BIS - MT · 10/07/2026

# Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

> *"Biết thì thưa thốt, không biết thì dựa cột mà nghe."*
>
> — Tục ngữ Việt Nam

*Khiêm tốn lắng nghe khi chưa hiểu rõ — sự khiêm tốn là khởi đầu của mọi sự học hỏi thật sự.*

## TIN TỨC NỔI BẬT

1

### Claude Code vs GitHub Copilot 2026: SWE-bench, Giá cả [Đã thử nghiệm]

*Claude Code vs GitHub Copilot 2026: SWE-bench, Pricing [Tested]*

tech-insider.org  Đọc bài viết →

Trong một so sánh gần đây, Claude Code và GitHub Copilot đã được kiểm tra về hiệu suất của chúng trên SWE-bench, một công cụ đánh giá hiệu suất cho phát triển phần mềm. Kết quả cho thấy Claude Code vượt trội so với GitHub Copilot ở nhiều lĩnh vực, bao gồm hoàn thành mã, tạo mã và năng suất tổng thể. Claude Code đã chứng minh sự hoàn thành mã chính xác và hiệu quả hơn, trong khi GitHub Copilot gặp khó khăn với lỗi và sự không nhất quán. Về giá cả, cả hai công cụ đều cung cấp các mô hình khác nhau. GitHub Copilot có sẵn dưới dạng dịch vụ dựa trên đăng ký, với phí hàng tháng là 10 đô la cho cá nhân và 25 đô la cho nhóm. Mặt khác, Claude Code cung cấp kế hoạch miễn phí với tính năng hạn chế, cũng như kế hoạch trả phí với khả năng bổ sung. Chi tiết giá của Claude Code không được chỉ định trong bài viết. So sánh này làm nổi bật điểm mạnh và điểm yếu của từng công cụ, cung cấp thông tin cho các nhà phát triển và doanh nghiệp muốn tích hợp hỗ trợ mã hóa AI vào quy trình làm việc của họ.

2

### Lớp Lệnh AI Kho Hàng Đa Agent Mang Lại Hiệu Suất Vận Hành Vượt Trội và Thông Minh Chuỗi Cung Ứng

*Multi-Agent Warehouse AI Command Layer Enables Operational Excellence and Supply Chain Intelligence*

NVIDIA Developer  Đọc bài viết →

NVIDIA đã giới thiệu Lớp lệnh AI Nhà kho Đa tác nhân, được thiết kế để nâng cao hiệu quả hoạt động và trí tuệ chuỗi cung ứng. Công nghệ này kết hợp trí tuệ nhân tạo (AI) với hệ thống đa tác nhân, cho phép tối ưu hóa thời gian thực các hoạt động nhà kho. Hệ thống có thể phân tích dữ liệu từ nhiều nguồn khác nhau, bao gồm cảm biến, camera và hệ thống quản lý hàng tồn kho, để đưa ra quyết định thông minh. Lớp lệnh AI Nhà kho Đa tác nhân có thể tự động hóa các nhiệm vụ như quản lý hàng tồn kho, thực hiện đơn hàng và tối ưu hóa tuyến đường. Nó cũng có thể dự đoán và ngăn chặn các vấn đề tiềm ẩn, chẳng hạn như hết hàng hoặc tồn kho quá mức, bằng cách phân tích dữ liệu lịch sử và đọc cảm biến thời gian thực. Ngoài ra, hệ thống có thể cung cấp thông tin về hiệu suất chuỗi cung ứng, cho phép doanh nghiệp đưa ra quyết định dựa trên dữ liệu. Bằng cách tận dụng công nghệ AI của NVIDIA, Lớp lệnh AI Nhà kho Đa tác nhân nhằm cải thiện hiệu quả hoạt động, giảm chi phí và nâng cao sự hài lòng của khách hàng. Hệ thống được thiết kế để có thể mở rộng và linh hoạt, làm cho nó phù hợp với nhiều ngành công nghiệp và kích thước nhà kho khác nhau.

3

## MiniMax-M2 là vị vua mới của các LLM mã nguồn mở (đặc biệt xuất sắc trong agentic tool calling)

*MiniMax-M2 is the new king of open source LLMs (especially for agentic tool calling)*

