Ruth Cheesley

Early stage documentation: From chaos to clarity!

WriteTech Hub Bootcamp
5th June 2025

@RCheesley

Ruth Cheesley

Ruth Cheesley (she/her)

Mautic Project Lead & Co-Founder, Women of Open Source community

ruth.cheesley@mautic.org

speaking.ruthcheesley.co.uk for slides, recording, links and resources

@RCheesley

What we're covering.

By 'early-stage' I mean any project that doesn't already have a strongly defined documentation team responsible for maintaining and improving their documentation.

@RCheesley

1.

Documentation landscape in early-stage open source projects: what to expect, common patterns and the problems maintainers face.

2.

Making an impact as a technical writer in an early-stage open source project: how you can make high-impact contributions.

3.

Interactive hands-on breakouts: reviewing the documentation for early-stage projects, prioritising improvements, and making that first contribution.

It’s time to wake up, folks!

There’s going to be audience participation and breakout groups, so it’s time to pay attention!

My contribution journey

2003

Started using open source tools (Knoppix, Ubuntu, ClamAV, HijackThis) to fix infected computers for other students while at Uni, as ‘Essex Virus Removals’.

2007

My first job in IT after graduating as a physio! Working in a school as an IT Technician, I was asked to rebuild the website. Came across Content Management Systems (Drupal, Joomla!) and started using Joomla.

2008

Needing to pay off loans from University, I started building Joomla! websites and freelancing alongside work.

2009

Started a local Joomla user group as I wanted to meet other Joomla users and find ways to do things / have people to ask when I got stuck. We met monthly and quickly gained a committed group of members.

2012

Started working with Joomla as my main occupation, proposed my first talk for the Joomla World Conference in San Jose - created GitHub account!

2013

Created my first bug report on GitHub! Invited to join the Joomla Community Leadership Team to support User Groups and Marketing.

2014

Created my first PR at a Pizza, Bugs and Fun event to fix a bug in Joomla, which was merged the day after it was submitted! 🥳

Joined the Joomla Marketing Team.

2015

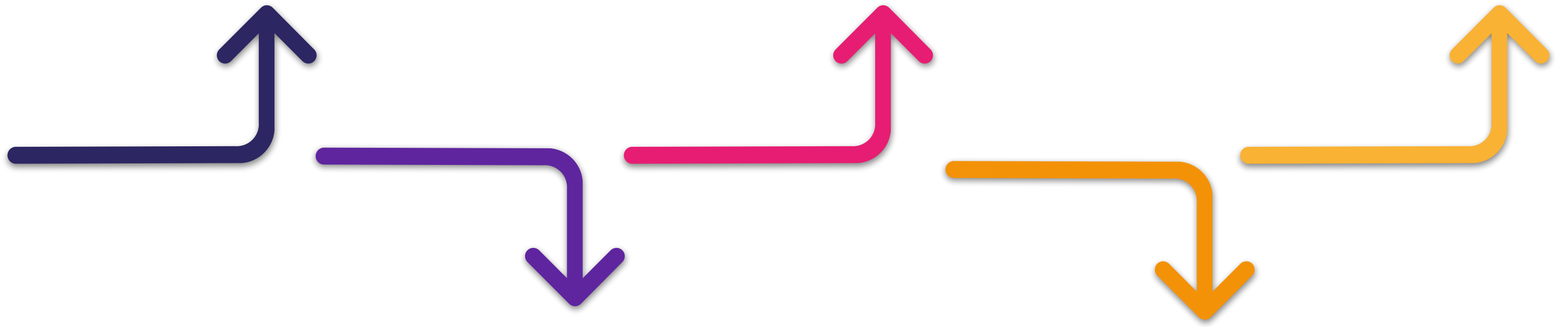
Raised my first issue and PR for Mautic - a project that has just launched and which I used for clients in our Digital Agency.

2019

Started work at Acquia as Mautic Community Manager, helping the open source community to establish its own governance and autonomy from the corporate entity which was acquired by Acquia in 2019.

