


The cs-techrep Example of its CTAN Package and \LaTeX Class: TechRep in Computer Science

Technical Report: CL-TR-2025-99, January 2025


Christoph P. Neumann 

CyberLytics-Lab at the Department of Electrical Engineering, Media, and Computer Science
Ostbayerische Technische Hochschule Amberg-Weiden
Amberg, Germany


Abstract—This paper demonstrates an example of a technical report in computer science or software engineering, based on the cs-techrep \LaTeX class. The example is intended for beginners, e. g., undergraduate students. It contains a basic outline template and usually fills it with dummy text, but some sections are describing the intent of the outline template and its sections. Graphic exclamation marks highlight important remarks. {  The abstract does neither mention a thesis in which context a technical report is written nor an institution or any other organizational aspects. It is a summary of the content of the technical report, thus, usually the objectives and architecture of a piece of software. Do NOT remove the abstract, this section is mandatory. Do NOT use special characters, symbols, or math in your title or abstract. Do NOT use references in the abstract, avoid abbreviations or acronyms. The abstract must look as one paragraph only. Ideally, end the abstract with one sentence stressing out the main output of the paper. }


Index Terms—template; lorem ipsum.

I. INTRODUCTION AND OBJECTIVES | FUNCTIONAL REQUIREMENTS | PROBLEM STATEMENT

The cs-techrep formatting is adopted both from IEEE [1] and IARIA [2] styles. The cs-techrep \LaTeX class is based on IEEEtran class [3]. In addition, be aware of the supplementary IARIA editorial rules [4]  that provide a beginner-friendly set of further advices. It is recommended to use a grammar tool, e. g., the LanguageTool [5] browser plugin in combination with Overleaf [6].

The pipe symbol “|” in the headings represents alternatives. Choose one and remove the others. The selectively provided quoted terms are special German alternatives.

The problem statement needs to be written from perspective of a subject-matter expert (“Fachkonzept”). Like an elevator pitch / mission statement . NOT from a technical perspective.

{  End the first section with a paragraph describing the structure of the paper. }

II. OPTIONAL: RELATED WORK | STATE OF THE ART | METHODS | DATA ACQUISITION

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consetetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum

placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

III. ARCHITECTURAL GOALS

Provides (1) a visualization of the external systems and users with which the system interacts (“Kontextabgrenzung”), (2) the most important technical and organizational preconditions (“Rahmenbedingungen”), (3) quality/non-functional requirements (“Qualitätsziele”), and/or (4) architectural style design decisions with formative patterns of the solution (“Architekturstil”) as well as (5) the applied programming language.

IV. ARCHITECTURE OF FANCYNAME | RESULTS | STRUCTURAL DESIGN | “BAUSTEINSICHT”

A. Technology Stack | Overall System

Provides (1) design decisions based on the previously defined requirements and (2) a visualization of the functional structure at top level including relationships (“Grobe Zerlegung”), thus, gives an overview on modules, frameworks, and middleware.

In discussions of multi-tier architecture, layer is often used interchangeably – and mistakenly – for tier. They aren’t the same. A “layer” refers to a functional division of the software, but a “tier” refers to a functional division of the software that runs on infrastructure separate from the other divisions. The Contacts app on your phone, for example, is a three-layer application, but a single-tier application, because all three layers run on your phone.

In discussions concerning multi-tier architecture, the term “layer” is frequently misused interchangeably with “tier”, despite their distinct meanings. A layer denotes a functional partition within the software, whereas a tier signifies a functional division that operates on separate infrastructure from other divisions/tiers. For instance, the Camera app or Settings app on your phone exemplifies a three-layer application but remains a single-tier application since all three layers run on your phone.

B. Presentation Tier | Frontend

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum,

magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

C. Application Tier | Backend | “Anwendungskern”

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

D. Data Tier | Persistence

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

E. Optional: Infrastructure and Deployment | Distribution Perspective | “Verteilungssicht”

Provides (1) information about configuration, exact software versions, SBOM, DevOps, Cloud, AWS, and others. Should add (2) security-related considerations or disclaimers. Could include (3) a software bill of materials (SBOM), at least for the major libraries or frameworks.

V. DISCUSSION | EVALUATION | LESSONS LEARNED | IMPEDIMENTS

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

VI. CONCLUSION AND FUTURE WORK | “FAZIT UND AUSBLICK”

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

{ **▲** Beginners’ guide against **kindergarten mistakes**: Expand each acronym only when first used; then use only the acronym (however, do not take the abstract into account of this rule). Insert at least a sentence between a section title and subsection title! In-text references (i.e., “text [x][y]...”), leave a space between “text” and “[x]”! Prefer references to footnotes. Explain abbreviations in the paper text. All “section x”, “figure x”, “table x” must be “Section x”, “Figure x”, “Table x” (i.e., capital letter)! Across the entire paper; always e.g., and i.e., are followed by a comma “,”! Across the entire paper; “which” is preceded by a comma “,” or replace a which-without-a-preceding-comma with “that”. Check your raster images for appropriate quality! }

{ **▲** You must not leave the bibliography blank! Add appropriate references to your related work. } A selection of previous technical reports of the CyberLytics lab [7–14] are included as reference and further example.

