

# Quản lý dự án phần mềm

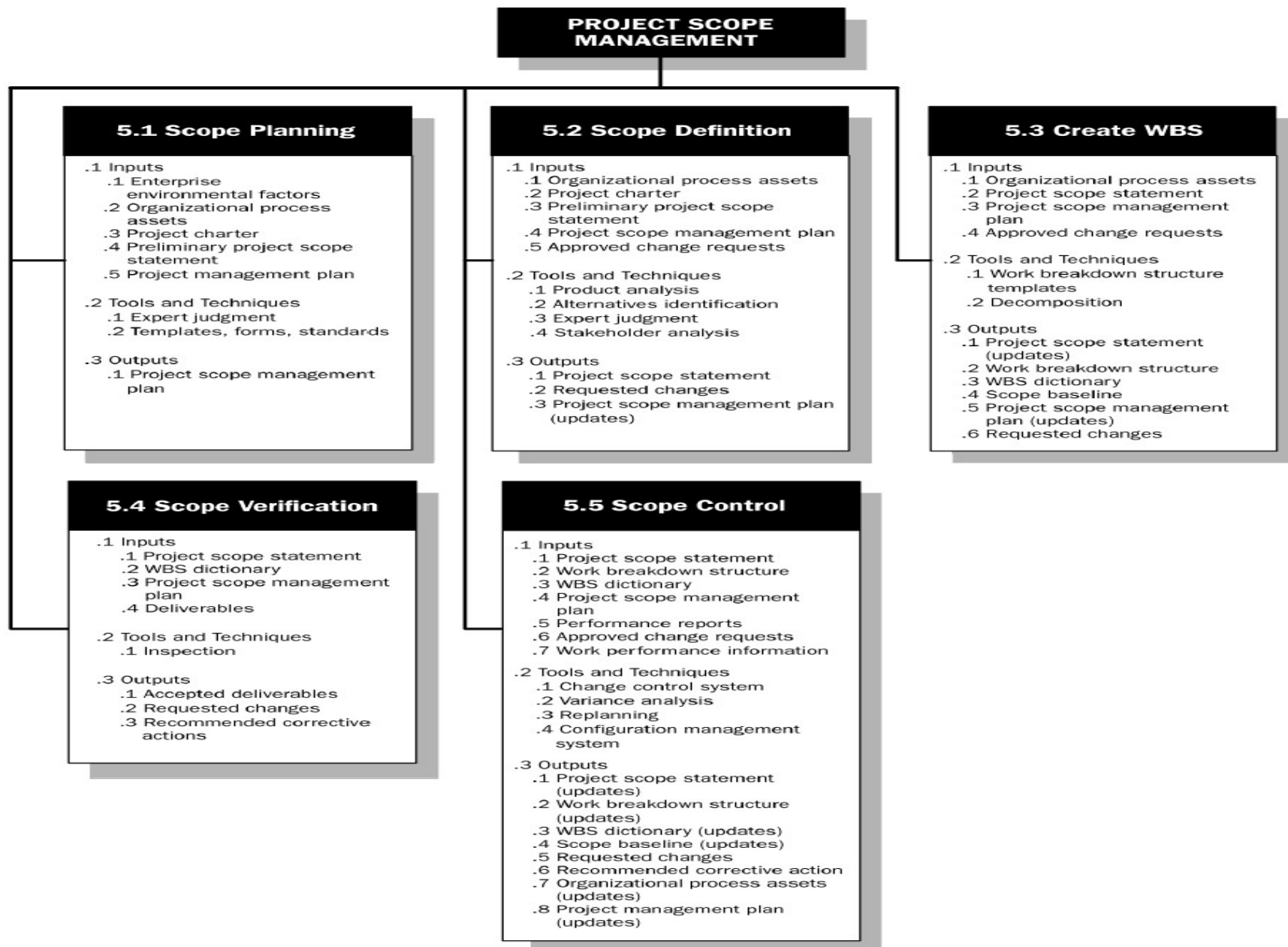
---

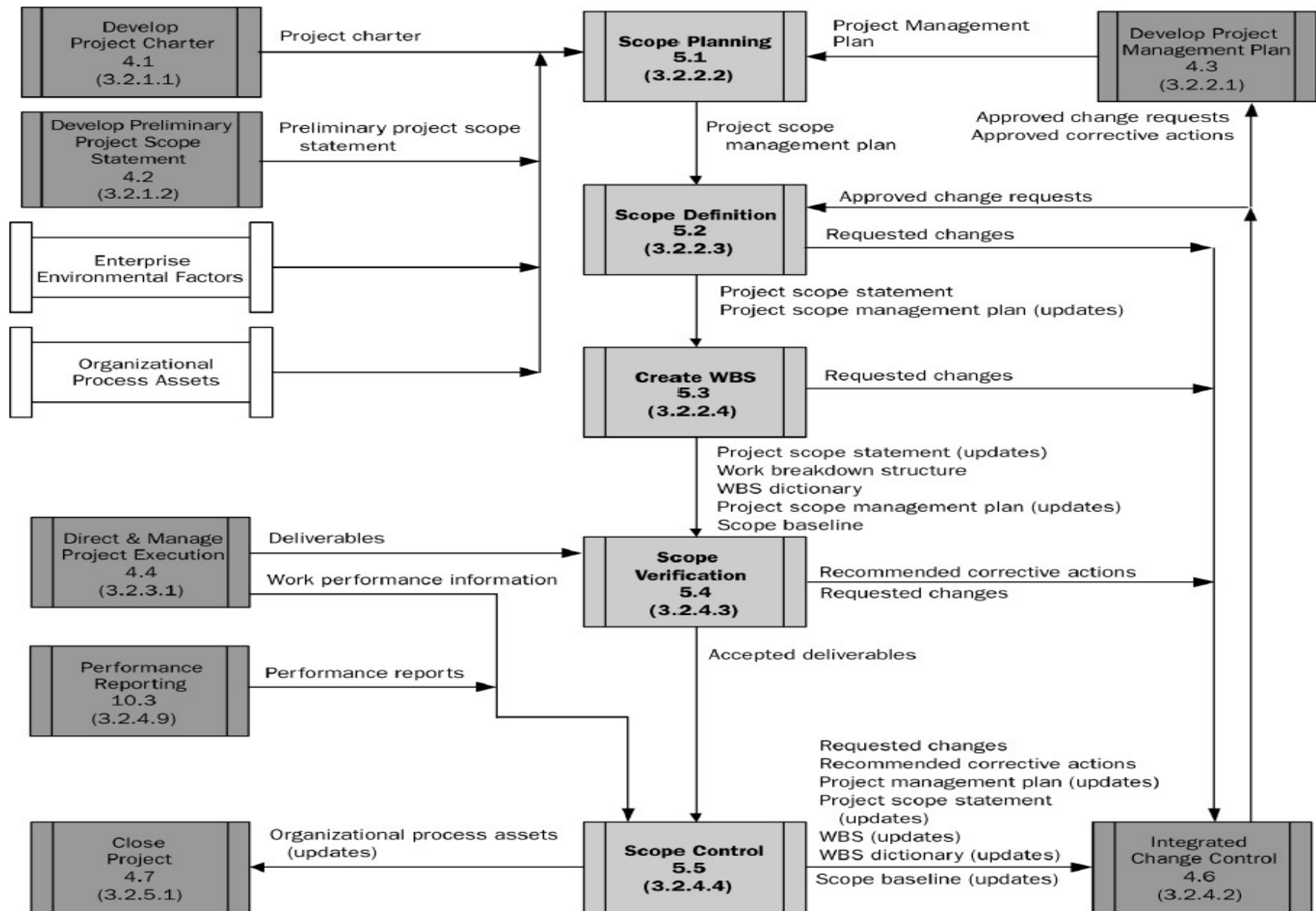
## Bài 4: Cấu trúc phân rã công việc (WBS), Ước lượng và Lập lịch

# Nội dung bài học

---

- Tóm tắt về quản lý phạm vi
- Cấu trúc phân rã công việc Work Breakdown Structures (WBS)
- Ước lượng
- Cơ bản về mạng
- Các kỹ thuật PERT & CPM
- Biểu đồ Gantt





Note: Not all process interactions and data flow among the processes are shown.

# Lập kế hoạch, ước lượng và lập lịch

---

- Sự khác nhau là gì?
- Lập kế hoạch: xác định các hoạt động. Không có ngày bắt đầu và kết thúc cụ thể.
- Ước lượng: xác định kích cỡ và khoảng thời gian cần thực hiện các hoạt động đó.
- Lập lịch: gắn thêm ngày bắt đầu và kết thúc cụ thể, các mối quan hệ và các tài nguyên.

