

# Nền tảng Quản lý Dự án Tự chủ bằng AI

Một "Project Manager ảo" biết đọc mã nguồn, tự rã yêu cầu thành công việc, tự cập nhật tiến độ và tự giao ban — để đội ngũ phát triển tập trung vào điều duy nhất quan trọng: viết ra sản phẩm.

Xóa bỏ nhập liệu thủ công · Tiến độ luôn phản ánh sự thật · Quyết định nhanh hơn

Tiếp nhận yêu cầu bằng ngôn ngữ tự nhiên

Thấu hiểu mã nguồn

Tự động hóa qua Git

Giao ban tự động

MỤC LỤC

Nội dung tài liệu

Tài liệu được viết theo hướng nghiệp vụ — mô tả nền tảng làm được gì và mang lại giá trị gì cho đội ngũ, không đi vào chi tiết kỹ thuật triển khai.

|    |                                             |                                                        |
|----|---------------------------------------------|--------------------------------------------------------|
| 01 | Tóm tắt điều hành                           | Nền tảng trong một trang                               |
| 02 | Bối cảnh & Vấn đề                           | Vì sao quản lý dự án truyền thống đang kìm hãm đội ngũ |
| 03 | Tầm nhìn & Giá trị cốt lõi                  | Triết lý "AI-Native"                                   |
| 04 | Đối tượng thụ hưởng                         | AI được lợi và lợi như thế nào                         |
| 05 | Bản đồ tính năng tổng thể                   | Sáu trụ cột năng lực                                   |
| 06 | Chi tiết từng nhóm tính năng                | Vấn đề · Cách hoạt động · Giá trị                      |
| 07 | Kịch bản thực tế — "Một ngày cùng nền tảng" | Từ ý tưởng đến hoàn thành                              |
| 08 | Bảo mật & Quyền riêng tư dữ liệu            | Cam kết với tài sản mã nguồn                           |
| 09 | Lộ trình & Mức độ sẵn sàng                  | Đang có gì, sắp có gì                                  |
| 10 | Lợi ích kinh doanh & Kết luận               | Vì sao đầu tư bây giờ                                  |

## Tóm tắt điều hành

Nền tảng thay thế toàn bộ phần việc vận hành lặp đi lặp lại của một Project Manager truyền thống — nhưng làm việc đó 24/7, không quên, không thiên vị, và luôn bám sát sự thật từ mã nguồn.

Trong các đội phát triển phần mềm, phần lớn "quản lý dự án" thực chất là công việc thủ công tốn thời gian: gõ lại yêu cầu thành ticket, ước lượng bằng cảm tính, kéo-thả thẻ trên bảng Kanban, nhắc nhau cập nhật trạng thái, và tổng hợp báo cáo mỗi sáng. Những việc này không tạo ra giá trị trực tiếp cho khách hàng, nhưng lại chiếm hàng giờ mỗi tuần của cả kỹ sư lẫn quản lý — và thường xuyên sai lệch so với thực tế.

**Nền tảng Quản lý Dự án AI-Native** tự động hóa trọn vẹn vòng đời đó. Người dùng chỉ cần **mô tả mong muốn bằng ngôn ngữ tự nhiên** — qua chat hoặc web — và hệ thống sẽ tự làm phần còn lại:

- **Hiểu bối cảnh mã nguồn thật** của dự án trước khi lập kế hoạch, thay vì đoán mò.
- **Tự rã yêu cầu** thành Epic và các Task chi tiết — kèm mô tả, yêu cầu kỹ thuật, tiêu chí nghiệm thu và ước lượng công sức.
- **Tự sắp xếp Sprint** dựa trên quỹ năng lực còn lại và chủ động đàm phán khi quá tải.
- **Tự cập nhật tiến độ** theo hoạt động Git thật (tạo nhánh, mở Pull Request, merge) — bảng công việc không bao giờ "cũ".
- **Tự tổ chức giao ban** mỗi sáng và tổng hợp báo cáo cho quản lý.

Con người vẫn giữ quyền quyết định ở những điểm quan trọng: mọi kế hoạch AI đề xuất đều ở dạng **bản nháp chờ duyệt** (draft → approve). AI lo phần nặng nhọc; con người giữ phần phán xét.

