

HIVEMIND: Human-centred collaborative multi-agent framework for accelerating software development and maintenance

Alba Bonet-Jover^{1,*†}, Alicia Picazo-Izquierdo^{2†}, Santiago Massé Palomo^{3†},
Aitana Delgado Cabanillas^{3†}, Andrés Montoyo^{1†}, Estela Saquete^{1†},
Patricio Martínez-Barco^{1†} and Daniel García Guirao^{3†}

¹Department of Software and Computing Systems, University of Alicante, Spain

²University of Alicante, Spain

³IDENER.AI, Spain

Abstract

HIVEMIND is a European project aimed at advancing the entire Software Development Lifecycle (SDLC) through an ethical and responsible human-centered approach. The project seeks to introduce a multi-agent environment based on Large Language Models (LLMs), Artificial Intelligence (AI), and data technologies, incorporating a Human-in-the-Loop (HITL) approach that enables collaboration between humans and AI agents. This ambitious project is led by a multidisciplinary consortium composed of 13 experts from the business, academic, and research sectors across 8 countries, all specialized in Software Engineering (SE) and AI. The HIVEMIND project promotes the importance of strengthening collaboration between humans and agents through a balanced approach, where the Human in the loop becomes an essential stage in many phases of development within the multi-agent environment.

Keywords

Multi-Agent Feedback, Human-Computer Interaction, Human-in-the-Loop, Natural Language Processing, Software Development, Reinforcement Learning with Human Feedback, Large Language Models

1. Introduction

Multi-agent systems (MAS) have long been studied in Software Engineering (SE) as a way to distribute tasks across specialised components [1]. Before the recent wave of Large Language Models (LLMs), these environments were typically based on rule-based agents [2], recommender systems [3], optimisation modules [4], or narrower Natural Language Processing (NLP) pipelines for tasks such as requirements classification, architectural recommendation, bug analysis, and verification support [5]. Currently, LLMs and Generative AI (GenAI) can play a significant role in streamlining different tasks during the Software Development Lifecycle (SDLC), as they can provide valuable capacity to process data, identify key information, and generate subsequent outputs at a scale and speed beyond human capability [6]. As a result, multi-agent SE is moving from isolated automation to collaborative, role-based ecosystems in which agents support requirements elicitation, architecture design, task planning, code generation, testing, deployment, and maintenance. At the same time, this shift has made human oversight more central than ever.

SEPLN 2026: 42nd International Conference of the Spanish Society for Natural Language Processing, León, Spain, 22-25 September 2026.

*Corresponding author.

†These authors contributed equally.

✉ alba.bonet@dlsi.ua.es (A. Bonet-Jover); alicia.picazo@ua.es (A. Picazo-Izquierdo); santiago.masse@idener.ai (S. M. Palomo); aitana.delgado@idener.ai (A. D. Cabanillas); montoyo@dlsi.ua.es (A. Montoyo); stela@dlsi.ua.es (E. Saquete); patricio@dlsi.ua.es (P. Martínez-Barco); daniel.garcia@idener.ai (D. G. Guirao)

ORCID 0000-0002-7172-0094 (A. Bonet-Jover); 0009-0006-0282-4259 (A. Picazo-Izquierdo); 0000-0002-2775-696 (S. M. Palomo); 0009-0005-2555-9673 (A. D. Cabanillas); 0000-0002-3076-0890 (A. Montoyo); 0000-0002-6001-5461 (E. Saquete); 0000-0002-3076-0890 (P. Martínez-Barco); 0000-0002-1206-4293 (D. G. Guirao)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Human feedback is therefore a core mechanism in these environments, so multifaceted Human-in-the-Loop (HITL) Machine Learning (ML) strategies are designed both to improve agent behaviour over time and to preserve human agency throughout the process. Within HIVEMIND, feedback is not limited to explicit ratings or manual corrections, but is conceived as a broader HITL learning process. The role of NLP is particularly relevant here, since most interactions in SE are mediated through language: requirements, design rationales, code comments, bug reports, documentation, and developer corrections are all textual signals.

Our contribution focuses on continuously improving each agent through the integration of human feedback in multiple forms. More specifically, NLP models are used to associate agent responses with human corrective actions and to infer implicit evaluations from those corrections. Combined with explicit feedback and the injection of expected responses, this process enables the construction of preference datasets for each agent. These datasets are then used to support Reinforcement Learning from Human Feedback (RLHF), adapting model behaviour to the needs, expectations, and working practices of human operators.