# Lập kế hoạch dự án: một chương trình 12 bước

---

- 1) thiết lập đích và phạm vi
- 2) lựa chọn chu trình sống
- 3) thiết lập kiểu nhóm /tổ chức
- 4) Lựa chọn đội làm việc
- 5) Xác định các rủi ro
- 6) Tạo WBS
- 7) Xác định các nhiệm vụ
- 8) Ước lượng kích cỡ
- 9) Ước lượng công
- 10) Xác định các sự phụ thuộc công việc
- 11) Gán tài nguyên
- 12) lập lịch thực hiện công việc

# Cách lập lịch

---

- 1. Xác định công việc cần được thực hiện
  - Cấu trúc phân rã công việc
- 2. Xác định kích cỡ
  - các kỹ thuật ước lượng kích cỡ
- 3. Xác định sự phụ thuộc giữa các công việc
  - Đồ thị phụ thuộc, biểu đồ mạng
- 4. Ước lượng tổng thời gian các công việc được thực hiện
  - Lịch thực tế

# WBS và việc ước lượng

---

- Bạn cảm thấy sao khi tôi hỏi
  - “Dự án của bạn sẽ thực hiện trong bao lâu?”
- Không dễ dàng có câu trả lời đúng
- Ít nhất không đúng trong trường hợp bạn là khách hàng thật của một dự án thực
- Bạn có thể giải quyết vấn đề này thế nào?



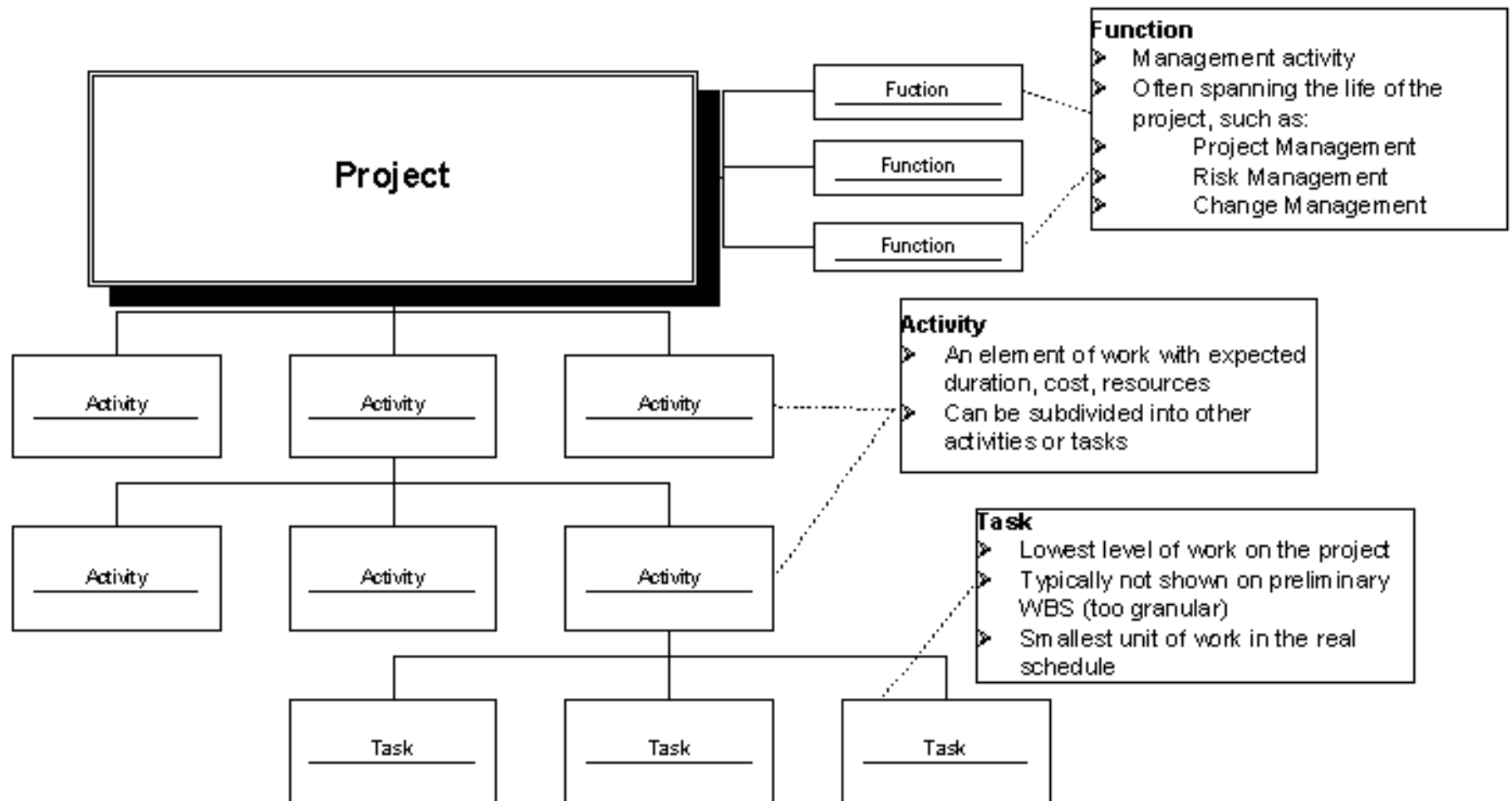
# Phân nhỏ dự án

---

- Bạn cần phân chia dự án thành nhiều phần có thể quản lý được
- **Tất cả** các dự án đều cần bước này
- Chia để trị
- Hai nguyên nhân chính dẫn đến dự án thất bại
  - Quên một số thứ thiết yếu
  - Những con số ước lượng trở thành mục tiêu
- Việc phân nhỏ dự án sẽ giúp gì cho việc này?

# Các nhân tố của dự án

- Một dự án: các chức năng, hoạt động, công việc



# Cấu trúc phân rã công việc: WBS

---

- Danh sách phân rã các hoạt động của dự án
- Hai định dạng
  - Các đầu mục (Outline)
  - Cây đồ họa (Biểu đồ tổ chức)
- Sử dụng một hệ thống đánh số thập phân
  - Ví dụ: 3.1.5
  - 0 thường biểu diễn mức cao nhất
- Bao gồm
  - Các nhiệm vụ phát triển, quản lý và hỗ trợ dự án
- Thể hiện các mối quan hệ "được chứa trong"
- Không thể hiện sự phụ thuộc hoặc khoảng thời gian cần thực hiện

# Cấu trúc phân rã công việc

---

- Định nghĩa:

Một nhóm phân cấp theo định hướng các sản phẩm phân phối của các nhân tố dự án tổ chức và định nghĩa toàn bộ phạm vi của dự án. Mỗi mức bên dưới thể hiện một định nghĩa dự án ở mức chi tiết hơn.

# WBS

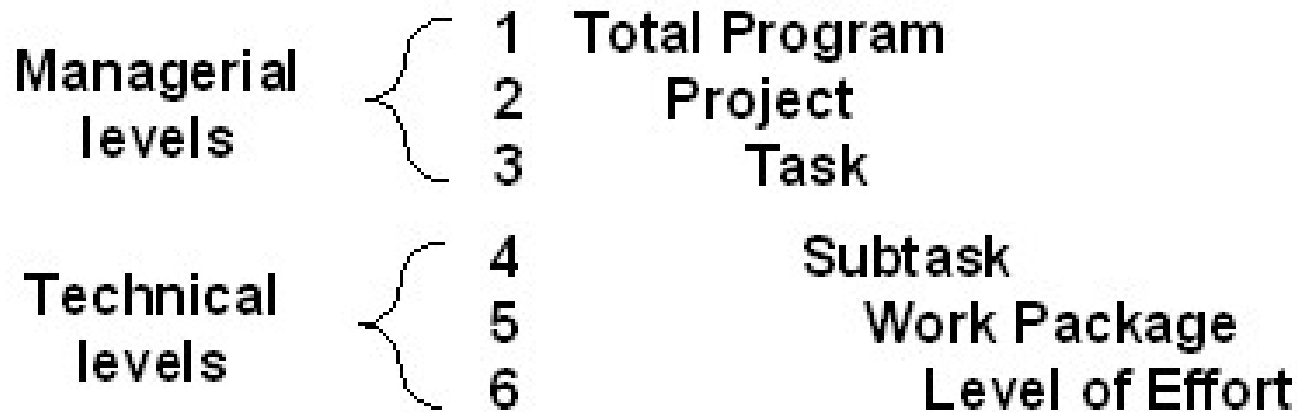
---

- WBS dạng hợp đồng (CWBS)
  - Hai hoặc ba mức đầu
  - Cần theo dõi các công việc ở mức cao
- WBS dạng dự án (PWBS)
  - Được định nghĩa bởi giám đốc dự án và các thành viên của đội
  - Các công việc được gắn với các sản phẩm phân phối
  - Cần theo dõi các công việc ở mức thấp nhất

