

# Collaboration and Transparency: A User-Generated Documentation for eScriptorium

–

Alix Chagué, *ALMAnaCH, Inria, Paris / Université de Montréal, Montreal / Ecole Pratique des Hautes Etudes, Paris.*

Floriane Chiffolleau, *ALMAnaCH, Inria, Paris / Université du Mans, Le Mans.*

Hugo Scheithauer, *ALMAnaCH, Inria, Paris / Ecole Pratique des Hautes Etudes, Paris.*

DH2024

Washington D.C., USA

August 7, 2024

–

{I'm presenting today a work done in collaboration with my two colleagues at Inria Paris, Floriane Chiffolleau and Hugo Scheithauer. We are presenting our solution to contribute to an open source project and its documentation. [→]}

The Digital Humanities are typically prone to producing research software. To name a few: Voyant Tools (Sinclair and Rockwell 2016) for text analysis, Recogito for semantic image annotation, or Transkribus (Muehlberger et al. 2019) for handwritten text recognition. There are multiple listings online, like the one offered by [Awesome Digital Humanities](#), and even the notorious [Digital Humanities Awards](#) organized by James Cummings gives a good idea of the productivity of the field. [→]

The software our team is particularly interested in is eScriptorium (Stokes, Kiessling, et al. 2021). It's an open source web application developed since 2019, originally by a team named eScripta, within the framework of the SCRIPTA-PSL project<sup>1</sup>, to facilitate the use of handwritten text recognition. The web application is coupled with Kraken, the software in charge of the deep-learning related features, developed by Ben Kiessling since 2015. This software is important at a time when automatic text recognition has clearly become an essential technology in the toolbox of patrimonial institutions and individuals in digital (or non-fully digital) humanities. [→]

However, eScriptorium definitely belongs to the category of “expert software”, as it offers a wide range of functionalities. This means that new users will be on a steep learning curve, having to familiarize themselves with the interface, with key concepts of machine learning and with good practices specific to automatic text recognition. To complicate the case of eScriptorium, up until last year, eScriptorium did not have a centralized user documentation. This is what Floriane, Hugo and I contributed to this project, not as members of this team (we are not) but as more or less expert users. Let's briefly see how we addressed the existing situation for the documentation, and then we'll discuss the contribution of such an initiative to accelerating the integration of open-source, research software to larger infrastructures.

The success of software extends beyond its ability to meet a specific need; it must also be welcoming to new users, which can be ensured by several means: the design of the user interface, the compatibility of the tool with the rest of the software environment, but also the availability of reliable documentation. [→]

As for eScriptorium, given the limited size of the team responsible for the development, no official extensive documentation was ever created. Instead, most of the available documentation was user-generated content scattered across the web and across various media, and was tailored to the needs of the user group which generated it.

- The code source provides a developer-oriented documentation<sup>2</sup>;
- eScripta's blog showcased key features in the form of short videos;
- A galaxy of tutorials in English<sup>3</sup>, French<sup>4</sup>, German<sup>5</sup> or Polish<sup>6</sup> written by non-members of the eScripta team was published on various websites but rarely updated<sup>7</sup>;
- Also, numerous hands-on workshops were conducted by eScripta<sup>8</sup> or by user groups sharing a specific interest in the application<sup>9</sup>.
- And lastly, a few channels, like on Gitter or via emails, allow for punctual technical discussions with contributors and users of eScriptorium.

This situation was challenging because not everyone has access to the workshops and locating tutorials or videos could be cumbersome. Plus, they might not cover all features and workflows, especially when they were not up-to-date. The absence of updates could be caused by inadequate publication formats (hard to edit), a lack of stake to do so for the editors or simply the unavailability of individuals to perform the updates, especially once the funding to a project having created a tutorial has run out. [→]

Considering the difficulty for eScripta to propose a comprehensive reference documentation (Jiang et al., 2022), we wanted to create a documentation website designed to meet three essential criteria:

- it needed to be easily maintainable since eScriptorium has yet to achieve a stable official release;
- it had to be open to external contributions;
- and it had to support multilingualism.

We consider that the key to creating an easily maintained documentation lies in its modularity, in the use of a lightweight markup language, and in a commitment to transparency. [→] We operationalized this vision by adopting the "continuous documentation" paradigm through ReadTheDocs,<sup>10</sup> with the source code openly accessible on GitHub (Chagué et al. 2023). [→]

This documentation is built from multiple Markdown text files, each addressing specific aspects of the application. When the source code is updated, ReadTheDocs activates Mkdocs, a static website builder, which builds web pages from the Markdown files. The result is published on a website served by ReadTheDocs ([escriptorium.readthedocs.io](https://escriptorium.readthedocs.io)).

For the organization of the content of the documentation, we used our experience as trainers of new users on eScriptorium. It was unthinkable to offer this documentation in the form of an extensive tutorial, so instead, we created an entry point in the form of a "quick-start"

tutorial, which allows a new user to focus only on the essential features and get familiar with the interface. The rest of the documentation is broken down into in-depth descriptions of each feature. This allows for the documentation to be a good supporting resource for training workshops while also facilitating future updates.

Choosing Markdown over more elaborate formats meant that any reader spotting a blind spot in the documentation can offer a correction by editing the source file and pushing a proposed update to the GitHub repository.<sup>11</sup> They can also more simply open an issue ticket.

