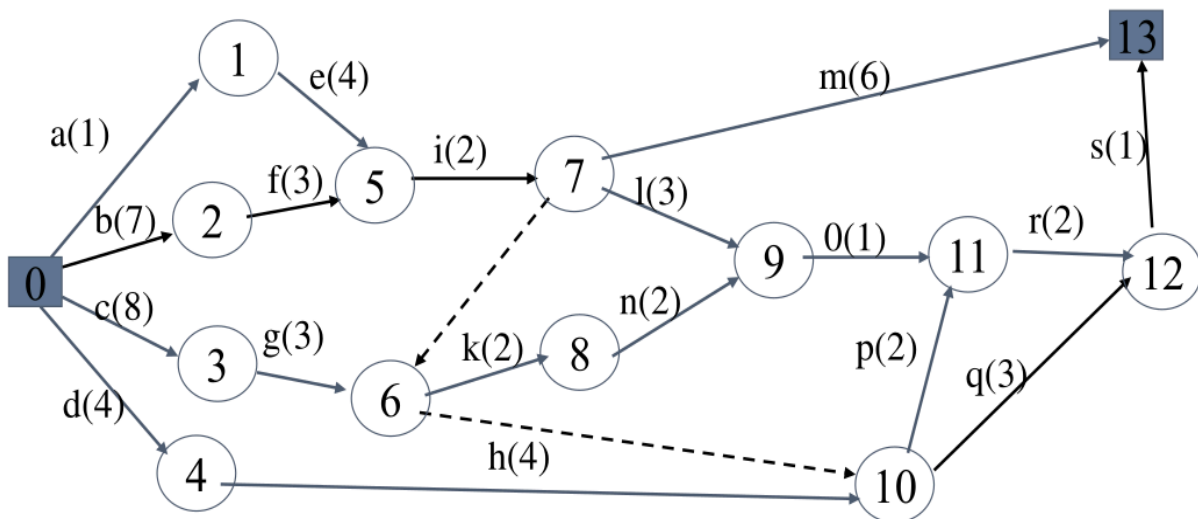
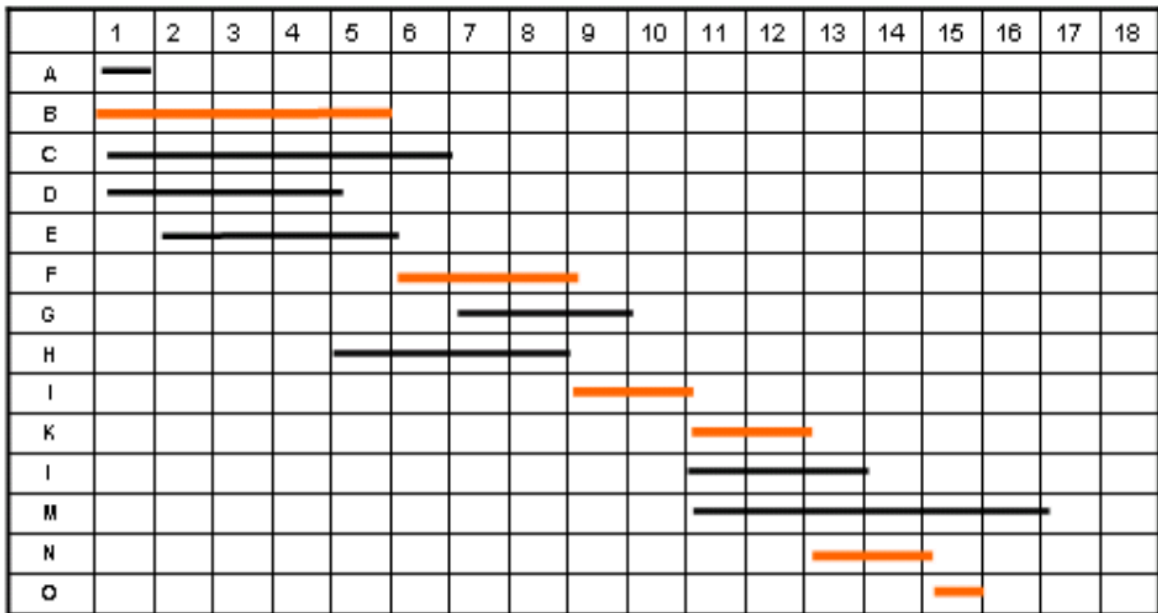


