

Câu hỏi và bài tập

1. Tại sao cần viết Proposal?

- Vì việc viết **Proposal (Đề xuất)** là một bước quan trọng trong quá trình lập kế hoạch và thực hiện dự án, **vì nó giúp:**
- **Xác định rõ mục tiêu và phạm vi:** Giúp định hình rõ ràng vấn đề cần giải quyết và định nghĩa phạm vi công việc, tránh tình trạng làm việc ngoài kế hoạch.
- **Truyền đạt ý tưởng một cách chuyên nghiệp:** Giúp người đọc (nhà đầu tư, sếp, khách hàng, đối tác) hiểu được đề xuất của bạn. Thể hiện tính logic và khả thi của dự án.
- **Thuyết phục người ra quyết định:** Nếu cần tài trợ, phê duyệt ngân sách hoặc nhân sự, proposal giúp bạn thuyết phục bên liên quan và cung cấp số liệu, phân tích để chứng minh tính hiệu quả.
- **Tạo cơ sở để theo dõi và đánh giá:** Đóng vai trò như một tài liệu tham chiếu trong quá trình triển khai và giúp kiểm soát tiến độ, đo lường kết quả so với kế hoạch.
- **Tăng tính minh bạch và giảm rủi ro:** Giúp các bên liên quan hiểu rõ quyền lợi và trách nhiệm và hạn chế hiểu lầm hoặc tranh chấp sau này.

2. Điểm chính nhất cần có trong Proposal là gì?

Điểm chính nhất trong một Proposal là **Giải pháp đề xuất (Proposed Solution)**.

Lý do:

- Đây là phần thể hiện cách giải quyết vấn đề mà Proposal hướng tới.
- Nếu giải pháp không rõ ràng, khả thi hoặc không thuyết phục, thì các phần khác (mục tiêu, ngân sách, rủi ro...) sẽ không có giá trị.
- Nhà quản lý, nhà đầu tư hoặc người ra quyết định sẽ tập trung vào phần này để đánh giá tính thực tế của Proposal.

Tuy nhiên, để giải pháp đề xuất có trọng lượng, cần có các phần khác để hỗ trợ cho Giải pháp đề xuất(Proposed Solution):

- Vấn đề cần giải quyết (nêu rõ lý do tại sao cần giải pháp).
- Mục tiêu và phạm vi (xác định rõ hướng đi).
- Kế hoạch thực hiện (chứng minh tính khả thi).
- Ngân sách & nguồn lực (đảm bảo khả năng thực hiện).

3. Để có bản kế hoạch tốt cần làm những việc gì?

Để có một bản kế hoạch tốt, bạn cần thực hiện các bước sau:

- **Xác định mục tiêu rõ ràng:** Xác định chính xác mục tiêu cuối cùng của kế hoạch. Sử dụng nguyên tắc SMART (Cụ thể, Đo lường được, Khả thi, Thực tế, Có thời hạn).
- **Nghiên cứu và thu thập thông tin:** Phân tích thị trường, khách hàng, đối thủ, hoặc tình hình nội bộ. Thu thập dữ liệu thực tế để đảm bảo kế hoạch có căn cứ.
- **Xác định phạm vi công việc:** Làm rõ những gì sẽ làm và những gì không làm. Tránh tình trạng "quá tải" hoặc thay đổi liên tục làm ảnh hưởng tiến độ.
- **Lập kế hoạch hành động:** Chia nhỏ công việc thành các bước cụ thể. Xác định ai làm gì, thời gian hoàn thành, và cách đo lường kết quả.
- **Phân bổ nguồn lực hợp lý:** Xác định nhân sự, ngân sách, và công cụ cần thiết. Đảm bảo tất cả yếu tố này đều có sẵn hoặc có kế hoạch bổ sung.
- **Dự báo rủi ro và phương án xử lý:** Xác định các rủi ro tiềm ẩn có thể ảnh hưởng đến kế hoạch. Đề xuất phương án dự phòng để giảm thiểu tác động.
- **Thiết lập thời gian và cột mốc quan trọng:** Xác định timeline chi tiết với các mốc quan trọng để theo dõi tiến độ. Đảm bảo các phần việc không bị chồng chéo hoặc trì hoãn.

- **Đánh giá và điều chỉnh:** Kiểm tra thường xuyên xem kế hoạch có đang đi đúng hướng không. Điều chỉnh linh hoạt nếu có thay đổi bất ngờ.