# Một cấu trúc WBS đầy đủ

---

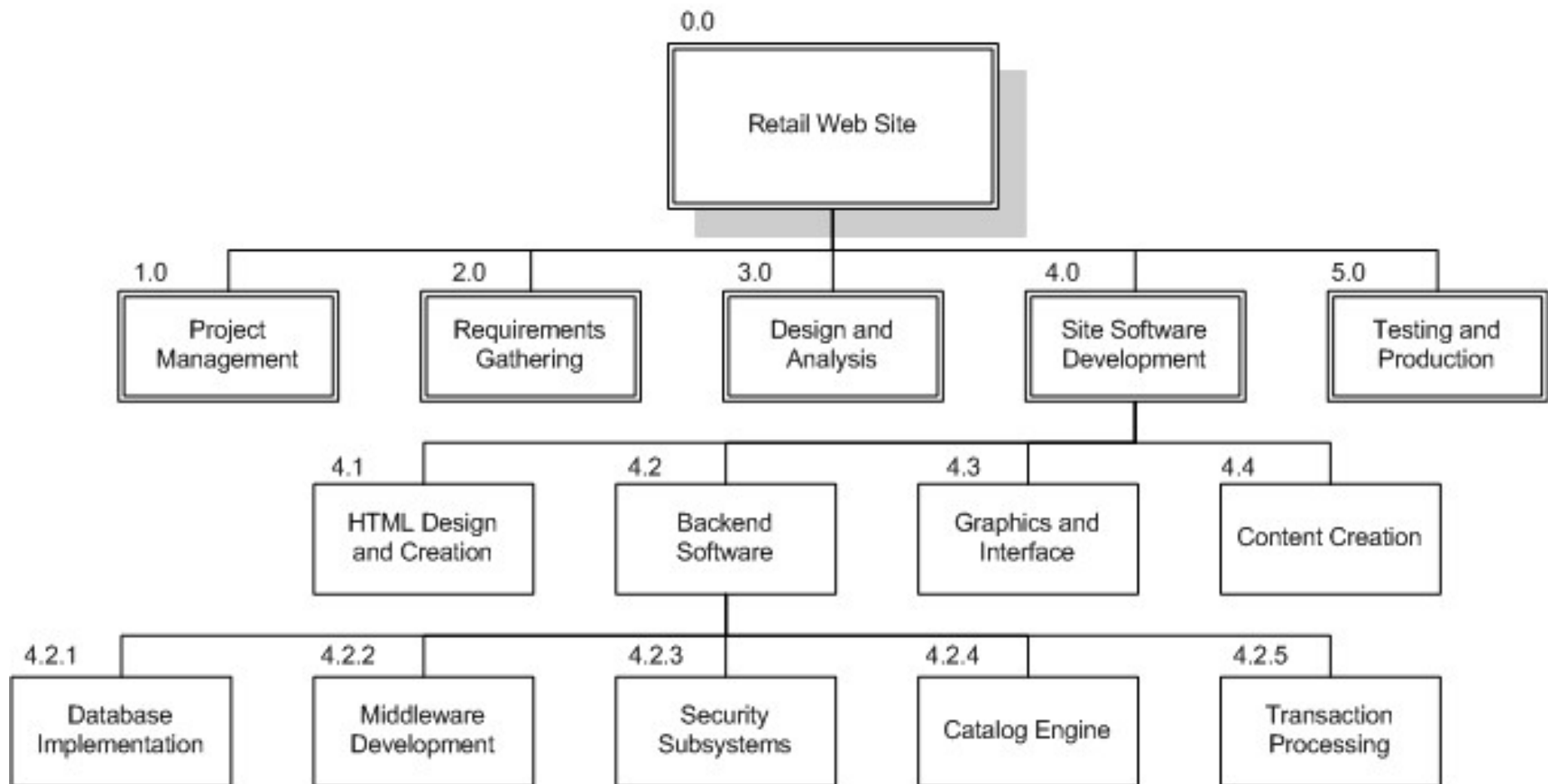
- Lên tới 6 mức (thường là 3-6) ví dụ như



- 3 mức trên có thể được sử dụng bởi khách hàng cho việc báo cáo
- Mức khác nhau có thể được áp dụng cho những người sử dụng khác nhau
  - Ví dụ: Mức 1 dành cho người trao quyền, mức 2 cho người làm ngân sách, mức 3 cho lập lịch

