On the Structuring of LaTeX Projects

Wouter ten Brinke[1], Bart Griepsma[1], Aleksandra Ignatovič[1], Nhat[2] and Vadim Zaytsev[2]

[1]Technical Computer Science, University of Twente, Enschede, The Netherlands
[2]Formal Methods & Tools (FMT), University of Twente, Enschede, The Netherlands

Abstract

In academia, LaTeX is a powerful typesetting system widely used for producing scientific documents such as research papers, theses and reports. It allows authors significant freedom and control over the structure and styling of their documents. However, this flexibility often leads to inconsistent internal project structures and coding styles, which can hinder maintainability and collaboration among co-authors.

In this paper, we investigate various existing traditions in structuring one's LaTeX projects. By analysing **29** academic users through interviews and surveys, we uncover prevalent practices and attitudes towards standardisation. Additionally, we mine **215** LaTeX repositories from GitHub to identify structural and stylistic patterns using feature extraction and clustering techniques. Finally, we introduce FlexiTeX, a system that allows users to maintain their preferred project structures while collaborating on shared content. FlexiTeX achieves this by parsing documents into an abstract tree representation and applying configurable transformation rules. Our preliminary findings suggest that while no universal standard exists, there is space for tool support in enhancing collaboration and maintainability in LaTeX projects.

1. Introduction

LaTeX [1] is a widely used typesetting system, particularly in academia, for producing high-quality scientific documents. Its strengths lie in its ability to handle complex formatting, mathematical notations, as well as bibliographies. LaTeX allows authors significant freedom in how they structure and organise their projects, and does not enforce any standards for folder layout, file naming conventions, coding styles, etc. Publishers often make use of their own *document classes* which impose some constraints on defining meta-information (authors' names, emails, title, subtitle, affiliations) and using certain packages, as well as *bibliography styles* which dictates which fields of BibTeX entries are used and how. A very occasional journal might employ a submission system that also limits font usage or requires all content to fit in one LaTeX file. Such unabashed flexibility can lead to inconsistent practices, making it challenging for collaborators to work together effectively, if they are used to drastically different folder structures or content clustering. Inconsistencies can also hinder maintainability, as authors may struggle in the future (when working on a resubmission, a camera ready version or an extended version of the same paper) to understand or modify documents that do not follow a clear and standardised format.

Despite its widespread use, there is currently no universally accepted standard for organising LaTeX projects. Authors often develop their own conventions for file structure, naming, and coding styles. These practices are often informal, ad hoc, and can vary widely across individuals and disciplines. This lack of standardisation leads to challenges in collaborative environments, where multiple authors may have different expectations and practices. In academia, such challenges are particularly pronounced, as scientific documents often involve multiple contributors(as is often the case for research papers) and require long-term maintenance activities (common for books and PhD theses). Services such as *Overleaf* aid collaboration by supporting various build configurations and providing templates, but they do not alleviate the issues one person's neatly curated setup is another's indecipherable labyrinth to navigate.

BENEVOL'25: Proceedings of the 24th Belgium-Netherlands Software Evolution Workshop, 17–18 November 2025, Enschede, The Netherlands

w.d.c.tenbrinke@student.utwente.nl (W. ten Brinke); b.griepsma@student.utwente.nl (B. Griepsma); a.ignatovic@student.utwente.nl (A. Ignatovič); research@nhat.run (Nhat); vadim@grammarware.net (V. Zaytsev)
https://grammarware.net/ (V. Zaytsev)
0009-0004-3110-9946 (Nhat); 0000-0001-7764-4224 (V. Zaytsev)

From functions and compile errors to version control and file structuring, LaTeX is closer to a *software project* than one might think. IDEs for it do exist, but less popular, more versatile and less standardised as in software development. In this project we explore how software engineering and evolution principles can be applied and adapted to the context of LaTeX writing. By treating LaTeX projects as software projects, we can leverage established software engineering and evolution practices to improve the experience and quality of LaTeX authoring.

