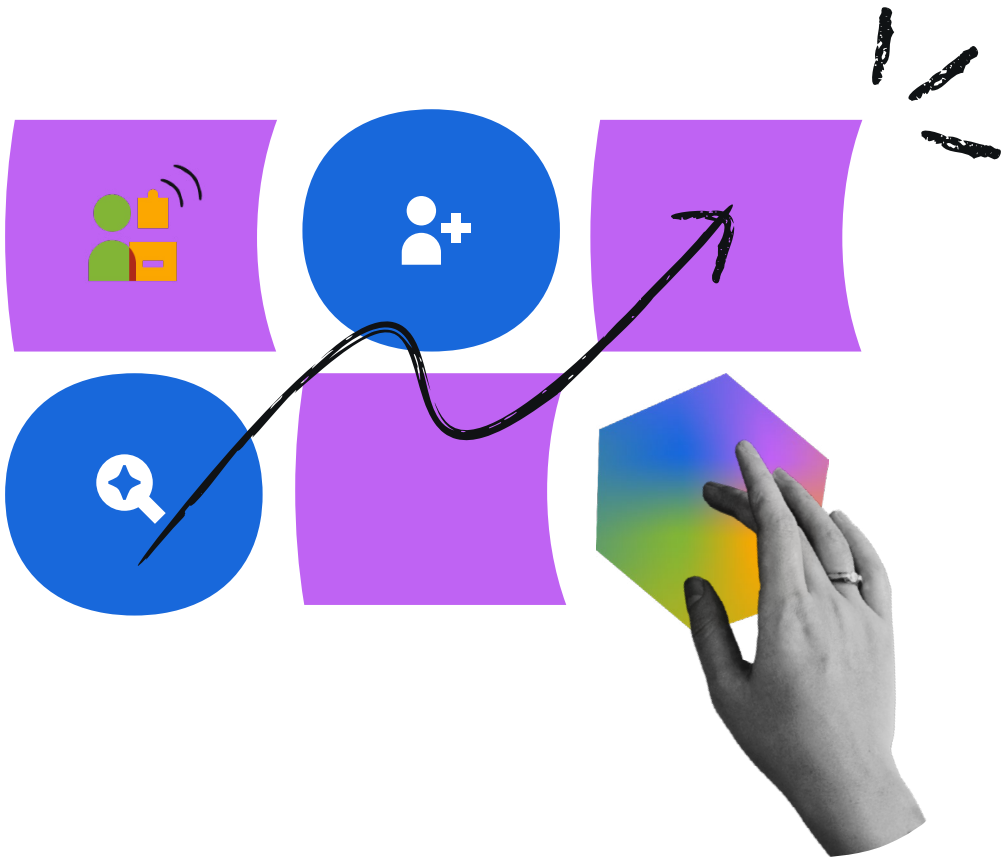
ATLASSIAN

candylio

Gold
Solution Partner



Cẩm nang Atlassian về Rovo Agents.

Tìm hiểu cách xây dựng, triển khai và mở rộng quy mô Rovo AI của bạn

Cẩm nang thực tiễn (playbook) với các bản thiết kế (blueprints), best practices, và các use case thực tế để đưa Rovo Agents vào các workflow của bạn.

P: +84 283 995 4888 | **E:** business@candylio.com | **Facebook, LinkedIn:** candylio | **www.candylio.com**

candylio

ATLASSIAN Gold Solution Partner

Table of contents

Hướng dẫn của Atlassian về Rovo Agents:	1
Tại sao cần Rovo Agents?	3
Hiểu về Rovo agents: Các khối xây dựng của automation	3
Các thành phần cốt lõi của một Rovo agent	4
Khi nào nên dùng chat, agents, hoặc cả hai?	7
Những yếu tố cần cân nhắc khi xây dựng agent hiệu quả	9
Best practices cho thiết kế agent và viết prompt	9
Conversation starters và scenario triggers	11
Khám phá các use case thực tế	12
Tạo ra Rovo Agent tuyệt vời tiếp theo	12
Thử Rovo ngay hôm nay	14

Tại sao cần Rovo Agents?

Rovo agents là những "đồng đội AI" giúp điều phối (orchestrate) công việc trên toàn bộ các công cụ doanh nghiệp, kết nối con người, kiến thức và workflow ở bất cứ nơi nào chúng tồn tại.

Bằng cách tự động hóa các tác vụ lặp lại và đưa đúng ngữ cảnh vào đúng thời điểm, chúng giúp các đội nhóm: hợp tác nhanh hơn, tập trung vào công việc có giá trị cao hơn, tạo ra kết quả tốt hơn – bất kể các công cụ nào đang nằm trong tech stack của bạn.