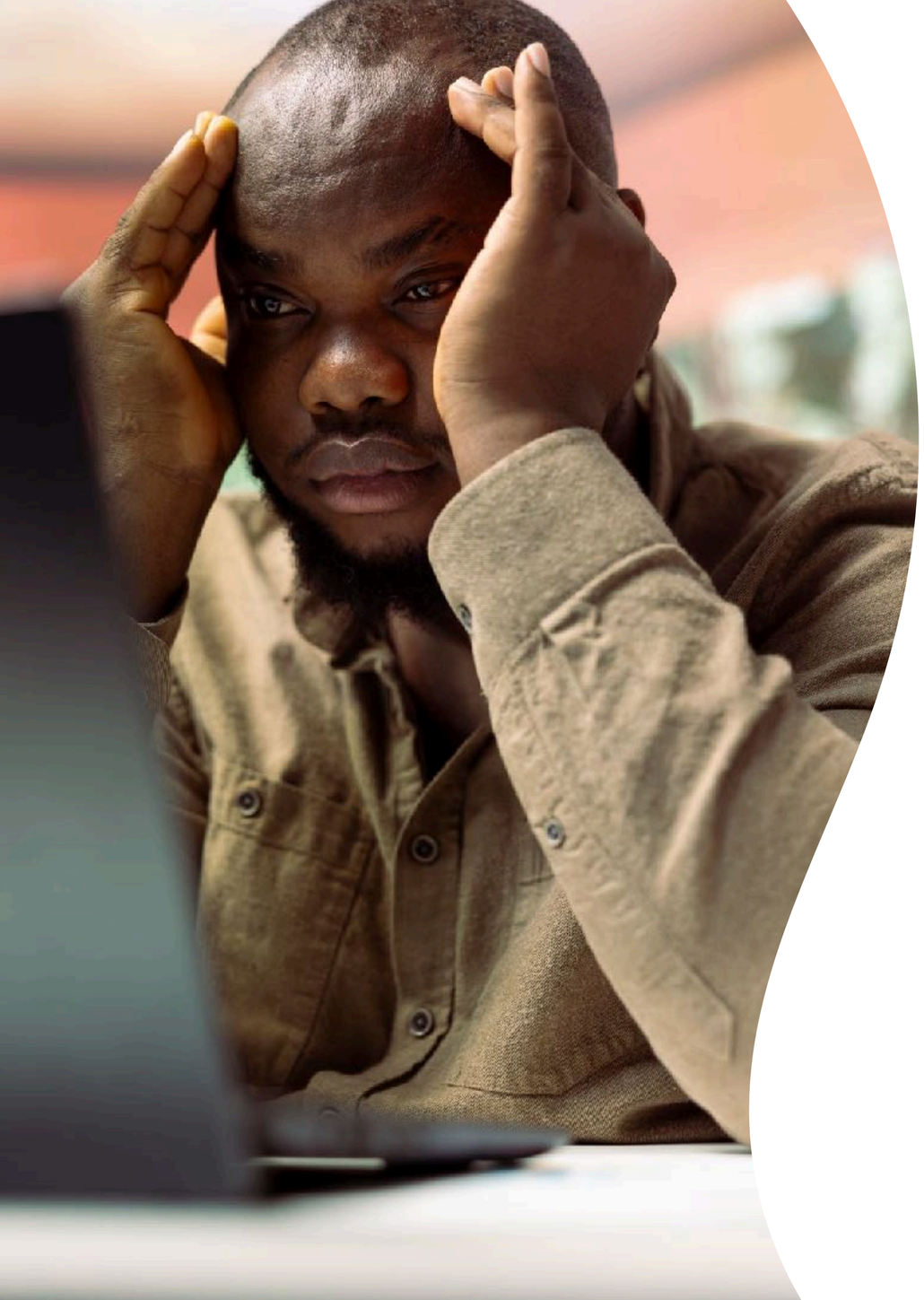
2020

Stepped up to Mautic Project Lead. Led my first release, established our community governance model, teams and values.