≈ 0

thao tác nhập liệu thủ công để tạo & cập nhật ticket

< 2s

độ trễ đồng bộ tiến độ lên bảng khi có sự kiện Git

100%

mã nguồn thô không được lưu trữ — chỉ đọc siêu dữ liệu

### ĐIỂM MẪU CHỐT

Nền tảng không phải "thêm một công cụ để đội ngũ phải cập nhật". Nó là lớp tự động loại bỏ chính việc cập nhật đó — biến bảng công việc từ nơi phải chăm sóc thành nơi phản ánh sự thật một cách tự nhiên.

## Bối cảnh & Vấn đề

Công cụ quản lý dự án hiện nay đặt gánh nặng nhập liệu lên con người. Kết quả là dữ liệu luôn trễ, luôn thiếu, và luôn cần một người "canh bằng".

### Năm nỗi đau phổ biến



#### Nhập liệu thủ công vô tận

CHI PHÍ ẨN LỚN NHẤT

Mỗi yêu cầu phải được gõ tay thành ticket, chia nhỏ, gắn nhãn, ước lượng. Kỹ sư giỏi dành thời gian làm thư ký thay vì làm sản phẩm.



#### Bảng công việc luôn "cũ"

MẤT NIỀM TIN VÀO DỮ LIỆU

Trạng thái ticket phụ thuộc việc ai đó nhớ kéo thẻ. Thực tế code đã xong nhưng thẻ vẫn "đang làm", hoặc ngược lại — báo cáo không đáng tin.



#### Ước lượng bằng cảm tính

KẾ HOẠCH SAI TỪ GỐC

Story point được đoán trong buổi họp, không dựa trên năng lực thật hay lịch sử của từng người. Sprint liên tục vỡ kế hoạch.



#### PM trở thành nút thắt cổ chai

CHẬM RA QUYẾT ĐỊNH

Mọi thay đổi phạm vi, mọi việc chèn task khẩn đều phải đợi PM sắp xếp thủ công. Đội ngũ chờ, còn PM thì quá tải.



#### Giao ban tốn sức, ít giá trị

MỆT MỎI VẬN HÀNH

Standup mỗi ngày ngốn thời gian cả đội để lặp lại thông tin lẽ ra đã nằm sẵn trong Git log và lịch sử task. Người ở khác múi giờ càng khó tham gia.

#### HỆ QUẢ KINH DOANH

Khi dữ liệu quản lý không phản ánh sự thật, mọi quyết định phía trên nó — dự báo tiến độ, phân bổ nguồn lực, cam kết với khách hàng — đều đứng trên nền móng lung lay. Chi phí thực sự không nằm ở vài phút nhập liệu, mà ở những quyết định sai vì thông tin sai.

Nền tảng được thiết kế để tấn công thẳng vào gốc rễ: **nếu tiến độ tự sinh ra từ hoạt động thật (chat, mã nguồn, Git), thì không còn khoảng cách giữa "điều đang diễn ra" và "điều được ghi nhận".**

# Tầm nhìn & Giá trị cốt lõi

"AI-Native" nghĩa là AI không phải một nút bấm phụ trợ gắn thêm — mà là trung tâm vận hành của toàn bộ quy trình.

## Bốn nguyên tắc nền tảng



### Ngôn ngữ tự nhiên là giao diện chính

Người dùng nói điều họ muốn bằng lời của mình. Không có form phức tạp, không cần học cú pháp công cụ. AI chịu trách nhiệm chuyển ý định thành cấu trúc.



### Quyết định dựa trên bối cảnh thật

AI đọc mã nguồn và lịch sử dự án trước khi đề xuất. Kế hoạch bám vào hiện trạng hệ thống, không phải giả định chung chung.



### Tiến độ tự sinh, không tự khai

Trạng thái công việc bắt nguồn từ sự kiện thật trên Git. Bảng công việc luôn khớp thực tế mà không cần ai duy trì.



### Con người giữ quyền phán xét

AI làm phần nặng nhọc và đề xuất; con người duyệt ở những điểm quan trọng. Tự động hóa có kiểm soát, không phải hộp đen.