Hiểu về Rovo agents: Các khối xây dựng của automation

Agents là các hệ thống được điều khiển bởi AI, có khả năng thực hiện các nhiệm vụ cụ thể bằng cách kết hợp **instructions (chỉ dẫn), knowledge (kiến thức), skills (kỹ năng), triggers (kích hoạt).** Sau đó được tăng cường bằng khả năng reasoning của AI. Trong hệ sinh thái Atlassian, Rovo agents đóng vai trò là orchestrator, giúp tự động hóa workflow, tạo ra kết quả xuyên suốt nhiều sản phẩm. Agents có thể xuất hiện trong chat, được gọi trong automation oặc được nhúng trực tiếp vào các business process.

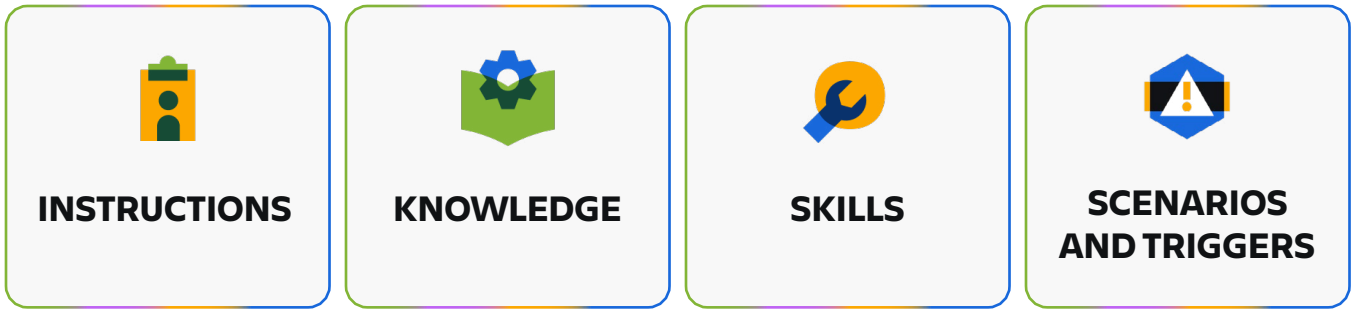
Ở cấp độ tổng thể, Rovo agents giống như những đồng đội ảo, có một vai trò rõ ràng, quyền truy cập vào knowledge của tổ chức, khả năng thực hiện hành động thay cho bạn

Các agent này chạy trên một lớp dữ liệu chung gọi là Atlassian Teamwork Graph, kết nối con người, công việc, mục tiêu, kiến thức trên Jira, Confluence, các công cụ bên thứ ba và nhiều hệ thống khác, điều này cho phép agent hiểu và reasoning dựa trên ngữ cảnh thực của tổ chức, thay vì coi mỗi tài liệu như một mảnh thông tin riêng lẻ.

Rovo agents không chỉ giới hạn trong các sản phẩm Atlassian. Chúng có thể kết nối toàn bộ hệ sinh thái công cụ của bạn thông qua hơn 50 connectors hoặc gần như bất kỳ ứng dụng nào thông qua Model Context Protocol (MCP). Bạn có thể tự động hóa workflow và điều phối công việc trên tất cả các công cụ yêu thích của mình. Với Rovo, AI teammates của bạn có cái nhìn toàn cảnh, không chỉ các silo dữ liệu rời rạc.

Các thành phần cốt lõi của một Rovo agent

Mỗi Rovo agent được xây dựng từ một số khối cấu trúc cơ bản.



Instructions (agent prompt/behavior)

Đây là "bản mô tả công việc" và "sổ tay vận hành" của agent. Instructions xác định:

- Vai trò và mục tiêu của agent (những gì nên và không nên làm).
- Các bước cần thực hiện để hoàn thành nhiệm vụ.
- Ngữ điệu và guardrails cần duy trì (ví dụ: chính xác vs mở rộng).
- Trong Rovo Studio, phần này nằm trong behavior của agent, scenario instructions.

Knowledge

Là dữ liệu và ngữ cảnh mà agent có thể sử dụng. Bao gồm:

- Các công cụ bên thứ ba kết nối vào Teamwork Graph (ví dụ: SharePoint, Google Drive).
- Các nguồn Atlassian như Jira projects, Confluence spaces.
- Các quy tắc nghiệp vụ hoặc template cụ thể được tham chiếu thông qua Confluence Smart Links.
- Knowledge có thể rộng (toàn bộ Teamwork Graph) hoặc giới hạn trong một domain cụ thể (ví dụ: knowledge base của IT helpdesk).

Skills (actions and plugins)

Là các công cụ mà agent có thể dùng để đọc hoặc thay đổi công việc:

- Dọc dữ liệu: truy vấn Jira issues, tìm hồ sơ khách hàng, lấy các ticket gần đây.
- Thực hiện hành động: tạo hoặc cập nhật Google Docs, tạo Jira issue, viết nháp Confluence page, đăng Slack message, gọi API bên ngoài qua Forge/MCP skills, gửi Microsoft Teams message, gửi email qua Google.
- Skills biến agent từ một chatbot chỉ trả lời thành một đồng đội chủ động có thể thúc đẩy công việc.

