

Using Github Issues to facilitate the project communication between developers and humanists.

Case study

Szulińska, Agnieszka

agnieszka.szulinska@ibl.waw.pl
Polish Academy of Sciences, Poland

Non-coders are coming

Digital humanities (DH) is opening up to non-coding participants. In many cases we see lowering barriers to uptake for new users not only in the form of tutorials, training, workshops, but also in *low-code* and *no-code* approaches. As such, Recogito, an open source software for collaborative annotation, informs on its homepage about “Semantic Annotation without the pointy brackets” (Recogito’s Homepage 2022), which indicates that no specific programming skills are required to use it. Another example, FairCopy Editor, a tool for creating digital scholarly editions, was promoted through the statement: “Transcribing isn’t just for XML experts anymore.” (Maryl et al, 2021). Thus, a large group of non-coders is emerging and wants to contribute actively to the DH communities

Inevitable yet invisible collaboration?

Yet, the programming skill set is still needed in DH projects at some point, thus collaboration between coding and non-coding participants is still in high demand. Zundert (2022, 164) observes, that apart from (digital) humanists: “(...) development of a high-end digital edition is often a collaboration involving computer engineers, graphical interface designers, an IT project manager, a data steward, and possibly several others with additional roles”, thus certain types of outputs depend upon interdisciplinary teams in DH projects.

One of the challenges in such teamwork are, as Pawlicka-Deger (2022) notices : “not all these groups are equally visible and recognised, and labor issues are increasingly salient” - not in terms of legal aspect, but rather “a question of scholarly accountability, responsibility, and value” (Zundert 2022, 167).

Crediting their work by enlisting developers on the project’s website should be a standard procedure, nonetheless it might be shown also by capturing and opening certain stages of project’s progress, i.e. the communication workflow between team members. Github seems to be an excellent meeting place for the coding and non-coding creators, especially one of its basic functionalities called Issues.

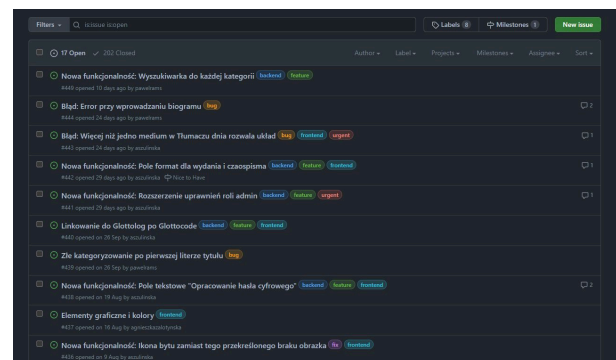
Github Issues in biblio-data lexicon and digital scholarly editions

Github Issues is a space, where digital project creators can plan their work, track potential problems and discuss new ideas Any Github user added to the project’s repository can create and comment on them openly.

In this paper, I will present two case studies using Github Issues as a core element of a collaboration between non-coding humanists and developers for collecting feedback and testing platforms in digital humanities projects. The first project, the Translator’s Lexicon, is a platform in the beta stage for digital (bio)bibliographical data projects with a focus on the translators’ work and multilingual entries.

The second one, TEI Panorama is an already-existing and operable platform for publishing digital scholarly editions, especially for Polish literary texts.

Digital humanities researchers involved in creating those platforms were obliged to formulate their observations regarding errors and new functionalities with Issues on Github, where developers answered their queries (and they could create and flag their issues too). As for the Translator’s Lexicon, 218 Issues were created (49 by non-coding participants, 169 by developers) and for TEI Panorama 70 Issues (42 by non-coding participants, 28 by developers) - and the numbers are still growing in both cases. Issues were divided into two main types: *An Error* and *A New Functionality*. Although sometimes categorization of the issue was questionable 1, in most cases it was sufficient and clear for everyone.



Picture 1: A list of issues from Translator’s Lexicon private repository

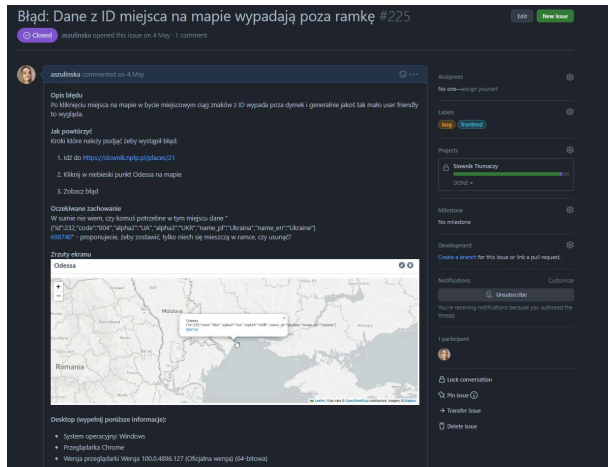
Add Github Issues to your workflow

Based on the analysis of the Github Issues for those projects I will propose a basic workflow for using this functionality in DH projects.

This short paper will consist of principles applied to this kind of collaboration, for instance: “always try to propose a solution for the error while reporting it” (for non-coders) or “check if a similar issue is not created already” (for both groups).

There will also be templates used for the two types of issues (*An Error* and *A New Functionality*), the usability of labels - like backend, frontend, pending, urgent - and milestones, connected

to the stages of the project's development in presented cases. It will also investigate potential extensions to this workflow by using Github Discussions and Github Issue forms.



Picture 2: An exemplary issue from from Translator's Lexicon private repository

Bibliography

Maryl, Maciej / Błaszczczyńska, Marta / Szulińska, Agnieszka / Buchner, Anna et al. (2021). 'OPERAS-P Deliverable D6.5: Report on the Future of Scholarly Writing in SSH'. <https://doi.org/10.5281/zenodo.4922512>.

Pawlicka-Deger, Urszula. (2022). 'Digital Humanities Needs Equality between Humanists and Technicians', July. <https://doi.org/10.5281/zenodo.7068110>.

Recogito Homepage, <https://recogito.pelagios.org/>, [23.10.2022].

Zundert, J. J. van. (2022). 'Scholarship in Interaction: Case Studies at the Intersection of Codework and Textual Scholarship'. Leiden University. <https://hdl.handle.net/1887/3464403>.