4. Giải thích các thuật ngữ: milestones, deliverables, releases

Milestones (Cột mốc quan trọng)

- Là điểm kiểm tra quan trọng trong kế hoạch dự án.
- Thường đánh dấu hoàn thành một giai đoạn lớn hoặc một sự kiện quan trọng.
- Không phải là công việc cụ thể mà là điểm xác nhận tiến độ.

Ví dụ:

- Hoàn thành thiết kế giao diện phần mềm.
- Ký kết hợp đồng với khách hàng.
- Hoàn tất kiểm thử nội bộ.

Deliverables (Sản phẩm bàn giao)

- Là kết quả cụ thể của một công việc hoặc một giai đoạn trong dự án.
- Có thể là tài liệu, phần mềm, mô hình, báo cáo...
- Thường là đầu ra có thể đo lường được.

Ví dụ:

- Báo cáo khảo sát thị trường.
- Mô hình thiết kế hệ thống.
- Một module phần mềm hoàn chỉnh.

Releases (Phiên bản phát hành)

- Là sản phẩm hoàn thiện được phát hành đến người dùng hoặc khách hàng.
- Thường gồm một hoặc nhiều deliverables đã hoàn tất.
- Trong phần mềm, mỗi release có thể là bản thử nghiệm (beta) hoặc bản chính thức (stable).

Ví dụ:

- Phiên bản 1.0 của ứng dụng di động.
- Bản cập nhật mới cho phần mềm với tính năng mới.
- Sản phẩm vật lý được xuất xưởng để bán ra thị trường.

5. Người quản lý dự án có cần có kỹ năng phân tích, thiết kế hay lập trình tốt không? (giải thích)

Không bắt buộc, nhưng nếu có sẽ là lợi thế lớn.

1. Kỹ năng phân tích – Rất quan trọng

- Quản lý dự án cần hiểu rõ yêu cầu, mục tiêu và rủi ro của dự án.
- Kỹ năng phân tích giúp họ đọc, đánh giá và đưa ra quyết định dựa trên dữ liệu.
- Họ không cần giỏi phân tích như một business analyst, nhưng cần đủ khả năng hiểu báo cáo và đánh giá hiệu suất.

2. Kỹ năng thiết kế – Không bắt buộc

- Nếu dự án liên quan đến phần mềm, kiến trúc hệ thống hoặc UI/UX, thì hiểu biết về thiết kế sẽ giúp quản lý dễ dàng giao tiếp với nhóm kỹ thuật.
- Tuy nhiên, việc thiết kế chi tiết nên do các chuyên gia đảm nhận (UI/UX Designer, System Architect, v.v.).
- Một quản lý giỏi chỉ cần hiểu khái niệm thiết kế ở mức cao để đưa ra quyết định phù hợp.

3. Kỹ năng lập trình – Không cần giỏi, nhưng có lợi thế

- Một quản lý dự án phần mềm không cần biết lập trình chuyên sâu, nhưng hiểu về công nghệ, quy trình phát triển phần mềm sẽ giúp họ:
 - o Giao tiếp tốt hơn với nhóm kỹ thuật.
 - o Hiểu được các thách thức về kỹ thuật.
 - o Đánh giá thời gian, nguồn lực và rủi ro chính xác hơn.

- Nếu không có nền tảng kỹ thuật, họ có thể dựa vào nhóm kỹ thuật để hỗ trợ đánh giá chuyên môn.