The project targets validation in five real-world use cases across both societal and industrial domains, with the broader goal of accelerating the SDLC without compromising explainability and responsibility. The expected impact is therefore threefold: (i) scientifically, by advancing responsible AI for software engineering; (ii) industrially, by reducing development and testing effort through adaptive multi-agent support; (iii) and socially, by promoting trustworthy and transparent human-AI collaboration in mission-critical settings.

2. Related Work

2.1. Generative AI in Software Engineering

The application of GenAI to software engineering tasks has grown rapidly since 2023. Nguyen-Duc et al. [7] published a research agenda identifying fine-tuning, hallucination mitigation, and context-awareness as the most pressing challenges for GenAI-assisted development. Hallucination remains a major obstacle; Xu et al. [8] showed that this issue intensifies when user-provided context contains inaccuracies. In code generation, Elgedawy et al. [9] compared four prominent LLMs (GPT-3.5, GPT-4, Bard, Gemini) and found that output quality is heavily dependent on the operator's expertise, suggesting that collaborative human-AI workflows outperform fully autonomous pipelines. Nascimiento et al. [10] reached a similar conclusion when comparing AI and human engineers on coding tasks, stressing the need for fine-tuning AI assistants to individual developers' expertise levels. More recently, multi-agent architectures have emerged as a way to decompose complex SE tasks across specialised LLM instances. Hong et al. [11] proposed MetaGPT, a framework that assigns distinct software-engineering roles to collaborating agents, while Rasheed et al. [12] introduced CodePori to automate coding through multi-agent collaboration. However, both approaches operate with limited human oversight once execution begins.

2.2. Human Feedback and Preference Alignment

Aligning LLM outputs with human expectations has become a central concern in NLP. Ouyang et al. [13] introduced RLHF, enabling InstructGPT to outperform GPT-3 despite having far fewer parameters. Since then, RLHF has become the *de facto* paradigm for preference alignment in both proprietary and open-source models. Nevertheless, Kirk et al. [14] observed that models are often fine-tuned with a limited number of feedback samples, leading to overconfidence when generalising to new situations, and called for methods that gather additional implicit signals to complement explicit annotations. Parameter-efficient fine-tuning techniques have made preference learning more accessible: Hu et al. [15] introduced Low-Rank Adaptation (LoRA), which freezes pre-trained weights and injects lightweight trainable matrices, dramatically reducing the computational cost of specialisation. Building on this, Shao et al. [16] proposed Group Relative Policy Optimisation (GRPO), a reinforcement-learning algorithm

that eliminates the need for a separate reward model by using group-based advantage estimation, offering a more efficient alternative to Proximal Policy Optimisation (PPO) for preference alignment. On the collection side, the distinction between explicit feedback (ratings, pairwise preferences) and implicit feedback (interaction patterns, behavioural signals) has been studied extensively in information retrieval [17] and is now being adopted in LLM training pipelines, although systematic methodologies for combining both modalities in multi-agent settings remain scarce.

2.3. NLP Techniques for Software Engineering Feedback

The intersection of NLP and SE encompasses several areas relevant to human feedback. In requirements engineering, Ronanki et al. [18] explored the potential of GenAI, determining that prompt engineering is critical for improving the completeness of elicited requirements, and Sabetzadeh et al. [19] demonstrated that structured prompt patterns significantly improve the accuracy of LLM-assisted requirements validation. Sentiment analysis has been applied to developer interactions to detect dissatisfaction and frustration signals [20], offering a pathway for implicit feedback collection. Semantic similarity measures, including dense vector representations from models such as Sentence-BERT [21], enable the detection of re-prompting behaviour and topical drift within conversational interactions. These NLP capabilities are fundamental to the feedback methodology that HIVEMIND proposes, as they allow the system to infer user satisfaction from natural interaction traces without requiring constant explicit annotations.

3. Project description

HIVEMIND (Human-centred collaborative multi-agent framework for accelerating software development and maintenance) is a project funded by the European Union's Horizon Europe research and innovation programme whose main goal is to promote the evolution of AI-based software engineering following a human-centric AI approach that enhances a balanced collaboration between humans and agents (Figure 1).

