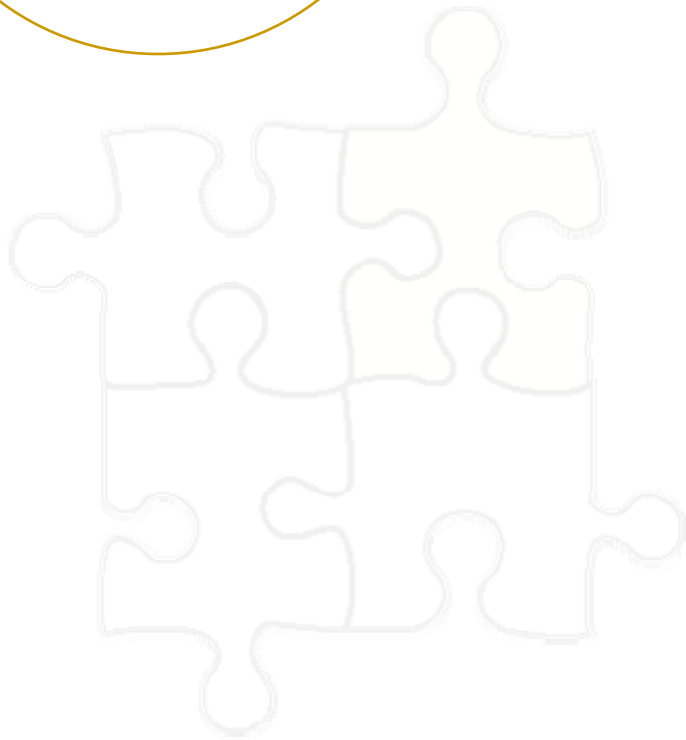


# Quản lý dự án phần mềm



# Nội dung

---

- Giới thiệu về quản lý dự án phần mềm
- Đo và ước lượng
- Lập lịch và theo dõi
- Đảm bảo chất lượng phần mềm
- Nghiên cứu khả thi
- Quản lý nhân sự
- Quản lý thay đổi
- Công cụ hỗ trợ quản lý dự án

# Tài liệu

---

- Pressman, *Software Engineering*, McGraw Hill (chapter 2 & 3)
- Sommerville, *Software Engineering*, Addison-Wesley (chapter 29)
- Ngô Trung Việt, *Phương pháp luận quản lý dự án CNTT*, NXB KHKT
- Giáo trình kỹ nghệ phần mềm (chương 6)
- Các tài liệu điện tử khác

# Tại sao phải quản lý dự án

---

- Các dự án thường:
  - Không hoàn thành đúng hạn
  - Chi phí xây dựng vượt quá dự toán
  - Chất lượng không đảm bảo

# Thống kê của Standish Group (2006)

---

- Có tới 50% trong số các dự án phần mềm thất bại
- Chỉ có 16.2% dự án là hoàn thành đúng hạn và nằm trong giới hạn ngân sách, đáp ứng tất cả tính năng và đặc tính như cam kết ban đầu
- Có 52.7% dự án được hoàn thành và đi vào hoạt động nhưng không hoàn thành đúng hạn và bội chi, thêm nữa không đáp ứng đầy đủ tính năng và đặc tính như thiết kế ban đầu
- Và có 31.1% dự án thất bại trước khi được hoàn thành
- -> hơn 83.8% dự án thất bại hoặc không đáp ứng những yêu cầu ban đầu

# Mục tiêu

---

- Quản lý các yếu tố:
  - Thời gian: đúng thời hạn
  - Chi phí: không vượt dự toán
  - Sản phẩm: đầy đủ các chức năng đã định
  - Thỏa mãn yêu cầu khách hàng
    - thỏa mãn về nhu cầu
    - thỏa mãn về tiến trình

# Nhiệm vụ, quyền hạn của người quản lý dự án

---

- Thời gian
  - lập lịch, điều chỉnh lịch
  - kiểm tra/đối chiếu các tiến trình con với lịch biểu
  - tạo độ mềm dẻo trong lịch biểu
- Tài nguyên
  - thêm tiền, thêm người, thêm thiết bị
- Sản phẩm
  - thêm, bớt, sửa chức năng
- Rủi ro
  - phân tích rủi ro
  - đề xuất giải pháp
  - thực hiện giải pháp và giám sát

# Các pha công việc

---

- Thiết lập: viết đề án
- Ước lượng (chi phí, người, thiết bị, ...)
- Phân tích rủi ro
- Lập kế hoạch
- Chọn người
- Theo dõi và kiểm soát tiến trình
- Viết báo cáo và trình diễn



# Các hoạt động thường xuyên

---

- Đảm bảo chất lượng phần mềm
  - đảm bảo sự đúng đắn
  - đảm bảo sự tuân thủ theo chuẩn
- Quản lý thay đổi/quản lý cấu hình phần mềm
  - Quản lý thay đổi về yêu cầu, thiết kế, mã nguồn...
  - Quản lý cấu hình (được phát triển phân tán)

# 1. Đo và ước lượng

---

- Cách thức tiếp cận quản lý: đo và ước lượng
- Đo phần mềm
  - kích thước, chi phí, hiệu năng, chất lượng
- Ước lượng
  - kích thước
  - chi phí
  - thời gian
- *Chỉ quản lý được các yếu tố có thể đo được*

