months ago	`[![DOI](https://zenodo.org/bad`
added more badges, sorted	2 years ago	
		`[![Latest release](https://img`
		`[![GitHub contributors](https:`
Add review discussion results	2 years ago	`[![GitHub commit activity](htt`
added more badges, sorted	2 years ago	
Update versions in README.md	9 months ago	`[![Ubuntu version](https://img`
added more badges, sorted	2 years ago	`[![Fedora package](https://img`
Add review discussion results	2 years ago	`[![Conda version](https://img.`
added more badges, sorted	2 years ago	`[![Homebrew version](https://i`
Update versions in README.md	9 months ago	`[![Docker Image Version](https`
		`[![Virtual applicance](https:/`

Plain text markup

- Well understood by developers
- You can use same editor (IDEs, vim, emacs etc.) as you would code

mdx

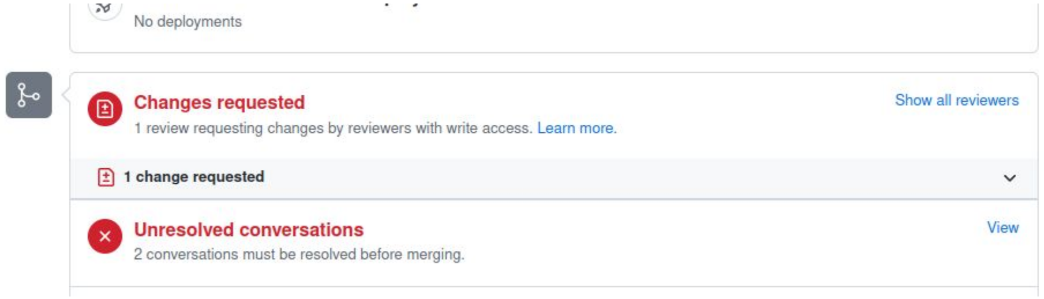
~~~
---
title: 'Example'
metaDescription: 'Example meta desc'
hidePage: true
---


### Code blocks

```js
async function main() {
  const allUsers = await
}
```
~~~

reStructured Text

```
.. _ref_label:

Some heading
-------------

.. code-block:: python

   import nest
```

Code and documentation reviews

- Pull(merge) requests that are documentation only OR code with documentation
- Select reviewers to oversee documentation aspects (language, links and references, style, structure . . .)
- Set guidelines for reviewers for ensuring documentation is correct


No deployments

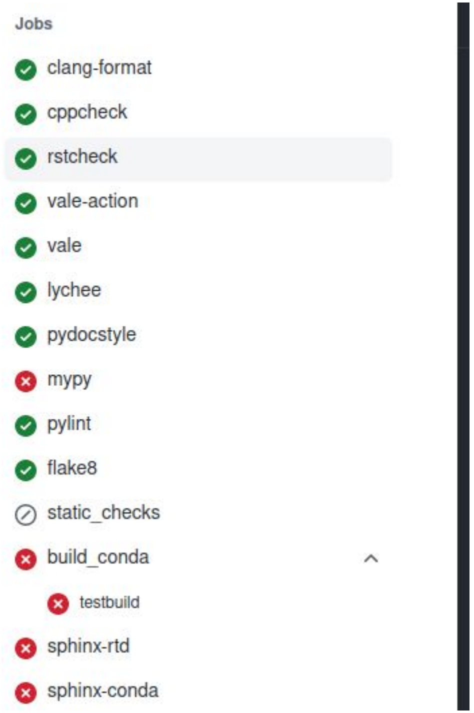
Changes requested
1 review requesting changes by reviewers with write access. Learn more.

1 change requested

Unresolved conversations
2 conversations must be resolved before merging.

Automated Tests

- Check documentation build
- Check links
- Lint prose
- Check format
- Test examples



Jobs

- ✓ clang-format
- ✓ cppcheck
- ✓ rstcheck
- ✓ vale-action
- ✓ vale
- ✓ lychee
- ✓ pydocstyle
- ✗ mypy
- ✓ pylint
- ✓ flake8
- ⊘ static_checks
- ✗ build_conda
  - ✗ testbuild
- ✗ sphinx-rtd
- ✗ sphinx-conda

Make documentation a point in meetings

Things to work on

Do you have a *good* README? Is your code documented?

Do you have a way to track documentation-related issues?

Are you missing guidelines for contributions or developers?

Do you have a documentation generator? Hosting platform?

Do you have content you want to include? Further develop?

Do you have automated tests for documentation (link checkers, build check, prose linter)

Content development tips

Encourage community involvement

- Use subject matter experts (SMEs) to develop content alongside developers
- Ensure documentation is understandable by audience (consider non-native speakers)

Look for pre-existing content

- Look for workshop / lecture material or publications that might have relevant content for documentation

Use various visual elements

- Make sure your text is illustrative with code blocks, figures, graphs, and tables.