Câu 1: Xây dựng sơ đồ mạng công việc. Tìm đường Gantt và tính chi phí. Vẽ giản đồ Gantt. (3đ)

Công việc	Thời gian	Đi sau công việc	Công việc	Thời gian	Đi sau công việc
a	1	-	k	2	g, i
b	7	-	m	3	i
c	8	-	l	3	i
d	4	-	n	2	k
e	4	a	o	1	l, n
f	3	b	p	2	g, i, h
g	3	c	q	3	g, i, h
h	4	d	r	2	o, p
i	2	e, f	s	1	r, q

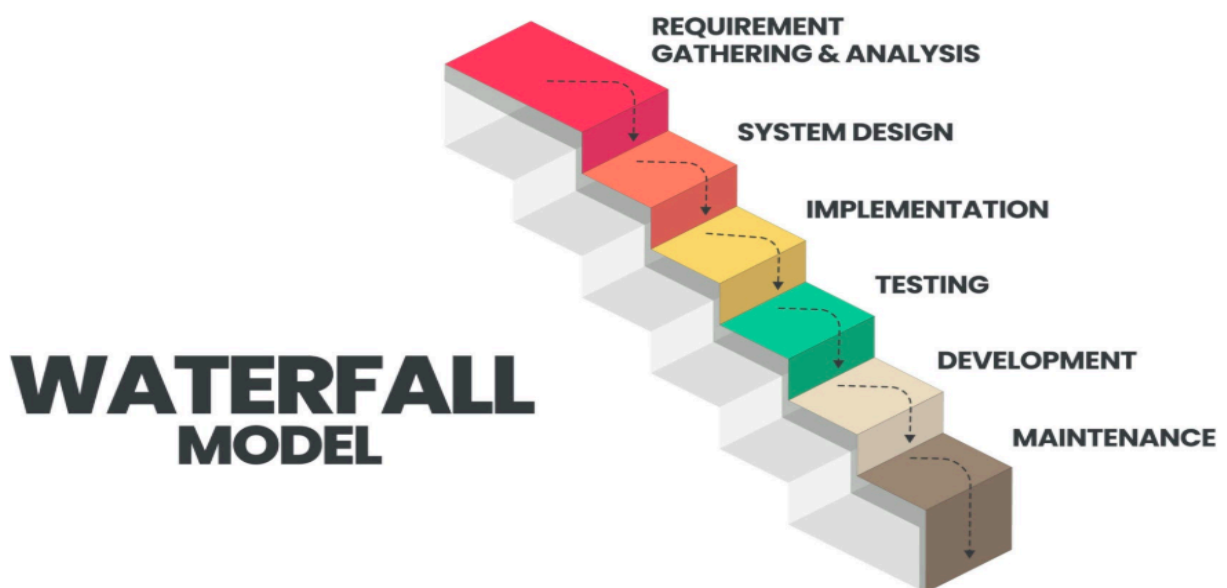




Câu 2: (1đ)

2.1. Các phương pháp phát triển phần mềm: Agile, thác nước, thác nước cải tiến => Ưu điểm, nhược điểm, phần mềm, giải thích,...