In this specific work, we focus on project structuring and aim to answer the following questions:

RQ1 How do academics across different disciplines structure their LaTeX projects?
RQ2 What structural patterns can be observed in real-world LaTeX repositories?
RQ3 How can we support LaTeX collaboration without sacrificing existing personal project structures?

2. Related Work

Although LaTeX has become a standard tool in academic writing, there is surprisingly little research on how users organise their LaTeX project files or whether there are best practices. Most of the existing work focuses on teaching the basics of LaTeX, promoting templates, or improving user accessibility rather than directly studying file and folder structures.

Several researchers highlight the strengths of LaTeX and its widespread use in academia. Igel emphasises the steep learning curve of LaTeX, particularly when managing bibliographic styles and formatting requirements, but notes that its open-source flexibility makes it highly adaptable [2]. Zheng discusses the advantages of LaTeX over Microsoft Word in academic settings, describing how structured workshops help users become familiar with templates and tools [3]. Both sources suggest that while LaTeX enables standardisation in output, it offers little guidance for project organisation behind the scenes.

More technical efforts show how templates can reduce confusion and error. Frank et al developed a LaTeX-based reporting workflow using modular templates and automation scripts to support reproducibility in pharmacokinetic analysis [4]. Similarly, at Carnegie Mellon University, librarians used Overleaf to collaboratively redesign internal documentation, gaining insight into file management, templating, and project clarity [5]. These initiatives illustrate how structured LaTeX setups can benefit collaboration and efficiency, especially in institutional or team contexts.

Guizani and Rodríguez-Simmonds describe the role of student-led workshops in making LaTeX more accessible and community-oriented. Their work shows how inconsistent assumptions about file organisation can create confusion, even within a single department [6]. Meanwhile, Santos et al focus on academic libraries and propose template-based standardisation aligned with national formatting standards in Brazil, advocating for librarians to support LaTeX as a formal document preparation tool [7].

In general, while no studies have yet proposed a universal LaTeX file structure, several papers recognise the value of standardisation and reusable templates. These findings support the motivation for this paper: to investigate how academic users actually structure LaTeX projects and whether informal conventions could evolve into widely accepted standards. One of the few direct calls for LaTeX standardisation is Verna's article *Towards LaTeX Coding Standards* [8], which proposes a set of informal conventions based on programming best practices. Verna highlights the inconsistency of LaTeX source files and argues for clearer structuring, modularisation, and naming. However, his proposals are not based on empirical analysis, and no large-scale studies have tested whether LaTeX authors actually follow patterns that could support a shared standard.

Several tools exist that convert LaTeX documents into other formats. Pandoc [9] supports many input and output formats and can turn LaTeX into HTML, Markdown, or DOCX. LaTeXML [10], which is used by arXiv, focuses on preserving semantic structure when rendering documents as HTML [10]. plasTeX [11] parses and interprets macros to produce detailed transformations into formats like HTML. pylatexenc [12] provides utilities for parsing LaTeX code and converting it to Unicode. Although these tools effectively extract structured information, they are designed for converting documents into other formats rather than reorganising or reformatting LaTeX while staying in the same format. They also do not preserve all commands or macros, since many are unnecessary when targeting non-LaTeX outputs.

Some services like DBLP [13], CSAuthors [14] or BibSLEIGH [15] sanitise and standardise collections of BibTeX entries. However, they all do it in their own way, and the state of the art is that each experienced BibTeX user organises their own `.bib` files as they please, and commonly edits them manually to minimise dissatisfaction. The paper on BibSLEIGH highlights some problems of maintenance of BibTeX collections (such as links expiring with each redesign of publishers' websites), plus many issues in sanitisation of available data and metadata, such as using correct and distinct symbols for μ-kernel (microkernel, hence the micro sign, U+00B5) and μ-calculus (mu calculus, hence the Greek small letter mu, U+03BC), as well as opportunities in community analysis and bridging [16]. In this paper we intentionally focus on LaTeX and leave any related and unrelated BibTeX issues out of our scope.