REFERENCES

- [1] IEEE. *Conference Template and Formatting Specifications*. 2018. URL: <https://www.ieee.org/content/dam/ieee-org/ieee/web/org/conferences/Conference-template-A4.doc>.
- [2] IARIA. *Formatting Rules*. 2014. URL: <http://www.iaria.org/formatting.doc>.
- [3] Michael Shell. *How to Use the IEEEtran L^AT_EX Class*. 2015. URL: http://mirrors.ctan.org/macros/latex/contrib/IEEEtran/IEEEtran_HOWTO.pdf.
- [4] IARIA. *Editorial Rules*. 2009. URL: <https://www.iaria.org/editorialrules.html>.
- [5] LanguageTooler GmbH. *LanguageTool: AI-based grammar checker*. URL: <https://languagetool.org/overleaf>.
- [6] Digital Science UK Limited. *Overleaf*. URL: <https://www.overleaf.com>.
- [7] Andreas Hecht, Linus Heise, Oliver Kneidl, Eva-Maria Maurer, and Christoph P. Neumann. *StockSentinel: AI-Powered Web Tool for Analyzing the Markets Perception of Stocks*. Technical Reports CL-2024-07. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, July 2024. DOI: 10.13140/RG.2.2.20457.30564.
- [8] Jakob Götz, Uwe Kölbel, Maximilian Schlosser, Oliver Schmidts, Jan Schuster, Philipp Seufert, Fabian Wagner, and Christoph P. Neumann. *Nautical Nonsense: Eine Phaser3- und FastAPI-basierte Webanwendung für Schiffe-Versenken mit Cloud-Deployment*. Tech. rep. CL-2023-07. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, July 2023. DOI: 10.13140/RG.2.2.17156.09601.

- [9] Rebecca Kietzer, Baran Baygin, Carl Küschall, Jonathan Okorafor, Luca Käsmann, Michael Zimmet, Michael Ippisch, and Christoph P. Neumann. *Stockbird: Eine React-basierte Webanwendung mit serverless Cloud-Deployment zur Analyse des Einfluss von Tweets auf Aktienkurs-Schwankungen*. Tech. rep. CL-2023-04. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, July 2023. DOI: 10.13140/RG.2.2.32675.02083.
- [10] Anastasia Chernysheva, Jakob Götz, Ardian Imeraj, Patrice Korinth, Philipp Stangl, and Christoph P. Neumann. *SGDb Semantic Video Game Database: Svelte- und Ontotext-basierte Webanwendung mit einer Graphen-Suche für Videospiele*. Tech. rep. CL-2023-02. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, Mar. 2023. DOI: 10.13140/RG.2.2.11272.60160.
- [11] Tobias Bauer, Fabian Beer, Daniel Holl, Ardian Imeraj, Konrad Schweiger, Philipp Stangl, Wolfgang Weigl, and Christoph P. Neumann. *Reddiment: Eine SvelteKit- und ElasticSearch-basierte Reddit Sentiment-Analyse*. Tech. rep. CL-2022-06. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, July 2022. DOI: 10.13140/RG.2.2.32244.12161.
- [12] Tobias Bauer, Albert Hahn, Lukas Kleinlein, Nicolas Proske, Leonard Wöllmer, Andrei Trukhin, and Christoph P. Neumann. *Covidash: Eine MEAN-Variation-basierte Webanwendung für Inzidenz-Zahlen und Impffortschritt in Deutschland*. Tech. rep. CL-2021-06. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, July 2021. DOI: 10.13140/RG.2.2.33921.84321.
- [13] Cameron Barbee, Tim Hoffmann, Christian Piffel, Tobias Schotter, Sebastian Schuscha, Philipp Stangl, Thomas Stangl, and Christoph P. Neumann. *FireForceDefense: Graphisches Tower-Defense-Spiel mit Kubernetes-Deployment*. Tech. rep. CL-2021-05. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, July 2021. DOI: 10.13140/RG.2.2.20500.07048.
- [14] Egidia Cenko, Madina Kamalova, Matthias Schön, Christoph Schuster, Andrei Trukhin, and Christoph P. Neumann. *Med-Planner: Eine Angular- und Django-basierte Webanwendung um ärztliche Termine übersichtlich zu verwalten*. Tech. rep. CL-2021-04. Ostbayerische Technische Hochschule Amberg-Weiden, CyberLytics-Lab an der Fakultät Elektrotechnik, Medien und Informatik, July 2021. DOI: 10.13140/RG.2.2.19409.71528.