Bridging Software Visualization and Human-Centered Learning Analytics

Birte Heinemann, Jasmin Hartanto and Matthias Ehlenz
RWTH Aachen University
heinemann@cs.rwth-aachen.de | hartanto@cs.rwth-aachen.de | ehlenz@lbz.rwth-aachen.de

ABSTRACT: Version control systems like GitLab are widely used in programming courses. They provide rich data, e.g. about collaborative learning, but often remain underexplored for learning analytics. This poster investigates how data from Git repositories can be used to support learning analytics in programming education. Building on prior work in software visualization and xAPI-based data collection, we investigate how insights from professional software development can inform the design of dashboards for educators and learners. These foundation highlight gaps in existing analytics approaches and point toward new educational perspectives. To address these gaps, we will present first prototypical visualizations on the poster, combining GitLab and learning management system data through xAPI, enabling cross-tool analyses of programming and reflection activities in the future.

Keywords: Software Visualization, Learning Analytics, GitLab, Software Evolution, xAPI.

1 INTRODUCTION

Software development environments such as GitLab and GitHub generate vast amounts of data that reflect not only technical progress but also learning processes. Every commit, pull request, or code review provides insight into how developers acquire skills, collaborate, and solve problems over time. This makes software engineering a data-rich domain in which methods from Learning Analytics (LA) can be meaningfully applied (Ehlenz & Röpke, 2024). In software visualization research, numerous approaches have been developed to understand how software evolves. However, these methods primarily focus on the system —the code, the artifacts, and the workflows—while largely neglecting the developers and their professional learning processes. In contrast, Human-Centered Learning Analytics (HCLA) places the human at the center of analysis, aiming to make learning visible and actionable through data-driven feedback, reflection, and guidance.

The poster explores how LA concepts, such as dashboards, explainable visualizations, and storytelling approaches, can be transferred to the context of software development. The poster will present visualization prototypes and outlines a set of research questions for connecting the monitoring of software evolution with human-centered analytics. By combining insights from software engineering and LA, we aim to foster a deeper understanding of how developers learn through practice, mentoring and collaboration.

2 FROM SOFTWARE VISUALIZATION TO HCLA

A growing body of research in software engineering visualization explores how development processes can be represented and understood through data. Classical approaches to monitoring software development, such as those by Armenti and Lanza (2024), aim to improve the understanding

Creative Commons License, Attribution - NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0)

of software evolution at the commit level. Similar to LA, these approaches explore the use of interactive and accessible visualizations to make complex processes more understandable. However, approaches from software engineering focus primarily on the software itself. At the process level, and without an explicit educational perspective, Schreiber et al. (2024) developed an interactive provenance-based dashboard that visualizes software development as a network of entities (artifacts such as files, issues, releases), activities (commits, builds), and agents (developers, tools). Their system aims to make relationships and dependencies transparent, helping project managers and developers monitor workflows, collaboration, and complexity. A complementary narrative approach is presented by Fiechter et al. (2021), who introduce Issue Tales—visual narratives that link issues, pull requests, commits, and contributors to depict the history of software development as interconnected stories. Their graph-based visualizations make dynamic relationships visible that are otherwise hidden in textual issue threads. These visualizations are designed for exploration and analysis, rather than for educational purposes.

Together, these examples demonstrate that software visualization research provides powerful methods to understand the evolution of software artifacts. However, these approaches focus on what happens to the code rather than on how developers learn and collaborate. Bridging software visualization and LA therefore opens up a range of new perspectives — for example, examining the professionalization processes of developers during studies or vocational training. Developed LA tools could also be used to study the effectiveness of AI-powered coding buddies (Nutalapati et al., 2024) or to analyze the interaction between learning material and version control systems. Likewise, methods for teaching programming – such as peer instruction or pair programming (Brown and Wilson, 2018) - could be investigated through a LA lens.

