

## DAFTAR PUSTAKA

- [1] L. Gazzola, D. Micucci, and L. Mariani, "Automatic Software Repair: A Survey," *IEEE Transactions on Software Engineering*, vol. 45, no. 1, pp. 34–67, 2019, doi: 10.1109/TSE.2017.2755013.
- [2] H. D. T. Nguyen, D. Qi, A. Roychoudhury, and S. Chandra, "SemFix: Program repair via semantic analysis," in *2013 35th International Conference on Software Engineering (ICSE)*, 2013, pp. 772–781. doi: 10.1109/ICSE.2013.6606623.
- [3] K. Liu *et al.*, "A critical review on the evaluation of automated program repair systems," *Journal of Systems and Software*, vol. 171, p. 110817, 2021, doi: <https://doi.org/10.1016/j.jss.2020.110817>.
- [4] C. S. Xia, Y. Wei, and L. Zhang, "Automated Program Repair in the Era of Large Pre-trained Language Models," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023, pp. 1482–1494. doi: 10.1109/ICSE48619.2023.00129.
- [5] Q. Zhang, C. Fang, Y. Ma, W. Sun, and Z. Chen, "A Survey of Learning-based Automated Program Repair," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 2, Dec. 2023, doi: 10.1145/3631974.
- [6] L. Belzner, T. Gabor, and M. Wirsing, "Large Language Model Assisted Software Engineering: Prospects, Challenges, and a Case Study," in *Bridging the Gap Between AI and Reality: First International Conference, AISoLA 2023, Crete, Greece, October 23–28, 2023, Proceedings*, Berlin, Heidelberg: Springer-Verlag, 2023, pp. 355–374. doi: 10.1007/978-3-031-46002-9\_23.
- [7] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly," *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024, doi: <https://doi.org/10.1016/j.hcc.2024.100211>.
- [8] C. S. Xia and L. Zhang, "Less training, more repairing please: revisiting automated program repair via zero-shot learning," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, in ESEC/FSE 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 959–971. doi: 10.1145/3540250.3549101.
- [9] R. Just, D. Jalali, and M. D. Ernst, "Defects4J: a database of existing faults to enable controlled testing studies for Java programs," in *Proceedings of the 2014 International Symposium on Software Testing and Analysis*, in ISSTA 2014. New York, NY, USA: Association for Computing Machinery, 2014, pp. 437–440. doi: 10.1145/2610384.2628055.

- [10] C. S. Xia and L. Zhang, “Conversational Automated Program Repair,” 2023. [Online]. Available: <https://arxiv.org/abs/2301.13246>
- [11] J. S. Andersen and W. Maalej, “Design Patterns for Machine Learning-Based Systems With Humans in the Loop,” *IEEE Softw*, vol. 41, no. 4, pp. 151–159, 2024, doi: 10.1109/MS.2023.3340256.
- [12] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, “Challenges and Applications of Large Language Models,” 2023. [Online]. Available: <https://arxiv.org/abs/2307.10169>
- [13] H. Cheng *et al.*, “Generative AI for Requirements Engineering: A Systematic Literature Review,” 2025. [Online]. Available: <https://arxiv.org/abs/2409.06741>
- [14] Y. Hu, U. Z. Ahmed, S. Mechtaev, B. Leong, and A. Roychoudhury, “Re-Factoring Based Program Repair Applied to Programming Assignments,” in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2019, pp. 388–398. doi: 10.1109/ASE.2019.00044.
- [15] D. Lin, J. Koppel, A. Chen, and A. Solar-Lezama, “QuixBugs: a multi-lingual program repair benchmark set based on the quixey challenge,” in *Proceedings Companion of the 2017 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*, in *SPLASH Companion 2017*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 55–56. doi: 10.1145/3135932.3135941.
- [16] D. Williams *et al.*, “User-Centric Deployment of Automated Program Repair at Bloomberg,” in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*, in *ICSE-SEIP ’24*. New York, NY, USA: Association for Computing Machinery, 2024, pp. 81–91. doi: 10.1145/3639477.3639756.
- [17] J. Zhang *et al.*, “PyDex: Repairing Bugs in Introductory Python Assignments using LLMs,” *Proc. ACM Program. Lang.*, vol. 8, no. OOPSLA1, Apr. 2024, doi: 10.1145/3649850.
- [18] T. Phung *et al.*, “Generating High-Precision Feedback for Programming Syntax Errors using Large Language Models,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.04662>
- [19] C. Le Goues, M. Pradel, and A. Roychoudhury, “Automated program repair,” *Commun. ACM*, vol. 62, no. 12, pp. 56–65, Nov. 2019, doi: 10.1145/3318162.
- [20] K. Huang *et al.*, “A Survey on Automated Program Repair Techniques,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.18184>
- [21] A. Fan *et al.*, “Large Language Models for Software Engineering: Survey and Open Problems,” in *2023 IEEE/ACM International Conference on*

*Software Engineering: Future of Software Engineering (ICSE-FoSE)*, Los Alamitos, CA, USA: IEEE Computer Society, May 2023, pp. 31–53. doi: 10.1109/ICSE-FoSE59343.2023.00008.

- [22] OpenAI, “GPT-4 Technical Report,” 2024. [Online]. Available: <https://arxiv.org/abs/2303.08774>
- [23] H. Touvron *et al.*, “LLaMA: Open and Efficient Foundation Language Models,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.13971>
- [24] L. Caruccio, S. Cirillo, G. Polese, G. Solimando, S. Sundaramurthy, and G. Tortora, “Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach,” *Intelligent Systems with Applications*, vol. 21, p. 200336, 2024.
- [25] G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, 2015, pp. 358–368. doi: 10.1109/ICSE.2015.55.
- [26] F. Zampetti, S. Geremia, G. Bavota, and M. Di Penta, “CI/CD Pipelines Evolution and Restructuring: A Qualitative and Quantitative Study,” in *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2021, pp. 471–482. doi: 10.1109/ICSME52107.2021.00048.
- [27] T. Kinsman, M. Wessel, M. A. Gerosa, and C. Treude, “How Do Software Developers Use GitHub Actions to Automate Their Workflows?,” in *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, 2021, pp. 420–431. doi: 10.1109/MSR52588.2021.00054.
- [28] J. White *et al.*, “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.11382>
- [29] C. Wohlin *et al.*, *Experimentation in software engineering*, vol. 236. Springer, 2012.
- [30] H. Krekel, B. Oliveira, R. Pfannschmidt, F. Bruynooghe, B. Laughner, and F. Bruhin, “pytest x.y,” 2004. [Online]. Available: <https://github.com/pytest-dev/pytest>