VentureBeat Đọc bài viết →

MiniMax-M2 đã nổi lên như mô hình ngôn ngữ lớn (LLM) mã nguồn mở hàng đầu, đặc biệt là xuất sắc trong việc gọi công cụ đại lý. Sự tiến bộ này có ý nghĩa quan trọng trong ngành công nghệ, nơi các LLM đang ngày càng được sử dụng cho nhiều ứng dụng khác nhau. Khả năng của MiniMax-M2 trong việc gọi công cụ đại lý cho phép nó tương tác và kiểm soát hiệu quả các công cụ bên ngoài, khiến nó trở thành một tài sản quý giá cho các nhà phát triển và doanh nghiệp. Bản chất mã nguồn mở của mô hình này cho phép phát triển và hợp tác được thúc đẩy bởi cộng đồng, có thể đẩy nhanh sự phát triển và cải tiến của nó. Mặc dù các chi tiết cụ thể về kiến trúc và dữ liệu đào tạo của MiniMax-M2 không được cung cấp, hiệu suất của nó trong việc gọi công cụ đại lý là đáng chú ý. Thành tựu này có thể có ý nghĩa đối với sự phát triển của các công cụ và ứng dụng được hỗ trợ bởi AI phức tạp hơn. Khi cảnh quan công nghệ tiếp tục phát triển, sự xuất hiện của MiniMax-M2 như một LLM mã nguồn mở hàng đầu là một sự phát triển quan trọng đáng được chú ý và khám phá thêm.

4

## Từ các dự án mã nguồn mở thành công đến OpenAI

*From open source hits to OpenAI*

Changelog  Đọc bài viết →

> Trong tập này, Max Stoiber, một developer làm việc trên thư mục plugin và nền tảng ứng dụng của ChatGPT tại OpenAI, thảo luận về các dự án mã nguồn mở và tác động của chúng đối với ngành công nghệ. Ông đề cập đến các dự án mã nguồn mở ít được biết đến, như react-boilerplate và styled-components, đã thu hút được sự chú ý đáng kể. Stoiber cũng chia sẻ kinh nghiệm của mình với Spectrum, một nền tảng cuối cùng đã trở thành một phần của GitHub và giúp định hình GitHub Discussions. Ngoài ra, ông cũng nói về sự phát triển của Stellate, một bộ nhớ đệm GraphQL đã được Shopify và The Guild mua lại. Cuộc trò chuyện cũng khám phá khái niệm về các ứng dụng ChatGPT như một bề mặt mới cho phát triển phần mềm.

5

## Lượng khí thải carbon của Microsoft tăng 25% vào năm ngoái

*Microsoft's carbon emissions went up 25 percent last year*

The Verge AI  Đọc bài viết →

> Tổng lượng khí thải carbon của Microsoft đã tăng 25 phần trăm vào năm 2025, đạt tổng cộng 34 triệu tấn. Theo báo cáo bền vững năm 2026 của công ty, sự gia tăng này chủ yếu là do việc mở rộng cơ sở hạ tầng trung tâm dữ liệu và quyết định ngừng mua chứng chỉ năng lượng tái tạo không bổ sung. Microsoft đã đặt mục tiêu trở nên trung hòa carbon vào năm 2030, nhưng sự cố này không phải là lần đầu tiên công ty phải đối mặt với những thách thức trong việc đạt được mục tiêu này. Báo cáo cũng nhấn mạnh rằng các giải pháp bền vững không được mở rộng nhanh enough để đáp ứng nhu cầu ngày càng tăng về năng lượng và tài nguyên do cơ sở hạ tầng AI thúc đẩy. Vấn đề này không chỉ riêng với Microsoft, vì Google và Amazon cũng đã báo cáo sự gia tăng đáng kể trong lượng khí thải của chuỗi cung ứng trong báo cáo bền vững năm 2026 của họ.

6

## Về việc sở hữu một codebase, và tại sao đó có thể là công việc khó khăn nhất trong ngành phần mềm

*On owning a codebase, and why it may be the hardest job in software*

Sourcegraph Blog  Đọc bài viết →

Việc sử dụng ngày càng nhiều các đại lý mã hóa AI đang tạo ra một lượng lớn mã mới, nhưng mặc dù vậy, thế giới vẫn phụ thuộc nặng nề vào các cơ sở mã khổng lồ đã có từ hàng thập kỷ. Những cơ sở mã di sản này đã được xây dựng theo thời gian và là xương sống của nhiều hệ thống phần mềm. Tuy nhiên, việc sở hữu và hiểu rõ những cơ sở mã phức tạp này là một nhiệm vụ đầy thách thức. Kích thước khổng lồ và tuổi thọ của những cơ sở mã này làm cho chúng khó hiểu, với nhiều cơ sở mã chứa công nghệ lỗi thời và mã lỗi thời. Độ phức tạp của những cơ sở mã này cũng làm cho việc xác định và sửa lỗi của các nhà phát triển trở nên khó khăn, dẫn đến gánh nặng bảo trì đáng kể. Kết quả là, việc sở hữu và hiểu rõ một cơ sở mã có thể được coi là công việc khó nhất trong phần mềm, đòi hỏi sự hiểu biết sâu sắc về mã, lịch sử của nó và sự tương tác với các hệ thống khác. Nhiệm vụ này rất quan trọng để đảm bảo sự ổn định và bảo mật của các hệ thống phần mềm, nhưng nó cũng là một thách thức đáng kể mà nhiều nhà phát triển phải đối mặt.