Moreover, LA approaches can be extended to other phases of the professionalism process—for instance, when a project is initiated, when newcomers are onboarded into a project, or when development is carried out collaboratively. Fagerholm et al. (2014) examined how mentoring and project characteristics influence factors such as a developer's first commit or first pull request. Hatley et al. (2018) proposed a model of design collaboration with mentoring that could be evaluated in entirely new ways through LA analyses. Their case study showed that mentoring and guidance play a crucial role in the professionalization of developers, concluding that these interactions deserve closer examination. Furthermore, in the field of mentoring, questions arise as to how far onboarding processes for new developers can be supported—or even replaced—by large language models (Azanza et al., 2024).

3 CONCLUSION AND OUTLOOK

Integrating LA with software engineering visualization offers the opportunity to shift the analytical focus from the software system to the learners and developers behind it. While existing visualization approaches successfully reveal how code evolves, a human-centered LA perspective can illuminate how developers learn, collaborate, and grow within these processes.

LA can enrich software visualization in several ways. First, by introducing reflective dashboards that combine process and learning data, enabling developers, mentors, and educators to interpret their coding behavior in relation to feedback, collaboration patterns, or mentoring interactions. Second, by linking cross-system data sources through standards such as xAPI, allowing activity traces from

Creative Commons License, Attribution - NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0)

GitLab to be analyzed together with information from learning management systems or coding assistants. Recent work by Ehlenz and Röpke (2024) demonstrates the potential of a human-centered design approach to collect and transform such GitLab data into xAPI statements, enabling more accessible and interoperable LA infrastructures. Finally, by applying explainable AI techniques or storytelling (Martinez-Maldonado et al., 2020), complex analytics can be transformed into interpretable insights that foster reflection.

This human-centered view opens up new possibilities for onboarding, mentoring, and developer professionalization in educational and professional contexts. By combining the analytical precision of software engineering with the interpretive depth of LA, future dashboards could help both educators and developers understand not only how software evolves—but how learning happens through software development itself.

REFERENCES

Azanza, M., Pereira, J., Irastorza, A., & Galdos, A. (2024). Can LLMs Facilitate Onboarding Software Developers? An Ongoing Industrial Case Study. 2024 36th International Conference on Software Engineering Education and Training (CSEE&T), 1–6. https://doi.org/10.1109/CSEET62301.2024.10662989

Brown, N. C. C., & Wilson, G. (2018). Ten quick tips for teaching programming. PLOS Computational Biology, 14(4), e1006023. https://doi.org/10.1371/journal.pcbi.1006023

Ehlenz, M., & Röpke, R. C. (2025). A Human-centered Approach on Collecting Learning Analytics Insights in GitLab. In Companion Proceedings of the 15th International Conference on Learning Analytics & Knowledge (LAK25). Society of Learning Analytics Research (SoLAR).

Fagerholm, F., Guinea, A. S., Münch, J., & Borenstein, J. (2014). The role of mentoring and project characteristics for onboarding in open source software projects. Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 1–10. https://doi.org/10.1145/2652524.2652540

Fiechter, A., Minelli, R., Nagy, C., & Lanza, M. (2021). Visualizing GitHub Issues. 2021 Working Conference on Software Visualization (VISSOFT), 155–159. https://doi.org/10.1109/VISSOFT52517.2021.00030

Hatley, L., Al-Freih, M., & Bannan, B. (2018). "I'm Just Guiding You": An Exploration of Software Design Mentorship within a Software Engineering Firm. Journal of Software Engineering and Applications, 11(6), 299–317. https://doi.org/10.4236/jsea.2018.116019

Martinez-Maldonado, R., Echeverria, V., Fernandez Nieto, G., & Buckingham Shum, S. (2020). From Data to Insights: A Layered Storytelling Approach for Multimodal Learning Analytics. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 1–15. https://doi.org/10.1145/3313831.3376148

Nutalapati, H., Velmurugan, S., & Tiglao, N. M. (2024). Coding Buddy: An Adaptive AI-Powered Platform for Personalized Learning. 2024 International Symposium on Networks, Computers and Communications (ISNCC), 1–6. https://doi.org/10.1109/ISNCC62547.2024.10759044

Schreiber, A., von Kurnatowski, L., Meinecke, A., & de Boer, C. (2024). Visualization of Software Development Provenance. In H. Mori & Y. Asahi (Hrsg.), Human Interface and the Management of Information (S. 121–139). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-60114-9_10

Creative Commons License, Attribution - NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0)