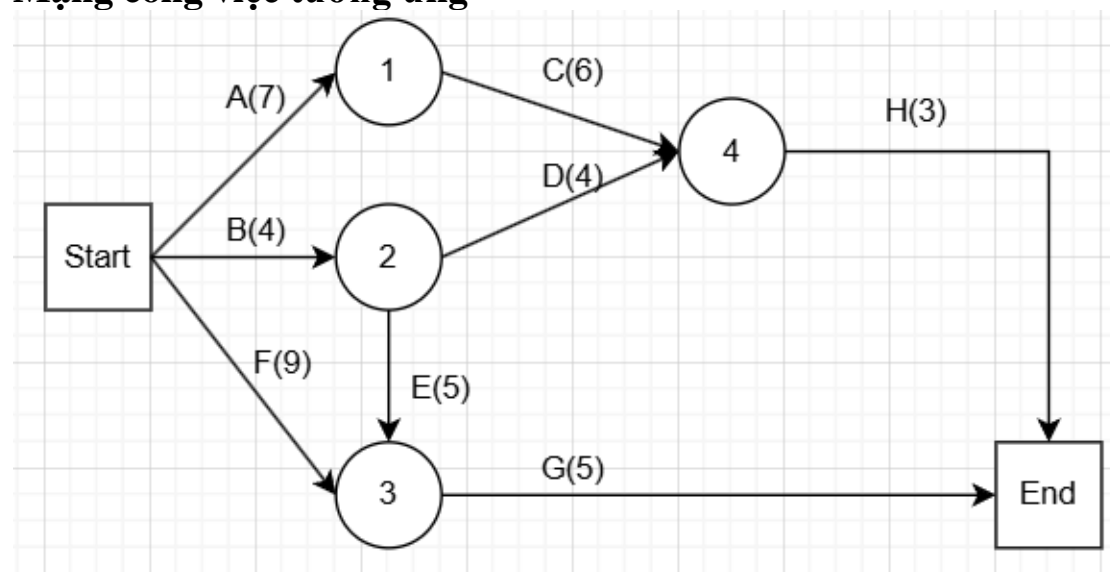
Bài tập

Bảng công việc đã được xác định

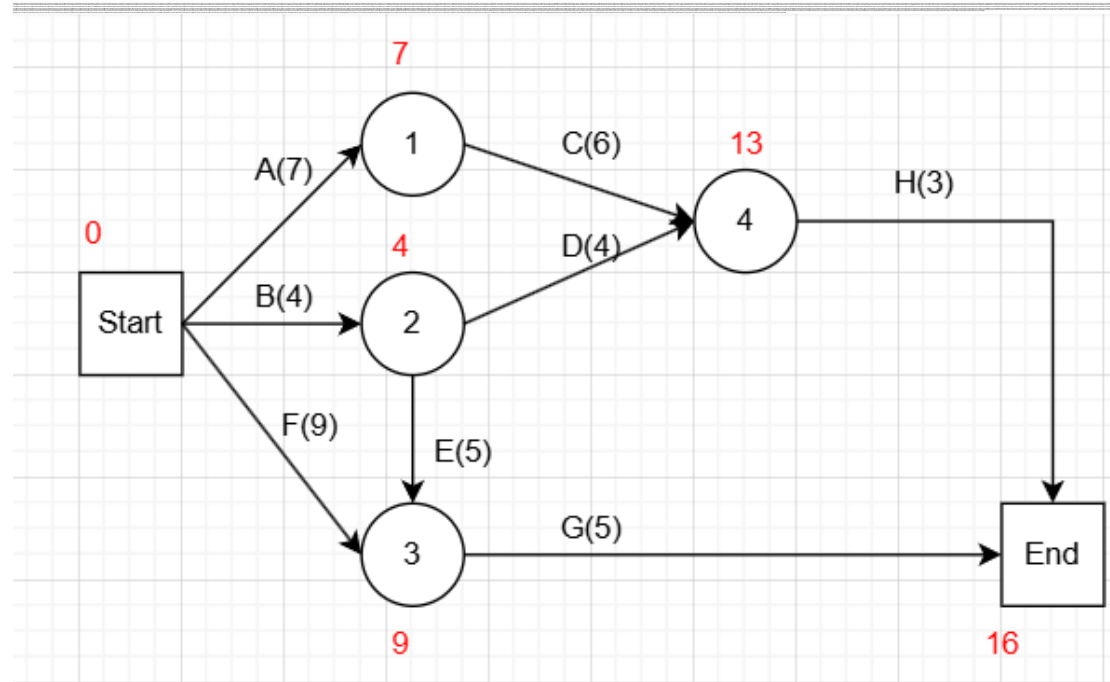
- Vẽ mạng công việc và tìm đường găng

Id.	Activity Name	Duration (weeks)	Precedents
A	Hardware selection	7	
B	Software design	4	
C	Hardware Installation	6	A
D	Coding	4	B
E	Data Preparation	5	B
F	User Documentation	9	
G	User Training	5	E,F
H	System Installation	3	C,D

