

Diagrams & Dragons

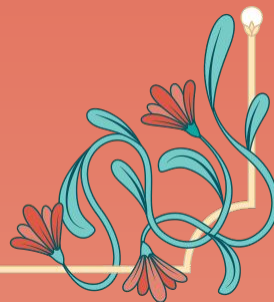
Teaching Technical Writing to University Students
the Open Source Way

Jiří Herrmann
Red Hat Czech



What we'll talk about:

- Tech writing courses for university students in Brno by Red Hat Czech tech writers
 - What we did and how we did it
 - What worked well and what didn't
 - What the challenges were
 - What the takeaways are



How it started...

(2019)



Tech writing education on the university level is really lacking in our country.

Also, it would be good for recruitment...

Pretends he knew that university education for tech writing actually exists

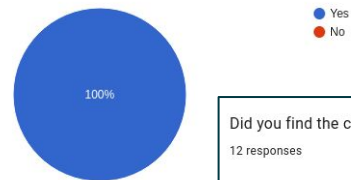


How it's going...



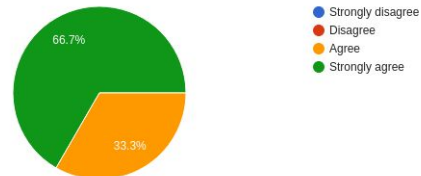
Do you think that the knowledge you gained from the course will be useful in your future career?

11 responses



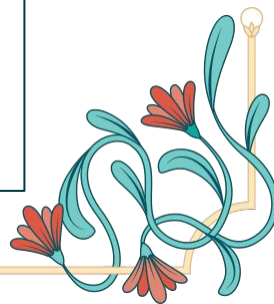
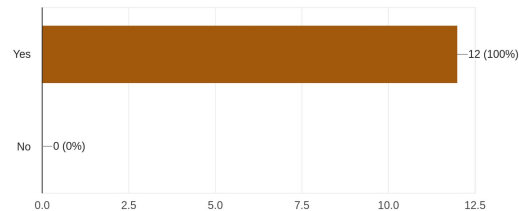
Did you find the course engaging?

12 responses



Would you recommend this technical writing course to other students?

12 responses



How we got there

- 2015 – 2020: Short workshops at the Red Hat Brno Student Open House event (mixed-major students)
- 2016: 3-day workshop for FF MUNI (postgraduate students of the “English language and literature” major)
- 2022: Lecture on tech writing as a part of a Red Hat course on contributing to open source at MUNI (IT major students)
- 2023 – 2024: Half-semester courses for FI MUNI (sophomore mixed-major students)
- 2024 – 2026: 2-week intensive courses for BUT (freshman students of the “English in electrotechnics and communication technologies” major)