## Kiến trúc trải nghiệm kép

Nền tảng gặp gỡ người dùng ở đúng nơi họ đang làm việc, qua hai bề mặt đồng bộ thời gian thực:

| Bề mặt                                                           | Vai trò nghiệp vụ                                                                                                                                |
|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Bot hội thoại</b><br><i>Slack · Microsoft Teams · Discord</i> | Điểm chạm chính hằng ngày: tiếp nhận yêu cầu, gửi thông báo, nhắc nhở, và điều phối buổi giao ban bất đồng bộ. Nơi kỹ sư "nói chuyện" với dự án. |
| <b>Web Dashboard</b><br><i>Bản đồ chiến lược</i>                 | Cái nhìn toàn cảnh: Bảng Kanban, Backlog, cấu hình Sprint, cấu trúc dự án. Đồng bộ liên tục với mọi hành động của AI và mọi thay đổi trong Git.  |

### GIÁ TRỊ CỐT LÕI TÓM GỌN

Chuyển đổi ngữ từ mô hình "con người phục vụ công cụ" (cập nhật, đồng bộ, báo cáo) sang "công cụ phục vụ con người" (tự vận hành, tự báo cáo, chỉ hỏi khi cần quyết định).

## Đối tượng thụ hưởng

Mỗi vai trò trong đội ngũ nhận về một lợi ích cụ thể và khác nhau — nhưng cùng hướng đến ít ma sát vận hành hơn.



### Lập trình viên (Developer)

NGƯỜI THỰC THI

Được giải phóng khỏi việc cập nhật ticket. Chỉ cần làm việc như bình thường trên Git — tạo nhánh, mở PR, merge — và tiến độ tự cập nhật. Standup chỉ mất vài giây xác nhận.



### Trưởng nhóm kỹ thuật (Tech Lead)

NGƯỜI ĐỊNH HƯỚNG

Yêu cầu thô được rã thành task có cấu trúc, bám mã nguồn thật — chỉ cần rà soát và duyệt thay vì tự ngồi bóc tách từng việc. Chất lượng phân rã đồng đều.



### Quản lý sản phẩm / Dự án (PM/PO)

NGƯỜI ĐIỀU PHỐI

Không còn là nút thắt cổ chai. Ai lo việc sắp xếp Sprint và đàm phán năng lực; PM tập trung vào ưu tiên sản phẩm và quan hệ khách hàng.



### Lãnh đạo / Quản lý cấp cao

NGƯỜI RA QUYẾT ĐỊNH

Có báo cáo tiến độ đáng tin cậy theo thời gian thực, tổng hợp tự động mỗi sáng — dự báo và cam kết dựa trên dữ liệu phản ánh đúng thực tế.

### NGUYÊN TẮC CHUNG

Nền tảng không thêm việc cho bất kỳ ai. Với mỗi vai trò, nó **gỡ bỏ** phần công việc thủ công đang tồn tại và trả lại thời gian cho công việc có giá trị cao.

## Bản đồ tính năng tổng thể

Toàn bộ năng lực của nền tảng được tổ chức thành sáu trụ cột, phối hợp thành một vòng khép kín: từ ý tưởng → kế hoạch → thực thi → báo cáo.

| # | Trụ cột năng lực                                                               | Nền tảng làm gì cho đội ngũ                                                             |
|---|--------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| 1 | <b>Tiếp nhận &amp; Rã việc tự động</b><br><i>NL Intake &amp; Decomposition</i> | Biến một câu mô tả thành Epic + Task chi tiết, đầy đủ tiêu chí nghiệm thu và ước lượng. |
| 2 | <b>Thấu hiểu mã nguồn</b><br><i>Codebase Awareness</i>                         | Đọc và ghi nhớ cấu trúc dự án để lập kế hoạch bám sát hiện trạng hệ thống thật.         |
| 3 | <b>Ước lượng theo bối cảnh</b><br><i>Context-Aware Estimation</i>              | Đề xuất công sức dựa trên lịch sử hoàn thành thật của từng thành viên, không đoán mò.   |
| 4 | <b>Quản lý Sprint tự chủ</b><br><i>AI Sprint Management</i>                    | Tự tính quỹ năng lực còn lại và đàm phán khi có việc mới chèn vào Sprint đang chạy.     |
| 5 | <b>Tự động hóa qua Git</b><br><i>Git-Driven Automation</i>                     | Chuyển trạng thái ticket theo hoạt động Git thật — tạo nhánh, mở PR, merge.             |
| 6 | <b>Giao ban bất đồng bộ</b><br><i>Async Standup Bot</i>                        | Tự soạn báo cáo mỗi sáng từ dữ liệu thật, thu thập xác nhận, tổng hợp cho quản lý.      |