# Độ đo và ước lượng

---

- Ước lượng phần mềm là công việc quan trọng hàng đầu trong quản lý dự án
  - kích cỡ, chi phí
  - thời gian, nhân lực
- Để ước lượng được cần có độ đo
  - kích cỡ, chất lượng, hiệu năng
- Nguyên lý: *cần phải xác lập độ đo cho mọi công việc*
  - *độ đo phải định lượng*

# Đo kích cỡ phần mềm

---

- Đo theo dòng lệnh (LOC – Lines Of Code)
  - trực quan
  - phụ thuộc ngôn ngữ
- Đo điểm chức năng (FP – Functional Points)
  - độc lập với ngôn ngữ
  - phụ thuộc các mô hình lựa chọn (tham số)
    - hiệu năng: KLOC/người-tháng
    - chất lượng: số lỗi/KLOC
    - chi phí: giá thành/KLOC



# Điểm chức năng

---

- Tổng hợp các đặc trưng của module
  - Input
  - Output
  - Interface
  - Files
- Đặt trọng số cho các đặc trưng
- Trọng số phụ thuộc vào ngữ cảnh (dự án) cụ thể
  - độ phức tạp của bài toán
  - Các yêu cầu về chất lượng, hiệu năng
  - Kích thước của dữ liệu sử dụng

# Điểm chức năng FP

---

- $FP = a_1I + a_2O + a_3E + a_4L + a_5F$ 
  - I : số Input
  - O: số Output
  - E: số yêu cầu
  - L: số tệp truy cập
  - F: số giao diện ngoại lai (devices, systems)

# Điểm chức năng: Ví dụ

---

- $FP = 4I + 5O + 4E + 10L + 7F$
  - Hàm: *Tính ước số chung lớn nhất của hai số nguyên*
    - Input  $I = 2$
    - Output  $O = 1$
    - Yêu cầu  $E = 1$
- > Điểm chức năng  $FP = 17$

# Độ đo về chất lượng dựa trên thống kê

---

- Độ tin cậy: MTBF – Mean Time Between Failures
  - thời gian chạy liên tục không có lỗi
- Thời gian khôi phục hệ thống
  - MTTR – Mean Time To Recover
- Tính sẵn có:  $\frac{MTBF}{MTBF + MTTR}$



# Độ đo hiệu quả phát hiện lỗi

---

- Hiệu quả khử lỗi:  $E/(E+D)$ 
  - E(error): lỗi phát hiện trước khi bàn giao
  - D(effect): lỗi phát hiện sau khi bàn giao
- $E/(E+d/0.9)$ 
  - d: số lỗi phát hiện trong 1 tháng sau khi bàn giao

# Ước lượng phần mềm

---

- Các yếu tố cần ước lượng
  - kích cỡ phần mềm
  - chi phí (công sức) phát triển
  - thời gian
  - số người tham gia
- Nguyên tắc ước lượng
  - phân rã chức năng
  - ước lượng với từng chức năng
  - dựa trên kinh nghiệm, dữ kiện quá khứ

# Ước lượng

---

- Kích cỡ
  - LOC: ước lượng trực tiếp với từng mô đun
  - FP: ước lượng gián tiếp thông qua ước lượng input/output, yêu cầu
- Công sức:
  - dựa trên kích cỡ, độ phức tạp
  - dựa vào dữ liệu quá khứ
  - đơn vị: person-day, person-week, person-month

## Ví dụ

---

- *Trang web xem kết quả học tập của sinh viên* bao gồm các mô đun/giao diện chính:
    - nhập thông tin tìm kiếm: 100 LOC
    - tìm kiếm trên CSDL sinh viên: 300 LOC
    - sinh kết quả: 100 LOC
- ➔ công sức: 01 person-week

Vậy phần mềm đào tạo 2000 LOC thì sao?

## Mô hình ước lượng COCOMO - Costructive Cost Model

---

- Ước lượng nỗ lực, thời gian, số người phát triển từ kích cỡ phần mềm.
- Sử dụng với các phần mềm lớn
- Mô hình cơ sở
  - Nỗ lực  $E = a * Lb$
  - Thời gian  $T = c * Ed$
  - Số người  $N = E/T$

L: số dòng lệnh (KLOC)  
a, b, c, d: tham số

# COCOMO: các bước tiến hành

---

- Thiết lập kiểu dự án
  - organic: đơn giản, không truy cập các thiết bị ngoại lai
  - semi-detached
  - embedded: phức tạp, truy cập thiết bị
- Xác lập (phân rã) mô đun và ước lượng số dòng lệnh
- Tính lại số dòng lệnh trên cơ sở tái sử dụng
- Tính nỗ lực phát triển E cho từng mô đun
- Tính lại E dựa trên độ phức tạp của dự án
  - độ tin cậy, độ lớn của CSDL
  - yêu cầu về tốc độ, bộ nhớ
- Tính thời gian và số người tham gia

# COCOMO: tham số cơ sở

---

	a	b	c	d
organic	3.2	1.05	2.5	0.38
semi-detached	3.0	1.12	2.5	0.35
embedded	2.8	1.2	2.5	0.32

