BÁO CÁO
TÌM KIẾM VÀ ĐÁNH GIÁ THÔNG TIN HỌC THUẬT

CHỦ ĐỀ : TÁC ĐỘNG CỦA TRÍ TUỆ NHÂN TẠO TẠO SINH ĐẾN NĂNG SUẤT VÀ CHẤT LƯỢNG PHÁT TRIỂN PHẦN MỀM GIAI ĐOẠN 2021 - 2026

1. Giới thiệu

Trong bối cảnh trí tuệ nhân tạo tạo sinh (Generative AI) phát triển mạnh từ sau sự xuất hiện của các mô hình ngôn ngữ lớn và công cụ lập trình như GitHub Copilot, ChatGPT, CodeWhisperer, Cursor, lĩnh vực phát triển phần mềm đang thay đổi rõ rệt. Các công cụ này có thể gợi ý mã nguồn, sinh kiểm thử, giải thích lỗi, hỗ trợ tài liệu hóa và giúp lập trình viên xử lý tác vụ lặp lại nhanh hơn. Tuy nhiên, hiệu quả thực tế không đồng đều vì còn phụ thuộc vào độ phức tạp của bài toán, năng lực kiểm chứng của lập trình viên, chất lượng dữ liệu huấn luyện và rủi ro bảo mật của mã sinh tự động.

• Mục tiêu nghiên cứu: Thu thập và đánh giá các tài liệu học thuật về tác động của AI tạo sinh đến năng suất, chất lượng mã nguồn và quy trình phát triển phần mềm.

• Phạm vi: Hoạt động phát triển phần mềm trong ngành Công nghệ thông tin giai đoạn 2021 - 2026, tập trung vào lập trình, kiểm thử, bảo trì mã nguồn và quản trị rủi ro khi dùng công cụ AI hỗ trợ.

• Câu hỏi nghiên cứu: (1) AI tạo sinh ảnh hưởng thế nào đến năng suất lập trình viên? (2) Công cụ AI có cải thiện chất lượng phần mềm hay tạo thêm rủi ro? (3) Tiêu chí nào cần dùng để đánh giá độ tin cậy của các nguồn thông tin về chủ đề này?

2. Quy trình tìm kiếm và thu thập thông tin

Quá trình tìm kiếm được thực hiện qua Google Scholar, Semantic Scholar, ACM Digital Library, IEEE Xplore, arXiv, O'Reilly Online Learning và các nguồn mở uy tín như Stack Overflow Developer Survey, Google Cloud DORA và GitHub Blog. Các từ khóa sử dụng bao gồm: "generative AI software engineering", "large language models for software engineering", "GitHub Copilot developer productivity", "AI-generated code security", "LLM software testing survey". Sau khi lọc theo mức độ liên quan, độ uy tín của tác giả/cơ quan xuất bản, phương pháp nghiên cứu, số trích dẫn và tính cập nhật, báo cáo lựa chọn 12 tài liệu gồm 8 bài báo khoa học, 2 sách chuyên khảo và 2 nguồn mở/báo cáo thực tiễn.

3. Danh sách tài liệu thu thập

STT	Tác giả/Nguồn	Năm	Tên tài liệu	Loại nguồn
1	Chen, M. et al.	2021	Evaluating Large Language Models Trained on Code	Bài báo học thuật / arXiv
2	Vaithilingam, P. et al.	2022	Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by LLMs	Bài báo khoa học / ACM CHI EA
3	Barke, S. et al.	2023	Grounded Copilot: How Programmers Interact with Code-Generating Models	Bài báo khoa học / PACM PL
4	Mastropaolo, A. et al.	2023	An Empirical Study on GitHub Copilot	Bài báo khoa học / ICSE
5	Peng, S. et al.	2023	The Impact of AI on Developer Productivity: Evidence from GitHub Copilot	Bài báo học thuật / thí nghiệm
6	Fan, A. et al.	2023	Large Language Models for Software Engineering: Survey and Open Problems	Bài báo khoa học / ICSE-FoSE
7	Wang, J. et al.	2024	Software Testing with Large Language Models: Survey, Landscape, and Vision	Bài báo khoa học / IEEE TSE
8	Fu, Y. et al.	2025	Security Weaknesses of Copilot-Generated Code in GitHub Projects	Bài báo khoa học / ACM TOSEM
9	Taulli, T.	2024	AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment	Sách chuyên khảo / O'Reilly
10	Winters, T. et al.	2020	Software Engineering at Google: Lessons Learned from Programming Over Time	Sách chuyên khảo / O'Reilly
11	Google Cloud DORA	2025	State of AI-Assisted Software Development	Báo cáo thực tiễn / nguồn mở
12	Stack Overflow	2025	Developer Survey 2025: AI	Khảo sát mở trên internet