Scenarios and triggers

Để tránh việc chỉ dùng một prompt duy nhất dễ bị lỗi, agent có thể có nhiều scenarios, mỗi scenario đại diện cho một nhiệm vụ cụ thể. (ví dụ: triage feedback, viết release notes, viết bài community). Mỗi scenario có:

- Instructions riêng, knowledge riêng, skills riêng.
- Một trigger để xác định khi nào scenario đó được sử dụng dựa trên user intent hoặc event context.
- Scenario mặc định (default scenario) sẽ được dùng khi không trigger nào khớp.

Kết hợp với admin và workplace settings của Rovo, các team có thể triển khai agent với guardrails rõ ràng. Bạn có thể quyết định agent kết nối công cụ nào, scenario nào khả dụng cho user nào, mức độ tự động hóa của agent.

Tất cả các thành phần này được điều phối bởi LLM (Large Language Model) - đóng vai trò như "bộ não" của agent, quyết định cách hiểu request, knowledge nào cần tìm, skill nào cần gọi, scenario nào phù hợp nhất.

Controls, guardrails, and governance

Scenarios và triggers cũng đóng vai trò như cơ chế kiểm soát mạnh mẽ. Bằng cách giới hạn chặt chẽ scenario instructions, knowledge sources, skills, bạn có thể:

- Giới hạn dữ liệu mà agent được truy cập.
- Kiểm soát hành động agent được phép thực hiện (ví dụ workflow nhạy cảm như HR hoặc finance) chỉ chạy khi có trigger cụ thể.
- Tạo scenario read-only và action-oriented riêng biệt giúp triển khai automation một cách an toàn theo từng giai đoạn.

Điều gì xảy ra khi user (hoặc automation) gửi một prompt?

Dù là con người nhập liệu trong chat hay automation rule gọi agent, luồng xử lý cơ bản đều giống nhau:

I. AI hiểu request và context

LLM đọc prompt, cuộc hội thoại hiện tại, input có cấu trúc (ví dụ: Jira issue fields), sau đó sẽ xác định intent của user và chọn scenario phù hợp nhất hoặc quay về lại default scenario.

I. Quyết định knowledge và skills cần dùng

Dựa trên cấu hình của scenario, agent sẽ:

i. Tìm kiếm knowledge sources phù hợp (ví dụ: Confluence space, Jira project, app đã kết nối), hoặc

ii. Chọn skills cần sử dụng, chẳng hạn nghiên cứu các issue liên quan, lấy lịch sử khách hàng, tổng hợp feedback gần đây.

II. Reasoning, orchestration và thực hiện hành động

Agent sau đó sẽ:

i. Tổng hợp thông tin từ nhiều nguồn bằng LLM.
ii. Áp dụng business rules trong instructions (và tham chiếu template hoặc rule pages bất kỳ).
iii. Sau đó có thể thực hiện hành động như:
    1. Tạo hoặc cập nhật Jira issues, tạo JPD ideas.
    2. Viết hoặc cập nhật Confluence pages.
    3. Gửi thông báo qua Slack hoặc tools khác nhau.

III. Xác nhận và tóm tắt kết quả

Cuối cùng agent sẽ:

i. Trả về tóm tắt dễ đọc cho người dùng và đưa ra khuyến nghị bước tiếp theo..

ii. Ngoài ra agent có thể xuất structured output chẳng hạn JSON để automation có thể phân nhánh quyết định (ví dụ trường hợp chỉ tạo Jira Product Discovery ideas mới khi feedback thật sự là theme mới).

Trong thực tế, Rovo Agents có thể làm nhiều thứ hơn thay vì chỉ trả lời câu hỏi, Rovo có thể giúp:

Loại bỏ công việc lặp lại, chẳng hạn tóm tắt meeting notes và nhóm hàng trăm feedback tickets.

Giải quyết vấn đề nhiều bước, chẳng hạn phân loại feedback khách hàng, so sánh với roadmap, và chỉ tạo idea mới khi cần.

Cải thiện quyết định bằng cách tổng hợp context từ Atlassian và các tool khác và trình bày theo dạng actionable.

Bằng cách kết hợp các instructions, knowledge, skills và các yếu tố kích hoạt trên nền Teamwork Graph, các tác nhân Rovo trở thành nền tảng cho quá trình tự động hóa đáng tin cậy, từ đầu đến cuối trên các sản phẩm của Atlassian và hơn thế nữa.

Khi nào nên dùng Chat, Agents, hoặc kết hợp?

Trong hệ sinh thái Rovo, chat và agents phục vụ hai mục đích khác nhau nhưng bổ trợ lẫn nhau. Hiểu rõ khi nào nên sử dụng từng loại sẽ giúp bạn tận dụng AI hiệu quả hơn.