# COCOMO: Ví dụ

---

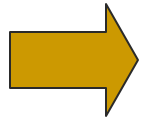
- Phần mềm kích cỡ 33.3 KLOC.
  - $a = 3.0$     $b = 1.12$     $c = 2.5$     $d = 0.35$
  - $E = 3.0 * 33.3^{1.12} = 152$  person-month
  - $T = 2.5 * E^{0.35} = 14.5$  tháng
  - $N = E/D = \sim 11$  người



# Khó khăn trong ước lượng

---

- Các thông số không trực quan
- Khó đánh giá tính đúng đắn của các tham số
- Không có mô hình tổng quát
- Các kỹ thuật ước lượng đang thay đổi



- Áp dụng các mục hớnh khỏc nhau
- Tiến hành ước lượng nhiều lần
- Ước lượng lại khi dự ỏn tiến triển

## 2. Lập lịch và theo dõi

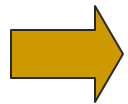
---

- Ước lượng cho chúng ta con số khái quát để làm cơ sở thực hiện dự án
  - Lịch trình cụ thể phụ thuộc vào mô hình lựa chọn
  - Số người tham gia thay đổi theo từng pha của dự án
- Cần phải phân tích chi tiết hơn và lập lịch để kiểm soát công việc

# Lập lịch và theo dõi

---

- Lập lịch để kiểm soát công việc (nhiệm vụ)
  - xác định nhiệm vụ
  - thời điểm bắt đầu, thời điểm kết thúc
  - người thực hiện
  - ràng buộc (mối liên hệ giữa các nhiệm vụ)



cần cú độ mềm dẻo về thời gian

# Xác định tài nguyên cho dự án

---

- Con người
  - là nhân tố quan trọng nhất
  - cần phải tập hợp các thành viên có năng lực
  - mỗi giai đoạn cần số người, năng lực khác nhau
- Phần mềm dùng lại được
  - Các thành phần đã được đóng gói (dễ dàng dùng lại)
  - Các thành phần đã có kinh nghiệm (dễ dàng sửa chữa để phục vụ cho dự án)
  - Các thành phần dùng lại ít có kinh nghiệm (chi phí cho sửa chữa lớn)
- Phần cứng/công cụ phần mềm
  - Phải chia sẻ phần cứng, công cụ

# Xác định nhiệm vụ

---

- Nhiệm vụ phải được xác định là:
  - Là công việc có kết quả bàn giao
  - Qui trách nhiệm cho một cá nhân
  - Có hạn định về thời gian
  - Có thể đo được (tiến độ, chất lượng)