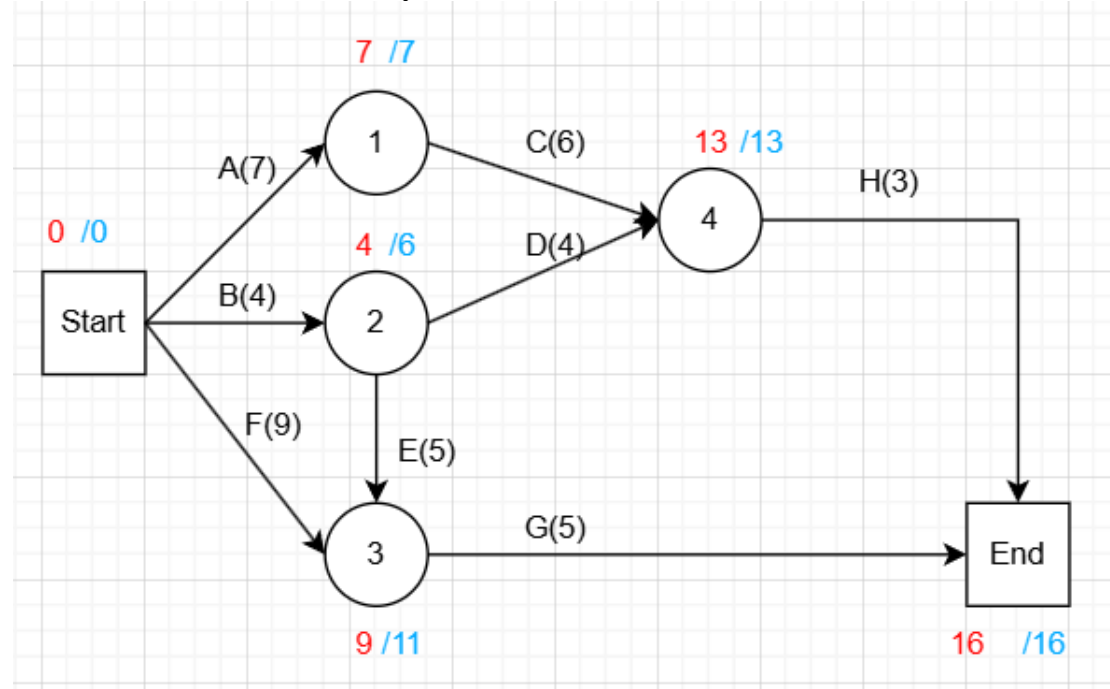
Mạng công việc tương ứng



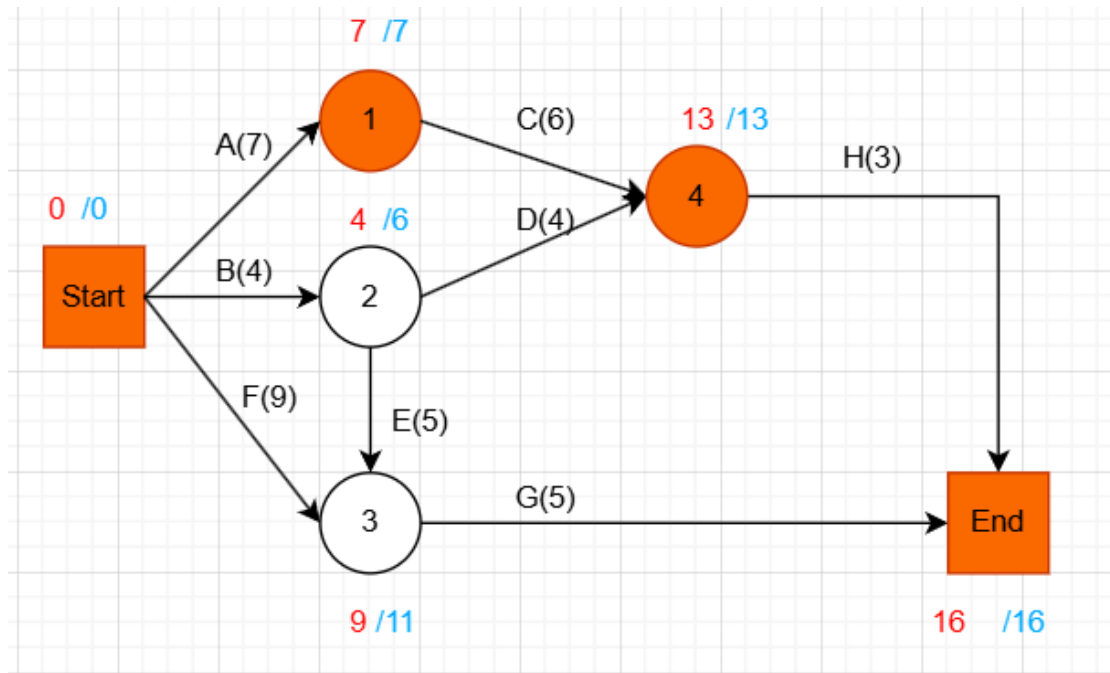
Thời điểm bắt đầu sớm nhất: t_s



Thời điểm bắt đầu muộn nhất: t_m



Công việc găng, đường găng



Biểu đồ lịch trình dự án (Gantt)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A																
B																
C																
D																
E																
F																
G																
H																