4. Đánh giá độ tin cậy của nguồn

Nhìn chung, nhóm bài báo từ ACM, IEEE và các hội nghị phần mềm có độ tin cậy cao vì có tác giả rõ ràng, phương pháp nghiên cứu cụ thể, được công bố trong hệ thống học thuật uy tín và có số trích dẫn lớn. Sách chuyên khảo của O'Reilly hữu ích để hiểu bối cảnh thực hành, nhưng không thay thế được bằng chứng định lượng. Các khảo sát mở như Stack Overflow và DORA có quy mô lớn, cập nhật, song cần đọc cùng hạn chế mẫu khảo sát và mục tiêu báo cáo ngành.

STT	Nguồn tài liệu	Tác giả/cơ quan xuất bản	Phương pháp, trích dẫn, cập nhật	Xếp hạng và hạn chế
1	Chen et al. 2021	OpenAI/arXiv; tác giả rõ, nền tảng Codex	Đánh giá mô hình trên HumanEval; trích dẫn rất cao; 2021	Rất cao - nền tảng kỹ thuật mạnh, nhưng chưa phản ánh đầy đủ môi trường doanh nghiệp.
2	Vaithilingam et al. 2022	Harvard/Purdue/ACM CHI EA	Nghiên cứu người dùng; trích dẫn cao; 2022	Rất cao - phân tích trải nghiệm tốt, cỡ mẫu không lớn.
3	Barke et al. 2023	UC San Diego/PACM PL OOPSLA	Grounded theory với 20 lập trình viên; trích dẫn cao; 2023	Rất cao - mô tả sâu cách dùng Copilot, nhưng thiên về định tính.
4	Mastropaolo et al. 2023	ICSE/IEEE-ACM	Thực nghiệm về GitHub Copilot; có DOI; 2023	Cao - phù hợp trực tiếp, nhưng tập trung vào một công cụ.
5	Peng et al. 2023	Microsoft/GitHub/academic working paper	Controlled trial; kết quả năng suất rõ; trích dẫn cao; 2023	Cao - bằng chứng định lượng tốt, nhưng tác vụ thí nghiệm hẹp.
6	Fan et al. 2023	Meta/University/ICSE-FoSE	Tổng quan lĩnh vực LLM4SE; nhiều trích dẫn; 2023	Rất cao - bao quát tốt, nhưng mang tính survey.
7	Wang et al. 2024	IEEE Transactions on Software Engineering	Survey 102 nghiên cứu về kiểm thử bằng LLM; 2024	Rất cao - nguồn tạp chí mạnh, cập nhật cho kiểm thử phần mềm.
8	Fu et al. 2025	ACM TOSEM	Empirical study trên code AI sinh trong GitHub; 2025	Rất cao - cập nhật và có dữ liệu thực, nhưng phụ thuộc cách nhận diện mã AI.
9	Taulli 2024	O'Reilly	Sách thực hành về quy trình AI-assisted programming; 2024	Cao - hữu ích cho triển khai, không phải nghiên cứu phản biện.
10	Winters et al. 2020	Google/O'Reilly	Sách nền tảng về quy trình kỹ nghệ phần mềm; 2020	Cao - nền tảng tốt, nhưng không chuyên về GenAI.
11	Google Cloud DORA 2025	Google Cloud/DORA	Khảo sát ngành quy mô lớn; cập nhật 2025	Cao - dữ liệu mới, nhưng có mục tiêu báo cáo thực tiễn.
12	Stack Overflow 2025	Stack Overflow	Khảo sát cộng đồng lập trình viên toàn cầu; 2025	Trung bình - Cao - mẫu lớn, nhưng tự báo cáo và không phản biện học thuật.

