**Bài tập**

***Quản lý dự án phần mềm***

**------o0o------**

**THÔNG TIN NHÓM SINH VIÊN THỰC HIỆN**

**- Nhóm: 11**

**- Thành viên nhóm:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** |
| 1 | Đặng Minh Nhật | 22110389 |
| 2 | Vi Quốc Thuận | 22110006 |
| 3 | Đỗ Phú Luân | 22110372 |
| 4 | Huỳnh Minh Mẫn | 22110377 |
| 5 | Nguyễn Tiến Dũng | 22110302 |

**Bài tập 03: Quản lý thời gian thực hiện dự án với phương pháp PDM (Precedence Diagramming Method)**

1. **Định nghĩa, công dụng**

* **Định nghĩa về phương pháp precedence diagramming method (PDM)**
* Phương pháp Precedence diagramming method (PDM) là một công cụ lập kế hoạch tiến độ dự án. Đây là một phương pháp xây dựng bản tiến độ dự án dựa theo phương pháp đường găng (CPM), mà các thành phần chính là các hộp thông tin công việc, được gọi là các nút công việc (activity node), để đại diện cho các công việc (hay hoạt động) và kết nối chúng với nhau bằng những mũi tên, thể hiện sự phụ thuộc giữa các công việc liền trước hay kế tiếp nhau giúp xác định thứ tự thực hiện công việc của dự án.
* **Công dụng của PDM**

Phương pháp Precedence diagramming method (PDM) có nhiều công dụng quan trọng trong việc lập kế hoạch tiến độ của dự án, cụ thể như sau:

* Xác định thứ tự cũng như trình tự các hoạt động trong lịch trình nhờ vậy chúng ta có thể phát triển chính xác hơn lịch trình của dự án cũng như tối ưu hóa được thời gian hoàn thành dự án.
* Cho thấy những hoạt động nào quan trọng, những hoạt động nào bị phụ thuộc hoặc phải theo sau những hoạt động khác, điều này giúp tập trung vào những nhiệm vụ chính, ngăn ngừa những trường hợp tắc nghẽn sau này trong dự án.
* Việc sắp xếp trình tự các hoạt động cũng có thể giúp xác định những khoảng trống, những hoạt động còn thiếu và những liên kết chưa hợp lý.
* Giúp hiển thị các quy trình và hoạt động quan trọng trong lịch trình.

1. **Hệ thống ký hiệu được sử dụng**

Các thành phần cấu tạo: [4]

Hộp hoạt động (Node): Đây là biểu tượng chứa tên hoạt động và thời gian thực hiện của nó. Mỗi hoạt động được biểu diễn bởi một hộp riêng biệt.

Mũi tên (Arrow): Mũi tên được sử dụng để biểu thị mối quan hệ giữa các hoạt động. Mũi tên xuất phát từ hoạt động tiền đề và kết thúc tại hoạt động kế cận. Chiều của mũi tên thể hiện thứ tự thực hiện hoạt động.

Thời gian thực hiện (Duration): Được hiển thị trên hộp hoạt động, thể hiện thời gian cần để hoàn thành hoạt động đó.

Loại mối quan hệ (Dependency Type): Mỗi mũi tên đi kèm với một loại mối quan hệ, như mối quan hệ bắt buộc (Mandatory), mối quan hệ tùy ý (Discretionary), mối quan hệ từ bên ngoài (External), hoặc mối quan hệ trong nội bộ (Internal. Loại mối quan hệ này xác định cách các hoạt động tương tác với nhau.

Ký hiệu mối quan hệ (Dependency Notation): Ký hiệu được sử dụng để chỉ rõ loại mối quan hệ giữa các hoạt động, chẳng hạn như mũi tên kép (FS, Finish-to-Start), mũi tên đơn (SS, Start-to-Start), và các biểu tượng tương tự.

Đường đi quan trọng (Critical Path): Đây là chuỗi các hoạt động có tổng thời gian trễ là 0. Đường đi quan trọng thể hiện thời gian tối thiểu để hoàn thành dự án.

Khoảng trễ (Float): Là thời gian mà một hoạt động có thể trễ mà không ảnh hưởng đến thời gian hoàn thành dự án.

Thời gian bắt đầu sớm (Early Start – ES) và thời gian kết thúc sớm (Early Finish – EF): Là thời điểm mà một hoạt động có thể bắt đầu và kết thúc dự kiến nếu không có trễ.

Thời gian bắt đầu muộn (Late Start – LS) và thời gian kết thúc muộn (Late Finish – LF): Là thời điểm mà một hoạt động phải bắt đầu và kết thúc để không ảnh hưởng đến thời gian hoàn thành dự án.