Chat

Chat phù hợp nhất khi bạn cần một tương tác ngay lập tức, mang tính ad-hoc và được điều khiển bởi con người. Nó được tối ưu cho việc tra cứu nhanh, các câu hỏi mang tính khám phá, làm rõ thông tin, và những quyết định một lần khi con người sẽ là người thực hiện hành động dựa trên câu trả lời.

- Lý tưởng cho các tìm kiếm nhanh, các bước khắc phục sự cố (troubleshooting), brainstorming, hoặc làm rõ các quy tắc sản phẩm/quy trình ngay tại thời điểm cần thiết.
- Phù hợp khi người dùng mong đợi sự trao đổi qua lại mang tính hội thoại và thực hiện các bước tiếp theo một cách thủ công (không có tác động tự động).
- Chi phí thiết lập thấp: không cần cấu hình phạm vi knowledge, skills, scenarios, hoặc automations.

VÍ DỤ:

Hỏi Rovo Chat "Làm thế nào để reset VPN (Virtual Private Network) của tôi?" - Bạn sẽ ngay lập tức nhận được các hướng dẫn rõ ràng theo từng bước hoặc các liên kết trực tiếp đến các tài nguyên liên quan trong knowledge base của bạn, để bạn không cần phải tự tìm kiếm trong tài liệu.

Rovo Chat hiểu các câu hỏi hỗ trợ IT phổ biến và có thể diễn giải các từ viết tắt như VPN, tự động hướng dẫn bạn đến các bước troubleshooting hoặc các tài nguyên hỗ trợ phù hợp.

Agents

Agents phù hợp nhất khi công việc cần lặp lại, mang tính xác định (deterministic), nhiều bước, hoặc tích hợp với các hệ thống khác. Chúng được thiết kế để điều phối các tác vụ, gọi các skills (tạo Jira items, cập nhật Confluence, gọi API của bên thứ ba), và chạy theo lịch hoặc theo trigger sự kiện.

- Sử dụng khi workflow có từ 2 bước trở lên, yêu cầu tổng hợp dữ liệu, tích hợp với các công cụ, hoặc phải chạy tự động (theo lịch hoặc theo sự kiện).
- Sử dụng khi bạn cần đầu ra có cấu trúc (ví dụ: JSON) để automations có thể phân nhánh, lặp và tạo các artifact một cách xác định.
- Sử dụng khi bạn muốn có giọng điệu/guardrails nhất quán cho nhiều tác vụ liên quan, thiết lập behavior chung và chia công việc thành các scenario để đảm bảo instructions luôn đáng tin cậy.

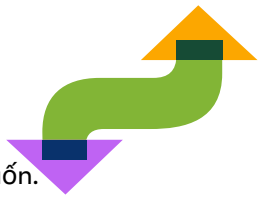
VÍ DỤ:

Bằng cách xây dựng một Rovo Agent đóng vai trò trợ lý phân loại feedback, bạn sẽ có một AI assistant liên tục quét các kênh phản hồi của khách hàng (như support tickets, bình luận NPS và các diễn đàn cộng đồng), nhóm các bình luận liên quan thành các theme rõ ràng, kiểm tra backlog hiện có trong JPD (Jira Product Discovery), và tự động tạo các work item idea mới khi phát hiện các nhu cầu chưa được đáp ứng hoặc các khoảng trống.

Thay vì phải đọc thủ công, gắn tag và loại bỏ trùng lặp các feedback, bạn có thể nhận được một cái nhìn có cấu trúc và được ưu tiên về những gì khách hàng đang yêu cầu, kèm theo các liên kết trực tiếp đến feedback gốc và đến các idea liên quan trong Jira Product Discovery (JPD – sản phẩm của Atlassian để ghi nhận, ưu tiên và quản lý các ý tưởng sản phẩm).

Khi nào nên sử dụng cả hai

Có nhiều trường hợp Rovo Chat và Agent kết hợp với nhau sẽ tối ưu nhất: bắt đầu với chat để khám phá, sau đó chuyển quy trình đã được xác nhận thành agent khi bạn cần khả năng lặp lại, automation hoặc tích hợp.



- Sử dụng chat để thử nghiệm prompt, thu thập ví dụ và xác nhận định dạng đầu ra mong muốn.
- Sau khi xác nhận, xây dựng một agent scenario sử dụng các instructions đó, gắn các knowledge sources và thêm skills cho automation.
- Ví dụ workflow: thử nghiệm "draft release notes" trong chat → chuyển prompt thành một agent scenario với trigger + skills của Confluence/Jira → lên lịch chạy bằng automation rule.

Checklist quyết định – "Tôi nên dùng Rovo Chat hay Agent?"