# Xác định ràng buộc nhiệm vụ

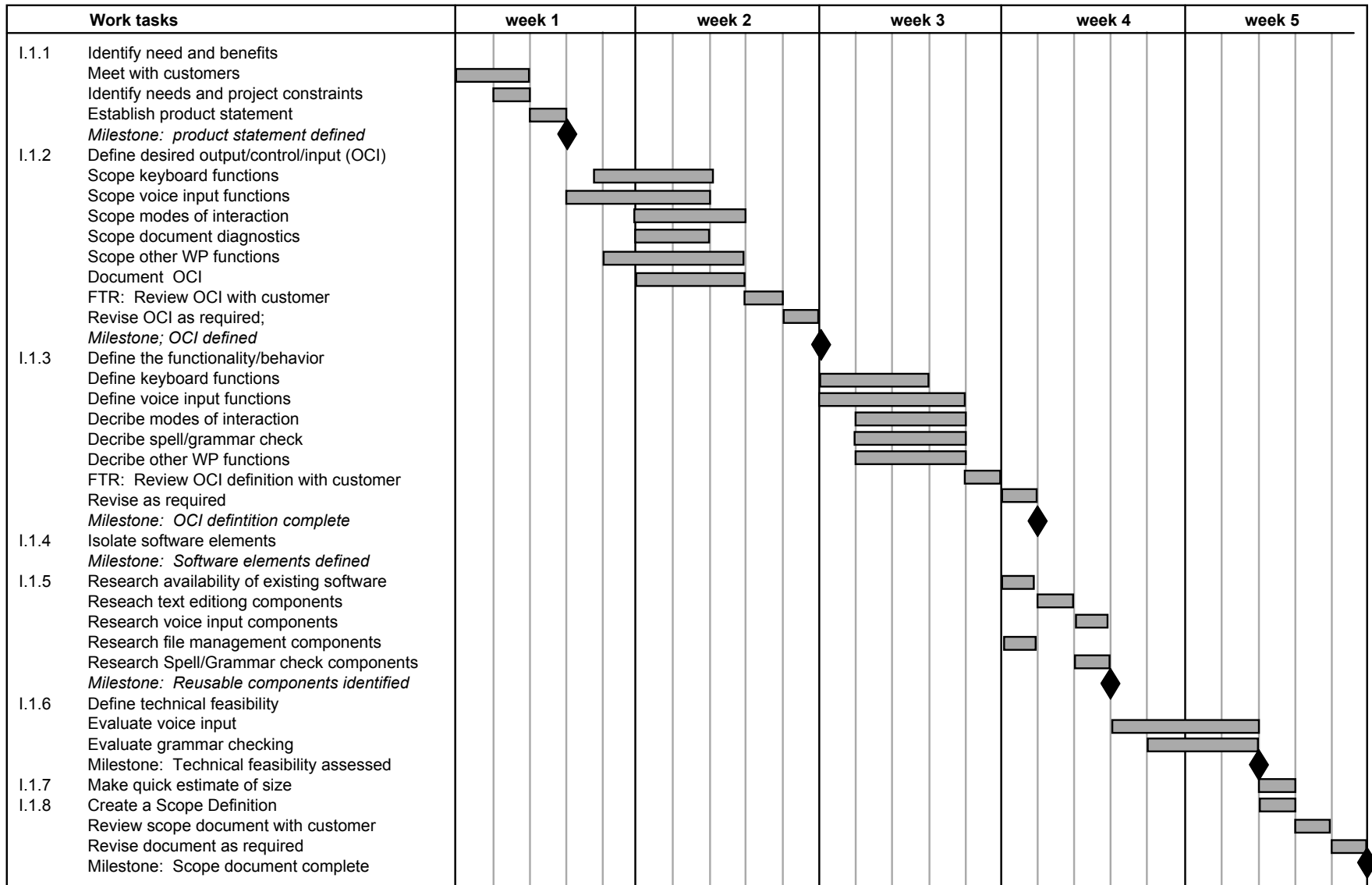
---

- Các ràng buộc về tài nguyên (con người, thiết bị)
- Ràng buộc về tiến trình
  - các nhiệm vụ phải được kết thúc trước
  - các nhiệm vụ có thể được thực thi kế tiếp
- Giảm tối đa các nhiệm vụ phụ thuộc
- Thực hiện các nhiệm vụ song song khi có thể

# Lập lịch nên

---

- Giảm tối đa thời gian thừa
- Tận dụng tối đa các nguồn lực có thể
- Điều phối tài nguyên (chỗ thừa/thiếu)
- Xem xét các hạn chế
  - phụ thuộc tiến trình
  - phụ thuộc tài nguyên
- Là một qui trình lặp lại
  - theo dõi thời gian biểu
  - sửa chữa, lập lại thời gian biểu
- Sử dụng các công cụ tự động





# Tham khảo

---

- Thời gian thực tế thường kéo dài hơn ước lượng từ 25% đến 40%.
- Lý do:
  - Một số công việc không ước lượng được
  - Một số công việc phải làm lại
  - Người phát triển tham gia đồng thời nhiều công việc

# 3. Đảm bảo chất lượng phần mềm

---

- Software Quality Assurance – SQA
  - Là công việc xuyên suốt quá trình phát triển phần mềm
- Thế nào là chất lượng?
  - Chất lượng của phần cứng = sự ổn định, sự đồng đều
  - Chất lượng phần mềm
    - Tin cậy, dễ sử dụng, hiệu quả, bảo trì
    - Khó đo đạc trực quan

