

# Conclusions & Final project peer reviews

The fundamentals of technical writing | MUNI 2024

The Red Hat Customer Content Services team

# What will happen today

- Recap of the course's takeaways
- Crowd-sourced peer review
- Group discussions
- Q & A
- Instructions for your presentations



## Objectives of technical writing

- WHAT? -> DOCUMENTATION
  - Make difficult information easy to understand and apply
  - Manuals/guides, troubleshooting/solutions, release notes/ patch notes
- WHY? -> USERS
  - Help users accomplish their goals and avoid problems
  - Write about features, but for users -> use cases, user stories
- WHO? -> TECH WRITERS
  - Bridge between developers and users
  - Planning, research, consulting, testing, writing, reviewing, publishing



## Tech writing style

- Focus on readability, findability, accessibility, and usability
- Best practices for style in tech writing:
  - Use simple, direct, translation-friendly language.
  - Use visual structure and formatting to make the text easy to read.
  - Focus on helping the reader accomplish a specific goal.
  - Say as much as possible in as few words possible (~minimalism).
  - Stick to a single style (guide)



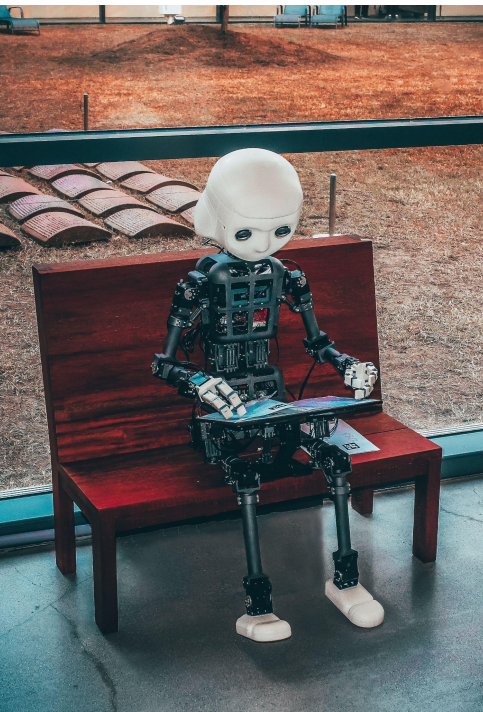
## The tools of technical writing

- “Docs as code” X Content management systems (e.g. AEM)
- Testing: virtual machines, containers, cloud instances
- Languages: Markdown, AsciiDoc, DITA/DocBook XML, RST
- Text editors: VS Code, IntelliJ, Vim, Emacs, Sublime Text...
- Version control: Git, SVN
- Publishing: GitHub/GitLab Pages, Antora, internal toolchains



## Hard skills & Soft skills in tech writing

- Supplement your product knowledge with teamwork
- Collaboration with SMEs (developers, QEs), customer support, product managers, marketing representatives
- Try to get involved early in the development cycle
- Be considerate yet constructive
- Make your tasks SMART as much as possible



## AI tools and large language models

- Feel free to use them for documentation, BUT...!
  - Verify and test all results
  - Experiment with prompts to optimize
  - Provide custom instruction sets
  - Keep in mind information security
  - Not useful for newly introduced features

# Crowd-sourced peer review





## Task: Documentation for desktop sharing and remote login in GNOME

- MR: [https://gitlab.gnome.org/GNOME/gnome-user-docs/-/merge\\_requests/176](https://gitlab.gnome.org/GNOME/gnome-user-docs/-/merge_requests/176)
- Previews:
  1. Download <https://drive.google.com/drive/folders/1HpD3zhbBSuUmRIVYksaht2vM3hkHuyY4?usp=sharing> (the entire directory)
  2. Open the HTML files in a browser
- Links^ available on Discord and in the course's GitHub page
- Feedback requested especially for:
  - The titles and introductory texts
  - The "important" note in Connecting (about port numbers)

# Group peer reviews & discussions



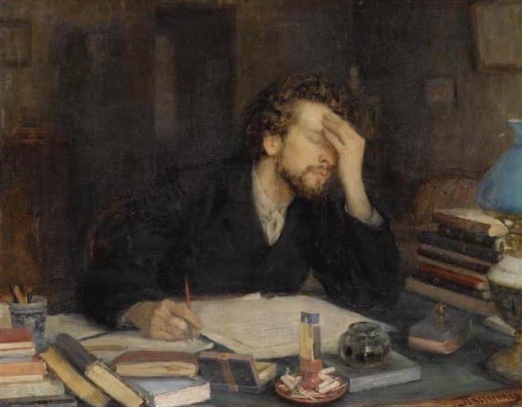
## What to discuss

- How is your work going?
- What blockers are you facing?
- Any tips & tricks that helped you (especially if they weren't mentioned in the classes)?



## Discussion retrospective

- What have you learned from your group conversations?
- What are you doing differently from others?
- What advice did you receive/give?
- Have you resolved any blockers?
- Have you hit a blocker you're still unable to resolve?



## Course retrospective

- What is the most useful thing you've learned during this course?
- Does your work on final project diverge from what you've been told at this course? If so, how?

# Q&A

## Final project deadline soon!

- Finalize the work on your project by **May 7, 9:59 AM**
  - The code must be merged or at least approved by SME
- Address feedback from peer review (if applicable)
- Review all the communication on your pull request
- Check pull request description and commit messages
- Remember: reviewers may take up to 2 workdays to get back to you, and are unlikely to respond on weekends!

# Final project requirements

- **Create documentation that sufficiently addresses the issue in the ticket**
  - Demonstrate an understanding of the subject matter and how your documentation will help to address user needs
  - Use technical writing style (focus on the UX, minimalism)
  - Use the appropriate markup correctly
- **Create a Pull Request (PR) for merging the updates and request reviews from the appropriate stakeholders**
  - Use well-structured commits and informative messages, name and label the PR correctly
  - Be direct but respectful towards the stakeholders
- **Address the feedback received from reviewers**
  - Constructive discussion with the stakeholders
  - Apply reviews correctly
- **Optional: Present your work in the final class of the course (5 min/20 points)**
  - Lessons learned and key takeaways
  - Address ensuing questions and discussion points



## Next week: Final project presentation

- Up to 20 bonus points to your final project score
- 5 - 10 minutes of presentation, ideally with slides or other visual aids
- Briefly introduce your task and its specifics (language, repo structure...)
- Focus on:
  - Problems you encountered & how you solved them
  - What tools and approaches proved useful
  - Other lessons learned
- If interested, contact any of the teachers on Discord or via email