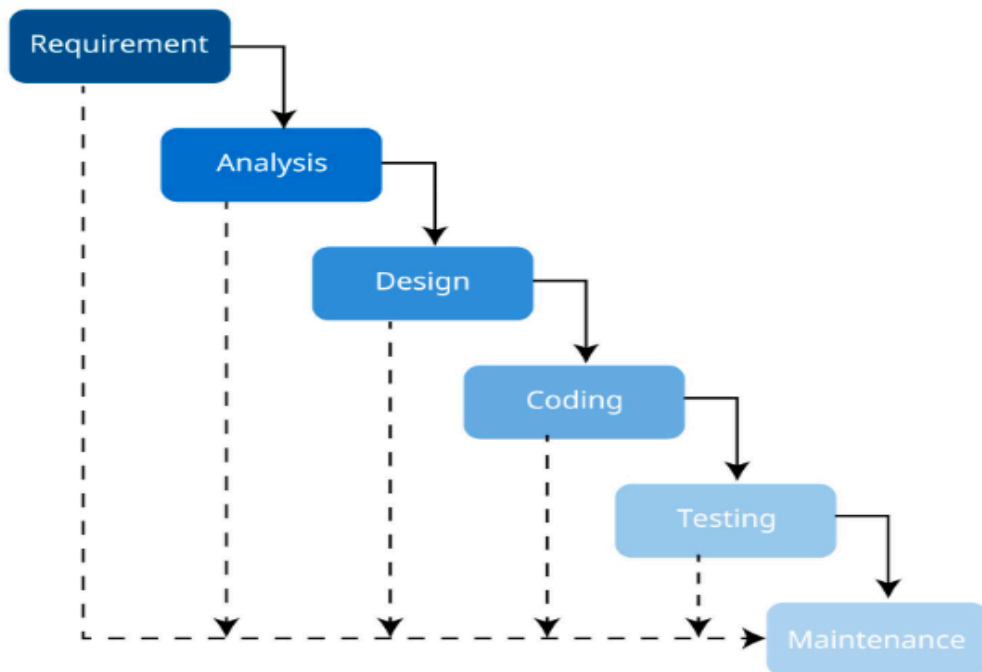
Thác nước



- Phương pháp Thác nước là một mô hình phát triển phần mềm tuyến tính, tuần tự, trong đó mỗi giai đoạn phải được hoàn thành trước khi giai đoạn tiếp theo bắt đầu. Giống như một dòng thác nước chảy xuống, các giai đoạn chỉ đi theo một chiều, không có sự quay lại. Các giai đoạn điển hình bao gồm:
 - + Phân tích yêu cầu (Requirement Analysis): Thu thập và tài liệu hóa tất cả các yêu cầu của hệ thống.
 - + Thiết kế (Design): Thiết kế kiến trúc tổng thể của hệ thống, cơ sở dữ liệu, giao diện người dùng, v.v.
 - + Phát triển/Triển khai (Implementation/Coding): Viết mã và phát triển phần mềm dựa trên các tài liệu thiết kế.
 - + Kiểm thử (Testing): Kiểm tra phần mềm để đảm bảo nó đáp ứng các yêu cầu và không có lỗi.
 - + Triển khai/Bảo trì (Deployment/Maintenance): Triển khai phần mềm và thực hiện các hoạt động bảo trì, sửa lỗi, nâng cấp sau này.
- Ưu điểm:
 - Dễ hiểu và dễ quản lý: Do tính tuần tự rõ ràng, mỗi giai đoạn có các đầu ra và thời hạn cụ thể.
 - Tài liệu hóa chặt chẽ: Yêu cầu và thiết kế được ghi chép chi tiết, giúp ích cho việc bảo trì và chuyển giao kiến thức.
 - Phù hợp với các dự án nhỏ, yêu cầu ổn định: Khi yêu cầu được xác định rõ ràng ngay từ đầu và ít thay đổi.
 - Tỷ lệ lỗi giảm dần ở các giai đoạn sau: Do có thời gian kiểm thử và sửa lỗi kỹ lưỡng.
- Nhược điểm:
 - Thiếu linh hoạt: Khó khăn trong việc thay đổi yêu cầu sau khi một giai đoạn đã hoàn thành. Nếu có thay đổi, có thể dẫn đến việc phải làm lại từ đầu.
 - Phát hiện lỗi muộn: Lỗi thường chỉ được phát hiện ở giai đoạn kiểm thử, gây tốn kém và khó sửa chữa.
 - Khách hàng ít tham gia: Khách hàng chỉ thực sự thấy sản phẩm vào cuối chu trình, có thể không đáp ứng đúng kỳ vọng.