### XUYỀN SUỐT TẤT CẢ — LUỒNG DUYỆT "CON NGƯỜI TRONG VÒNG LẶP"

Mọi đề xuất của AI (kế hoạch task, sắp xếp Sprint) đều đi qua bước **bản nháp → phê duyệt**. Đây không phải một trụ cột riêng mà là cơ chế an toàn bao trùm: đội ngũ luôn giữ quyền chỉnh sửa và quyết định cuối cùng.

Sáu trụ cột không hoạt động rời rạc. Chúng tạo thành một dây chuyền giá trị liên tục — được minh họa chi tiết trong **Mục 07 — Kịch bản thực tế**.

## Chi tiết từng nhóm tính năng

Mỗi tính năng được trình bày theo ba câu hỏi nghiệp vụ: **giải quyết vấn đề gì, hoạt động ra sao dưới góc nhìn người dùng**, và **mang lại giá trị gì**. Nhấn trạng thái cho biết mức độ sẵn sàng hiện tại.

### 1 Tiếp nhận ngôn ngữ tự nhiên & Tự động rải việc

ĐANG PHÁT TRIỂN

#### VẤN ĐỀ

Kỹ sư và quản lý mất hàng giờ để chuyển một ý tưởng thành các ticket đủ chi tiết để bắt tay làm. Chất lượng phân rải phụ thuộc vào người viết, không đồng đều.

#### CÁCH HOẠT ĐỘNG

Người dùng gõ (hoặc nói) mong muốn bằng lời tự nhiên — "Thêm đăng nhập bằng Google cho ứng dụng web". AI đọc bối cảnh mã nguồn liên quan rồi tự tạo ra **một Epic** (nếu yêu cầu lớn) cùng **các Task con**. Mỗi Task đầy đủ: *Tiêu đề · Mô tả · Yêu cầu kỹ thuật · Tiêu chí nghiệm thu (Acceptance Criteria)*. Kết quả hiện ra dưới dạng bản nháp để rà soát.

#### GIÁ TRỊ NGHIỆP VỤ

Rút ngắn thời gian từ "ý tưởng" đến "sẵn sàng thực thi" từ hàng giờ xuống còn vài phút. Chuẩn hóa chất lượng đầu vào của mọi công việc — mọi task đều có tiêu chí nghiệm thu rõ ràng ngay từ đầu, giảm hiểu lầm và làm lại.

### 2 Thấu hiểu mã nguồn (Codebase Awareness)

ĐÃ SẴN SÀNG

#### VẤN ĐỀ

Công cụ PM thông thường không biết gì về hệ thống thật. Chúng đề xuất chung chung, không nhận ra rằng một tính năng đã có sẵn, hay một thay đổi sẽ đụng đến phần nào của hệ thống.

#### CÁCH HOẠT ĐỘNG

Đội ngũ kết nối kho mã nguồn (GitHub / GitLab / Bitbucket) một lần. Nền tảng tự đọc và lập chỉ mục toàn bộ cấu trúc dự án để "ghi nhớ" hiểu biết về hệ thống. Khi có yêu cầu mới, AI tra cứu đúng những phần liên quan để lập kế hoạch chính xác. Kho được cập nhật lại khi mã nguồn thay đổi.

#### GIÁ TRỊ NGHIỆP VỤ

Kế hoạch bám sát thực tế hệ thống thay vì lý thuyết. Giảm rủi ro "lập task cho việc đã tồn tại" hoặc bỏ sót ảnh hưởng dây chuyền. Đây là nền móng khiến mọi đề xuất khác của AI trở nên đáng tin.