3. Methodology and Contributions

This research project employs a mixed-methods approach, combining qualitative and quantitative techniques to explore LaTeX project structuring practices and develop a collaborative editing system. The contributions are divided into three main components:

- We conducted a qualitative study [17] involving semi-structured interviews and surveys with 29 academic users from various academic disciplines to understand their practices and attitudes towards LaTeX project structuring and standardisation. The study revealed dynamic cultural practices, shaped partly by syntax, and partly by the habits and preferences of individual members. Despite the diverging personal workflows, their responses converged on the need for a lightweight, flexible, modular, and community-driven framework that facilitates onboarding for newcomers and collaboration among co-authors.
- We performed a quantitative analysis [18] of 215 LaTeX repositories from *GitHub*, extracting features related to project structure and coding styles, and applying clustering techniques to identify common patterns with K-means as the clustering algorithm and Principal Component Analysis (PCA) for dimensionality reduction and visualisation. Structurally, the analysis revealed a wide range of practices, with no clear standard emerging. This enforces the lack of standardisation observed in the qualitative study. Stylistically, cluster variations suggested the absence of a common coding style, with differences in comment density, line lengths, and indentation styles.
- We introduced FLEXíTEX [19], a system that allows users to maintain their preferred project structures while collaborating on shared content. FLEXíTEX achieves this by parsing documents into an abstract tree representation and applying configurable transformation rules expressed in *YAML*. The transformation is designed to be reversible and idempotent, while preserving the ability to compile the document. Two collaborative workflows were proposed, one supporting turn-based collaboration and the other enabling mergeable collaboration when combined with a tool like `git diff`. A proof-of-concept was developed and demonstrated in a *GitHub* repository, showcasing the proposed workflows. Evaluation of this implementation on 324 real-world LaTeX projects from *GitHub* showed that FLEXíTEX performed well for projects with common macro usage and typical structure, but struggled with other cases due to parser limitations. Overall, FLEXíTEX demonstrates a promising approach to flexible LaTeX collaboration, though further work is needed to improve robustness and handle edge cases.

4. Results and Findings

4.1. RQ1: How do academics structure their projects?

All the statistics and findings are hard to fit in here, but we can include a few interesting points discussed during interviews we have conducted (number of interviewees included in brackets):

- Some (10) LaTeX users prefer to minimise folder usage or just use the root folder.
- Many (22) split their project in folders per section/chapter.

- Many have a dedicated folder for references (20), figures (22), code (9), tables (6), front matter (5).
- Half (15) of interviewed users stayed in one `main.tex` file without any modularity.
- Most popular file naming conventions are `snake_case` (7) and `CamelCase` (6), as well as numerical prefixes (14).
- When asked about strategies of separating content over files (e.g., file per section), no specific strategy (4) and a "mixed approach" (6) were surprisingly popular answers.
- The majority (19) supported possible future initiatives on standardising project structures.

4.2. RQ2: What structural patterns can be observed in repositories?

First, we collect relevant repositories from GitHub using its public API and our own bespoke script, expressing the following inclusion and exclusion criteria. We set the language filter to LaTeX and limit our search to repositories that included keywords such as "Thesis" and "PhD". To avoid test files and simple templates, we exclude any repositories with keywords like "template", "example", "sample", or "class" in their titles or descriptions. This step helps us narrow the focus to projects that reflect real-world usage of LaTeX for academic writing, rather than generic or instructional codebases. Then, we enforce the minimum repository size to 1MB, which serves as a rough indicator of content richness and helps screen out projects that are either incomplete or too small to provide meaningful structural insights. We clone all repositories selected this way, to make sure the complete folder structure and all associated files are available for processing. Our dataset is publicly available to enable replications [20].