Có 4 loại phụ thuộc trong quan hệ logic. [5]

Finish-to-start (FS). Quan hệ này chỉ ra công việc sau không thể bắt đầu cho đến khi công việc trước đó hoàn thành. Ví dụ, việc cài đặt và vận hành hệ thống trên máy tính không thể bắt đầu nếu phần cứng chưa được lắp ráp

Finish-to-finish (FF). Quan hệ này chỉ ra công việc sau không thể kết thúc cho đến khi công việc trước kết thúc. Ví dụ, viết tài liệu (công việc trước) được yêu cầu kết thúc trước khi chỉnh tài liệu (công việc sau) kết thúc.

Start-to-start (SS). Quan hệ trong đó công việc sau không thể bắt đầu cho đến khi công việc trước bắt đầu. Ví dụ, cấp bê tông không thể bắt đầu cho đến khi công việc đổ móng bắt đầu.

Start-to-finish (SF). Quan hệ trong đó công việc sau không thể kết thúc cho đến khi công việc trước bắt đầu. Ví dụ, hệ thống thanh toán mới phải bắt đầu trước khi hệ thống thanh toán cũ kết thúc.

1. **Các nguyên tắc vẽ [6]**

*Bước 1: Tách thành Cấu trúc phân chia công việc (WBS) và Cấp độ hoạt động*

* Thu thập thông tin chi tiết về thời gian và ngày tháng của các hoạt động trong dự án.
* Phân tách dự án thành các phần nhỏ hơn, gọi là Cấu trúc phân chia công việc (WBS).
* Xác định các hoạt động cụ thể trong mỗi phần và sắp xếp chúng theo cấp độ.

*Bước 2: Đề xuất bảng liệt kê*

* Tạo một bảng liệt kê tất cả các hoạt động bạn đã xác định ở Bước 1.
* Ghi lại thứ tự dự kiến mà các hoạt động sẽ được thực hiện.

*Bước 3: Thêm Mối quan hệ và Phụ thuộc*

* Dựa vào bảng bạn đã tạo, thêm mối quan hệ giữa các hoạt động.
* Xác định xem hoạt động nào phải hoàn thành trước khi hoạt động khác có thể bắt đầu.

*Bước 4: Tạo Sơ đồ PDM*

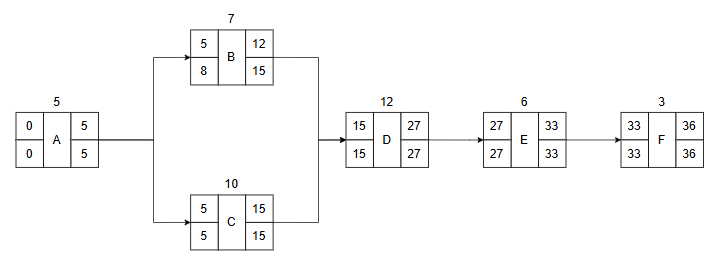
* Sử dụng hình dạng và biểu tượng thích hợp để biểu diễn mỗi hoạt động
* Sử dụng các mũi tên để thể hiện mối quan hệ giữa chúng. Sơ đồ này sẽ giúp bạn thấy rõ hơn cách các hoạt động tương tác và ảnh hưởng đến nhau.
* Tính toán các giá trị của mỗi hoạt động
* Xác định đường đi quan trọng (Critical Path)

1. **Một vài ví dụ cụ thể**

Ví dụ: Phát triển một ứng dụng di động  
Mô tả dự án: Một nhóm phát triển phần mềm đang xây dựng một ứng dụng di động với các tính năng cơ bản như đăng nhập, hiển thị danh sách sản phẩm và đặt hàng.

|  |  |  |  |
| --- | --- | --- | --- |
| Công việc | Mô tả | Thời gian (ngày) | Quan hệ phụ thuộc |
| A | Xác định yêu cầu | 5 | Bắt đầu đầu tiên |
| B | Thiết kế UI/UX | 7 | A (FS) |
| C | Xây dựng API backend | 10 | A (FS) |
| D | Phát triển frontend | 12 | B, C (FS) |
| E | Kiểm thử tích hợp | 6 | D (FS) |
| F | Triển khai ứng dụng | 3 | E (FS) |

Sơ đồ PDM:



1. **Cách tính thời gian hoàn thành dự án (sớm nhất, trễ nhất, khả năng rút ngắn) [7]**
2. **Cách xác định thời gian hoàn thành dự án sớm nhất (Early Start - ES và Early Finish - EF):**

* **Bước 1:** Bắt đầu từ hoạt động đầu tiên, gán thời gian bắt đầu sớm nhất (ES) là 0 hoặc ngày bắt đầu dự án.
* **Bước 2:** Tính thời gian kết thúc sớm nhất (EF) cho mỗi hoạt động:

EF = ES + Thời gian thực hiện

* **Bước 3:** Đối với các hoạt động kế tiếp:
  + Nếu một hoạt động chỉ có một tiền nhiệm, thời gian bắt đầu sớm nhất (ES) bằng thời gian kết thúc sớm nhất (EF) của hoạt động trước đó.
  + Nếu một hoạt động có nhiều tiền nhiệm, ES là giá trị EF lớn nhất trong các tiền nhiệm.