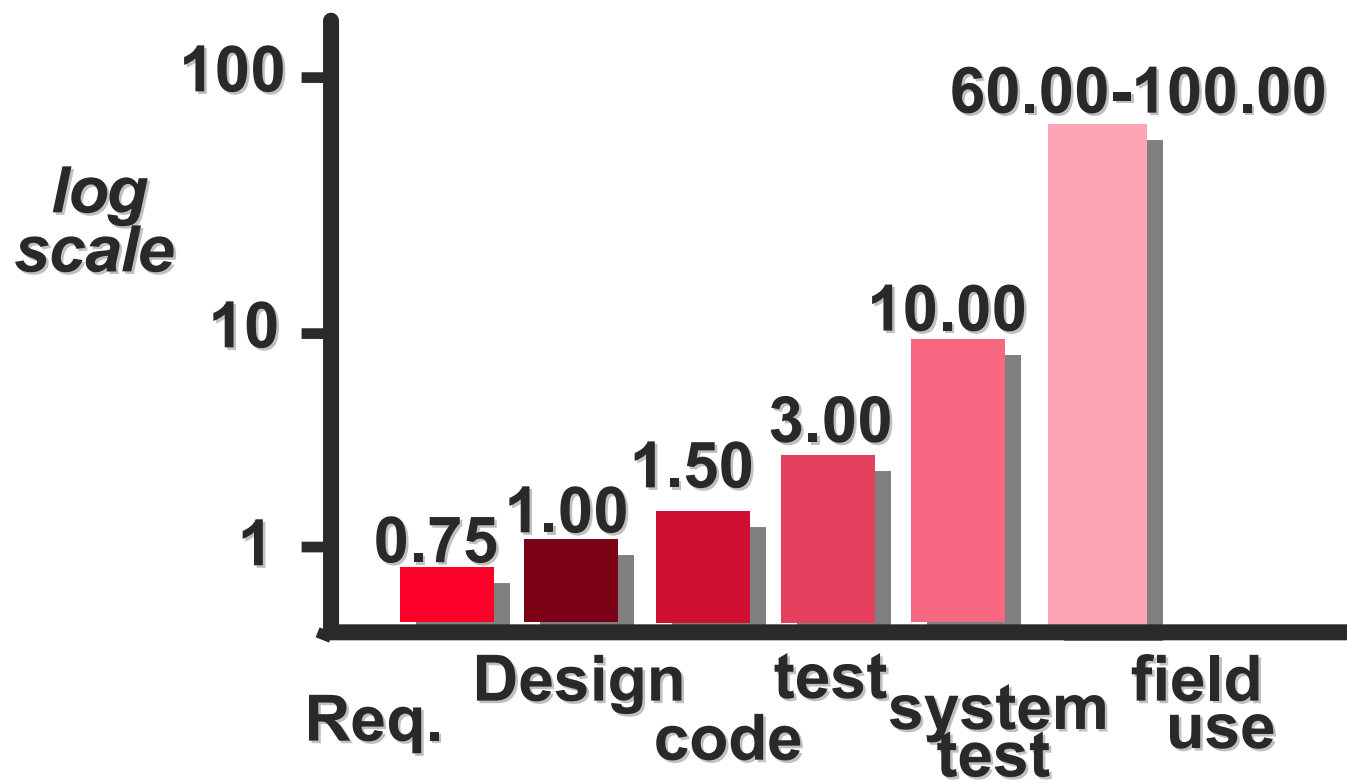
# Đảm bảo chất lượng

---

- Đảm bảo chất lượng khi bắt đầu dự án
  - Con người
  - Quy trình
  - Công cụ
- Đảm bảo chất lượng trong quá trình thực hiện dự án
  - tuân thủ quy trình (các chuẩn, các tài liệu)
  - họp xét duyệt
  - kiểm thử sản phẩm

# Giá trị cho tìm và sửa lỗi

---



# Xét duyệt

---

- Tại mỗi pha công việc, cần họp xét duyệt để đảm bảo chất lượng
  - không để lỗi truyền sang pha sau
- Thực hiện theo nhóm
- Xét duyệt các tài liệu
  - Phân tích
  - Thiết kế
  - Mã nguồn
  - Tài liệu người dùng
  - ...

## 4. Nghiên cứu khả thi

---

- *Xác định, phân tích các yếu tố*
  - Phạm vi phần mềm
  - Khả thi về kinh tế
  - Khả thi về kỹ thuật
  - Khả thi về pháp lý
  - Các rủi ro và biện pháp khắc phục

# Khả thi về kinh tế

---

- *Phân tích lợi ích - chi phí*
  - chi phí xây dựng
  - phí tổn vận hành
  - hiệu quả kinh tế
  - vị trí của sản phẩm
  - khả năng tài chính của khách hàng
- *Khách hàng và nhà phát triển có cách nhìn khác nhau về tính kinh tế, nhà phát triển cần thuyết phục khách hàng về tính kinh tế*

# Khả thi về kỹ thuật

---

- *Khó đánh giá ở giai đoạn phân tích*
  - có công nghệ để thực hiện không?
  - có năng lực thực hiện không?
  - có tài nguyên (phần cứng) để thực hiện không?
  - khách hàng có vận hành được không



# Khả thi về pháp lý

---

- Khả thi về pháp lý là yếu tố ít quen thuộc đối với người phát triển
  - Vi phạm bản quyền
    - sử dụng mã nguồn của người khác...
    - cung cấp âm nhạc trực tuyến...
  - Vi phạm tự do cá nhân
    - kiểm duyệt email, phá mật khẩu...
  - Gây hại đối với bên thứ ba
    - virus, spam email, DoS
  - Các vi phạm pháp luật khác
    - cung cấp các dịch vụ cấm,...

# Rủ ro và biện pháp

---

- *Các nhân tố có thể làm thất bại dự án*
  - rủi ro kỹ thuật: quá khó
  - rủi ro kinh tế: quá đắt
  - rủi ro thời gian: thời gian quá ngắn
- ➔ *phần hoạch yêu cầu*
  - cần thiết
  - mong muốn
  - phụ (optional)

# Báo cáo khả thi

---

- *cần đưa ra quyết định*
  - làm
  - không làm
  - xem xét lại

# Quản lý rủi ro

---

- Rủi ro là các sự kiện khiến dự án thất bại
  - chi phí quá cao
  - thời gian quá dài
  - tính năng quá kém
- Là các yếu tố có thể quản lý được
- Nhiệm vụ của người quản lý dự án
  - xác định (dự đoán) rủi ro
  - phân tích rủi ro (khả năng và thiệt hại)
  - quản lý rủi ro (đưa ra giải pháp)
  - giám sát (theo dõi sự xuất hiện, tác động của rủi ro) và thực hiện biện pháp quản lý

# Quản lý rủi ro

---

- Dựa trên phân hoạch yêu cầu
  - chức năng cần thiết
  - chức năng mong muốn
  - chức năng phụ thêm
- Nguyên lý Pareto (80-20)
- Phân tích, đưa ra quyết định có áp dụng biện pháp quản lý cần thiết hay không
  - dựa trên thống kê (kinh nghiệm)
  - dùng cây quyết định

## 5. Quản lý nhân sự

---

- Con người là yếu tố quan trọng nhất trong phát triển phần mềm
- Các thành viên rất khác nhau về năng lực
- Một số các công việc đặc thù không phải ai cũng làm được
  - lập trình hệ thống
  - giao diện đồ họa cao cấp
  - điều khiển thiết bị
  - cơ sở dữ liệu