After light screening of projects with unusual patterns and outliers, we pass them to the automated feature extraction script. We extract features by basically counting everything we can think of: number of LaTeX files, of `Makefiles`, of lines per file, of characters per line, of `\input/\include` commands, on average and maximum, etc. Then we use a combination of K-Means clustering to reveal latent structures in the data, and Principal Component Analysis (PCA) to reduce dimensionality. This yields four clusters (Figure 1, left):

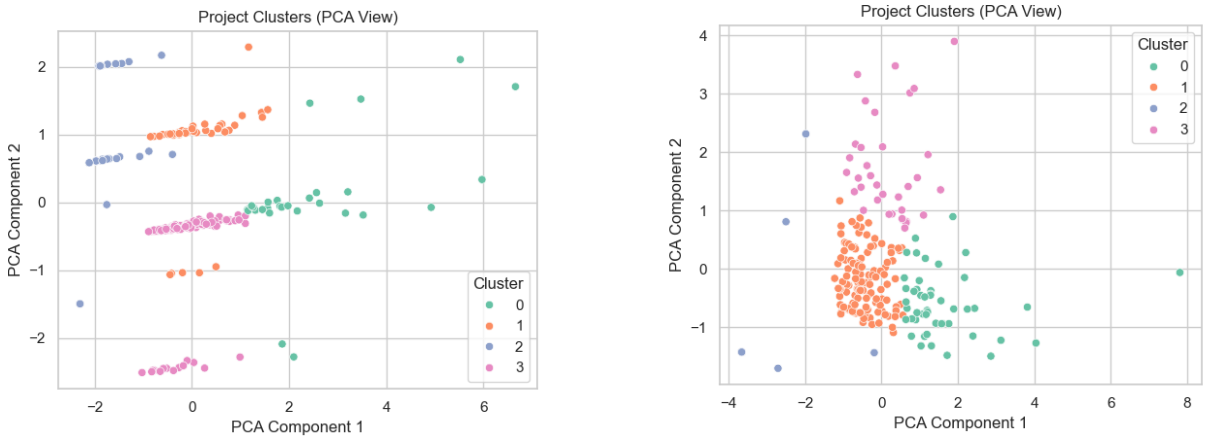

Figure 1: Clustering based on structural (left) and stylistic (right) features.

Cluster 0 stands out for its scale and complexity. Projects in this group often comprise dozens of `.tex` files spread across deep folder hierarchies, with total line counts exceeding 13000. Inclusion commands are consistently present in these repositories. In contrast, **Cluster 2** represents projects on the opposite end of the spectrum: comparatively small, averaging fewer than five `.tex` files and around 2200 lines, with shallow folder structures. *None* of the projects in Cluster 2 use inclusion commands. Between these two extremes lie Clusters 1 and 3, both of which reflect moderate project scales. **Cluster 1** projects typically contain around 21 `.tex` files and about four folders, accompanied by frequent use of inclusion commands. This group also shows a higher prevalence of Makefiles and README files. **Cluster 3**, while similar in size to Cluster 1, includes slightly fewer folders and projects, and rarely

uses Makefiles, although README files remain common. Clusters 0–2 have 30–40 projects each, while Cluster 3 has over 100 projects.

We repeat the same process focusing on different features, leading to different clusters. For instance, Figure 1, right, shows clusters based on features of readability and style. **Cluster 0** includes projects with very long lines (up to 2600 symbols), which indicates either generated content or tool support with soft newlines (like Overleaf offers). **Cluster 1** has much shorter lines (around 650 symbols) and even lower comment ratio. **Cluster 2** has lines of around 300 symbols maximum and just 24 on average, and tend to use tabs for indentation. **Cluster 3** has consistent line lengths (about 57 both for maximum and average) and has around 20% of lines in the project commented out as opposed to 5–6% in other clusters.

4.3. RQ3: How can we support LaTeX collaboration?

If Alice and Bob have each their own style of project organisation, but want to collaborate, this can be a case of applying bidirectional transformations (bx) [21] techniques, in particular forming a network of interacting bx [22]. That way, the `main` branch can host something in a "common style" with enough information to accommodate both Alice's and Bob's needs, who continue to work on their branches. Just like often the case with bx [23, 24], one can design a system based on states (*turn-based* on Figure 2) or on changes (*diff-based* on Figure 2).