The expected impacts of this project can be framed from three distinct perspectives: scientific, technical, and social.

- **Scientific approach:** the project aims to advance responsible SE methods integrating GenAI and HITL by establishing a multi-agent LLM-based framework for SE roles, where open-access scientific publications (≥ 20 expected) constitute one of the key impacts at this level.
- **Technical approach:** at this level, HIVEMIND seeks to produce open-source profitable, scalable and replicable solutions (TRL5), validated in 5 industrial environments and commercialised as a service, with the objective of reducing software development time (cycle time and testing).
- **Social approach:** finally, some of the greatest impacts at the social level lie in promoting a human-centric AI paradigm grounded in explainability, transparency, and safety, alongside an ethical and gender-balanced development process. The project also fosters an open-source strategy to enhance accessibility and ecosystem growth and drives the application of these solutions within public services.

3.1. European Consortium

HIVEMIND brings together a multidisciplinary consortium of 13 expert organisations from 8 European countries: Germany, Ireland, Romania, Spain, Sweden, Switzerland, Turkey, and the United Kingdom (see Figure 2).

The team includes top universities, research institutes, Small and Medium Enterprises (SMEs), and industry leaders, which are the following: the private research AI company IDENER.AI, administrative and technical coordinator of the project; the German Research Center for Artificial Intelligence (DFKI);



Figure 1: HIVEMIND logo.

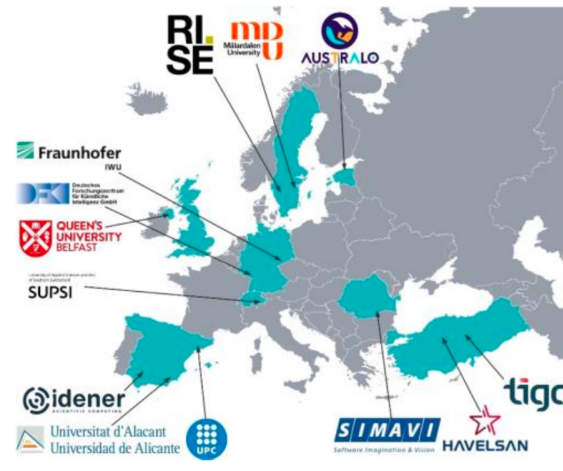


Figure 2: European consortium of the HIVEMIND's project.

the University of Alicante (UA); the Universitat Politècnica de Catalunya (UPC); the Research Institutes of Sweden (RISE); the Romanian company Software Imagination & Vision (SIMAVI); the Queen's University Belfast; the leading Turkish defence and advanced technology company (HAVELSAN); the Fraunhofer Institute for Machine Tools and Forming Technology IWU; the Turkish leader in healthcare IT Tiga Information Technologies (TIGA); the University of Applied Sciences and Arts of Southern Switzerland (SUPSI); the Mälardalen University (MDU) and the marketing agency AUSTRALO.

3.2. Main objectives

This project aims to accelerate all stages of the SDLC, leveraging novel AI and data technologies, and proposing a multi-agent environment where LLMs collaboratively work as multiple agents to solve complex SE tasks. As can be seen in the multi-agent framework proposed by this project in Figure 3, the ecosystem is composed of 8 distinct agents that represent a specific stage in the software development workflow for requirements engineering, architecture definition, task creation, code development, quality control, test generation, deployment management and maintenance of software systems.

The HIVEMIND project aims to develop a unified open-source framework targeted to reach TRL5. To that end, the project addresses a range of societal (citizen engagement, disaster response, healthcare) and industrial sectors (automated mobility, manufacturing), to ensure that the HIVEMIND technologies are validated in five relevant environments: i) citizen engagement (represented by SIMAVI); ii) automated mobility (led by Queen's University Belfast); iii) disaster management and response (driven by HAVELSAN); iv) manufacturing (guided by FRAUNHOFER) and v) healthcare (led by the TIGA).