Consider that every page is page one

- A user needs to navigate to their goal from any point in your documentation.
- Use table of contents, breadcrumb trails, previous / next buttons, and appropriate links

Write docs as you write code

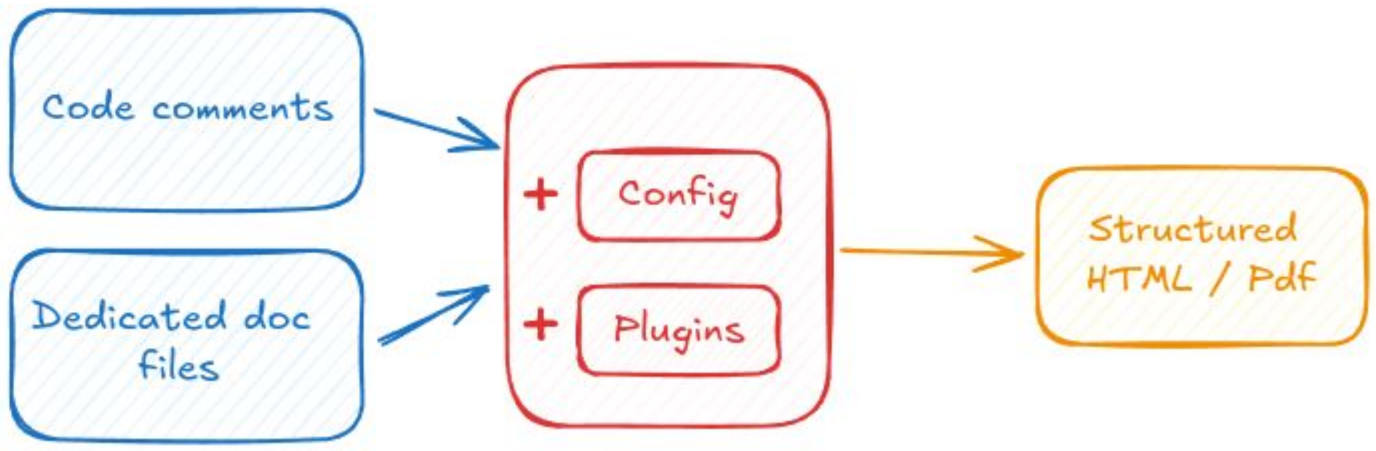
- An example of tutorial-driven development
    - Tutorial-driven development puts the focus on user needs rather than code implementation.
    - The tutorial is written first, then the code.

AI the docs

- Generate comments / docstrings
- Check and improve language clarity (deepl.com/write)
- Write drafts or outlines

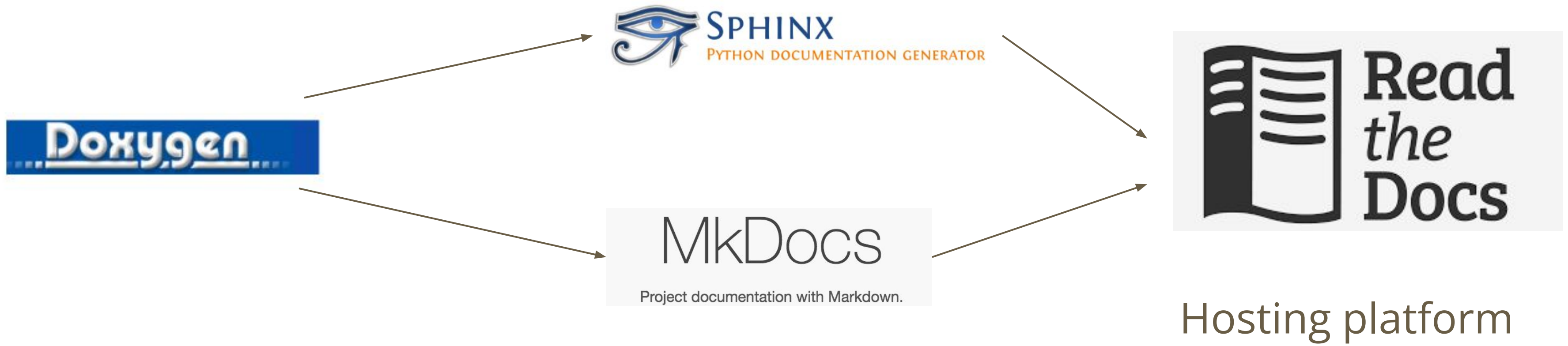
Always requires human review!

Documentation Generators



Documentation Generator	Compatible Languages	Markup language
Doxygen	C++ (C, Python, PHP, Java, C#, Objective-C, Fortran, VHDL, Splice, IDL, and Lex)	Custom syntax / Markdown
Sphinx	Python (C++, C, Javascript)	ReStructured Text / Markdown
MkDocs	Python	Markdown
Documenter.jl	Julia	Markdown
FORD	Fortran	Markdown
roxygen2 / Rmarkdown	R	Custom syntax / Markdown

Sphinx and MkDocs have Doxygen plugins and are supported by Read the Docs


Doxygen
SPHINX
PYTHON DOCUMENTATION GENERATOR
MkDocs
Project documentation with Markdown.
Read the Docs
Hosting platform

Hosting platforms

Read the Docs

GitLab

Pages

GitHub Pages

Provide free hosting for open-source projects

Things to work on

Do you have a *good* README? Is your code documented?

Do you have a way to track documentation-related issues?

Are you missing guidelines for contributions or developers?

Do you have a documentation generator? Hosting platform?

Do you have content you want to include? Further develop?

Do you have automated tests for documentation (link checkers, build check, prose linter)