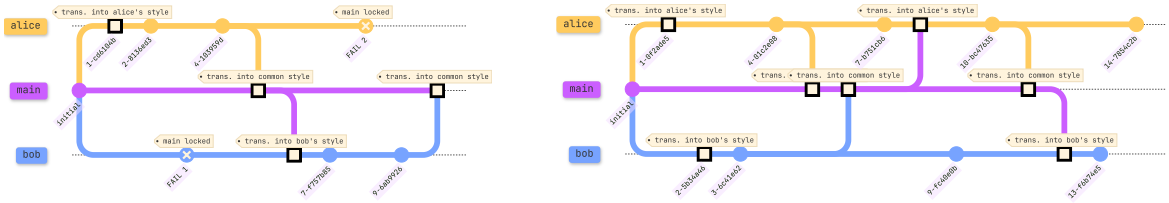

Figure 2: Proposed setups with FLEXiTEX: (left) turn-based, (right) diff-based.

We are currently conducting more experiments with FLEXiTEX [25], which is our prototype tool. It takes a configuration file specifying desired folder structure and conditions for content splitting, and is supposed to be run either locally as a git hook or remotely as an action. Preliminary results indicate that it can handle many real projects, but in particular tracking paths to files which are referred to through bespoke LaTeX commands, remains a challenge.

5. Concluding Remarks

In this work we explored how project structuring practices in LaTeX can be understood and improved through the lens of software engineering and evolution using a mixed-methods approach. Both the qualitative and quantitative analyses revealed a lack of universal standards, but also highlighted emerging informal conventions that could inform the development of flexible, community-driven guidelines. Building on these insights, we introduced FLEXiTEX, a system that enables users to maintain their preferred project structures while collaborating on shared content. For more information about the parts of this project, we advise consulting corresponding Bachelor theses that formed the core of this research [17, 18, 19].

Declaration on Generative AI

The authors have not employed any Generative AI tools to create, change or rephrase the content of this document.

References

[1] D. Knuth, L. Lamport, et al., LaTeX — A Document Preparation System, https://www.latex-project.org, 1984.

[2] C. Igel, Academic Writing with LaTeX, 2019. doi:10.22541/au.156080179.95968195.

[3] Y. Zheng, Academic Writing by Using LaTeX: A Hands-on Workshop, in: Proceedings of the 24th Annual Conference on Information Technology Education, SIGITE, ACM, New York, NY, USA, 2023, p. 90–91. doi:10.1145/3585059.3611425.

[4] T. Frank, S. Gastine, K. Lindauer, H. Speth, A. Strougo, A. Kovar, LaTeX Tutorial for the Standardization and Automation of Population Analysis Reports, CPT: Pharmacometrics & Systems Pharmacology 10 (2021) 1310–1322. doi:https://doi.org/10.1002/psp4.12705.

[5] H. C. Gunderman, D. Scherer, K. Behrman, Leveraging Library Technology Resources for Internal Projects, Outreach, and Engagement: A Case Study of Overleaf, LaTeX, and the KiltHub Institutional Repository Service at Carnegie Mellon University Libraries, College & Undergraduate Libraries 27 (2020) 164–175. doi:10.1080/10691316.2021.1885549.

[6] N. Guizani, H. E. Rodriguez-Simmonds, Developing Personal and Community Graduate Student Growth through the Implementation of a LaTeX Workshop, in: 2016 ASEE Annual Conference & Exposition, ASEE Conferences, New Orleans, Louisiana, 2016. doi:10.18260/p.26768.

[7] F. E. P. Santos, J. S. Lima, E. M. Rodrigues, I. L. d. Santos, K. Y. S. Feitosa, Desafios e possibilidades da atividade mediadora do bibliotecário na normalização de trabalhos acadêmicos: o uso do LaTeX, InCID: Revista de Ciência da Informação e Documentação 9 (2018) 25–51. doi:10.11606/issn.2178-2075.v9i1p25-51.