In summary, some of the main objectives defined by HIVEMIND are as follows:

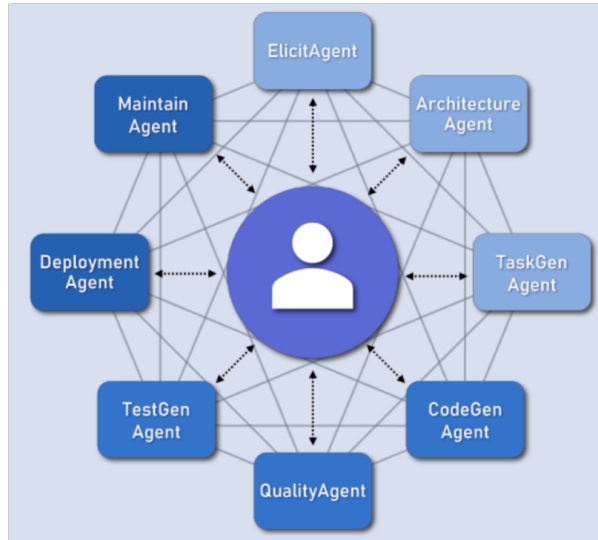


Figure 3: Human-centred collaborative multi-agent ecosystem of HIVEMIND.

- To advance responsible software engineering methods, tools and best practices leveraging, among others, novel GenAI and data technologies.
- To accelerate the SDLC.
- To ensure human oversight throughout the SDLC, aligning results with ethical and responsible practices.
- To ensure that the developed solutions have broad relevance and impact in the software engineering landscape, training LLM-based agents on data and specific use case scenarios.

4. Methodology and Timeline

Regarding the project implementation, HIVEMIND defines a 4-stage methodology to create an efficient evolution from TRL2 to TRL5, which includes clear checkpoints over a three-year period, as illustrated in Figure 4.

- **Stage 0. Team alignment, technical scouting and planning:** definition development task specifications and their interrelation, specifications and their interrelation; responsible practices in software engineering leveraging AI technologies and protocols and strategies for ethical and robust AI development.
- **Stage 1. Experimental proof of concept:** development of evidence supporting the proper behaviour of context-aware multi-agent ecosystems for streamlining software development in proven experiments.
- **Stage 2. Expanding the reach of self-collaborative LLM agents with Human-in-the-loop:** development of evidence of proper behaviour for experimentally proved human-centred multi-agent AI systems in a controlled environment, while aligning future development with the needs of validation sites.
- **Stage 3. Validation in relevant environments:** TRL5 is consolidated during this phase through the validation in 3 societal and 2 industrial relevant environments (citizen engagement, automated mobility, disaster response, manufacturing, healthcare) of a human-centred multi-agent ecosystem.

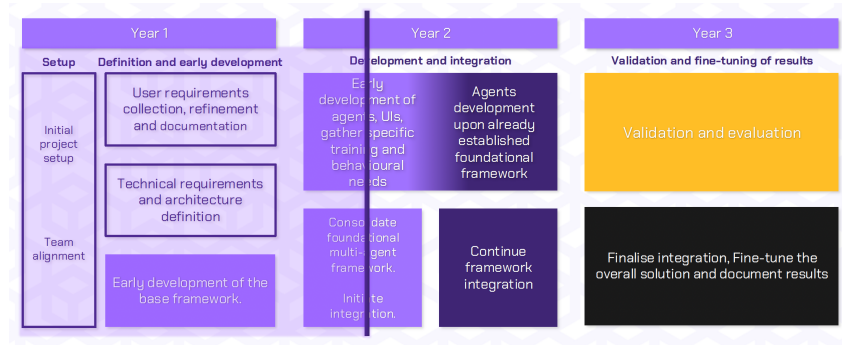


Figure 4: Methodology for the implementation of the project and current status of the project.

5. Feedback Methodology Proposal for Multi-Agent Environments in Software Development

Methodologically, feedback is regarded as a loop integrated into a broader customisation strategy. Our proposal defines four complementary layers for agent specialisation: fine-tuning with organisational data, prompt engineering of behavioural instructions, retrieval-augmented generation, and HITL machine learning. Within this architecture, human feedback plays a distinct role: whereas fine-tuning and RAG improve contextual grounding, and prompt engineering shapes role-specific behaviour, human feedback provides direct evidence about whether the produced outputs are actually useful, correct, and acceptable from the operator’s perspective.