- Thời gian triển khai dài: Dự án có thể kéo dài nếu các giai đoạn phải chờ đợi nhau.
- Phần mềm phù hợp:
 - Các dự án có yêu cầu rất rõ ràng, ổn định và ít có khả năng thay đổi.
 - Các dự án nhỏ hoặc các dự án mà các công nghệ và kiến trúc đã được thiết lập.
 - Ví dụ: Các hệ thống nhúng (embedded systems) với chức năng cố định, các phần mềm quân sự, y tế có quy định chặt chẽ.

Thác nước cải tiến



- Thác nước cải tiến là một biến thể của mô hình thác nước, kết hợp yếu tố lặp lại (iterative) để khắc phục nhược điểm của mô hình thác nước truyền thống. Thay vì chỉ có một chu trình duy nhất, dự án được chia thành nhiều chu trình nhỏ hơn hoặc các "phiên bản" (iterations), mỗi phiên bản đi qua các giai đoạn của thác nước (yêu cầu, thiết kế, triển khai, kiểm thử). Sau mỗi phiên bản, một phần chức năng của sản phẩm được phát hành hoặc xem xét, và phản hồi được sử dụng để cải thiện các phiên bản tiếp theo.

- Ưu điểm:
 - Linh hoạt hơn Waterfall truyền thống: Cho phép điều chỉnh yêu cầu trong các vòng lặp sau.
 - Phát hiện lỗi sớm hơn: Do có các vòng lặp kiểm thử nhỏ, lỗi có thể được tìm thấy và sửa chữa sớm hơn.
 - Khách hàng tham gia nhiều hơn: Khách hàng có thể cung cấp phản hồi sau mỗi vòng lặp.
 - Giảm rủi ro: Rủi ro được phân tán và quản lý tốt hơn qua từng vòng lặp.
- Nhược điểm:
 - Vẫn còn tính tuần tự: Mỗi vòng lặp vẫn tuân thủ trình tự của thác nước, có thể gây mất thời gian nếu có sự thay đổi lớn.
 - Khó xác định số vòng lặp: Việc quyết định bao nhiêu vòng lặp là đủ có thể khó khăn.
 - Có thể phát sinh thêm chi phí: Nếu có quá nhiều thay đổi giữa các vòng lặp.
- Phần mềm phù hợp:
 - Các dự án có yêu cầu tương đối rõ ràng nhưng vẫn có khả năng thay đổi ở một mức độ nhất định.
 - Các dự án lớn có thể được chia thành các phần nhỏ hơn, có thể quản lý được.
 - Ví dụ: Phát triển các ứng dụng web phức tạp, hệ thống quản lý doanh nghiệp (ERP) với các module riêng biệt.

Phương pháp Agile

- Agile (Linh hoạt) là một triết lý phát triển phần mềm dựa trên Tuyên ngôn Agile, tập trung vào sự cộng tác, phản hồi liên tục, thích ứng với thay đổi và phân phối giá trị thường xuyên. Thay vì đi theo một kế hoạch cứng nhắc, Agile ưu tiên phản ứng nhanh với các thay đổi và sự tham gia của khách hàng. Các phương pháp Agile phổ biến bao gồm Scrum, Kanban, Extreme Programming (XP)
- Ưu điểm:

- Linh hoạt cao: Khả năng thích ứng nhanh chóng với các yêu cầu thay đổi, ngay cả ở các giai đoạn muộn của dự án.
 - Phản hồi liên tục từ khách hàng: Khách hàng được tham gia thường xuyên, giúp đảm bảo sản phẩm cuối cùng đáp ứng đúng nhu cầu.
 - Phân phối giá trị thường xuyên: Phần mềm hoạt động được phân phối theo các chu kỳ ngắn (sprints), cho phép khách hàng sử dụng và đưa ra phản hồi sớm.
 - Chất lượng cao hơn: Do có kiểm thử liên tục và khả năng sửa lỗi sớm.
 - Giảm thiểu rủi ro: Rủi ro được giảm thiểu do khả năng điều chỉnh nhanh chóng.
 - Tăng cường sự cộng tác: Giữa các thành viên trong nhóm và với khách hàng.
- Nhược điểm:
- Yêu cầu sự tham gia tích cực của khách hàng: Nếu khách hàng không tham gia đủ, dự án có thể gặp khó khăn.
 - Khó dự đoán thời gian và chi phí ban đầu: Do tính linh hoạt cao, việc đưa ra ước tính chính xác từ đầu có thể khó.
 - Tài liệu hóa ít hơn: Có thể gây khó khăn cho việc bảo trì hoặc chuyển giao dự án nếu không được quản lý tốt.
 - Cần đội ngũ tự quản lý và kinh nghiệm: Đòi hỏi các thành viên có kỹ năng tốt và khả năng tự tổ chức cao.
- Phần mềm phù hợp:
- Các dự án có yêu cầu không rõ ràng ngay từ đầu, hoặc có khả năng thay đổi cao.
 - Các dự án cần thời gian đưa ra thị trường nhanh chóng.
 - Các dự án mà sự hợp tác liên tục với khách hàng là rất quan trọng.
 - Ví dụ: Phát triển các ứng dụng di động, phần mềm khởi nghiệp (start-up), các sản phẩm phần mềm dựa trên ý tưởng mới, các dự án liên tục cần cải tiến và cập nhật.

So sánh tổng quan:

Đặc điểm	Thác nước (Waterfall)	Thác nước cải tiến (Iterative Waterfall)	Agile
Tính tuần tự	Rất cao, tuyến tính	Vẫn có tính tuần tự trong mỗi vòng lặp	Linh hoạt, lặp đi lặp lại
Linh hoạt	Thấp	Trung bình	Rất cao
Khách hàng	Ít tham gia, chỉ ở đầu và cuối	Tham gia nhiều hơn ở cuối mỗi vòng lặp	Tham gia liên tục, là một phần của nhóm
Phát hiện lỗi	Muộn, ở giai đoạn kiểm thử	Sớm hơn, ở cuối mỗi vòng lặp	Rất sớm, liên tục
Rủi ro	Cao, đặc biệt khi yêu cầu thay đổi	Trung bình, được phân tán	Thấp, do thích ứng nhanh
Tài liệu hóa	Rất chặt chẽ	Tốt	Vừa phải, tập trung vào phần mềm hoạt động
Phù hợp với	Dự án yêu cầu ổn định, rõ ràng	Dự án yêu cầu tương đối rõ, có thể thay đổi	Dự án yêu cầu thay đổi cao, cần nhanh

2.2. Các quy trình phát triển phần mềm (5 quy trình - pha) => Đặc điểm, ưu, nhược

Pha Phân tích Yêu cầu

- Đặc điểm: Thu thập, hiểu, và tài liệu hóa chi tiết nhu cầu của hệ thống và người dùng.
- Ưu điểm: Đảm bảo sản phẩm đúng nhu cầu, giảm rủi ro làm lại.
- Nhược điểm: Tốn thời gian nếu yêu cầu phức tạp, khó quản lý thay đổi.

Pha Thiết kế

- Đặc điểm: Lập kế hoạch cách hệ thống sẽ được xây dựng (kiến trúc, CSDL, giao diện, module).
- Ưu điểm: Cung cấp lộ trình rõ ràng, phát hiện vấn đề sớm, đảm bảo tính nhất quán.

- Nhược điểm: Tốn thời gian cho hệ thống phức tạp, đòi hỏi kinh nghiệm, khó thay đổi lớn.