[8] D. Verna, Towards LaTeX Coding Standards, TUGboat 32 (2011) 309–328. URL: https://www.tug.org/TUGboat/tb32-3/tb102verna.pdf.

[9] J. MacFarlane, A. Krewinkel, J. Rosenthal, Pandoc, GPL-2.0 license, https://github.com/jgm/pandoc, 2006.

[10] B. R. Miller, D. Ginev, LaTeXML, National Institute of Standards and Technology, Public Domain license, https://math.nist.gov/~BMiller/LaTeXML/, 2004.

[11] K. Smith, PlasTeX, As-is license, https://github.com/plastex/plastex, 2007.

[12] P. Faist, Pylatexenc, MIT license, https://github.com/phfaist/pylatexenc, 2015.

[13] M. Ley, DBLP: Computer Science Bibliography, https://dblp.org, 1993.

[14] R. P. Barazzutti, CSAuthors.Net, https://www.csauthors.net, 2014.

[15] V. Zaytsev, BibSLEIGH, http://bibtex.github.io, 2015.

[16] V. Zaytsev, BibSLEIGH: Bibliography of Software (Language) Engineering in Generated Hypertext, in: A. H. Bagge, T. Mens, H. Osman (Eds.), Post-proceedings of the Eighth Seminar in Series on Advanced Techniques and Tools for Software Evolution (SATToSE 2015), volume 1820 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 54–64. URL: http://ceur-ws.org/Vol-1820/paper-06.pdf.

[17] A. Ignatovič, How Academics Organize LaTeX Projects — and Whether Structure Should Be Standardized, Bachelor's thesis, Universiteit Twente, Enschede, The Netherlands, 2025. URL: http://purl.utwente.nl/essays/107820.

[18] B. Griepsma, Can We Standardize LaTeX? Discovering Patterns in Real-World Repositories, Bachelor's thesis, Universiteit Twente, Enschede, The Netherlands, 2025. URL: http://purl.utwente.nl/essays/107264.

[19] W. ten Brinke, FLEXíTEX: LaTeX Collaboration Without Giving Up Personal Project Structure, Bachelor's thesis, Universiteit Twente, Enschede, The Netherlands, 2025. URL: http://purl.utwente.nl/essays/107262.

[20] B. Griepsma, LaTeX Academic Dataset, CC-0 license, https://github.com/Bart0TW/LaTeX_academic_dataset, 2025.

[21] K. Matsuda, R. Eramo, M. Johnson, V. Zaytsev (Eds.), Bidirectional Transformations — Foundations and Applications, National Institute of Informatics, 2025. URL: https://shonan.nii.ac.jp/seminars/231/.

[22] H. Giese, G. Karsai, V. Zaytsev, WG4: Multiple Interacting Bidirectional Transformations, in: A. Cleve, E. Kindler, P. Stevens, V. Zaytsev (Eds.), Report from Dagstuhl Seminar 18491 on Multidirectional Transformations and Synchronisations (MX Dagstuhl), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019, pp. 10–11.

[23] Z. Diskin, Y. Xiong, K. Czarnecki, From State- to Delta-Based Bidirectional Model Transformations: the Asymmetric Case, Journal of Object Technology 10 (2011) 6: 1–25. doi:10.5381/JOT.2011.10.1.A6.

[24] Z. Diskin, Y. Xiong, K. Czarnecki, H. Ehrig, F. Hermann, F. Orejas, From State- to Delta-Based Bidirectional Model Transformations: The Symmetric Case, in: J. Whittle, T. Clark, T. Kühne (Eds.), Proceedings of the 14th International Conference on Model Driven Engineering Languages and Systems (MoDELS), volume 6981 of *LNCS*, Springer, 2011, pp. 304–318. doi:10.1007/978-3-642-24485-8_22.

[25] W. ten Brinke, FLEXITEX, MIT License, https://github.com/wtb04/FlexiTeX, 2025.