## CAM KẾT BẢO MẬT KÈM THEO

Nền tảng **không lưu trữ mã nguồn thô**. Nó chỉ ghi nhận siêu dữ liệu (cấu trúc, vị trí, dấu hiệu ngữ nghĩa) đủ để hiểu và tra cứu; khi cần xem đoạn mã cụ thể, hệ thống lấy đúng đoạn đó theo thời điểm rồi bỏ đi. Chi tiết ở **Mục 08**.

### 3 Ước lượng công sức theo bối cảnh

ĐANG PHÁT TRIỂN

#### VẤN ĐỀ

Ước lượng thường là con số đoán trong phòng họp, không gắn với năng lực thật của người sẽ làm. Kế hoạch Sprint vì thế sai lệch ngay từ khâu ước tính.

#### CÁCH HOẠT ĐỘNG

Với mỗi task mới, AI đề xuất mức công sức (Story Point theo thang Fibonacci hoặc số giờ). Ban đầu dựa trên phân tích độ phức tạp của yêu cầu; theo thời gian, hệ thống **học từ lịch sử hoàn thành thật** của từng thành viên để hiệu chỉnh cho sát hơn. Nguồn gốc ước lượng luôn được ghi nhận (do AI suy luận hay dựa trên lịch sử).

#### GIÁ TRỊ NGHIỆP VỤ

Kế hoạch Sprint đáng tin hơn, ít vỡ tiến độ hơn. Ước lượng ngày càng chính xác khi hệ thống tích lũy dữ liệu — biến kinh nghiệm ngầm của đội thành tài sản đo lường được.

### 4 Quản lý Sprint tự chủ & Đàm phán nguồn lực

TRÊN LỘ TRÌNH

#### VẤN ĐỀ

Khi có việc khẩn chèn vào giữa Sprint, PM phải thủ công tính lại quỹ thời gian và quyết định đánh đổi — chậm và dễ sai. Sprint phình ra âm thầm cho đến khi vỡ.

#### CÁCH HOẠT ĐỘNG

Khi một task mới được duyệt, AI tự quét quỹ năng lực còn lại của Sprint hiện tại. Nếu vượt tải, AI không tự ý quyết định mà **mở một cuộc hội thoại đàm phán** với các phương án rõ ràng: (A) *Chèn thẳng vào Sprint*, (B) *Đưa vào Backlog*, hoặc (C) *Hoán đổi với một task khác* — kèm hệ quả của mỗi lựa chọn.

#### GIÁ TRỊ NGHIỆP VỤ

Quyết định về phạm vi Sprint trở nên minh bạch và tức thời. Đội ngũ luôn thấy rõ cái giá của việc chèn thêm việc, tránh cam kết quá tải một cách vô thức. PM ra quyết định thay vì làm phép tính.

## 5 Tự động hóa tiến độ bằng Git

TRÊN LỘ TRÌNH

### VẤN ĐỀ

Trạng thái ticket chỉ đúng khi có người nhớ cập nhật. Trên thực tế, bảng công việc và mã nguồn luôn lệch nhau.

### CÁCH HOẠT ĐỘNG

Nền tảng lắng nghe hoạt động thật trên kho mã nguồn và tự chuyển trạng thái ticket tương ứng — không cần ai kéo thẻ:

| Hành động trên Git     | Trạng thái task | Hành động tự động của AI                                                |
|------------------------|-----------------|-------------------------------------------------------------------------|
| Tạo nhánh mới cho task | ĐANG LÀM        | Gán người tạo nhánh làm người phụ trách, chuyển ticket sang "đang làm". |
| Mở Pull Request        | ĐANG REVIEW     | Tự gán người review, đăng liên kết PR vào ticket, gửi nhắc nhở.         |
| PR được merge          | HOÀN THÀNH      | Đóng ticket, tạo changelog và thông báo lên nhóm chat.                  |

### GIÁ TRỊ NGHIỆP VỤ

Bảng công việc trở thành tấm gương phản chiếu trung thực của công việc thật, cập nhật trong dưới hai giây. Xóa bỏ hoàn toàn công đoạn "cập nhật trạng thái" và các cuộc tranh luận "thẻ này thực ra xong chưa?".