Đây có phải là một câu hỏi một lần hoặc thỉnh thoảng mới xuất hiện không?	→	**CHAT**
Nhiệm vụ này có được lặp lại thường xuyên hoặc ở quy mô lớn không?	→	**AGENT**
Có cần trải qua nhiều bước hoặc nhiều hệ thống (Jira, Confluence, API của bên thứ ba) không?	→	**AGENT**
Bạn có cần đầu ra ở định dạng có cấu trúc (JSON) cho automation phía sau không?	→	**AGENT**
Bạn có muốn nó chạy tự động theo lịch hoặc theo sự kiện không?	→	**AGENT** kết hợp Jira automations
Bạn có cần sự trao đổi, tìm hiểu qua lại hoặc phán đoán ad-hoc bởi con người không?	→	**CHAT** Trong một số trường hợp có thể kết hợp: chat để khám phá, agent để vận hành chính thức (production).

Nếu bạn nhận thấy mình lặp lại cùng một prompt hoặc cùng một quy trình nhiều lần, hoặc nếu workflow của bạn bao gồm nhiều bước, trigger hoặc tích hợp, thì xây dựng một agent sẽ là cách tiếp cận tốt nhất.

Các yếu tố để có thể build một agent hiệu quả

Trước khi bắt đầu, hãy làm rõ vấn đề bạn muốn giải quyết. Hãy vạch ra workflow, xác định các bước và tìm các cơ hội để tự động hóa hoặc đơn giản hóa. Một problem statement tập trung và một bức tranh rõ ràng về quy trình hiện tại sẽ giúp việc thiết kế agent dễ dàng hơn nhiều, và giúp agent mang lại giá trị đáng tin cậy thay vì chỉ là một "AI helper" mơ hồ.

- Chia các prompt phức tạp thành các scenario dễ quản lý để tăng độ tin cậy.
- Xác định rõ instructions, knowledge sources và skills cần thiết cho từng scenario.
- Sử dụng behaviors để thiết lập giọng điệu và sự nhất quán của agent giữa các scenario.
- Thử nghiệm và lặp lại (test and iterate).

Cuối cùng, hãy đảm bảo cả đội ngũ cùng nhau hợp tác với bạn, vì agents có thể trở thành các tài sản lâu dài. Agents có thể được đồng sở hữu và quản lý bởi các nhóm, không chỉ bởi cá nhân. Diều này cho phép bạn phân chia trách nhiệm cập nhật, giám sát và phản ứng với sự cố cho nhiều thành viên trong nhóm.

Hãy đảm bảo rằng bạn tài liệu hóa cách agent hoạt động, chỉ định rõ người chịu trách nhiệm, và khuyến khích các nhóm đóng góp thêm scenario, ví dụ và cải tiến để agent tiếp tục phát triển cùng với workflow của bạn.

Các yếu tố quan trọng khi xây dựng agent hiệu quả

Thiết kế một agent hiệu quả bắt đầu từ trước khi bạn viết instruction đầu tiên.

Những agent thành công nhất được đối xử giống như một product: chúng có mục đích rõ ràng, đối tượng người dùng xác định và các prompt được viết để đảm bảo độ tin cậy, không chỉ để sáng tạo. Trong phần này, chúng ta tập trung vào cách định hình "job description" của agent, viết instructions rõ ràng và có thể lặp lại, sử dụng examples, templates và structured outputs để đạt được kết quả nhất quán.

Hãy coi những thực hành mẫu này như một playbook để biến một mô hình AI mạnh mẽ thành một đồng đội ảo đáng tin cậy, hành xử theo đúng kỳ vọng của đội ngũ mỗi khi được gọi.

Cách vận dụng agent của bạn giống như một thành viên mới trong nhóm

Xác định rõ vai trò và mục tiêu

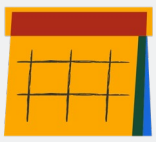
Hãy coi đây như một bản mô tả công việc: agent chịu trách nhiệm giải quyết vấn đề gì, những gì nằm ngoài phạm vi, và bạn sẽ đo lường thành công như thế nào? Một vai trò được giới hạn chặt chẽ (ví dụ: “tóm tắt cập nhật tuần của team” hoặc “phân loại yêu cầu đến thành ba loại”) sẽ giúp việc thiết kế prompt, chọn nguồn dữ liệu phù hợp và tránh hành vi ngoài mong đợi dễ dàng hơn nhiều.

Viết instructions như thể agent không có bất kỳ ngữ cảnh nào trước đó

Hãy viết rõ ràng và theo từng bước. Hãy giả định rằng đó là một người vừa gia nhập team vào ngày đầu tiên và chưa từng thấy các quy trình của bạn trước đây. Hãy mô tả đầy đủ chuỗi hành động, các điểm quyết định và các quy tắc nghiệp vụ mà bạn thường giải thích cho một nhân viên mới. Càng loại bỏ sự mơ hồ (“trước tiên làm X, sau đó làm Y, chỉ khi điều kiện Z xảy ra”) thì agent càng dễ dự đoán và đáng tin cậy.