5.1. Human-in-the-Loop Supervision in Multi-Agent Software Environments

To align the behaviour of specialised SE agents with the needs, expectations, and quality standards of human operators, we adopt a HITL methodology. Within the project, human feedback is not treated as an external evaluation step performed only at the end of development, but as a continuous source of supervision that supports the iterative improvement of agent outputs. This is relevant in software engineering scenarios, where correctness, maintainability, security, and actionability are critical, and where fully autonomous generation may still produce hallucinations, omissions, or context-inappropriate recommendations. In this sense, human operators are active collaborators who monitor, assess, and refine the outputs of each HIVEMIND agent. This human-centred perspective is explicitly embedded in the project architecture, in the sense that it defines the multifaceted inclusion of human feedback in LLM-based agents as a core mechanism for continuously improving agent performance.

5.2. Human Feedback Collection and Annotation Methodology

HITL-based systems are in a continuous development, since the human input addresses the limits of previous AI solutions and serves as an interaction flyover between machines and humans. The purpose of using this kind of technology roots in the initiative scale the processing of AI-driven applications to vast amount data while relying on human intelligence to perform very complex tasks, such in the case of natural language understanding (NLU) [22]. As a methodology, different researchers use HITL to boost efficiency in data collection (see Fanton et al. [23] and Cañizares-Díaz et al. [24]). A continuous executive loop is created, driving higher accuracy and stronger system robustness.

The data collection methodology proposed for HIVEMIND is built around conversational interactions between a human user and an AI agent through a dedicated chat interface. These raw interactions constitute the primary source of preference data and are later transformed into structured records suitable for retraining and evaluation. Our framework organises this process as a two-pronged strategy that balances signal quality and signal volume through the complementary use of explicit and implicit feedback. Explicit feedback provides a high-quality and unambiguous supervisory signal, while implicit

feedback captures weaker but more frequent indicators of user satisfaction or dissatisfaction that emerge naturally during interactions.

5.2.1. Explicit Feedback

Explicit feedback can be elicited through three main mechanisms, available through an interactive user interface. First, users can provide binary approval signals through thumbs up or thumbs down buttons attached to each agent output. Importantly, negative ratings are not reduced to a simple rejection event: when an operator selects thumbs down, the interface can request structured reasons linked to software engineering quality criteria, such as inaccuracy, lack of clarity, poor modularity, unsafe practices, or limited actionability, among others. Second, users may be presented with two alternative responses to the same prompt and asked to select the preferred one. Third, operators can rate responses on Likert scales, producing more granular annotations that help model nuanced judgments.

A major strength of the methodology is that explicit feedback is guided by a common quality framework rather than by ad hoc user impressions. Our user guide defines four foundational principles for evaluating software engineering outputs such as maintainability and actionability, among others. These principles serve as annotation criteria across agents and tasks, ensuring that feedback remains consistent and comparable even when it is collected for very different roles, such as requirements elicitation, code generation, quality assurance, testing, deployment, or maintenance. For each agent and purpose, the guide includes specific-stage instructions regarding the development lifecycle.

5.2.2. Implicit Feedback

In parallel, implicit feedback strategies are being explored in order to extend supervision beyond the cases in which users provide direct ratings. These include the refinement loop, when a user re-prompts the system several times on the same topic, indicating poor performance, or the copy-to-clipboard behaviour, interpreted as a weak positive indication that an artifact was useful and actionable. Therefore, conversation abandonment or abrupt topic changes after an unsatisfactory answer can be treated as weak negative evidence. Furthermore, sentiment analysis is being explored to classify human interactions as positive, negative or neutral. Finally, follow-up statements produced immediately after an agent response are stored and analysed to infer whether the previous output was satisfactory. In summary, these interaction traces make it possible to gather a larger volume of feedback without overburdening users with constant annotation requests.

5.2.3. Annotation Criteria and Preference Data Structuring

For the purpose of making these heterogeneous signals usable for learning, it is essential to convert them into standardised preference data schemas. The framework specifies structured records for thumbs annotations, chosen-versus-rejected pairs, and Likert evaluations. In this way, human judgments are not only collected, but formalised as machine-readable supervision signals linked to prompts, responses, annotators, timestamps, and feedback type.

A particularly relevant methodological component is the processing of implicit feedback through NLP-based signal detection. The framework defines two parallel systems: the first checks semantic relevance between the user's new prompt and the preceding agent response, and the second performs sentiment analysis on immediate follow-up utterances, contributing to a richer picture of user satisfaction.