Nhận xét tổng hợp: Bộ tài liệu đáp ứng đủ bốn nhóm nguồn theo yêu cầu: cơ sở dữ liệu học thuật, tạp chí/hội nghị khoa học chuyên ngành, sách chuyên khảo và nguồn mở trên internet. Nhóm nguồn đáng tin cậy nhất là các bài báo ACM/IEEE vì có phản biện, DOI và phương pháp rõ. Các nguồn ngành như DORA, Stack Overflow và O'Reilly nên dùng để bổ sung bối cảnh triển khai, không nên dùng làm bằng chứng duy nhất cho kết luận học thuật.

5. Kết luận

Các tài liệu thu thập cho thấy AI tạo sinh đang trở thành công cụ hỗ trợ quan trọng trong phát triển phần mềm, đặc biệt ở các tác vụ gợi ý mã, sinh kiểm thử, giải thích lỗi, tài liệu hóa và giảm công việc lặp lại. Tuy nhiên, bằng chứng học thuật chưa cho phép kết luận rằng AI luôn làm tăng năng suất trong mọi bối cảnh. Tác động tích cực thường rõ hơn ở tác vụ ngắn, có phạm vi rõ, còn với hệ thống lớn hoặc yêu cầu bảo mật cao thì lập trình viên vẫn phải kiểm chứng chặt chẽ. Vì vậy, cách dùng hợp lý là xem AI như công cụ hỗ trợ, kết hợp với code review, kiểm thử tự động, phân tích bảo mật và năng lực hiểu hệ thống của con người.

6. Danh mục tham khảo (Harvard Style)

- Barke, S., James, M.B. and Polikarpova, N. (2023) 'Grounded Copilot: How Programmers Interact with Code-Generating Models', Proceedings of the ACM on Programming Languages, 7(OOPSLA1), pp. 85-111. doi: 10.1145/3586030.
- Chen, M. et al. (2021) 'Evaluating Large Language Models Trained on Code'. arXiv:2107.03374. Available at: https://arxiv.org/abs/2107.03374 (Accessed: 4 June 2026).
- Fan, A. et al. (2023) 'Large Language Models for Software Engineering: Survey and Open Problems', 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), pp. 31-53.
- Fu, Y. et al. (2025) 'Security Weaknesses of Copilot-Generated Code in GitHub Projects: An Empirical Study', ACM Transactions on Software Engineering and Methodology, 34(8), Article 218. doi: 10.1145/3716848.
- Google Cloud DORA (2025) State of AI-Assisted Software Development. Available at: https://dora.dev/dora-report-2025/ (Accessed: 4 June 2026).
- Mastropaolo, A. et al. (2023) 'An Empirical Study on GitHub Copilot', 2023 IEEE/ACM International Conference on Software Engineering (ICSE). doi: 10.1109/ICSE48619.2023.00181.
- Peng, S., Kalliamvakou, E., Cihon, P. and Demirer, M. (2023) 'The Impact of AI on Developer Productivity: Evidence from GitHub Copilot'. arXiv:2302.06590. Available at: https://arxiv.org/abs/2302.06590 (Accessed: 4 June 2026).
- Stack Overflow (2025) Developer Survey 2025: AI. Available at: https://survey.stackoverflow.co/2025/ai (Accessed: 4 June 2026).
- Taulli, T. (2024) AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment. Sebastopol, CA: O'Reilly Media.
- Vaithilingam, P., Zhang, T. and Glassman, E.L. (2022) 'Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models', CHI Conference on Human Factors in Computing Systems Extended Abstracts, pp. 1-7. doi: 10.1145/3491101.3519665.
- Wang, J. et al. (2024) 'Software Testing with Large Language Models: Survey, Landscape, and Vision', IEEE Transactions on Software Engineering, 50(4), pp. 911-936. doi: 10.1109/TSE.2024.3368208.
- Winters, T., Manshreck, T. and Wright, H. (2020) Software Engineering at Google: Lessons Learned from Programming Over Time. Sebastopol, CA: O'Reilly Media.