## 6 Bot giao ban bất đồng bộ (Async Standup)

TRÊN LỘ TRÌNH

### VẤN ĐỀ

Standup trực tiếp tốn thời gian cả đội, lặp lại thông tin đã có sẵn, và loại trừ người ở khác múi giờ.

### CÁCH HOẠT ĐỘNG

**8h30 sáng** — bot tự quét Git log và lịch sử task để soạn sẵn bản nháp báo cáo cho từng người ("hôm qua đã làm gì, hôm nay dự kiến gì"). Mỗi người chỉ cần xác nhận hoặc gõ thêm điểm vướng mắc (blocker). **9h00 sáng** — AI tổng hợp thành một báo cáo duy nhất cho quản lý, hiển thị trên web và gửi vào nhóm chung.

### GIÁ TRỊ NGHIỆP VỤ

Giao ban gói gọn trong vài giây mỗi người, không cần họp đồng thời. Đội phân tán theo múi giờ vẫn đồng bộ. Điểm vướng mắc được phát hiện sớm, quản lý có bức tranh tiến độ hằng ngày mà không cần đi hỏi từng người.



## Luồng "Bản nháp → Phê duyệt" (Human-in-the-loop)

ĐANG PHÁT TRIỂN

### CÁCH HOẠT ĐỘNG

Mọi thứ AI tạo ra — kế hoạch task, phương án Sprint — đều xuất hiện dưới dạng **bản nháp có thể chỉnh sửa**. Người phụ trách xem lại, sửa trực tiếp nếu cần, rồi mới *phê duyệt* để đưa vào backlog chính thức; hoặc *từ chối* để bỏ qua.

### GIÁ TRỊ NGHIỆP VỤ

Kết hợp tốc độ của tự động hóa với sự an tâm của kiểm soát con người. AI không bao giờ là một "hộp đen" đưa ra quyết định không thể can thiệp — đội ngũ luôn nắm quyền cuối cùng.

## Kịch bản thực tế — "Một ngày cùng nền tảng"

Theo chân một yêu cầu đi trọn vòng đời — từ lời nói của một quản lý sản phẩm đến khi tính năng được giao — để thấy sáu trụ cột phối hợp ra sao.

### 1 09:10 — Yêu cầu xuất hiện trong chat

*"Khách hàng muốn xuất báo cáo doanh thu ra file Excel. Làm giúp tôi nhé."*

Quản lý sản phẩm gõ một câu ngắn vào bot trên Slack. Không mở công cụ nào khác, không điền form. (Trụ cột 1 — Tiếp nhận)

### 2 09:10 — AI đối chiếu với mã nguồn

Ngay lập tức, AI tra cứu hệ thống đã lập chỉ mục: tìm thấy module báo cáo hiện có, thư viện xử lý dữ liệu đang dùng, và điểm sẽ cần chỉnh sửa. Kế hoạch sắp tạo ra sẽ bám vào những chi tiết thật này. (Trụ cột 2 — Thấu hiểu mã nguồn)

### 3 09:11 — Bản nháp kế hoạch hiện ra

AI đề xuất một Epic "Xuất báo cáo doanh thu (Excel)" với ba task con: *bổ sung điểm truy xuất dữ liệu · sinh file Excel theo định dạng · thêm nút tải xuống trên giao diện*. Mỗi task kèm tiêu chí nghiệm thu và một mức ước lượng đề xuất. (Trụ cột 1 & 3)

### 4 09:13 — Tech Lead rà soát & duyệt

Tech Lead xem bản nháp, chỉnh nhẹ một tiêu chí nghiệm thu, rồi bấm duyệt. Ba task chính thức vào backlog. (Luồng Bản nháp → Phê duyệt)

### 5 09:14 — AI đàm phán Sprint

*"Sprint hiện tại chỉ còn 3 điểm trống, việc này cần 8 điểm. Anh muốn: (A) chèn thẳng và dời task X sang sau, (B) đưa vào Backlog cho Sprint tới, hay (C) hoán đổi với một task ưu tiên thấp?"*

Quản lý chọn phương án B. Quyết định mất mười giây, hoàn toàn minh bạch về đánh đổi. (Trụ cột 4 — Quản lý Sprint)