Why focus on early-stage projects?

The docs landscape in early-stage projects



The jargon-heavy introduction.

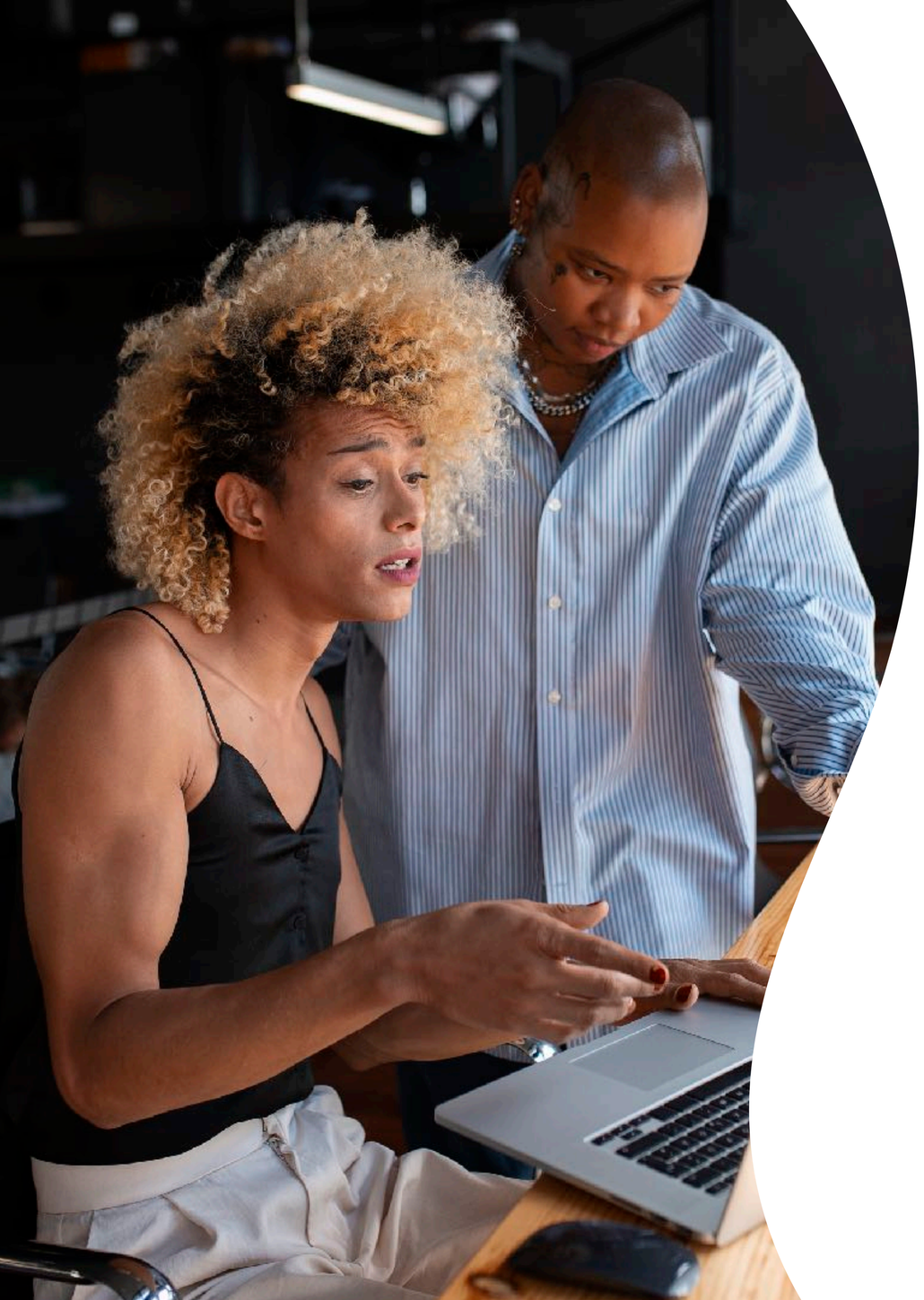
- Documentation is packed with unexplained technical acronyms
- Assumes expert knowledge of the area, without linking to beginners tutorials for assumed knowledge
- No clear explanation of the project's purpose, what it does, how it does it and why it's used
- Limited 'quick start' guidance or getting started steps

The README-or-else project!