# Ví dụ về sơ đồ WBS

---



# Ví dụ về WBS dạng đầu mục

---

0.0 Retail Web Site

1.0 Project Management

2.0 Requirements Gathering

3.0 Analysis & Design

4.0 Site Software Development

4.1 HTML Design and Creation

4.2 Backend Software

4.2.1 Database Implementation

4.2.2 Middleware Development

4.2.3 Security Subsystems

4.2.4 Catalog Engine

4.2.5 Transaction Processing

4.3 Graphics and Interface

4.4 Content Creation

5.0 Testing and Production



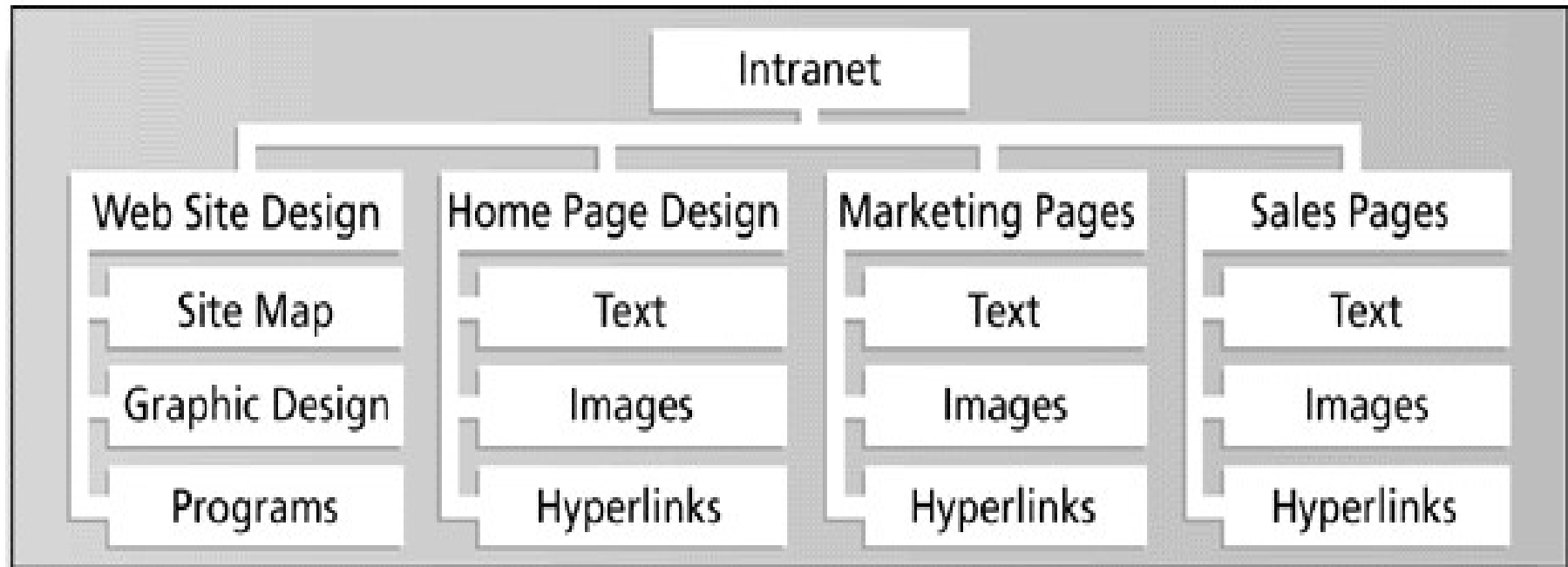
# Các loại WBS

---

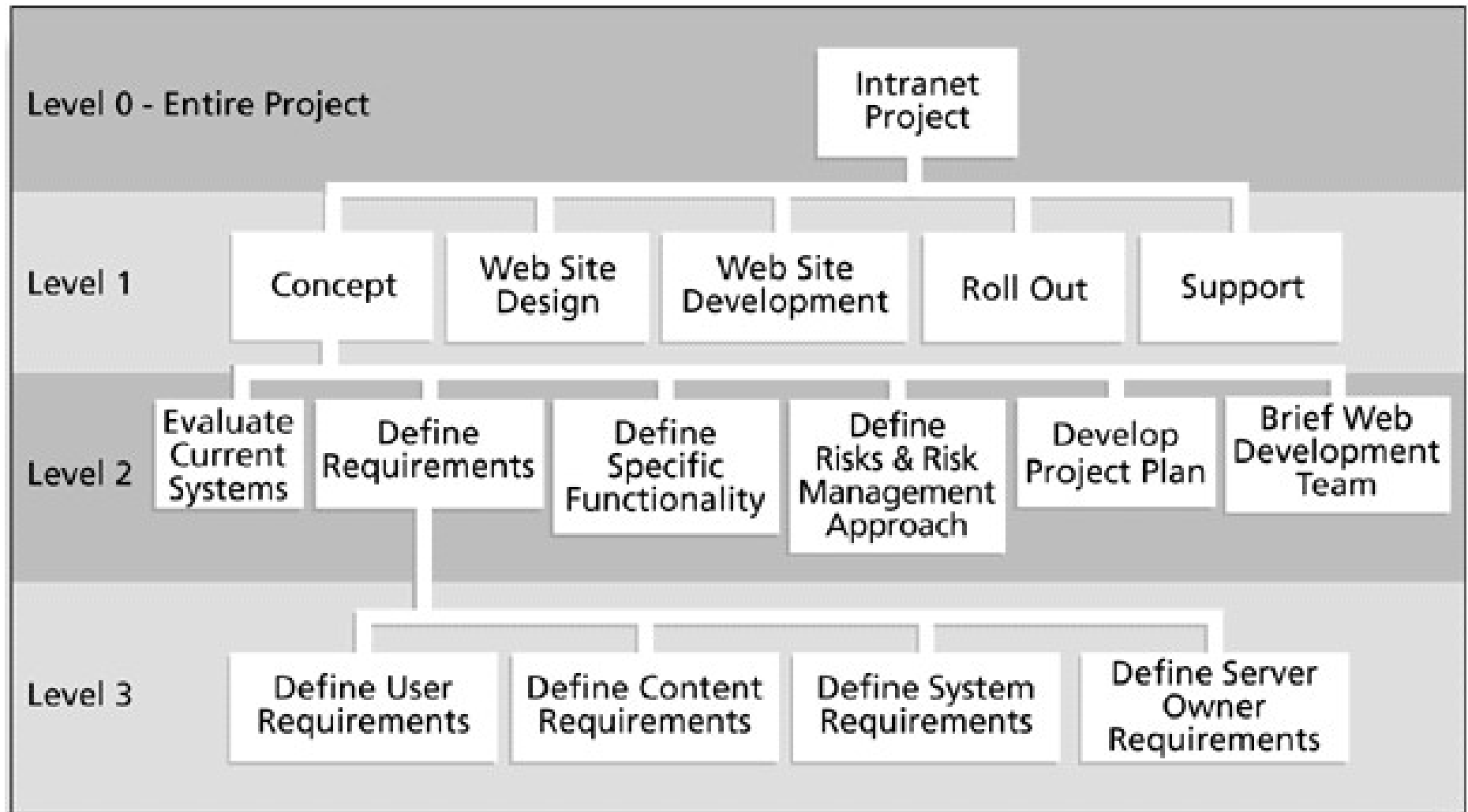
- WBS theo tiến trình
  - hay định hướng hoạt động
  - Ví dụ: Yêu cầu, Phân tích, Thiết kế, Kiểm thử
  - Diễn hình được sử dụng bởi giám đốc dự án
- WBS theo sản phẩm
  - hay định hướng thực thể
  - Ví dụ: bộ máy tài chính, hệ thống giao diện, cơ sở dữ liệu
  - Thường được dùng cho người quản lý kỹ thuật
- WBS kết hợp giữa hai loại trên:
  - Thường ít được sử dụng
  - Ví dụ: các pha chu trình sống ở mức cao với các thành phần hoặc đặc tính cụ thể bên trong mỗi pha
  - thường thể hiện các tiến trình sinh các sản phẩm