Pha Phát triển / Triển khai

- Đặc điểm: Chuyển bản thiết kế thành mã nguồn thực tế, viết code, kiểm thử đơn vị.
- Ưu điểm: Hình thành sản phẩm, tạo phần mềm hoạt động.
- Nhược điểm: Dễ phát sinh lỗi, tốn kém nhân lực nếu không quản lý tốt.

Pha Kiểm thử

- Đặc điểm: Kiểm tra phần mềm để đảm bảo đúng yêu cầu, không lỗi, đáp ứng chất lượng.
- Ưu điểm: Phát hiện lỗi sớm, đảm bảo chất lượng, tăng sự hài lòng người dùng.
- Nhược điểm: Tốn kém, không thể tìm hết lỗi, nếu lỗi phát hiện muộn sẽ rất tốn kém sửa chữa.

Pha Triển khai & Bảo trì

- Đặc điểm:
 - Triển khai: Đưa phần mềm vào môi trường hoạt động thực tế.
 - Bảo trì: Sửa lỗi, cập nhật tính năng, nâng cấp, thích ứng với môi trường thay đổi.
- Ưu điểm: Đưa sản phẩm đến người dùng, đảm bảo hoạt động liên tục, kéo dài giá trị.
- Nhược điểm: Chi phí bảo trì cao theo thời gian, đòi hỏi đội ngũ chuyên trách.

2.3. Các vị trí trong ngành CNTT? Job Description của vị trí đó.

1. Kỹ sư Phát triển Phần mềm (Software Engineer / Developer)

- **Tổng quan:** Là người thiết kế, phát triển, kiểm thử và duy trì các ứng dụng phần mềm. Họ có thể chuyên về Front-end, Back-end, Full-stack, Mobile, Desktop, Game, hoặc AI/ML.
- **Mô tả công việc:**
 - Phân tích yêu cầu và thiết kế kiến trúc phần mềm.
 - Viết mã sạch, hiệu quả và dễ bảo trì bằng các ngôn ngữ lập trình (ví dụ: Java, Python, C#, JavaScript, Go, Swift, Kotlin).
 - Thực hiện kiểm thử đơn vị (unit testing) và kiểm thử tích hợp (integration testing).
 - Tham gia vào quy trình review code để đảm bảo chất lượng.
 - Hợp tác với các thành viên khác trong nhóm (QA, BA, PM) để đưa ra giải pháp.
 - Nghiên cứu và áp dụng các công nghệ mới.
 - Sửa lỗi (bug fixing) và tối ưu hiệu suất phần mềm.

2. Kỹ sư Kiểm thử Phần mềm (Software Quality Assurance - QA / Tester)

- **Tổng quan:** Đảm bảo chất lượng của sản phẩm phần mềm thông qua việc kiểm tra, phát hiện lỗi và xác minh các yêu cầu chức năng cũng như phi chức năng. Có thể là Manual Tester (kiểm thử thủ công) hoặc Automation Tester (kiểm thử tự động).
- **Mô tả công việc:**
 - Phân tích tài liệu yêu cầu để hiểu chức năng sản phẩm.
 - Lập kế hoạch kiểm thử, viết các trường hợp kiểm thử (test cases).
 - Thực hiện kiểm thử chức năng, kiểm thử phi chức năng (hiệu năng, bảo mật...).
 - Ghi nhận, theo dõi và báo cáo lỗi (bug) chi tiết cho nhóm phát triển.
 - Xác minh lỗi đã được sửa (retest).

- Phát triển và duy trì các script kiểm thử tự động (đối với Automation QA).
- Tham gia vào quy trình cải tiến chất lượng phần mềm.