- All the documentation exists in a single README file
- Usually very technical
- Often assumes knowledge of technology used
- Additional information and new changes is scattered across PRs, issues and chat conversations
- Substantial ‘technical docs debt’ due to lack of quality standards

Inadequate configuration docs.

- No explanation of configuration options
- Missing default values or configuration snippets
- Lack of clarity for required v optional settings
- Ambiguous comments (e.g. ‘add custom plugins here’ - how?)
- No examples of complete configurations



Outdated tutorials.

- Refers to old versions or incorrect instructions
- Outdated screenshots which don’t match the product
- Referencing features which have changed or no longer exist
- No indication of changelog or what has changed since it was written

Fragmented documentation.

- Information scattered across multiple locations
- No clear path for users to understand how to get started or contribute
- Users forced to hunt for the basic information
- Inconsistent formats across documentation resources
- Relies on internal or pre-assumed knowledge

Are they already asking for help?

Finding existing tasks.

- Check for **labels** like ‘documentation’ - especially if they’re combined with ‘good first issue’
- Search **discussions** (chat, GitHub discussions, mailing lists etc), project management tools etc. for references to documentation being needed
- Search PRs for references to needing documentation - sometimes this might be a label, sometimes a passing comment that the docs need writing or are missing/outdated.

https://github.com/**organization/repo/labels**

Finding ways to contribute

CLEAR?

A suggestion for how to review and assess a project's documentation quality

Content

Layout

Examples

Audience

Recency

https://bit.ly/clear-framework

CLEAR?

A good place to start is with the actual content written in whatever documentation a project has available.

Content

Layout

Examples

Audience

Recency

CLEAR?

Here's some tips for where to start when reviewing the content of an open source project's documentation resources.

Content Assessment.

1.

Does the README explain what the project is, what it does, how to install/configure/use it? Are pre-requisites or dependencies mentioned?

2.

Do the steps provided to get started actually work? Could you get it installed and configured without any external searching?

3.

Are there any links that don't work, references to outdated code, or code samples which don't match the currently supported version?

CLEAR?

Layout is an important aspect when it comes to documentation of open source projects. Good content with a poor layout can make for a painful experience!

Content

Layout

Examples

Audience

Recency

CLEAR?

Here's some areas relating to layout and structure where we commonly see open source projects struggling, or not optimising the experience for their users.

Layout and structure assessment.

1.

Could you easily find the documentation, or is it fragmented across multiple locations or not linked clearly from the README? Is it logically laid out, with clear sections and linked headings?

2.

Do longer documents have a table of contents? Are all the links working, and going to the right anchors? Does the structure make sense?

3.

Does the navigation make sense? Can you easily find what you're looking for? Are there links between sections, where relevant (e.g. if mentioning a different feature)

CLEAR?

Examples can be critical for new users, whether it's specific configurations to be copy/pasted or use cases for how the product is used, it all helps with onboarding.

Content

Layout

Examples

Audience

Recency

CLEAR?

Here’s some areas where you can review the examples and explanations that are used by an open source project.

Examples and explanations.

1.

Is technical language explained or avoided? Could you understand everything that was written, as a newcomer to the project?

2.

Are there sufficient examples for key features and functions? Do the examples take you from a new user through to a more seasoned, experienced user?

3.

Are there examples and explanations which cover different use cases, helping people to understand where this project fits their specific needs?

CLEAR?

Open source projects often have to cater to several audiences - developers, users, implementors and more. It's an important dimension to consider when reviewing documentation.

Content

Layout

Examples

Audience

Recency

CLEAR?

Here’s some areas where open source projects often don’t optimise their documentation for specific audiences.