1. **Cách xác định thời gian hoàn thành dự án muộn nhất (Late Start - LS và Late Finish - LF):**

* **Bước 1:** Bắt đầu từ hoạt động cuối cùng của dự án và tính toán ngược về hoạt động khởi đầu, gán thời gian kết thúc muộn nhất (LF) bằng thời gian kết thúc sớm nhất (EF) của nó (hoặc ngày kết thúc dự án nếu đã xác định).

LF = EF

* **Bước 2:** Tính thời gian bắt đầu muộn nhất (LS) cho mỗi hoạt động:

LS = LF − Thời gian thực hiện

* **Bước 3:** Đối với các hoạt động tiền nhiệm
  + Nếu một hoạt động chỉ có một kế nhiệm, thời gian kết thúc muộn nhất (LF) bằng thời gian bắt đầu muộn nhất (LS) của hoạt động kế tiếp.
  + Nếu một hoạt động có nhiều kế nhiệm, LF là giá trị LS nhỏ nhất trong các kế nhiệm.

1. **Xác định Đường găng (Critical Path):**

* Đường găng là chuỗi các hoạt động mà nếu bị trì hoãn sẽ ảnh hưởng trực tiếp đến thời gian hoàn thành dự án.
* Các hoạt động trên đường găng có độ chênh lệch tổng (Total Float) bằng 0, tức là:

Total Float = LS − ES = LF − EF = 0

***Nguyên tắc rút ngắn: rút ngắn đường găng [8]***

**6.** **Ưu điểm**

* Linh hoạt trong biểu diễn mối quan hệ: PDM cho phép biểu diễn 4 loại mối quan hệ (FS, SS, FF, SF), mang lại tính linh hoạt cao hơn so với các phương pháp khác như AOA (Activity-On-Arrow).
* Dễ hiểu và dễ sử dụng: Cấu trúc sơ đồ trực quan, dễ hiểu ngay cả với người không chuyên.
* Dễ cập nhật và điều chỉnh: Việc thêm/bớt công việc không đòi hỏi phải vẽ lại toàn bộ sơ đồ.
* Biểu diễn chi tiết: Có thể biểu diễn nhiều thông tin trên mỗi nút (ES, EF, LS, LF, Float...).
* Xác định đường găng chính xác: Giúp xác định rõ ràng các công việc tới hạn và tập trung nguồn lực.
* Phổ biến trong phần mềm quản lý dự án: Được hỗ trợ bởi hầu hết các phần mềm quản lý dự án hiện đại (MS Project, Primavera...).

**7. Khuyết điểm**

* Phức tạp khi số lượng công việc lớn: Với dự án có nhiều công việc, sơ đồ có thể trở nên phức tạp và khó theo dõi.
* Khó khăn trong tính toán thủ công: Việc tính toán ES, EF, LS, LF cho mạng lưới phức tạp đòi hỏi nhiều công sức khi thực hiện thủ công.
* Hạn chế trong biểu diễn thời gian thực: Không biểu diễn được thời gian thực của dự án trên sơ đồ (cần có biểu đồ Gantt đi kèm).
* Không trực quan về thời gian: Mũi tên chỉ biểu thị mối quan hệ, không biểu thị thời gian, nên khó hình dung được khoảng thời gian giữa các công việc.
* Đòi hỏi kiến thức chuyên môn: Để sử dụng hiệu quả, người lập kế hoạch cần có kiến thức về quản lý dự án và phương pháp PDM.
* Khó tối ưu hóa nguồn lực: PDM tập trung vào mối quan hệ thời gian, không trực tiếp giải quyết vấn đề phân bổ nguồn lực.

**8. Tài liệu sử dụng:**

[1] Bài giảng chương 3 (Trương Mỹ Dung)

[2] Một số tài liệu đính kèm (lấy từ LMS).

[3] Các nguồn khác trên Internet.

[4] <https://pma.edu.vn/blogs/precedence-diagram-method-la-gi/>

[5] <https://fmit.vn/tu-dien-quan-ly/lap-so-do-trinh-tu-la-gi>

[6] <https://pma.edu.vn/blogs/precedence-diagram-method-la-gi/>

[7]<https://www.linkedin.com/pulse/precedence-diagram-method-pdm-l%C3%A0-g%C3%AC-pmavietnam/>

[8] <https://youtu.be/qNSTP88FHWA>