## 7 CEO AGI Deployment của OpenAI, Fidji Simo, từ chức

*OpenAI's CEO of AGI Deployment, Fidji Simo, Is Stepping Down*

Wired Đọc bài viết →

Fidji Simo, giám đốc điều hành về việc triển khai AGI của OpenAI, đang rời bỏ vai trò toàn thời gian tại công ty để trở thành cố vấn bán thời gian. Quyết định này đến sau khi Simo nghỉ ốm do tình trạng bệnh tự miễn thần kinh ngày càng tồi tệ, một tình trạng mà cô đã sống chung trong bảy năm. Cô gia nhập hội đồng quản trị của OpenAI vào tháng 3 năm 2024 và đảm nhiệm các tổ chức sản phẩm và kinh doanh vào năm 2025. Trước đó, Simo từng giữ các vị trí lãnh đạo tại Instacart và Meta. Sự ra đi của cô là một phần của sự thay đổi lớn trong ban lãnh đạo của OpenAI, công ty đã tái tổ chức các đội sản phẩm và hợp nhất các đội làm việc trên ChatGPT, trình duyệt AI của công ty và tác nhân mã hóa AI. OpenAI đang tập trung vào một số sản phẩm cốt lõi trước khi IPO dự kiến vào năm 2027, nhắm đến mức định giá 1 nghìn tỷ đô la.

## 8 Tại sao các LLM nên ngừng "suy nghĩ thành tiếng" (và điều gì đến sau chain-of-thought)

*Why LLMs should stop thinking out loud (and what comes after chain-of-thought)*

BD Tech Talks Đọc bài viết →

> Ngành công nghệ đang đối mặt với một thách thức đáng kể khi chi phí đào tạo và chạy các mô hình ngôn ngữ lớn (LLM) tăng vọt ngoài tầm kiểm soát. Các công ty như Uber, Meta và Amazon đang□□ đánh giá các quy trình làm việc AI của mình và áp đặt giới hạn ngân sách do chi phí đào tạo LLM cao. Phần lớn chi phí này được quy cho phương pháp nhắc chuỗi suy nghĩ (Chain-of-Thought, CoT), phương pháp này hướng dẫn LLM "suy nghĩ từng bước" trước khi đưa ra câu trả lời cuối cùng. Mặc dù CoT ban đầu là một giải pháp thực tế, nhưng nó đã trở thành tiêu chuẩn ngành mặc dù là một cách tiếp cận có lỗi. Nghiên cứu đã chỉ ra rằng các token trung gian được tạo ra bởi CoT không phải là một đại diện thực sự của quá trình suy luận của mô hình, mà là một ràng buộc cấu trúc mô phỏng các định dạng đầu ra giống con người. Cách tiếp cận này tạo ra một nút thắt kỹ thuật và tài chính, làm chậm tốc độ suy luận và che giấu bản chất thực sự của tính toán máy. Để mở rộng AI một cách bền vững, ngành công nghệ cần phải vượt qua CoT và khám phá các cơ chế suy luận thay thế.

---

TIPS & TRICKS CHO DEV

### Tăng tốc viết code

**Vấn đề:** Không đủ thời gian để viết code cho dự án.

**Cách làm:** Sử dụng Claude Code để tự động sinh code, ví dụ `claude code --lang python --prompt "Viết hàm tính tổng"`.

**Đánh giá:** Hiệu quả khi cần viết code nhanh, nhưng cần kiểm tra lại chất lượng code.

### Tối ưu hóa code

**Vấn đề:** Code chưa tối ưu, cần refactoring.

**Cách làm:** Sử dụng Claude Code với lệnh `claude refactor --prompt "Tối ưu hóa đoạn code này"`.

**Đánh giá:** Tiết kiệm thời gian khi cần tối ưu hóa code, nhưng cần xem xét lại logic code.

### Debug code hiệu quả

**Vấn đề:** Khó tìm ra lỗi trong code.

**Cách làm:** Sử dụng Claude Code với lệnh `claude debug --prompt "Tìm lỗi trong đoạn code này"`.

**Đánh giá:** Giúp tìm ra lỗi nhanh chóng, nhưng cần kiểm tra lại output và logic code.

BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
2. Dev cần biết cách tối ưu chi phí và hiệu năng của các mô hình ngôn ngữ lớn (LLM) để áp dụng hiệu quả trong các dự án. Điều này giúp giảm thiểu chi phí tính toán và tăng tốc độ xử lý. Việc tối ưu hóa LLM cũng giúp cải thiện hiệu suất của ứng dụng.
3. Ví dụ, có thể sử dụng kỹ thuật fine-tuning và LoRA (Low-Rank Adaptation) để điều chỉnh mô hình LLM cho phù hợp với use case cụ thể, giúp giảm kích thước mô hình và tăng tốc độ xử lý.
4. Tip hoặc bước tiếp theo: Sử dụng các công cụ như Hugging Face Transformers để thực hiện fine-tuning và tối ưu hóa LLM, và đánh giá hiệu suất của mô hình trên các tập dữ liệubenchmark để đảm bảo hiệu suất tốt nhất.