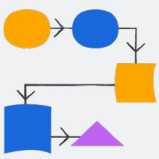
Cung cấp cả ví dụ tốt và ví dụ xấu

Diều này giúp agent có khung tham chiếu để đưa ra kết quả nhất quán hơn. Hãy sử dụng liên kết đến template hoặc quy tắc nghiệp vụ khi có thể. Giống như khi bạn cho một nhân viên mới xem các ticket, tài liệu hoặc công việc trước đây, hãy cung cấp cho agent các ví dụ cụ thể về “kết quả tốt” trông như thế nào, cũng như những điều cần tránh.



Sử dụng markdown hoặc bảng cho structured outputs

Cấu trúc là yếu tố biến một câu trả lời dạng tự do thành thứ có thể tái sử dụng và tự động hóa. Hãy yêu cầu agent phản hồi bằng tiêu đề, danh sách hoặc các cột bảng cố định để ta có thể đọc nhanh và các công cụ phía sau có thể phân tích kết quả đáng tin cậy. Đối với workflow phức tạp hơn, bạn thậm chí có thể định nghĩa các schema đơn giản (ví dụ: các section được đặt tên hoặc các cặp key-value) mà agent phải luôn tuân theo.



Lặp lại dựa trên thử nghiệm thực tế và phản hồi

Phiên bản đầu tiên chỉ là điểm khởi đầu, không phải sản phẩm cuối cùng. Hãy quan sát cách người dùng thực sự sử dụng agent, ghi nhận các trường hợp thất bại và tinh chỉnh instructions, ví dụ hoặc knowledge sources theo thời gian. Hãy coi mỗi lần lặp lại giống như huấn luyện một đồng đội mới: cung cấp hướng dẫn rõ ràng hơn, mẫu tốt hơn và quy tắc cập nhật khi quy trình của bạn thay đổi.

Conversation starters và scenario triggers

Trong khi các ví dụ conversation starter có thể giúp người dùng tương tác với agent, trọng tâm nên nằm ở scenario triggers, tức là những câu lệnh rõ ràng dựa trên ý định (intent) giúp hướng agent đến đúng scenario. Hãy coi conversation starters như các điểm bắt đầu thân thiện ("Hỏi trợ lý HR về quyền lợi của bạn"), còn scenario triggers là tín hiệu chính xác mà agent dùng để quyết định phải làm gì ("Kiểm tra số ngày PTO của tôi trong quý này"). Các scenario trigger được thiết kế tốt sẽ giảm sự mơ hồ, cải thiện độ chính xác khi định tuyến, làm cho hành vi của agent trở nên dễ dự đoán hơn.

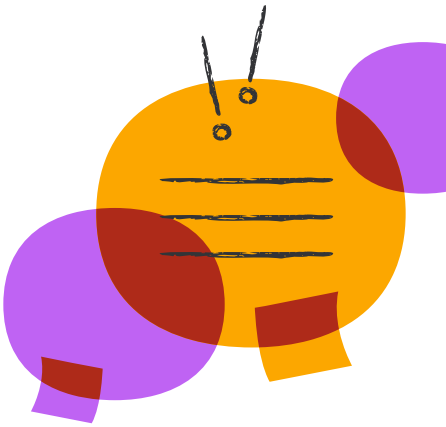
Hãy sử dụng cả ví dụ tích cực và tiêu cực để tinh chỉnh triggers. Hãy cho agent (và đội của bạn) thấy thế nào là một trigger "tốt" cho mỗi scenario, chẳng hạn nó phải cụ thể, định hướng hành động và gắn với một kết quả rõ ràng thay vì các trigger "xấu" vì quá mơ hồ hoặc thiếu định hướng.

VÍ DỤ:

"Phân loại ticket này thành incident, service request hoặc question" là một trigger tốt, còn chẳng hạn như "Giúp với ticket" thì không như vậy. Theo thời gian, bạn có thể thu thập các truy vấn thực tế của người dùng, gắn nhãn truy vấn nào nên hoặc không nên kích hoạt scenario, và đưa chúng trở lại làm ví dụ huấn luyện để cải thiện độ chính xác.

Tránh sử dụng các trigger chỉ gồm một từ; thay vào đó, hãy cung cấp những cụm từ giàu ngữ cảnh phản ánh chính xác ý định thực sự của người dùng. Một từ đơn lẻ như "report" hoặc "access" có thể mang nhiều ý nghĩa khác nhau, khiến agent khó lựa chọn đúng scenario một cách đáng tin cậy. Thay vào đó, hãy sử dụng các cụm từ ngắn bằng ngôn ngữ tự nhiên thể hiện rõ cả hành động và đối tượng, chẳng hạn như "Tạo báo cáo sự cố hằng tuần cho nhóm của tôi" hoặc "Yêu cầu quyền truy cập vào dashboard marketing."