What we focused on

Writing style



- Readability, findability, accessibility
- Style guides

Hard skills



- Product knowledge & research
- Cross-functional stakeholder know-how

Tooling

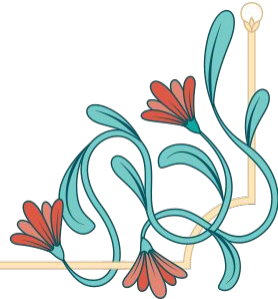


- WYSIWYG x docs as code
- Markup languages, CMSs
- Version control

Soft skills

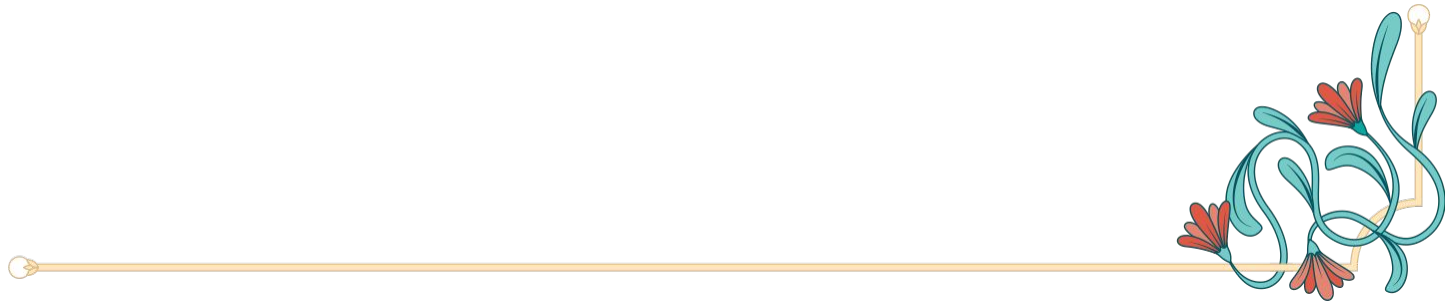


- Proactivity & negotiation
- Time management



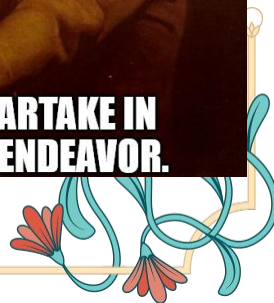
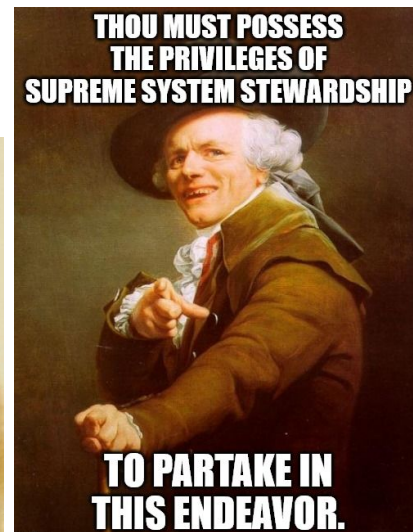
What didn't work

- Remote lecturers, different for each class
- Classic-style graded homework
- Command line-based classwork

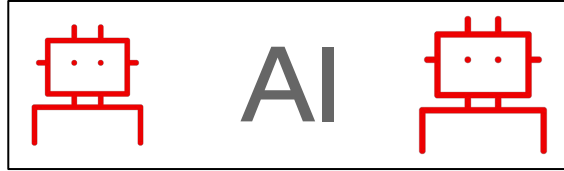


What worked well

- Semi-arbitrary point system (with secret prizes)
- Roleplay (Dungeons & Dragons-style)
- Real-world(ish) final project
- Memes!



What the biggest challenge has been

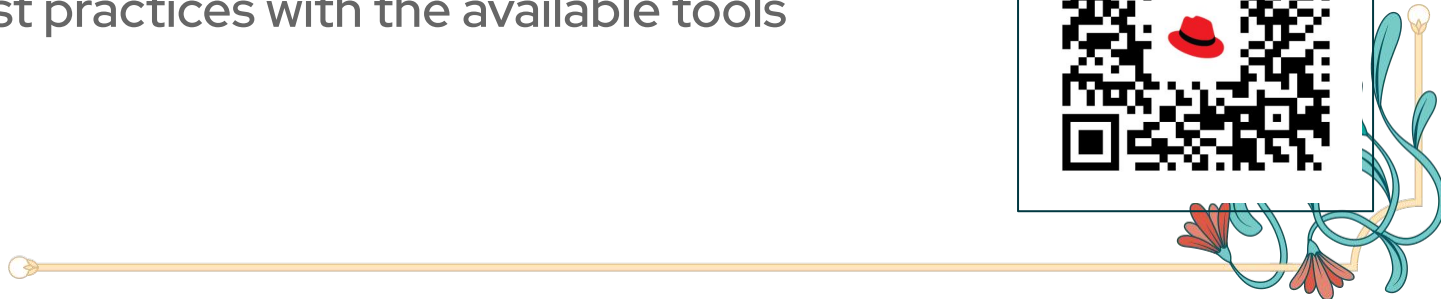


(...duh)

- > Emphasise basic skills required to review AI output
- > Demonstrate strengths and shortcomings
- > Instruct on prompt engineering
- > Show off best practices with the available tools

Notes from the
Tech Writing+AI BoF

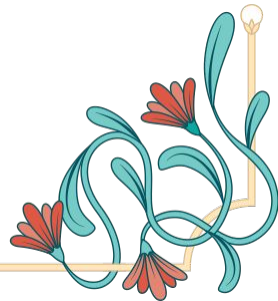
red.ht/tw-ai-oss-26



Want to see for yourself?

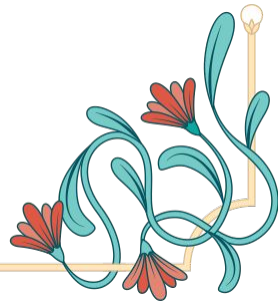
Course materials publicly available on GitHub:

- [2023 - 2024 MUNI courses](#)
- [BUT 2024 course](#)
- [BUT 2025 course](#)
- [BUT 2026 course](#)



What this all means

- Learning by doing (still) works best
- Assume your students are beginners no matter their background
- Stick to 2-3 teachers for the entire course (ideally in-person)
- Avoid tooling-heavy classwork
- Use homeworks for revision and preparation, not grading
- Gamification usually helps
- Count on your students using chatbots (even if you tell them not to)
- Tips on navigating the AI-fication of tech writing help a lot (at least for a time)



THE LINUX FOUNDATION



**EMBEDDED
LINUX
CONFERENCE**

EUROPE