3. Kỹ sư DevOps (DevOps Engineer)

- **Tổng quan:** Là cầu nối giữa phát triển (Dev) và vận hành (Ops), nhằm tự động hóa các quy trình từ xây dựng (build), kiểm thử (test), triển khai (deploy) đến giám sát (monitor) phần mềm, giúp tăng tốc độ và độ tin cậy của việc phát hành sản phẩm.
- **Mô tả công việc:**
 - Thiết lập và quản lý các công cụ CI/CD (Continuous Integration/Continuous Delivery) như Jenkins, GitLab CI/CD, Azure DevOps.
 - Xây dựng và duy trì cơ sở hạ tầng (Infrastructure as Code - IaC) sử dụng Terraform, Ansible, Chef.
 - Quản lý và triển khai ứng dụng trên các nền tảng đám mây (AWS, Azure, GCP) hoặc on-premise.
 - Sử dụng công nghệ containerization (Docker, Kubernetes) để đóng gói và quản lý ứng dụng.
 - Thiết lập hệ thống giám sát và cảnh báo (Prometheus, Grafana, ELK Stack).
 - Tối ưu hóa quy trình release và khắc phục sự cố hệ thống.
 - Đảm bảo tính bảo mật của hệ thống và dữ liệu.

4. Chuyên viên Phân tích Nghiệp vụ (Business Analyst - BA)

- **Tổng quan:** Đóng vai trò là cầu nối giữa các bên liên quan (khách hàng, đội ngũ kỹ thuật) để thu thập, phân tích và tài liệu hóa các yêu cầu nghiệp vụ, giúp chuyển đổi chúng thành các yêu cầu chức năng để phát triển phần mềm.
- **Mô tả công việc:**

- Gặp gỡ khách hàng/stakeholders để thu thập và làm rõ yêu cầu nghiệp vụ.
- Phân tích, đánh giá tính khả thi của các yêu cầu.
- Viết tài liệu đặc tả yêu cầu phần mềm (SRS, User Stories, Use Cases).
- Vẽ các sơ đồ luồng nghiệp vụ (workflow diagrams), mô hình dữ liệu.
- Làm việc với đội ngũ kỹ thuật để truyền đạt và làm rõ yêu cầu.
- Hỗ trợ trong giai đoạn kiểm thử chấp nhận người dùng (UAT).
- Đề xuất các giải pháp cải tiến quy trình nghiệp vụ.

5. Quản lý Dự án CNTT (IT Project Manager - PM)

- **Tổng quan:** Chịu trách nhiệm lập kế hoạch, thực thi và giám sát các dự án CNTT để đảm bảo chúng hoàn thành đúng thời hạn, ngân sách và phạm vi.
- **Mô tả công việc:**
 - Xác định phạm vi dự án, mục tiêu và các yếu tố có thể ảnh hưởng.
 - Lập kế hoạch dự án chi tiết (thời gian, nguồn lực, ngân sách).
 - Quản lý nhóm dự án, phân công nhiệm vụ và theo dõi tiến độ.
 - Quản lý rủi ro và giải quyết các vấn đề phát sinh.
 - Giao tiếp liên tục với các bên liên quan (khách hàng, đội ngũ).
 - Đảm bảo chất lượng sản phẩm bàn giao.
 - Báo cáo tiến độ và hiệu suất dự án.
 - Sử dụng các phương pháp quản lý dự án (Agile, Waterfall, Scrum...).

6. Kiến trúc sư Phần mềm (Software Architect)

- **Tổng quan:** Là người chịu trách nhiệm thiết kế cấu trúc tổng thể và các thành phần chính của hệ thống phần mềm, đảm bảo tính bền vững, khả năng mở rộng, hiệu suất và bảo mật.
- **Mô tả công việc:**
 - Phân tích yêu cầu nghiệp vụ và phi nghiệp vụ để xác định kiến trúc hệ thống phù hợp.
 - Thiết kế kiến trúc tổng thể (macro-level design) và các thành phần chính của hệ thống.
 - Lựa chọn công nghệ, framework, và các thành phần kỹ thuật.
 - Định hướng và hướng dẫn đội ngũ phát triển theo các nguyên tắc kiến trúc.
 - Đánh giá rủi ro kỹ thuật và đề xuất giải pháp.
 - Đảm bảo tuân thủ các tiêu chuẩn và quy tắc phát triển.
 - Nghiên cứu và đưa ra các giải pháp công nghệ mới.