Audience adaptation.

1.

Is there helpful documentation for people at every stage of their journey with the project, from beginners to experts?

2.

Are advanced / specialised topics separated from the basics to avoid confusing newcomers? Is it easy to find specific advanced technical information?

3.

Are there any assumptions made about pre-required knowledge? Is there a clear getting started guide, both for using the project and also for contributing and setting up a local environment?

CLEAR?

Keeping documentation up to date is a never-ending task for open source projects, and many fall behind quite easily.

Content

Layout

Examples

Audience

Recency

CLEAR?

A suggestion for how to review and assess a project's documentation quality

Recency and relevance.

1.

Are there code samples and screenshots which don't represent the current state - for example outdated code or screenshots which don't match the current experience?

2.

Have the documentation resources been reviewed and updated to ensure that they're culturally appropriate, written without bias and using inclusive language?

3.

Have new features introduced in recent releases been added to the documentation with clear information about what they do, how to use them, and how to troubleshoot?

Time to wake up!

Let’s take a look at some open source project readme’s and identify some documentation gaps or issues that could be fixed.

Other teams will use your findings in later breakouts!

Join us on Canva!

https://www.canva.com/design/DAGouh3RWi4/CYDrvi6Dls1g2kV73LTWfg/edit

Welcome back!

Hope you had fun reviewing some open source projects and finding areas for improvement!

Making an impact as a technical writer

How you can help maintainers better serve their users as a technical writer

CLEAR?

How can you make an impact by improving content for an open source project?

Content

Layout

Examples

Audience

Recency

Making an impact: Content.

- Write a clear, user-friendly project overview - ‘what is this’, ‘who is this for’, etc
- Create a quick-start guide/tutorial
- Fix any broken, incorrect or outdated content you come across
- Suggest or create a documentation style guide (where appropriate - larger projects)
- Implement a linter to flag content issues and improve the quality and consistency of docs

CLEAR?

What can you help to improve when it comes to the layout of an open source project's documentation?

Content

Layout

Examples

Audience

Recency

Making an impact: Layout.

- Consider whether a separate documentation system (e.g. Read the Docs or similar) would be beneficial, especially with larger/more complex projects with different versions
- Improve the experience with table of contents, organising using sections with headings, and centralising scattered documentation
- Add internal and external links to help people find relevant information, know where to go next, etc.

CLEAR?

Examples and explanations are critical in helping new users to understand and adopt open source tools, how can you make an impact?