5.3. From Feedback to Agent Improvement in the HIVEMIND Project

The final stage of the methodology is the use of these collected and processed signals to improve the agents. According to the HIVEMIND proposal, explicit feedback, implicit feedback inferred from corrective actions, and injected expected responses are combined to create preference datasets for each AI agent. These datasets are then used to apply GRPO, modifying the underlying model weights so that future outputs become better aligned with human operators' needs. The project also plans to support



Figure 5: QR code linking to the project website.

this continuous improvement process through model versioning, monitoring, and automated retraining as the datasets grow. As a result, feedback collection is not a one-off evaluation exercise, but part of an ongoing operational cycle of alignment and refinement.

This methodology is designed to support the full HIVEMIND ecosystem of specialised agents, including ElicitAgent, ArchitectureAgent, TaskGenAgent, CodeGenAgent, QualityAgent, TestGenAgent, DeploymentAgent, and MaintainAgent. Since these agents operate across different stages of the SDLC, a human-centred and structured feedback methodology is essential to ensure that improvements are not only technically measurable but also meaningful in practice.

6. Project website

All information related to the project, the team, the use cases, resources and updates is presented in its website, which can be easily accessed either through the QR code (see Figure 5) or via the following link.¹

7. Conclusions

The HIVEMIND project proposes an open-source, human-centric approach that fosters collaboration and ethical knowledge sharing, with the aim of promoting responsible SE practices that accelerate all stages of the SDLC.

In this context, LLMs, GenAI and NLP can play a significant role in enhancing interaction between humans and agents, specially through HITL approaches and feedback methodologies (i.e. direct feedback, suggestions and sentiment-aware corrections) that allow agents to learn and evolve.

The HIVEMIND team is composed of top AI researchers, software engineers, and industry leaders across Europe, forming a qualified multidisciplinary consortium capable of delivering real, ethical and human-centred solutions aligning model behaviour to human needs.

Declaration on Generative AI

The author(s) have employed Generative-AI tools for proofreading and spelling checking.

Acknowledgments

This work has been supported by funding from the HIVEMIND project – Horizon Europe call HORIZON-CL4-2024-DIGITAL-EMERGING-01 under Grant Agreement Number 101189745.