# Nhóm và đặc trưng

---

- Phần mềm phát triển theo nhóm
- Kích thước tối ưu của nhóm: 3~8 người
- Tổ chức nhóm
  - lập trình viên
  - chuyên gia giao diện
  - chuyên gia miền ứng dụng
  - thủ thư phần mềm (quản lý cấu hình)
  - kiểm thử viên
- Cần có
  - team leader
  - technical leader

# Nhóm và đặc trưng

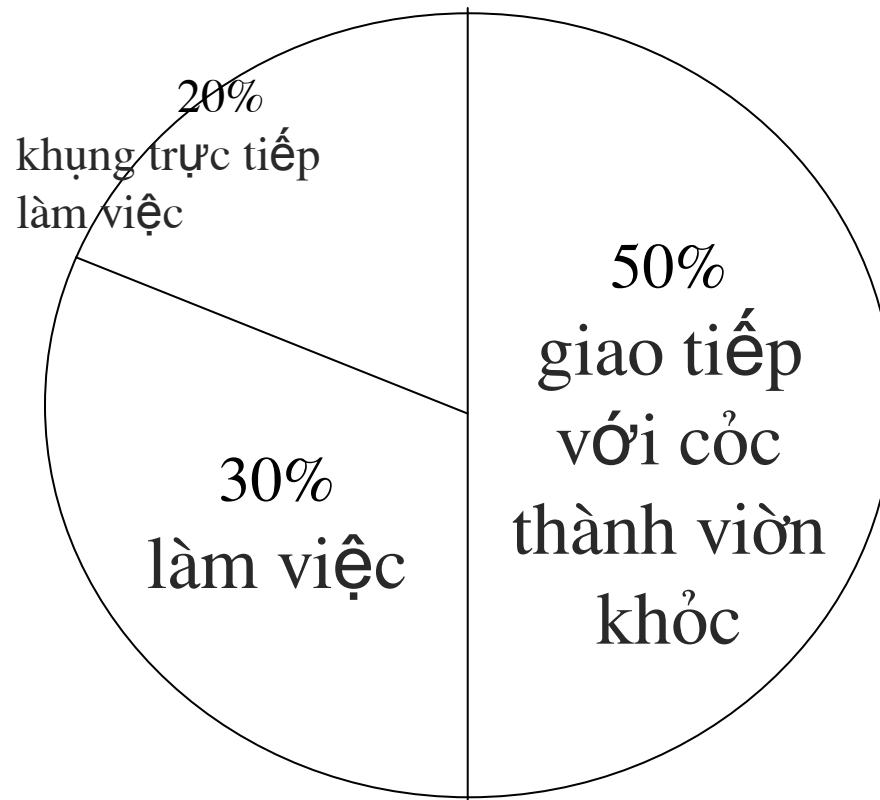
---

- Không nên tổ chức nhóm quá lớn
  - thời gian cho giao tiếp sẽ tăng cao
  - khó tăng tốc độ bằng cách thêm người
- Một số công việc chỉ nên để cho một người thực hiện
- Cần phân rã dự án lớn thành các dự án nhỏ



# Phân bổ thời gian làm việc

---



# Một số cách tổ chức nhóm

---

- Nhóm ngang quyền (democratic team)
  - Công việc được thảo luận và các thành viên thống nhất giải pháp chung
  - Các thành viên đều có kinh nghiệm và năng lực
- Nhóm XP
  - Một dạng của ngang quyền, lập trình đôi và chịu trách nhiệm chung
- Nhóm quyền lực tập trung (chief programmer team)
  - Nhóm trưởng có năng lực vượt trội và là người thiết kế chính
  - Các thành viên khác thực hiện công việc chi tiết

# Nhóm làm việc hiệu quả

---

- Các mục đích được thống nhất
- Thành viên tin tưởng vào vai trò và mục tiêu
- Chấp nhận mục tiêu và tiêu chí chất lượng
- Có phương thức trao đổi thông tin hiệu quả
- họp, trao đổi ý tưởng, kiểm soát thay đổi
- Xác lập được mối quan hệ hợp tác giữa các thành viên

## 6. Quản lý thay đổi

---

- Một trong các lý do khiến cho dự án thất bại
- Luôn có sự thay đổi
  - yêu cầu, thiết kế, mã hóa, sửa lỗi...
  - phần mềm luôn tiến hóa
- Không nhận ra sự thay đổi của vấn đề
- không có phương pháp hiệu quả để quản lý sự thay đổi

# Quản lý thay đổi

---

- Định nghĩa thay đổi với bất cứ hoạt động nào:
  - phạm vi
  - kết quả bàn giao
  - kiến trúc cơ bản
  - chi phí
  - lịch trình
- Lập tài liệu đầy đủ về các thay đổi, đảm bảo các thành viên hiểu rõ về các thay đổi
  - sử dụng công cụ hỗ trợ

# Quản lý cấu hình phần mềm

---

## Configuration management

- Nhiệm vụ của quản lý cấu hình:
  - quản lý phiên bản phần mềm
  - lưu trữ tài liệu, mã nguồn, dữ liệu
  - tạo điểm truy cập duy nhất (đảm bảo tính thống nhất của mã nguồn)
- Trên diện hẹp, còn gọi là quản lý mã nguồn