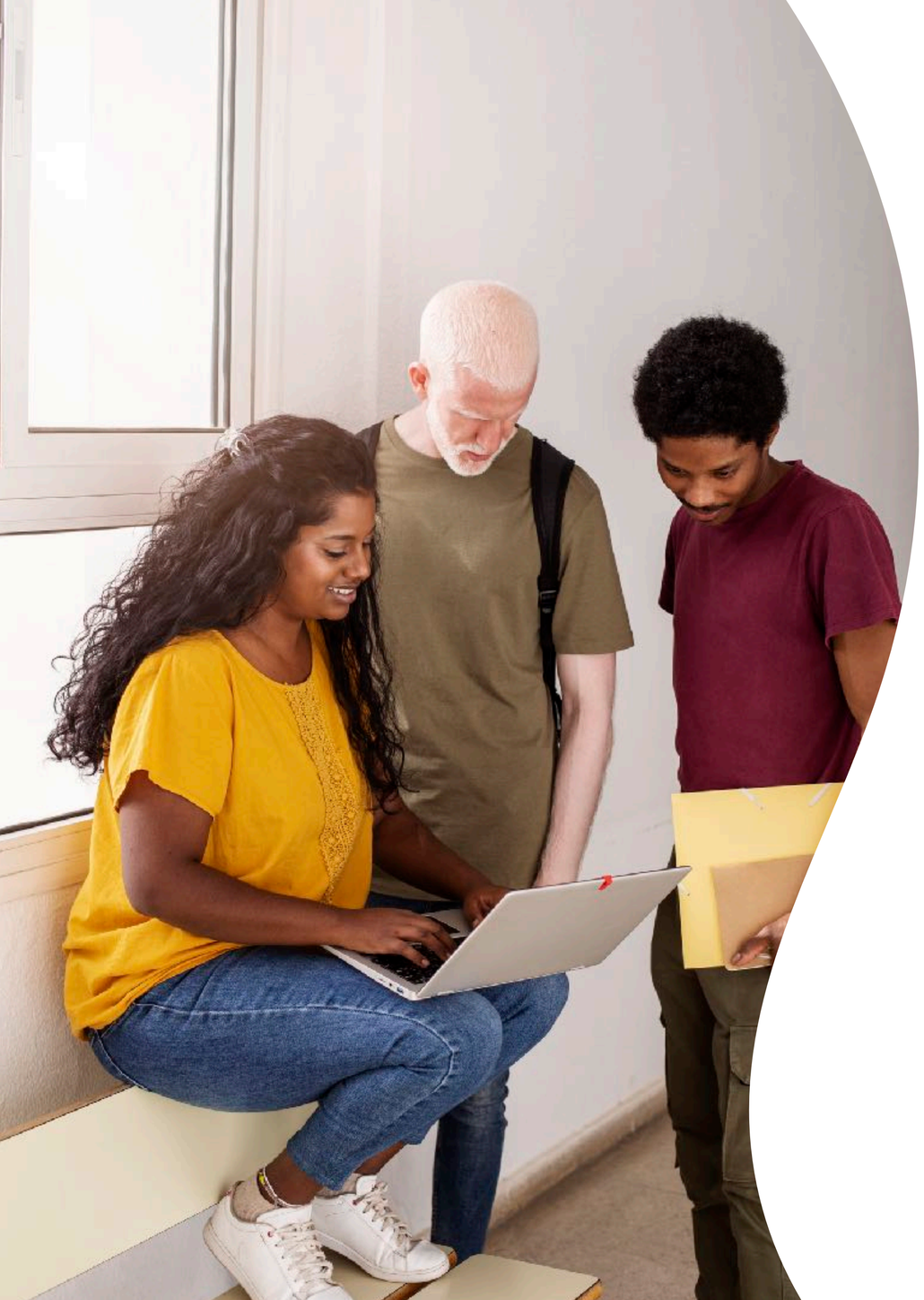
Content

Layout

Examples

Audience

Recency



Making an impact: Examples.

- Add beginner-friendly examples and step-by-step walkthroughs including copy-paste code samples to help newcomers
- Expand example coverage to ensure all features, configuration options and settings are documented
- Clarify technical language, especially if it’s something specific to this kind of technology or project

CLEAR?

Open source projects often have to cater to several audiences - developers, users, implementors and more. It's an important dimension to consider when reviewing documentation.

Content

Layout

Examples

Audience

Recency

Making an impact: Audience.

- Separate beginner, intermediate and expert content into different sections
- Add or improve contributor guides to help new users set up their local environment, find tasks to work on, understand coding standards and tooling used, and creating PRs
- List prerequisites and clearly state assumed knowledge, linking to where to learn more
- Create FAQs for common questions of different experience levels

CLEAR?

How can you have an impact by helping an open source project with keeping its documentation up to date?

Content

Layout

Examples

Audience

Recency

Making an impact: Recency.

- Check recent releases and ensure that all new features are fully documented - if not, offer to write the docs
- Refresh code samples, screenshots and instructions to match the current version
- Add a documentation changelog so users can tell when resources were last updated
- Review the docs for inclusivity, tone and accessibility against current industry and cultural expectations

Time to wake up!

Using the notes that were taken in the first round of breakouts, we're now going to consider what we could contribute.

Join a different breakout room to the one you started in!

Let’s go back to Canva!