7. Chuyên viên An toàn Thông tin (Information Security Analyst / Cyber Security Analyst)

- **Tổng quan:** Bảo vệ hệ thống máy tính và dữ liệu của tổ chức khỏi các mối đe dọa mạng, tấn công và truy cập trái phép.
- **Mô tả công việc:**
 - Giám sát hệ thống và mạng để phát hiện các hoạt động đáng ngờ.
 - Phân tích lỗ hổng bảo mật và đề xuất biện pháp khắc phục.
 - Phát triển và triển khai các chính sách và quy trình bảo mật.
 - Thực hiện kiểm tra thâm nhập (penetration testing) và đánh giá lỗ hổng.
 - Phản ứng và điều tra các sự cố an ninh mạng.
 - Đào tạo và nâng cao nhận thức về bảo mật cho nhân viên.
 - Đảm bảo tuân thủ các quy định về bảo mật dữ liệu.

8. Kỹ sư Dữ liệu (Data Engineer)

- **Tổng quan:** Thiết kế, xây dựng và quản lý các hệ thống cơ sở hạ tầng dữ liệu quy mô lớn, đảm bảo dữ liệu luôn sẵn sàng, sạch sẽ và có thể truy cập được cho các phân tích và ứng dụng.
- **Mô tả công việc:**
 - Thiết kế và xây dựng các pipeline ETL/ELT (Extract, Transform, Load) để thu thập, xử lý và di chuyển dữ liệu.
 - Quản lý và tối ưu hóa các kho dữ liệu (data warehouse) và hồ dữ liệu (data lake).
 - Làm việc với các công cụ Big Data (ví dụ: Apache Spark, Hadoop, Kafka).
 - Đảm bảo chất lượng dữ liệu (data quality) và tính nhất quán.
 - Hợp tác với Data Scientists và Business Analyst để đáp ứng nhu cầu dữ liệu.
 - Phát triển và duy trì các API dữ liệu.

9. Chuyên viên Khoa học Dữ liệu (Data Scientist)

- **Tổng quan:** Sử dụng các kỹ thuật thống kê, lập trình và kiến thức chuyên môn để phân tích các tập dữ liệu lớn, khám phá ra các mô hình, xu hướng và đưa ra những hiểu biết sâu sắc giúp hỗ trợ ra quyết định kinh doanh.
- **Mô tả công việc:**
 - Thu thập và làm sạch dữ liệu từ nhiều nguồn khác nhau.
 - Áp dụng các kỹ thuật thống kê và máy học để xây dựng mô hình dự đoán hoặc phân loại.
 - Phân tích dữ liệu để tìm ra insight và xu hướng.
 - Trực quan hóa dữ liệu và trình bày kết quả cho các bên liên quan.
 - Phát triển các thuật toán và mô hình học máy.
 - Hợp tác với Data Engineers để xây dựng pipeline dữ liệu và triển khai mô hình

10. Chuyên viên Hỗ trợ Kỹ thuật (Technical Support Specialist)

- **Tổng quan:** Cung cấp hỗ trợ kỹ thuật cho người dùng cuối hoặc khách hàng về các vấn đề liên quan đến phần mềm, phần cứng hoặc hệ thống.
- **Mô tả công việc:**
 - Tiếp nhận và giải quyết các yêu cầu hỗ trợ qua điện thoại, email hoặc trực tiếp.
 - Chẩn đoán và khắc phục sự cố phần cứng, phần mềm, mạng.
 - Cài đặt và cấu hình phần mềm, hệ điều hành.
 - Hướng dẫn người dùng sử dụng sản phẩm/hệ thống.
 - Ghi nhận và theo dõi các vấn đề được báo cáo.
 - Leo thang các vấn đề phức tạp lên cấp cao hơn.