### 6 Hôm sau — Lập trình viên bắt tay làm

Đến Sprint mới, một dev tạo nhánh Git cho task đầu tiên. Ngay lập tức ticket tự chuyển sang **Đang làm** và gán đúng dev đó — không ai phải kéo thẻ. (Trụ cột 5 — Tự động hóa Git)

### 7 Vài ngày sau — Review & hoàn thành

Dev mở Pull Request → ticket sang **Đang review**, người review được gán tự động. PR được merge → ticket **Hoàn thành**, changelog tự đăng lên nhóm. Bảng công việc luôn khớp thực tế. (Trụ cột 5)

### 8 Mỗi sáng 09:00 — Bức tranh tiến độ tự đến

Quản lý mở nhóm chat và thấy sẵn báo cáo standup tổng hợp: ai đang làm gì, việc gì vừa xong, điểm vướng mắc nào cần gỡ. Không họp, không đi hỏi từng người. (Trụ cột 6 — Giao ban)

## KẾT QUẢ

Trong toàn bộ hành trình, con người chỉ tham gia ở đúng ba điểm cần phán xét: **nêu yêu cầu**, **duyet kế hoạch**, và **chọn phương án Sprint**. Mọi thao tác vận hành còn lại — bóc tách, ước lượng, cập nhật trạng thái, báo cáo — do nền tảng tự làm.

## Bảo mật & Quyền riêng tư dữ liệu

Mã nguồn là tài sản nhạy cảm nhất của một tổ chức phần mềm. Nền tảng được thiết kế với nguyên tắc "chạm tối thiểu" ngay từ gốc.

### Ba cam kết cốt lõi



#### Không lưu trữ mã nguồn thô

Nền tảng chỉ lưu **siêu dữ liệu** đủ để hiểu và tra cứu hệ thống (cấu trúc, vị trí, dấu hiệu ngữ nghĩa) — không lưu bản thân đoạn mã. Khi cần đọc một đoạn cụ thể để lập kế hoạch, hệ thống lấy đúng đoạn đó theo thời điểm rồi giải phóng ngay, không giữ lại.



#### Khóa truy cập được mã hóa

Thông tin kết nối kho mã nguồn (access token) không bao giờ được lưu ở dạng đọc được. Nó được mã hóa và cất trong kho bí mật riêng; hệ thống chỉ tham chiếu đến nó khi cần, và không bao giờ trả lại token qua giao diện hay báo lỗi.



#### Tiến độ đồng bộ tức thời & đáng tin

Khi có sự kiện Git, tiến độ được cập nhật lên bảng trong **dưới 2 giây**, dùng đối chiếu chính xác để gắn đúng thay đổi với đúng task — tránh nhầm lẫn giữa các công việc.

### VÌ SAO ĐIỀU NÀY QUAN TRỌNG VỚI NGHIỆP VỤ

Nhiều tổ chức không thể áp dụng công cụ AI vì lo mã nguồn bị sao chép ra ngoài. Kiến trúc "chỉ đọc siêu dữ liệu, lấy mã theo yêu cầu" cho phép tận dụng sức mạnh AI mà vẫn giữ mã nguồn trong tầm kiểm soát — một điều kiện tiên quyết để triển khai trong môi trường doanh nghiệp.

## Lộ trình & Mức độ sẵn sàng

Nền tảng được xây theo từng lát cắt giá trị. Bảng dưới cho biết trung thực trạng thái hiện tại của từng năng lực để hỗ trợ ra quyết định.