¹<https://hivemind-project.eu/>

References

- [1] A. Dorri, S. S. Kanhere, R. Jurdak, Multi-agent systems: A survey, *IEEE Access* 6 (2018) 28573–28593. URL: <https://api.semanticscholar.org/CorpusID:49185093>.
- [2] M. E. H. Charaf, M. Benattou, S. Azzouzi, A rule-based multi-agent system for testing distributed applications, in: 2012 International Conference on Multimedia Computing and Systems, 2012, pp. 967–972. doi:10.1109/ICMCS.2012.6320205.
- [3] J. Neto, A. J. Morais, R. Gonçalves, A. L. Coelho, Multi-agent-based recommender systems: A literature review, in: X.-S. Yang, S. Sherratt, N. Dey, A. Joshi (Eds.), *Proceedings of Sixth International Congress on Information and Communication Technology*, Springer Singapore, Singapore, 2022, pp. 543–555.
- [4] G. Santos, T. Pinto, I. Praça, Z. Vale, Mascem: Optimizing the performance of a multi-agent system, *Energy* 111 (2016) 513–524. URL: <https://www.sciencedirect.com/science/article/pii/S0360544216307654>. doi:<https://doi.org/10.1016/j.energy.2016.05.127>.
- [5] J. A. Chudziak, M. Wawer, ElliottAgents: A natural language-driven multi-agent system for stock market analysis and prediction, in: N. Oco, S. N. Dita, A. M. Borlongan, J.-B. Kim (Eds.), *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, Tokyo University of Foreign Studies, Tokyo, Japan, 2024, pp. 961–970. URL: <https://aclanthology.org/2024.paclic-1.91/>.
- [6] S. Chen, Y. Liu, W. Han, W. Zhang, T. Liu, A survey on llm-based multi-agent system: Recent advances and new frontiers in application, 2025. URL: <https://arxiv.org/abs/2412.17481>. arXiv:2412.17481.
- [7] A. Nguyen Duc, B. Cabrero-Daniel, A. Przybylek, C. Arora, D. Khanna, T. Herda, U. Rafiq, J. Melegati, E. Guerra, K.-K. Kemell, M. Saari, Z. Zhang, H. Le, T. Quan, P. Abrahamsson, Generative artificial intelligence for software engineering—a research agenda, *Software: Practice and Experience* 55 (2025) 1806–1843. doi:10.1002/spe.70005.
- [8] Z. Xu, S. Jain, M. Kankanhalli, Hallucination is inevitable: An innate limitation of large language models, 2025. URL: <https://arxiv.org/abs/2401.11817>. arXiv:2401.11817.
- [9] R. Elgedawy, P. Dosch, J. Sadik, S. Dutta, A. Gautam, K. Georgiou, F. Gholamrezae, F. Ji, K. Lim, Q. Liu, S. Ruoti, Ocasionally secure: A comparative analysis of code generation assistants, 2025. URL: <https://arxiv.org/abs/2402.00689>. arXiv:2402.00689.
- [10] N. Nascimento, P. Alencar, D. Cowan, Comparing software developers with chatgpt: An empirical investigation, 2023. URL: <https://arxiv.org/abs/2305.11837>. arXiv:2305.11837.
- [11] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, C. Zhang, J. Wang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, C. Wu, J. Schmidhuber, Metagpt: Meta programming for a multi-agent collaborative framework, 2024. URL: <https://arxiv.org/abs/2308.00352>. arXiv:2308.00352.
- [12] Z. Rasheed, M. A. Sami, K.-K. Kemell, M. Waseem, M. Saari, K. Systä, P. Abrahamsson, Codepori: Large-scale system for autonomous software development using multi-agent technology, 2024. URL: <https://arxiv.org/abs/2402.01411>. arXiv:2402.01411.
- [13] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. URL: <https://arxiv.org/abs/2203.02155>. arXiv:2203.02155.
- [14] H. R. Kirk, A. M. Bean, B. Vidgen, P. Röttger, S. A. Hale, The past, present and better future of feedback learning in large language models for subjective human preferences and values, in: H. Bouamor, J. Pino, K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, 2023, pp. 2409–2430. URL: <https://aclanthology.org/2023.emnlp-main.148/>. doi:10.18653/v1/2023.emnlp-main.148.
- [15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. URL: <https://arxiv.org/abs/2106.09685>. arXiv:2106.09685.
- [16] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, D. Guo, Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL:

<https://arxiv.org/abs/2402.03300>. arXiv:2402.03300.

- [17] D. Kelly, J. Teevan, Implicit feedback for inferring user preference: a bibliography, SIGIR Forum 37 (2003) 18–28. URL: <https://doi.org/10.1145/959258.959260>. doi:10.1145/959258.959260.
- [18] K. Ronanki, C. Berger, J. Horkoff, Investigating ChatGPT’s potential to assist in requirements elicitation processes, in: Proceedings of the 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2023, pp. 354–361.
- [19] D. Luitel, S. Hassani, M. Sabetzadeh, Improving requirements completeness: Automated assistance through large language models, 2024. URL: <https://arxiv.org/abs/2308.03784>. arXiv:2308.03784.
- [20] B. Lin, N. Cassee, A. Serebrenik, G. Bavota, N. Novielli, M. Lanza, Opinion mining for software development: A systematic literature review, ACM Trans. Softw. Eng. Methodol. 31 (2022). URL: <https://doi.org/10.1145/3490388>. doi:10.1145/3490388.
- [21] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL: <https://arxiv.org/abs/1908.10084>. arXiv:1908.10084.
- [22] G. Demartini, S. Mizzaro, D. Spina, Human-in-the-loop artificial intelligence for fighting online misinformation: Challenges and opportunities., IEEE Data Eng. Bull. 43 (2020) 65–74.
- [23] M. Fanton, H. Bonaldi, S. S. Tekiroğlu, M. Guerini, Human-in-the-loop for data collection: a multi-target counter narrative dataset to fight online hate speech, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 3226–3240.
- [24] H. Cañizares-Díaz, A. Piad-Morffis, S. Estevez-Velarde, Y. Gutiérrez, Y. Almeida-Cruz, A. Montoyo, R. Muñoz, Active learning for assisted corpus construction: A case study in knowledge discovery from biomedical text, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), 2021, pp. 216–225.