Những trigger giàu ngữ cảnh như vậy giúp agent dễ dàng phân biệt các workflow tương tự nhau, giảm các trường hợp kích hoạt sai và đảm bảo người dùng nhanh chóng đi vào trải nghiệm thực sự giải quyết vấn đề của họ.



Explore practical use cases

Rovo agents hiện đã hỗ trợ một loạt workflow từ engineering đến HR đến product management. Chúng phát huy hiệu quả nhất ở những nơi công việc lặp lại, nhiều bước hoặc cần tổng hợp ngữ cảnh từ nhiều công cụ và biến nó thành các kết quả rõ ràng có thể hành động.

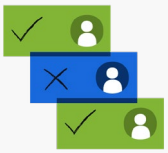
Những ví dụ nổi bật:

Issue organizer

Tự động xem xét các issue trong backlog, nhóm các công việc liên quan, di chuyển issue vào sprint phù hợp và gán chúng cho epic hoặc owner đúng. Điều này giúp các team giữ cho board luôn sạch và tập trung mà không cần nhiều giờ grooming thủ công.

Release notes drafter

Lấy thông tin từ các Jira issue như summary, labels và fix versions, sau đó biến chúng thành các release notes rõ ràng, thân thiện với người dùng. Agent có thể đề xuất bản nháp đầu tiên để team nhanh chóng review, chỉnh sửa và publish lên Confluence hoặc chia sẻ với bên liên quan.



Employee onboarding

Tạo các kế hoạch onboarding được cá nhân hóa, kết hợp các chính sách HR, tài liệu theo vai trò và các nghi thức (ritual) của team. Agent có thể tạo checklist, danh sách tài liệu cần đọc và kế hoạch theo từng tuần để giúp nhân viên mới bắt nhịp nhanh hơn và nhất quán hơn.

Đây chỉ là các điểm khởi đầu: khi bạn hiểu cách mỗi use case kết hợp instructions, knowledge và skills, bạn có thể kết hợp lại để tạo ra các agent phù hợp với workflow riêng của team.

Bắt đầu khởi tạo Rovo Agent

Sẵn sàng chuyển từ “zero” đến “hero” với Rovo chưa? Hãy bắt đầu bằng cách triển khai một nhóm thực hành tập trung gọn nhẹ với một dự án nhỏ nhưng có giá trị cao. Học các khái niệm cốt lõi, theo dõi một ví dụ end- to-end, sau đó triển khai một agent đơn giản và cải tiến dựa trên cách sử dụng thực tế.

Bắt đầu nhanh một project

Xây dựng một agent duy nhất nhưng có tác động lớn trong 4 bước. Mỗi bước đủ nhỏ để có thể hoàn thành trong 1 hay 2 ngày.

1. Viết một problem statement dài một đoạn, xác định rõ người dùng và kết quả mong muốn.
2. Soạn một scenario đơn giản với 3–4 behavior (greeting, thu thập input, thực hiện hành động, xác nhận).
3. Triển khai scenario trong Rovo Studio và chạy 5 bài test nội bộ với đồng nghiệp.
4. Phát hành cho một nhóm pilot nhỏ, thu thập phản hồi và lặp lại.

Hãy bắt đầu với một use case có phạm vi hẹp (ví dụ: phân loại ticket đến, tóm tắt meeting notes hoặc tự động hóa quy trình phê duyệt).

Phạm vi hẹp = học nhanh hơn và tác động có thể dễ đo lường được.

Best practice cần tuân theo

Bắt đầu đơn giản và đo lường sớm.

- Giữ agent đầu tiên chỉ với một mục tiêu chính.
- Thêm telemetry để theo dõi usage, success rate và hiển thị thông báo phản hồi sau mỗi session.
- Sử dụng các thay đổi có version để có thể rollback nhanh.
- Thu thập ít nhất một user quote định tính mỗi tuần trong giai đoạn pilot để giúp ưu tiên cải tiến.

Kế hoạch đề xuất 30 / 60 / 90 ngày

0–30 Ngày	Hoàn thành khóa học nền tảng, xem video và đăng bài trong hub. Soạn problem statement và scenario.
31–60 Ngày	Xây dựng agent trong Studio, chạy thử nghiệm và triển khai pilot cho 5–10 người dùng. Thu thập phản hồi định lượng và định tính.
61-90 Ngày	Lặp lại dựa trên phản hồi, thêm một behavior hoặc knowledge source, và chuẩn bị kế hoạch rollout cho team.

Rovo agents mở ra khả năng quy mô hóa tự động trong tổ chức khi bạn kết hợp việc học với những dự án nhỏ nhưng thực tế. Hãy chọn một vấn đề rõ ràng, làm theo các bước ở trên và bắt đầu xây dựng ngay hôm nay để mang lại giá trị có thể đo lường được cho đội ngũ của bạn.