# Lợi ích

---

- Cung cấp cho người phát triển phiên bản mới nhất của phần mềm
- Quản lý các mã nguồn được lưu trữ phân tán
- Quản lý các phiên bản khác nhau
- Ghi chú lý do của sửa đổi mã nguồn
- Dễ dàng truy cập các phiên bản cũ
- Tích kiệm không gian đĩa

# Quản lý phiên bản

---

- Thế nào là phiên bản phần mềm
  - version
  - variant
  - release
- Phải đặt ra các tiêu chí để xác định phiên bản, tiêu chí để định danh phiên bản



# Phương thức hoạt động

---

- Lưu trữ tập trung
  - mã nguồn, tài liệu, công cụ
- Lưu trữ duy nhất (logic)
- Quản lý sửa đổi
  - không cho phép sửa đổi đồng thời
  - lưu trữ phiên bản cũ
  - thông tin sửa đổi: lý do, người thực hiện, thời điểm

# Nội dung lưu trữ

---

- Tài liệu
  - phân tích, thiết kế, tài liệu người dùng...
- Mã nguồn
- Công cụ phát triển
  - cần công cụ cũ để biên dịch lại các mã nguồn cũ cho việc bảo trì
- Các bộ dữ liệu test

*Với phần mềm lớn, phải quản lý hàng nghìn tài liệu*

# Chia sẻ mã nguồn

---

Nhiều người đồng thời phát triển một tệp mã nguồn

- Sử dụng cơ chế lock/unlock chỉ cho phép một người được quyền sửa đổi tại một thời điểm
  - lock (check out): mở tệp để sửa đổi
  - unlock (check in): kết thúc sửa đổi