# Ví dụ về WBS theo sản phẩm

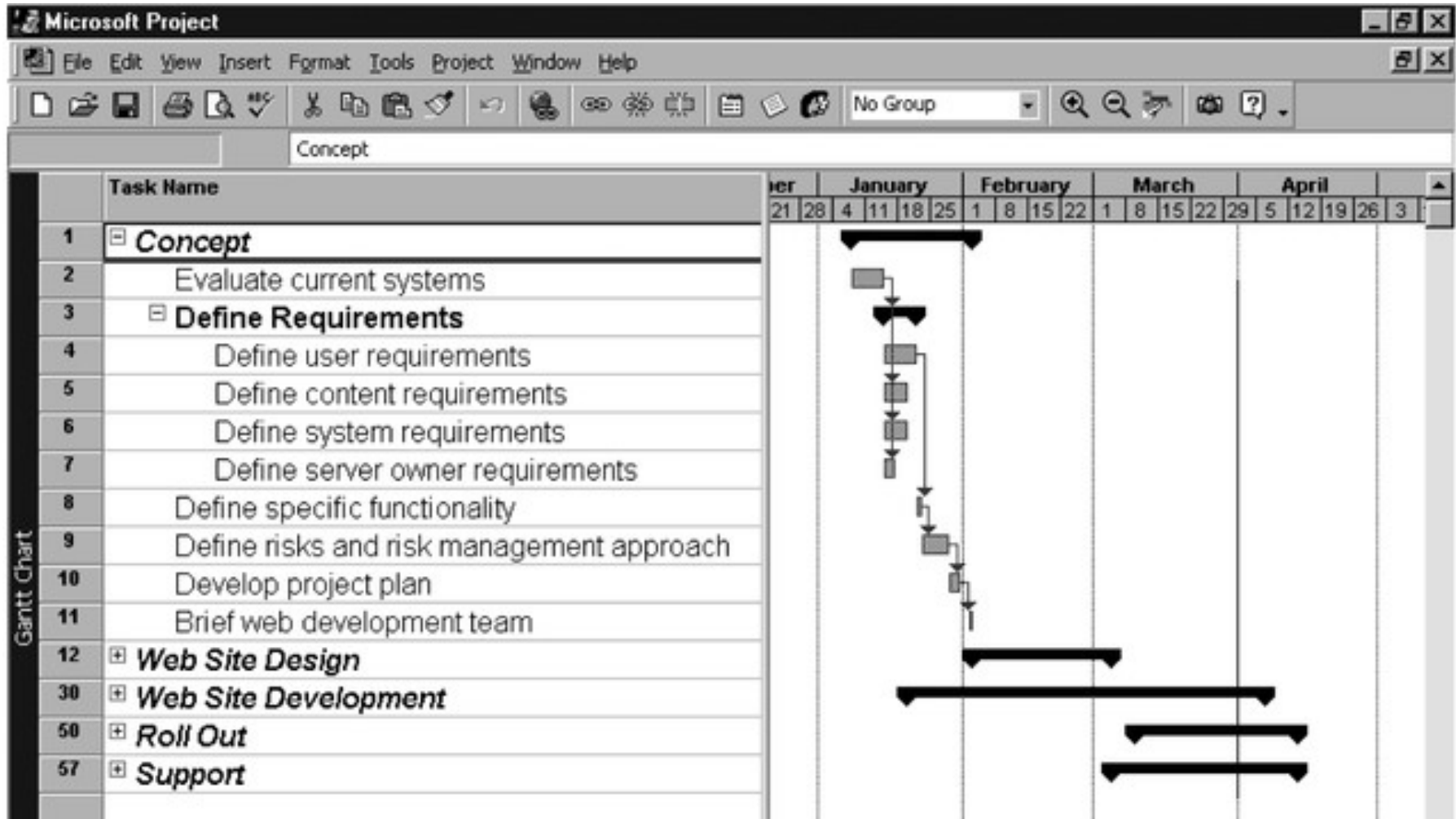
---



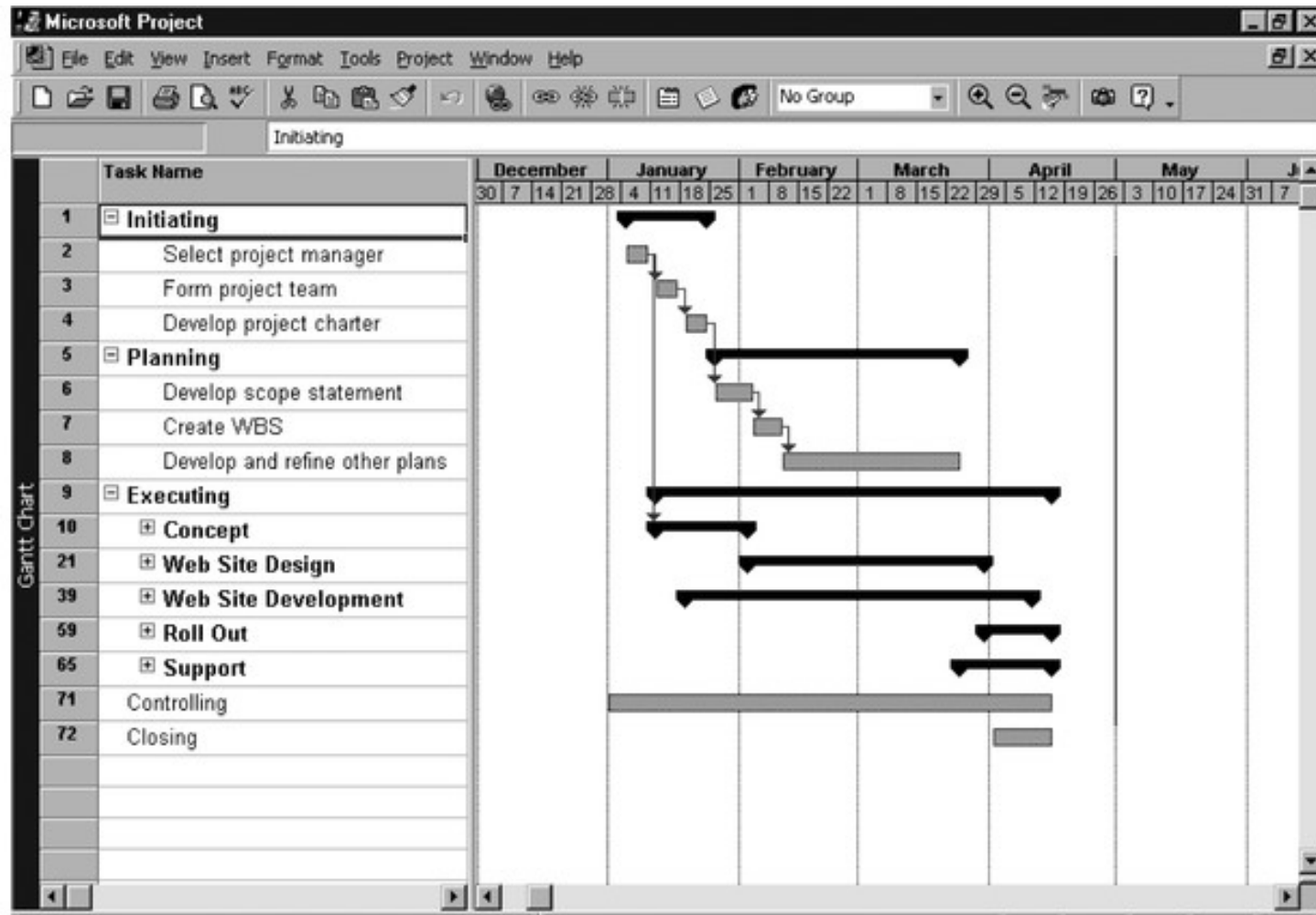
# Ví dụ về WBS theo tiến trình



# Ví dụ về WBS theo đầu mục với biểu đồ Gantt



# WBS theo các nhóm tiến trình của tổ chức PMI



# Các loại WBS khác

---

- Các loại thay thế ít được sử dụng
  - WBS theo tổ chức
    - Nghiên cứu, Thiết kế sản phẩm, Kỹ thuật, Thao tác
    - Có thể có ích cho các dự án mà độ đan chéo cao giữa các chức năng
  - WBS theo địa lý
    - Có thể có ích với các đội dự án phân tán về địa lý
    - Đội NYC, San Jose, Off-shore

# Các gói công việc

---

- Khái niệm chung cho các nhiệm vụ riêng biệt với các kết quả cuối có thể định nghĩa được
- Thường là các "lá" trên cây
- Luật "1 tới 2"
  - Thường tại: 1 hoặc 2 người làm trong 1 đến 2 tuần
- Tiền đề cho tiến trình theo dõi và báo cáo
  - Có thể gắn với các mục ngân sách (những con số chi tiêu)
  - Các tài nguyên con người được gán
- Lý tưởng hoá, ngắn hơn là dài
  - Dài thì cần thêm các ước lượng tiến độ
  - Mang tính chủ quan hơn là tính "hoàn thành"
  - Cho các dự án phần mềm, 2-3 tuần là nhiều nhất
  - 1 ngày là nhỏ nhất (đôi khi là nửa ngày)
  - Không quá nhỏ tới quản lý vi mô

# WBS

---

- Danh sách các công việc, không phải các vấn đề
- Danh sách các việc có thể lấy từ nhiều nguồn:
  - Phát biểu bảo toán SOW, bản đề xuất, buổi lấy ý kiến, người tham gia, đội dự án
- Mô tả các hoạt động sử dụng "ngôn ngữ nút"
  - có ý nghĩa nhưng các nhãn (terse labels)
- Tất cả các đường dẫn trong WBS không phải đi tới cùng một mức
- Không nên mô tả chi tiết hơn mức độ có thể quản lý được



# WBS và Phương pháp luận

---

- PM phải ánh xạ các hoạt động tới chu trình vòng đời dự án đã được chọn
- Mỗi loại chu trình có tập các hoạt động khác nhau
- Các hoạt động của tiến trình tích hợp xuất hiện cho toàn bộ quá trình
  - Lập kế hoạch, quản lý cấu hình, kiểm thử
- Các pha thực hiện và bảo trì thường không có trong kế hoạch (được coi là sau dự án)
- Vài mô hình được dùng đương nhiên cho WBS
  - Mô hình hình xoắn ốc, và các loại lặp lại khác
  - Chuỗi tuyến tính vài lần
- Các sản phẩm phân phối của các nhiệm vụ thay đổi tùy theo phương pháp luận

# Các kỹ thuật tạo WBS

---

- Trên xuống
- Dưới lên
- Tương tự
- Cuộn sóng
  - Lượt thứ nhất: đi sâu xuống mức 1-3
  - Thu thập thêm yêu cầu hoặc dữ liệu
  - Đưa thêm vào nhiều chi tiết hơn sau đó
- Xuất chúng lên tường

# Các kỹ thuật tạo WBS

---

- Trên xuống
  - Bắt đầu ở mức cao nhất
  - Phát triển chi tiết ở mức cao hơn một cách hệ thống
  - Tốt nhất nếu
    - Bài toán được hiểu thấu đáo
    - Kỹ thuật và phương pháp luận không mới
    - Giống với một dự án hay một bài toán trước đó
  - Nhưng được áp dụng trong hầu hết các trường hợp

# Các kỹ thuật tạo WBS

---

- Dưới lên
  - Bắt đầu từ mức thấp nhất
  - Kết hợp các công việc thành mức cao hơn và tổng quát hơn
  - Nhược điểm
    - Tốn thời gian
    - Cần tìm hiểu nhiều yêu cầu để hoàn thành
  - Ưu điểm
    - Rất chi tiết

# Các kỹ thuật tạo WBS

---

- Tương tự
  - Dựa trên WBS của một dự án tương tự
  - Sử dụng một mẫu sẵn có
  - Tương tự cũng có thể là tiền đề cho việc ước lượng
  - Ưu điểm
    - Dựa trên kinh nghiệm thực tế trong quá khứ
  - Nhược điểm
    - Cần có dự án tương tự

# Các kỹ thuật tạo WBS

---

- Tổng hợp ý tưởng
  - Tạo ra tất cả các hoạt động cần thực hiện cho dự án mà bạn có thể nghĩ đến
  - Nhóm chúng vào thành các loại khác nhau
- Cả hai loại trên xuống và tổng hợp ý tưởng có thể được sử dụng trên cùng một WBS
- Nhớ rằng nên kéo tất cả những người đang thực hiện công việc liên quan vào

# WBS – Nền tảng cho nhiều quá trình

---

- Lập lịch theo dạng mạng
- Tính chi phí
- Phân tích rủi ro
- Cấu trúc tổ chức
- Kiểm soát
- Đo đạc tính toán

# Hướng dẫn tạo WBS

---

- Nên dễ hiểu
- Một vài công ty có chuẩn cho sơ đồ này
- Một vài mục ở mức cao nhất như quản lý dự án nên có trong WBS của mỗi dự án
  - Các mục khác thay đổi theo từng dự án
- Vấn đề gây ảnh hưởng xấu nhiều nhất là cái đang thiếu
- Phân rã các mục đến khi bạn có thể đưa ra được sự ước lượng chính xác về thời gian và chi phí
- Đảm bảo mỗi yếu tố liên quan tới một sản phẩm phân phối