It took about 4 months to have a solid first version of the documentation and as of April 2023 (eScriptorium v0.13.6), the ReadTheDocs-hosted eScriptorium documentation that we created was indeed added to eScriptorium's homepage as its official documentation. [\[→\]](#)

Our solution comes with inherent limitations. Similar to the existing resources at the time we began working on the documentation, it is susceptible to obsolescence as the developer team integrates new features. Also, as primarily users of eScriptorium ourselves, the content we propose may initially be exposed to blind spots: there are features we rarely use. Thus, a question worth exploring is that of the trustworthiness of a documentation generated by users. We think the transparency of the process and its openness to any contributions are the keys to remediate these limitations.

Our efforts focused on a first proposition written in English only, but its compatibility with multilingualism makes it possible to imagine adding a French version later, or even integrating the German and Polish tutorials. On our side, this undertaking made us realize that contributions to open-source software can come in diverse forms (Puhlfürß et al., 2022), including in the form of rationalizing its documentation.

Finally, to answer our initial question: the creation of a comprehensive reference documentation for project-generated open-source software, such as the eScriptorium documentation initiative discussed here, can serve as a catalyst for accelerated integration into larger infrastructures. Firstly, by facilitating knowledge diffusion and enhancing accessibility, (well-)documented projects break down entry barriers, ensuring that a broader audience can understand and engage with the software. Secondly, the collaborative nature of documentation projects has the potential to foster community engagement, which could lead to the creation of a network of users actively involved in the software's development and adoption (Santos & Correia, 2022). Lastly, clear documentation can also attract developers and organizations looking for software solutions that align with their existing systems. [\[→\]](#)

So we know it's possible to contribute to an open source software project with something other than code, we've discussed what we think should be considered when building a documentation, and now we do have a comprehensive documentation for eScriptorium in English. The next step is going to gather volunteers to create an actual multilingual documentation for eScriptorium.

---

[1] See <https://psl.eu/en/scripta>.

[2] The manual and Docker-based installations are described here:  
<https://gitlab.com/scripta/escriptorium/-/wikis>.

[3] See <https://lectaurep.hypotheses.org/documentation/escriptorium-tutorial-en>.

[4] See <https://lectaurep.hypotheses.org/documentation/prendre-en-main-escriptorium>.

[5] See  
[https://ub-mannheim.github.io/eScriptorium\\_Dokumentation/Nutzungsanleitung\\_eScriptorium.html](https://ub-mannheim.github.io/eScriptorium_Dokumentation/Nutzungsanleitung_eScriptorium.html).

[6] See  
[https://github.com/pjaskulski/escriptorium\\_tutorial/blob/master/escriptorium\\_tutorial.md](https://github.com/pjaskulski/escriptorium_tutorial/blob/master/escriptorium_tutorial.md).

[7] For example, the English tutorial was published in February 2021, and was referred to on the homepage of eScriptorium. However, while new features appeared on the application, it was never updated.

[8] One such workshop occurred during the DH2022 conference (Stokes and Stökl Ben Ezra 2022). Another recent workshop was held at the University of Pennsylvania (Stokes 2023).

[9] For example, the HTRomance project organized a series of three workshop at the French National Library (BNF) at the end of 2023 (<https://bnf.hypotheses.org/35711>).

[10] Refer to <https://docs.readthedocs.io/>.

[11] GitHub's Pull Request system allows for editorial control of proposed updates.

## References

- Chagué, Alix / Chiffolleau, Floriane / Scheithauer, Hugo / Carrow, Jennifer** (2023): *eScriptorium Documentation (source code)*. <https://escriptorium.readthedocs.io> [last accessed on June 6, 2024].
- Jiang, Huaxi / Zhu, Jie / Yang, Li / Liang, Geng / Zuo, Chun** (2022): *DeepRelease: Language-agnostic Release Notes Generation from Pull Requests of Open-source Software*. arXiv. <http://arxiv.org/abs/2201.06720> [last accessed on May 31, 2024].
- Muehlberger, Guenter et al.** (2019): "Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study", in: *Journal of Documentation* 75 (5): 954–976. [10.1108/JD-07-2018-0114](https://doi.org/10.1108/JD-07-2018-0114).
- Pianosi, Francesca / Sarrazin, Fanny / Wagener, Thorsten** (2020): "How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox", in: *Environmental Modelling & Software* 124: 104579. [10.1016/j.envsoft.2019.104579](https://doi.org/10.1016/j.envsoft.2019.104579).
- Puhfürß, Tim / Montgomery, Lloyd / Maalej, Walid** (2022): *An Exploratory Study of Documentation Strategies for Product Features in Popular GitHub Projects*. arXiv. <http://arxiv.org/abs/2208.01317> [last accessed on May 31, 2024].
- Santos, João / Correia, Filipe** (2022): *Patterns for Documenting Open Source Frameworks*. arXiv. <http://arxiv.org/abs/2203.13871> [last accessed on May 31, 2024].
- Sinclair, Stéfan / Rockwell, Geoffrey** (2016): *Voyant Tool*. <http://voyant-tools.org/> [last accessed on June 6, 2024].
- Stokes, Peter Anthony / Kiessling, Benjamin / Stökl Ben Ezra, Daniel / Tissot, Robin / Gargem, El Hassane** (2021): "The eScriptorium VRE for Manuscript Cultures", in: *Classics@ Journal* 18:.
- Stokes, Peter Anthony / Stökl Ben Ezra, Daniel** (2022): *Hands-on Introduction to eScriptorium, an Open-Source Platform for HTR (WT-20)*. Tokyo, Japan. <https://dh2022.adho.org/workshops-and-tutorials/wt-20> [last accessed on June 6, 2024].
- Stokes, Peter Anthony** (2023): *How to Transcribe a Million Manuscripts with eScriptorium*. in: *Penn Libraries*. <https://www.library.upenn.edu/events/escriptorium> [last accessed on June 6, 2024].