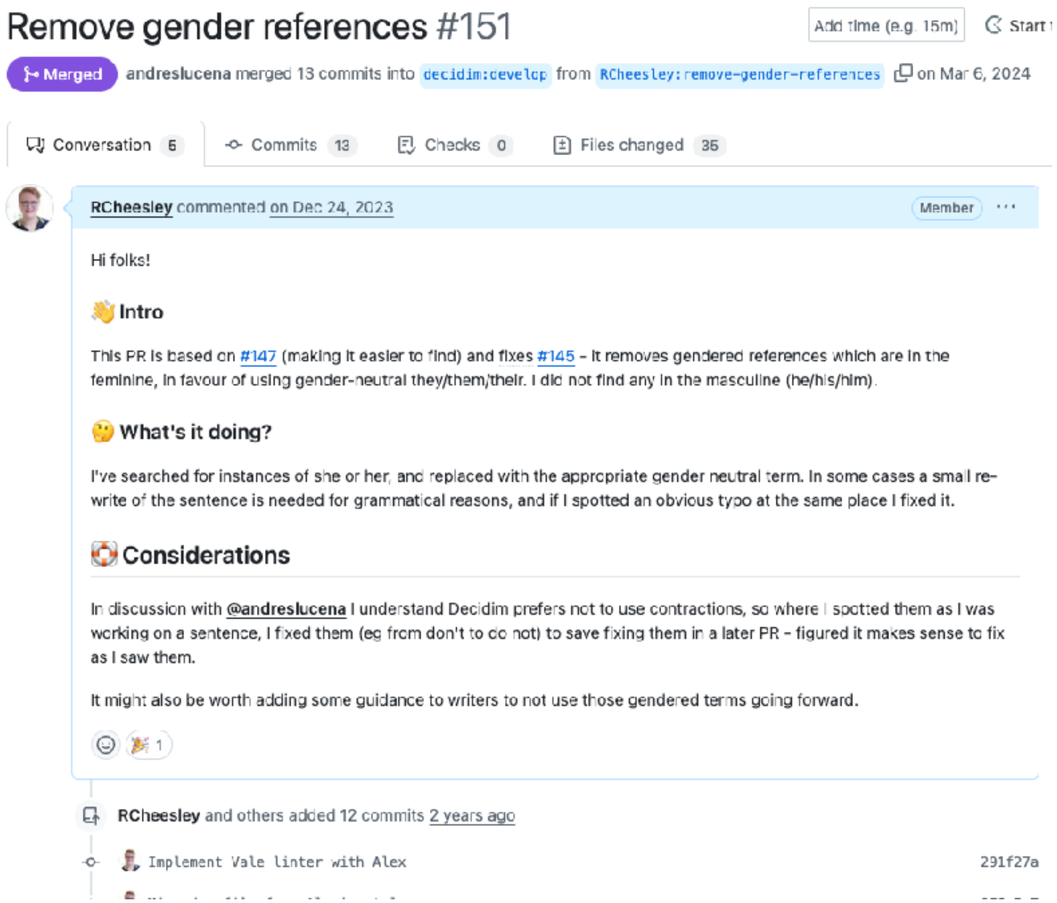
https://www.canva.com/design/DAGouh3RWi4/CYDrvi6Dls1g2kV73LTWfg/edit

Making that first outreach

How to reach out and offer to help maintainers of open source projects you're new to

Making the first contribution:

- Don’t lead with **how awful their documentation is** or a laundry list of the **things you’ve found wrong** - none of us like having our faults pointed out to us!
- Introduce yourself, tell them why you want to help them with this project, and outline what you’ve found and fixed in your first contribution
- Start out with small, well scoped tasks which are quick to review and merge - wait for the first to be merged before submitting more
- Listen to feedback and reply promptly!