# Hướng dẫn tạo WBS

---

- Chi tiết nên đến mức nào?
  - **Không nên quá chi tiết**
  - Mỗi mức không nên có quá 7 mục
  - Nó có thể tăng trưởng qua thời gian
- Công cụ nào nên được sử dụng?
  - Excel, Word, Project
  - Công cụ vẽ sơ đồ tổ chức (Visio, Visual Paradigm)
  - Các ứng dụng thương mại đặc thù
- Dùng lại một mẫu nếu bạn có

# Ước lượng

---

- Rất khó thực hiện nhưng thường rất cần
- Được tạo ra, sử dụng và chuẩn hoá trong quá trình
  - Lập kế hoạch chiến lược
  - Nghiên cứu tính khả thi và/hoặc phát biểu bài toán
  - Đưa bản đề xuất của dự án
  - Đánh giá đơn vị trung gian và hợp đồng cấp 2
  - Lập kế hoạch dự án (lập lại)
- Quá trình cơ bản
  - 1) Ước lượng kích cỡ của sản phẩm
  - 2) Ước lượng công sức thực hiện (theo man-months)
  - 3) Ước lượng lịch thực hiện
  - Chú ý: Không phải tất cả các bước này đều luôn được thực hiện một cách tường minh

# Ước lượng

---

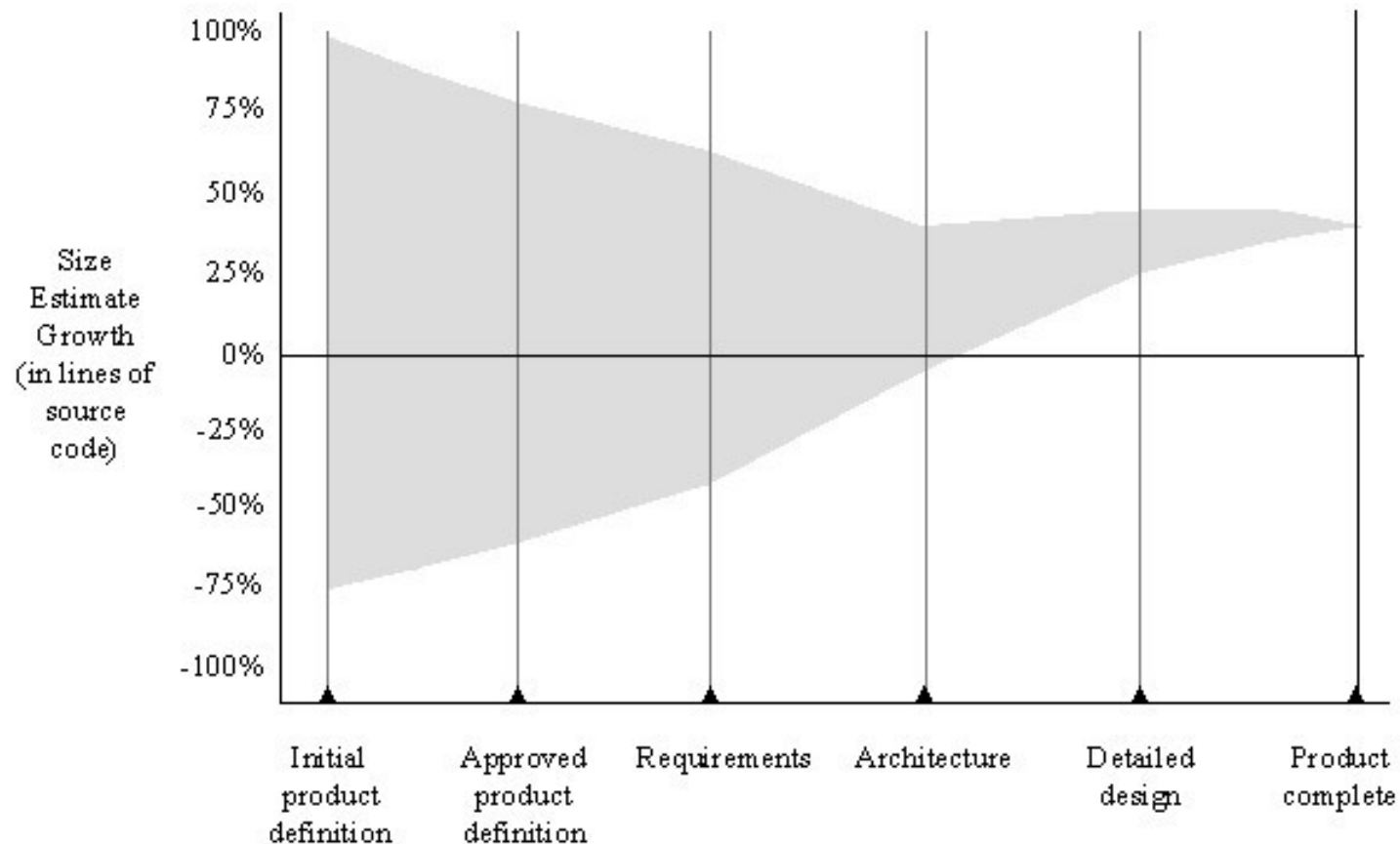
- Nhớ rằng một ước lượng chính xác là một điều không tưởng
- Thử ước lượng bạn sẽ tốn bao nhiêu thời gian để từ trường về đến nhà hôm nay
  - Dựa trên những tiền đề nào bạn làm được như vậy?
  - Kinh nghiệm?
  - Giống như một xác suất "trung bình"
  - Đối với hầu hết các dự án phần mềm, không có khái niệm "trung bình" như vậy
- Hầu hết các ước lượng đều sai khoảng 25-100%

# Ước lượng

---

- Các ngày mục tiêu (Target) và ngày cam kết (Committed)
  - Mục tiêu: được đề xuất bởi công việc kinh doanh và quản cáo với khách hàng
  - Đừng nên cam kết ngày mục tiêu đó quá nhanh!
  - Cam kết: Đội dự án đồng ý với điều đó
  - Sau khi bạn phát triển xong một lịch thực hiện

# Biểu đồ hình nón của sự không chắc chắn



Copyright 1998 Steven C. McConnell. Reprinted with permission from *Software Project Survival Guide* (Microsoft Press, 1998).

# Ước lượng

---

- Kích cỡ:
  - Dự án nhỏ (10-99 FPs), thay đổi khoảng 7% so với ước lượng sau khi tìm hiểu yêu cầu
  - Trung bình (100-999 FPs), thay đổi 22%
  - Lớn (1000-9999 FPs) thay đổi 38%
  - Rất lớn (> 10K FPs) thay đổi 51%

# Các phương pháp luận ước lượng

---

- Trên xuống
- Dưới lên
- Tương tự
- Phán đoán của chuyên gia
- Phương pháp thuật toán hoặc tham số
  - Sử dụng công thức và các biểu thức

# Ước lượng trên-xuống

---

- Dựa trên đặc tính toàn bộ của dự án
  - Một vài phương pháp khác có thể là kiểu trên-xuống (tương tự, phán đoán chuyên gia và các phương pháp dùng thuật toán)
- Ưu điểm
  - Dễ tính
  - Có hiệu quả sớm (như ước lượng chi phí ban đầu)
- Nhược điểm
  - Một vài mô hình còn nghi vấn hoặc có thể không phù hợp
  - Kém chính xác vì nó không tập trung vào chi tiết



# Ước lượng dưới lên

---

- Tạo WBS
- Thêm vào từ dưới lên
- Ưu điểm
  - Tốt nếu các hoạt động được hiểu rõ
- Nhược điểm
  - Các hoạt động cụ thể không phải luôn luôn được hiểu rõ
  - Tốn thời gian hơn

# Phán đoán của chuyên gia

---

- Sử dụng một ai đó có kinh nghiệm đối với một dự án tương tự gần đây
- Độ chính xác phụ thuộc vào độ chuyên gia thực sự của họ
- Các ứng dụng tương ứng phải được lựa chọn chính xác
  - Tính hệ thống
- Có thể sử dụng một hệ thống đánh trọng số

# Các đơn vị đo lường thuật toán

---

- Số dòng lệnh (LOC)
- Các điểm chức năng
- Điểm đặc tính hoặc điểm đối tượng
- Các đơn vị khác có thể là
  - Số lượng bọt nổi lên trong một cây tìm kiếm theo chiều sâu
  - Số lượng thực thể của mô hình thực thể liên kết
  - Số lượng các tiến trình trên một biểu đồ cấu trúc
- LOC và các điểm chức năng là thường được dùng
  - (theo các cách tiếp cận thuật toán)
- Phần lớn các dự án không sử dụng phương pháp nào ở trên cả