Hãy thử Rovo ngay hôm nay

Các tổ chức đã đạt những thành công nhất định ngay cả khi Rovo vẫn còn ở giai đoạn đầu: những team coi agent như đồng đội thực sự với vai trò rõ ràng, instructions mạnh mẽ và vòng lặp học-xây dựng chặt chẽ sẽ đạt được tác động nhanh nhất và bền vững nhất.

Hãy bắt đầu nhỏ, chọn một workflow có giá trị cao và sử dụng các thực hành trong hướng dẫn này để thiết kế một agent mà team của bạn có thể tin tưởng và tiếp tục phát triển theo thời gian.

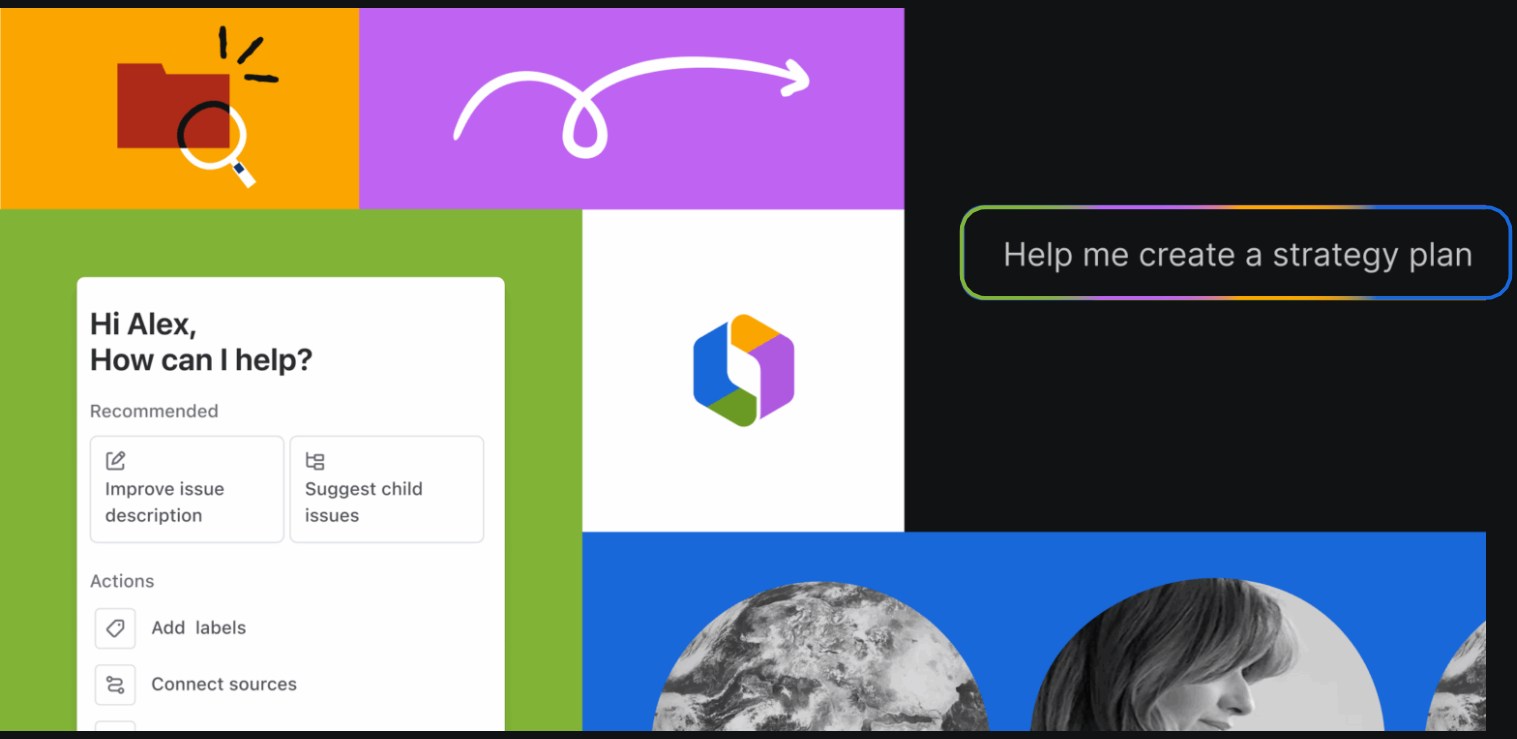
Hãy bắt đầu thử nghiệm với các đồng đội AI của riêng bạn và khám phá cách Rovo Agents có thể chuyển đổi workflow của bạn.

Để xem Rovo Agents có thể làm được những gì, hãy liên hệ với chúng tôi ngay hôm nay.

P: +84 283 995 4888
E: business@candylio.com
Facebook, LinkedIn: candylio
www.candylio.com

candylio

ATLASSIAN Gold Solution Partner



©2026 Atlassian. All rights reserved. CSC-3610.

ATLASSIAN