| Năng lực                                        | Trạng thái      | Ghi chú nghiệp vụ                                                                        |
|-------------------------------------------------|-----------------|------------------------------------------------------------------------------------------|
| Nền tảng dữ liệu & quản lý dự án cơ bản         | Đã sẵn sàng     | Tạo và quản lý dự án; cấu trúc dữ liệu lõi đã vận hành.                                  |
| Kết nối kho mã nguồn & Thấu hiểu mã nguồn       | Đã sẵn sàng     | Kết nối GitHub/GitLab/Bitbucket, tự lập chỉ mục hệ thống với bảo mật "chỉ siêu dữ liệu". |
| Lớp AI & xử lý nền (LLM, hàng đợi công việc)    | Đã sẵn sàng     | Bộ máy AI và xử lý bất đồng bộ — nền tảng cho mọi tính năng thông minh phía sau.         |
| Tra cứu ngữ cảnh mã nguồn cho lập kế hoạch      | Đang phát triển | Cho phép AI lấy đúng phần mã liên quan khi rà việc — mắt xích kế tiếp.                   |
| Tiếp nhận yêu cầu & Tự rà Epic/Task + Ước lượng | Đang phát triển | Trái tim tạo giá trị: biến lời nói thành backlog có cấu trúc, kèm luồng duyệt.           |
| Web Dashboard (Kanban, Backlog, Sprint)         | Trên lộ trình   | Bản đồ chiến lược trực quan, đồng bộ thời gian thực.                                     |
| Tự động hóa qua Git                             | Trên lộ trình   | Chuyển trạng thái ticket theo hoạt động Git thật.                                        |
| Bot hội thoại (Slack / Teams / Discord)         | Trên lộ trình   | Điểm chạm hằng ngày cho tiếp nhận yêu cầu và thông báo.                                  |
| Quản lý Sprint tự chủ & Đàm phán năng lực       | Trên lộ trình   | Tự cân đối quỹ Sprint và đối thoại phương án khi quá tải.                                |
| Bot giao ban bất đồng bộ                        | Trên lộ trình   | Tự soạn & tổng hợp báo cáo standup mỗi sáng.                                             |

### TRIẾT LÝ PHÁT TRIỂN

Nền tảng ưu tiên xây **phần lõi tạo giá trị và bảo mật trước** (thấu hiểu mã nguồn, bộ máy AI), rồi mở rộng dần ra các bề mặt trải nghiệm (dashboard, bot). Cách tiếp cận "lát cắt dọc" này đảm bảo mỗi bước đều tạo ra giá trị dùng được, không phải chờ đến khi hoàn thiện toàn bộ.

## Lợi ích kinh doanh & Kết luận

Tổng hợp giá trị mà nền tảng mang lại, quy về những gì lãnh đạo quan tâm: thời gian, độ tin cậy và năng lực ra quyết định.

### Bốn nhóm lợi ích đo lường được



#### Tiết kiệm thời gian trực tiếp

Loại bỏ hàng giờ mỗi tuần dành cho nhập liệu, cập nhật trạng thái và họp giao ban. Thời gian đó quay lại với việc phát triển sản phẩm.



#### Dữ liệu đáng tin theo thời gian thực

Tiến độ tự sinh từ hoạt động thật, không phụ thuộc trí nhớ con người. Mọi báo cáo và dự báo đứng trên dữ liệu phản ánh đúng thực tế.



#### Ra quyết định nhanh hơn

Đàm phán Sprint tức thời và báo cáo tự động giúp gỡ nút thắt cổ chai ở tầng quản lý. Đội ngũ phản ứng nhanh với thay đổi.



#### Chất lượng & nhất quán

Mọi task đều có tiêu chí nghiệm thu rõ ràng; ước lượng ngày càng chính xác nhờ học từ lịch sử. Giảm hiểu lầm, giảm làm lại.

### Điều làm nên khác biệt

- **AI-Native, không phải AI gắn thêm:** AI là trung tâm vận hành, không phải một nút bấm phụ trợ.
- **Bám mã nguồn thật:** mọi đề xuất dựa trên hiểu biết về hệ thống hiện có, không phải giả định.
- **Tự động có kiểm soát:** con người luôn giữ quyền duyệt ở những điểm quan trọng.
- **Bảo mật ngay từ thiết kế:** không lưu mã nguồn thô — điều kiện tiên quyết cho môi trường doanh nghiệp.

Mục tiêu cuối cùng không phải là một công cụ quản lý tốt hơn — mà là loại bỏ nhu cầu "quản lý công cụ" khỏi công việc hằng ngày của đội ngũ, để họ tập trung tạo ra sản phẩm.

#### BƯỚC TIẾP THEO ĐỀ XUẤT

Kết nối một kho mã nguồn thử nghiệm để trải nghiệm năng lực "Thấu hiểu mã nguồn" đã sẵn sàng, và cùng thống nhất thứ tự ưu tiên cho các tính năng đang trên lộ trình dựa trên nỗi đau lớn nhất của đội ngũ hiện tại.