# Ước lượng dựa trên mã nguồn

---

- Ưu điểm của LOC
  - Đơn vị đo lường nhìn chung dễ hiểu
  - Cho phép so sánh cụ thể
  - Dễ đo đạc trong thực tế
- Nhược điểm của LOC
  - Khó ước lượng sớm trong chu trình phát triển
  - Số lượng thay đổi tùy theo ngôn ngữ
  - Nhiều chi phí chưa được quan tâm đến (ví dụ các nhiệm vụ xác định yêu cầu)
  - Các lập trình viên có lẽ được thưởng dựa trên điều đó
    - Có thể sử dụng :  $\# \text{ defects} / \# \text{ LOC}$
  - Các phần mềm sinh mã nguồn thường sản sinh mã thừa

# Các vấn đề ước lượng LOC

---

- Làm thế nào bạn biết được có bao nhiêu ngay từ đầu?
- Còn về các ngôn ngữ khác nhau?
- Về các kiểu lập trình khác nhau?
- Con số thống kê: năng suất trung bình của một lập trình viên 3,000 LOC/năm
- Hầu hết các phương thức tiếp cận theo thuật toán có hiệu quả hơn sau khi quá trình tìm hiểu yêu cầu (thường phải sau yêu cầu)

# Điểm chức năng

---

- Kích cỡ phần mềm được đo lường bởi số lượng và độ phức tạp của các hàm thực hiện
- Phương pháp luận tốt hơn đếm LOC
- Tương tự đối với ngôi nhà
  - Diện tích ngôi nhà  $\sim$  số dòng lệnh của phần mềm
  - Số lượng phòng tắm và phòng ngủ  $\sim$  điểm chức năng
  - Con số trước chỉ là kích cỡ, con số sau là kích cỡ và chức năng
- Sáu bước cơ bản

# Quá trình xác định điểm chức năng

---

- 1. Đếm số chức năng nghiệp vụ của mỗi loại
  - Các loại: đầu ra, đầu vào, truy vấn dữ liệu, cấu trúc tệp hoặc dữ liệu và giao diện
- 2. Thiết lập yếu tố độ phức tạp cho mỗi chức năng và sử dụng
  - Đơn giản, Trung bình, Phức tạp
  - Thiết lập một trọng số nhân có mỗi yếu tố (0->15)
  - Dẫn tới một tổng số điểm chức năng không điều chỉnh được nữa
- 3. Tính toán một nhân tố ảnh hưởng và áp dụng nó
  - Nó chạy từ 0.65 đến 1.35, và dựa vào 14 yếu tố
- 4. Kết quả thành "tổng điểm chức năng"
  - Điều này có thể được sử dụng trong việc ước lượng tương thích

# Vấn đề của phương pháp tham số

---

- Nhớ rằng: hầu hết các dự án bạn sẽ gặp không sử dụng những phương pháp này
- Cái gì là "bình thường", vì vậy đừng ngạc nhiên
  - Hoặc bước vào một công việc mới và nói “Nào, hãy sử dụng phương pháp COCOMO”
- Có hiệu quả hơn đối với những dự án lớn
  - Nơi mà một tiền đề lịch sử quá khứ tồn tại
- Vấn đề chính cho hầu hết các dự án là
  - Thiếu các dự án tương tự
    - Vì vậy thiếu dữ liệu tương thích
- Bắt đầu thế nào



# Việc dùng lại mã nguồn và ước lượng

---

- Không phải miễn phí
- Các loại mã: mới, sửa đổi, dùng lại
- Nếu mã nguồn có lớn hơn 50% sửa chữa thì được coi là mới
- các yếu tố dùng lại có nhiều phạm vi
  - Mã dùng lại chiếm 30% công của mã mới
  - Mã sửa đổi là 60% mã mới
- Nhân công cho tích hợp với mã sửa đổi vẫn nhiều như với mã mới

# Ước lượng công

---

- Đến hiện giờ khi bạn biết "kích cỡ", cần xác định "công" để xây dựng dự án
- Nhiều mô hình khác nhau: kinh nghiệm, toán học, chủ quan
- Được thể hiện theo đơn vị thời gian thực hiện
  - Ví dụ đơn vị là công một người làm trong một tháng

# Ước lượng công

---

- Bảng chuyển đổi kích cỡ ra công
- Như với ước lượng kích cỡ theo tham số, những kỹ thuật này thực hiện tốt hơn với dữ liệu lịch sử
- Lại một lần nữa, đừng nhìn vào các dự án "trung bình"
- Thường thì các bước ước lượng kích cỡ và công sức được kết hợp lại (điều này không phải được khuyến khích nhưng thường xảy ra)
- Lập lịch “dựa trên cam kết” thường được thực hiện
  - yêu cầu người lập trình cam kết một ước lượng (của chính người đó)

# COCOMO (COnstructive COst MOdel)

---

- Mô hình chi phí xây dựng
- Đưa ra theo công sức của một người trong tháng
- Cost drivers sử dụng Cao/Trung bình/Thấp và bao gồm
  - Động cơ
  - Khả năng của đội dự án
  - Kinh nghiệm làm ứng dụng
- Điểm yếu lớn nhất?
  - Đòi hỏi đầu vào là kích cỡ ước lượng của sản phẩm theo LOC

# Các vấn đề của ước lượng

---

- Ước lượng chất lượng là cần sớm nhưng thông tin bị hạn chế
- Dữ liệu ước lượng chính xác có sẵn vào lúc cuối nhưng không cần thiết
- Ước lượng tốt nhất là dựa trên kinh nghiệm trước đó
- Nhiều dự án phần mềm thường có ít hoặc không có
  - Các công nghệ thay đổi
  - Dữ liệu lịch sử không sẵn có
  - Sự đa dạng trong kinh nghiệm/ kiểu
  - Bản chất chủ quan của việc ước lượng phần mềm

# Ước lượng thiếu và thừa

---

- Các vấn đề ước lượng thừa
  - Dự án sẽ không có đủ tiền
    - Những ước lượng bảo thủ để đảm bảo 100% thành công có thể dẫn đến xác suất có nguồn tiền đáp ứng là 0.
  - Luật Parkinson: Công việc mở rộng trong khoảng thời gian cho phép
  - Nguy hiểm của đặc tính và thiếu phạm vi
  - Nhận thức được “double-padding”: thành viên đội + quản lý
- Các vấn đề ước lượng thiếu
  - Các vấn đề về chất lượng (các pha ngắn quan trọng như kiểm thử)
  - Khả năng không đáp ứng được hạn thời gian
  - Các vấn đề khác về ý thức và động cơ của đội dự án

# Các chỉ dẫn về ước lượng

---

- Ước lượng lặp đi lặp lại!
  - Quá trình làm tốt dần dần
  - Tiến hành các ước lượng tốt nhất ở mỗi giai đoạn lập kế hoạch
  - Làm tốt dần các ước lượng và thay đổi các kế hoạch dần dần
  - Các kế hoạch và các quyết định có thể được thay đổi để đáp ứng
  - Cần cân đối: quá nhiều phiên bản với quá ít phiên bản chỉnh sửa

# Nhận thức được các hạn về thời gian

---

- Chúng là "hạn thực tế" không?
  - Được gắn với một sự kiện bên ngoài
  - Cần có để đảm bảo dự án thành công
  - Ví dụ: kết thúc năm tài chính, hết hạn hợp đồng, hạn đến năm 2000
- Hoặc có phải "hạn giả" không?
  - thiết lập bởi người có quyền bất kỳ
  - Có thể có một vài sự linh hoạt



# Thể hiện sự ước lượng

---

- Cách bạn thể hiện sự ước lượng thế nào có thể gây ra ảnh hưởng lớn
  - Các kỹ thuật
    - Các nhân tố Cộng hoặc trừ
      - 6 tháng +/-1 tháng
    - Các khoảng
      - 6-8 tháng
    - Định lượng rủi ro
      - +/- với các thông tin được thêm vào
      - +1 tháng với các công cụ mới không làm việc như mong đợi
      - -2 tuần cho khoảng thời gian chậm trễ để thuê lập trình viên mới
    - Các tình huống
      - Tốt nhất/ Theo kế hoạch/ Hiện tại/ Tồi nhất
    - Các ngày hoạt động
      - Q3 02: quý 3 năm 2002
    - Các yếu tố khẳng định
      - Tháng 4 xác suất 1-10%, tháng 7 xác suất 1-50%
- Principles of Project Management, Fall 2008