What makes a PR easy to accept:

- **Always use the PR template** and ensure you include all relevant details
- Highlight if you're fixing an open issue or building on an existing / earlier PR
- Call out specific changes which the maintainer should be aware of
- Use the right branch as your base
- Keep your PRs small, concise and focused. Don't flood maintainers with lots at once or huge PRs

Implement vale linter #147

Add time (e.g. 15m) | Start t

Merged | andresluceno merged 10 commits into `decidim:develop` from `RCheesley:implement-vale-linter` on Feb 13, 2024

Conversation 14 | Commits 10 | Checks 0 | Files changed 51

RCheesley commented on Sep 8, 2023 • edited | Member

Hi folks!

👋 Intro

This PR relates to #148 and implements the Vale linter to pick up spelling mistakes, grammar issues and potentially problematic word use.

I hope that this will help to iteratively improve the documentation and maintain quality going forward.

🤔 What's it doing?

Vale is a tool which lints - checks - text against certain pre-specified rules - much like similar tools for software development highlight and/or fix code style issues.

This helps to improve the quality of written text, pick up problems such as typos or grammar issues, and also can help to flag up words or phrases which might in some situations or cultures be inappropriate.

I have implemented two style plugins from Vale - Google and Alex.

Google's technical writers styleguide is a well-used developer technical style guide which covers a huge amount of grammar issues and best practices, and it also picks up some inappropriate word use.

Alex is more focused on the cultural sensitivity and appropriateness of words used.

This allows writers to check their text as they're writing (for example there are plugins for VSCode, PHPStorm etc) and it also allows us to check PRs which are made so that if we haven't fixed those files yet, they can be cleaned as we go and new contributions can be checked to ensure they adhere to the style guides.

✍️ Impact for writers

I have made one type correction in a file and left another error in the same line, so that you can see what it looks like to the writer when they get feedback on their PR that something needs fixing:

In this case, you could replace the word 'execute' with something else that is not going to flag up a warning for being potentially problematic, for example 'implement'.

🛟 Considerations

Some things that I have found from using Google's styleguide in particular in the Mautic docs is that it does promote writing in the active voice rather than passive voice for documentation which can be tricky even for native English speakers at times (but ChatGPT is great at helping to convert text from passive to active and there's various resources online to help, too). So it might do well to have some documentation for writers who want to contribute, explaining what your style guide, tone of voice etc is, and how to deal with feedback about improving the grammar.

It is also possible to add your own specific words, for example 'Decidim', so that it's not flagged up as a typo and is always written in its capitalised form. Likewise you can also have words which will also flag up a failure - eg if people commonly mis-spell something.

I hope that this is helpful, I'm planning to work through some of the errors that are thrown up with it and make a PR by section to fix them (but they can also be fixed iteratively over time, too - doesn't have to be addressed straight away), in addition to the PR which addresses the gendered terms.

How to propose larger changes:

- Talk with maintainers before starting
- Explain what you're proposing, why, and how it will help the project
- Call out directly the impact it will have and how it will improve things
- Be willing to listen to feedback
- Ensure that it's maintainable **without you being around** going forward - link to docs and resources to help the maintainers future-proof your contributions

Respecting maintainers:

- **Don’t ghost them** - if you’re only able to contribute for a few months (e.g. as part of a project or a course) be clear about that upfront
- Check that your prioritisation aligns with their vision for improvement - **don’t assume you know what’s most important**
- **Do your own homework** - don’t expect them to teach you about the tool unless you’ve already made efforts to learn for yourself
- **Use AI responsibly**, if at all, and always declare when you’ve used it and why

Everyone still with us?

Let’s agree on your take-home tasks

Tasks for this week.

Make your first steps toward contributing to a project you've encountered today or one you already know about

1.

Review the Canva board, and if something takes your interest from the second round of contribution ideas, write your name on the sticky note and make that contribution

2.

Apply the CLEAR framework with an open source project that you know of and want to help, see what you find, and begin the process of contributing to the project

3.

Have a conversation with a maintainer of a project that you're interested in. If you would like an introduction to any of the projects listed in the Canva board, ping me

Ruth Cheesley

Ruth Cheesley (she/her)

What questions can I answer?

ruth.cheesley@mautic.org

speaking.ruthcheesley.co.uk for slides, recording, links and resources

@RCheesley