# Các yếu tố ước lượng khác

---

- Kinh nghiệm hoặc kỹ năng của đội dự án
  - Lên tới một điểm
  - Thường cần nhiều hơn cho các thành viên ở mức thấp, ví dụ như một người mới hoặc tập sự
- Tính toán tới thời gian "không cho dự án" và các nhiệm vụ chung khác
  - Họp, nói chuyện điện thoại, lướt web, các ngày nghỉ
- Có các công cụ ước lượng thương mại sẵn có
  - Chúng thường yêu cầu các cấu hình dựa trên các dữ liệu trong quá khứ

# Các chú ý về ước lượng khác

---

- Nhớ rằng: "quản lý các mong đợi"
- Luật Parkinson
  - “Mở rộng công việc trong phạm vi thời gian cho phép”
- Bệnh sinh viên
  - Nước đến chân mới nhảy

# Phân tích tài chính cho dự án

---

- Việc cân nhắc về tài chính thường là rất quan trọng trong việc lựa chọn dự án ban đầu
- Ba phương pháp chính để xác định giá trị tài chính của các dự án:
  - Phân tích giá trị thực: Net present value (NPV)
  - Phân tích các giá trị trả trên đầu tư: Return on investment (ROI)
  - Phân tích giá trị kiếm được: Payback analysis

# Phân tích NPV

---

- NPV: một phương pháp tính toán số tiền mong đợi kiếm được hoặc bị mất đi từ dự án bằng cách lấy tổng tiền thu vào trừ đi tổng tiền chi ra cho đến thời điểm hiện tại
- Các dự án với một giá trị NPV dương sẽ được cân nhắc để lựa chọn nếu giá trị tài chính là một tiêu chí quan trọng với dự án này
- Giá trị NPV càng cao thì càng tốt

# Ví dụ về NPV

	A	B	C	D	E	F	G	H	I	J
2										
3	AN. INT. RATE -->	10%								
4										
5	<b>PROJECT 1</b>	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	TOTAL			
6	REVENUES	\$0	\$2,000	\$3,000	\$4,000	\$5,000	\$14,000			
7	COSTS	\$5,000	\$1,000	\$1,000	\$1,000	\$1,000	\$9,000			
8	CASH FLOW	(\$5,000)	\$1,000	\$2,000	\$3,000	\$4,000	\$5,000			
9	NPV	<b>\$2,316</b>								
10		Formula =npv(b3,b8:f8)								
11										
12	<b>PROJECT 2</b>	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	TOTAL			
13	REVENUES	\$1,000	\$2,000	\$4,000	\$4,000	\$4,000	\$15,000			
14	COSTS	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$10,000			
15	CASH FLOW	(\$1,000)	\$0	\$2,000	\$2,000	\$2,000	\$5,000			
16	NPV	<b>\$3,201</b>								
17		Formula =npv(b3,b15:f15)								
18										
19	RECOMMEND PROJECT 2 BECAUSE IT HAS THE HIGHER NPV.									
20										
21	IF STATEMENT -->	=IF(B9>B16,A5,A12)								
22	RESULT -->	PROJECT 2								

Notice that cash flow totals are the same, but NPVs are different.

# Phân tích ROI

---

- ROI: tỉ lệ thu nhập trên số tiền đầu tư  
$$\text{ROI} = (\text{tổng thu nhập} - \text{tổng chi phí}) / \text{tổng chi phí}$$
- Giá trị ROI càng cao thì càng tốt
- Nhiều tổ chức quy định một tỉ lệ ROI cần thiết hoặc chấp nhận được cho các dự án

# Phân tích Payback

---

- Một cân nhắc tài chính quan trọng khác nữa là phân tích giá trị kiếm được
- Giai đoạn hưởng giá trị kiếm được là khoảng thời gian lợi nhuận bắt đầu lớn hơn chi phí (lượng tiền đã đầu tư) cho dự án
- Giai đoạn hưởng thụ này xảy ra khi tổng tích lũy của lợi nhuận và chi phí lớn hơn 0
- Nhiều tổ chức muốn các dự án công nghệ thông tin có giai đoạn hưởng thụ đến sớm



# NPV, ROI, giai đoạn Payback: ví dụ 1

	A	B	C	D	E	F	G	H
1								
2	DISCOUNT RATE →	10%		Years				
3		1	2	3	4	5	TOTAL	
4	COSTS	(\$5,000)	(\$1,000)	(\$1,000)	(\$1,000)	(\$1,000)	-9,000	
5	DISCOUNT FACTOR	0.91	0.83	0.75	0.68	0.62		
6	DISCOUNTED COSTS	-4,545	-826	-751	-683	-621	-7,427	
7								
8	BENEFITS	\$0	\$2,000	\$3,000	\$4,000	\$5,000	14,000	
9	DISCOUNT FACTOR	0.91	0.83	0.75	0.68	0.62		
10	DISCOUNTED BENEFITS	0	1,653	2,254	2,732	3,105	9,743	
11								
12	DISCOUNTED BENEFITS + COSTS	-4,545	826	1,503	2,049	2,484	2,316 ← NPV	
13	CUMULATIVE BENEFITS + COSTS	-4,545	-3,719	-2,216	-167	2,316	4,633	
14						↑		
15	ROI	31%				Payback in this year		

# NPV, ROI, giai đoạn Payback: ví dụ 2

	A	B	C	D	E	F	G	H
1								
2	DISCOUNT RATE →	10%		Years				
3		1	2	3	4	5	TOTAL	
4	COSTS	(\$2,000)	(\$2,000)	(\$2,000)	(\$2,000)	(\$2,000)	-10,000	
5	DISCOUNT FACTOR	0.91	0.83	0.75	0.68	0.62		
6	DISCOUNTED COSTS	-1,818	-1,653	-1,503	-1,366	-1,242	-7,582	
7								
8	BENEFITS	\$1,000	\$2,000	\$4,000	\$4,000	\$4,000	15,000	
9	DISCOUNT FACTOR	0.91	0.83	0.75	0.68	0.62		
10	DISCOUNTED BENEFITS	909	1,653	3,005	2,732	2,484	10,783	
11								
12	DISCOUNTED BENEFITS + COSTS	-909	0	1,503	1,366	1,242	3,201 ← NPV	
13	CUMULATIVE BENEFITS + COSTS	-909	-909	594	1,960	3,201	6,403	
14				↑				
15	ROI	42%		Payback in this year				

# Mô hình tính điểm đánh trọng số

---

- Một mô hình tính điểm theo trọng số là một công cụ cung cấp một quá trình thống nhất để lựa chọn dự án dựa trên nhiều tiêu chí
  - Thứ nhất, xác định các tiêu chí quan trọng tới quá trình lựa chọn dự án
  - Sau đó gán trọng số (phần trăm) tới mỗi tiêu chí sao cho tổng các trọng số là 100%
  - Tiếp đến gán điểm cho mỗi tiêu chí của mỗi dự án
  - Nhân các điểm với các trọng số tương ứng để tính tổng số điểm có trọng số
- Tổng điểm có trọng số này càng cao thì càng tốt

# Mô hình ví dụ về tính điểm với trọng số

	A	B	C	D	E	F
1	Criteria	Weight	Project 1	Project 2	Project 3	Project 4
2	Supports key business objectives	25%	90	90	50	20
3	Has strong internal sponsor	15%	70	90	50	20
4	Has strong customer support	15%	50	90	50	20
5	Realistic level of technology	10%	25	90	50	70
6	Can be implemented in one year or less	5%	20	20	50	90
7	Provides positive NPV	20%	50	70	50	50
8	Has low risk in meeting scope, time, and cost goals	10%	20	50	50	90
9	<b>Weighted Project Scores</b>	<b>100%</b>	<b>56</b>	<b>78.5</b>	<b>50</b>	<b>41.5</b>



# Bài tập WBS

---

- Bạn chỉ cần làm tới mức 2 (hoặc 3 nếu bạn muốn làm khá chi tiết hoặc điều đó giúp làm rõ các hoạt động ở mức cao hơn). Đi xuống mức 2 có nghĩa là bắt đầu với mức 0 (một nút đơn- tên dự án của bạn), mức 1, mức cao nhất của sự phân chia (4-7 nút) và thêm một mức dưới đó.
- Như đã trình bày trên lớp, bạn có thể sử dụng một cách tiếp cận theo tiến trình, theo sản phẩm hoặc cách lai giữa chúng. Đối với hầu hết các dự án, dùng cách tiếp cận theo tiến trình sẽ hoạt động tốt nhất ở mức cao nhất.
- Bạn có thể sử dụng các phần mềm hoặc Excel, hoặc Word, hoặc Project. Tuân thủ theo cơ chế đánh số chuẩn cho cấu trúc phân rã công việc.
- Hãy suy nghĩ điều này cho thật thấu đáo vì bạn đang là giám đốc dự án.

# Câu hỏi?

---