Đồng thời sửa đổi / hệ thống giải nối các sửa đổi một cách tự động

# Các công cụ

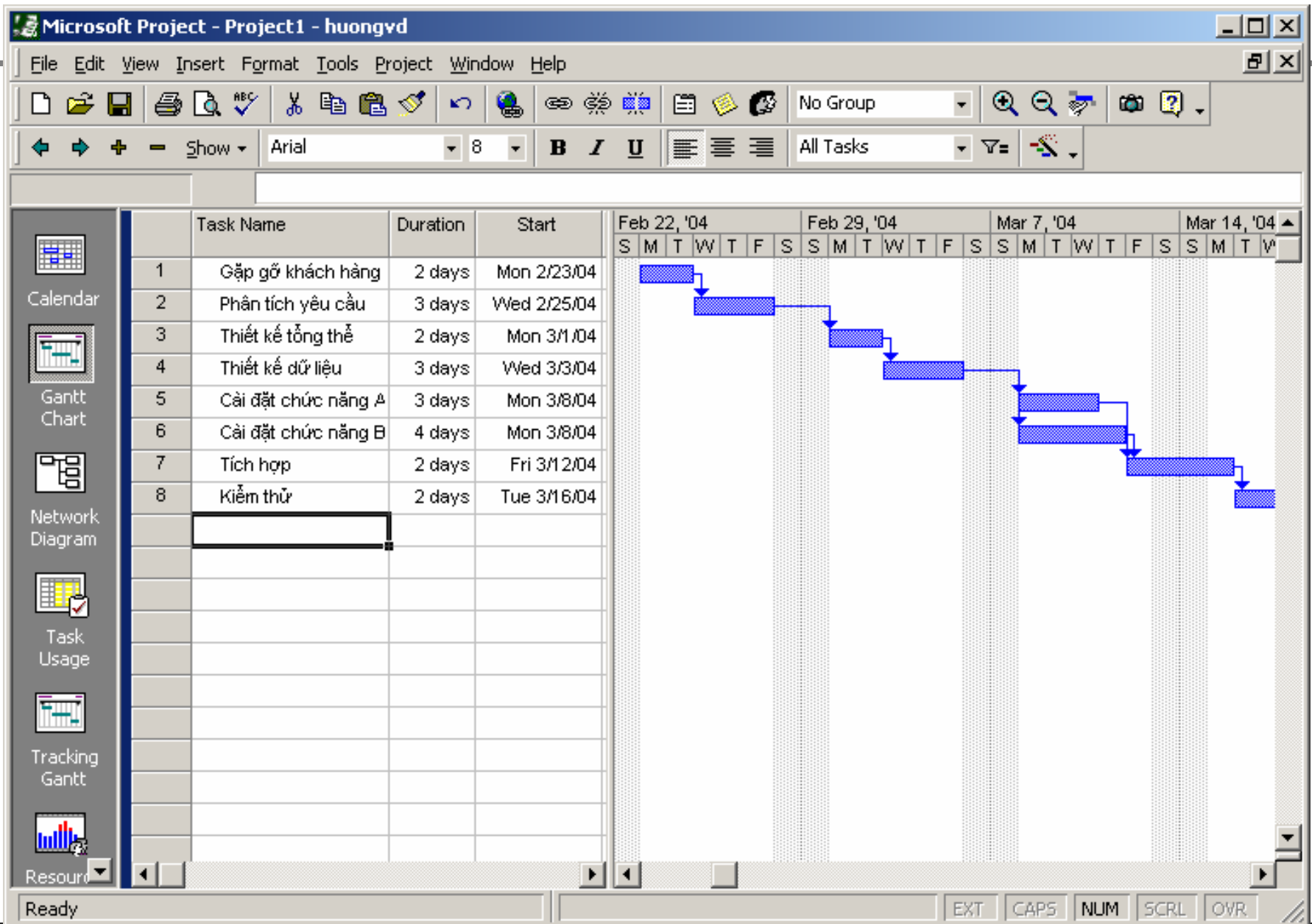
---

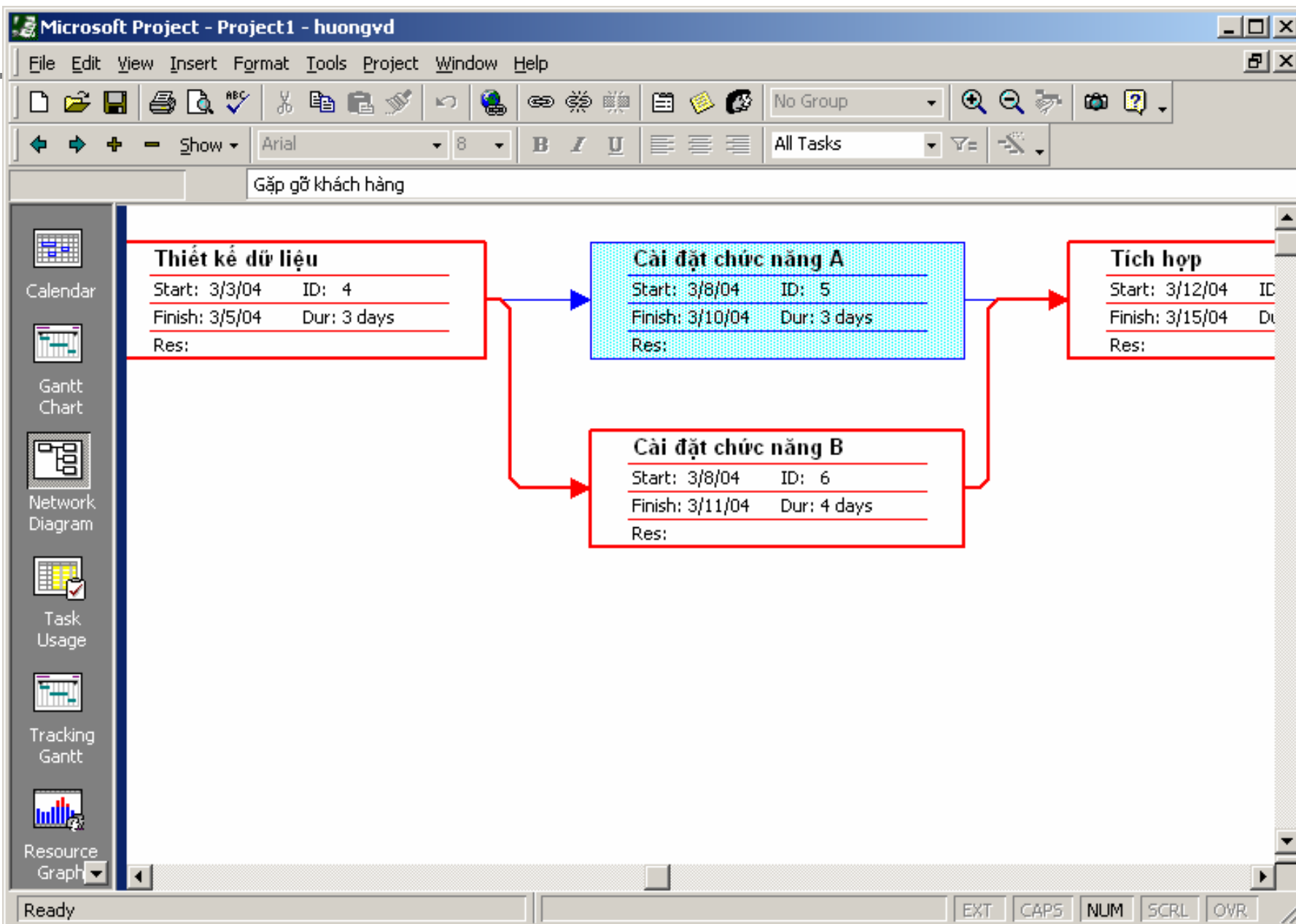
- RCS
  - chạy trên hệ điều hành Solaris
  - quản lí tệp
  - check out/check in
- CVS
  - dựa trên RCS
  - dùng cơ chế ghép nối sửa đổi
- Visual SourceSafe
  - trong bộ công cụ Visual Studio của Microsoft
  - quản lí project
  - check out/check in

## 7. Công cụ hỗ trợ quản lý dự án

---

- Microsoft Project 2000
  - Hỗ trợ quản lý dự án phần mềm
- Microsoft SourceSafe
  - Quản lý cấu hình, mã nguồn
- Visio 2000
  - Tạo bảng biểu, mô hình
- <http://www.startwright.com/project1.htm>





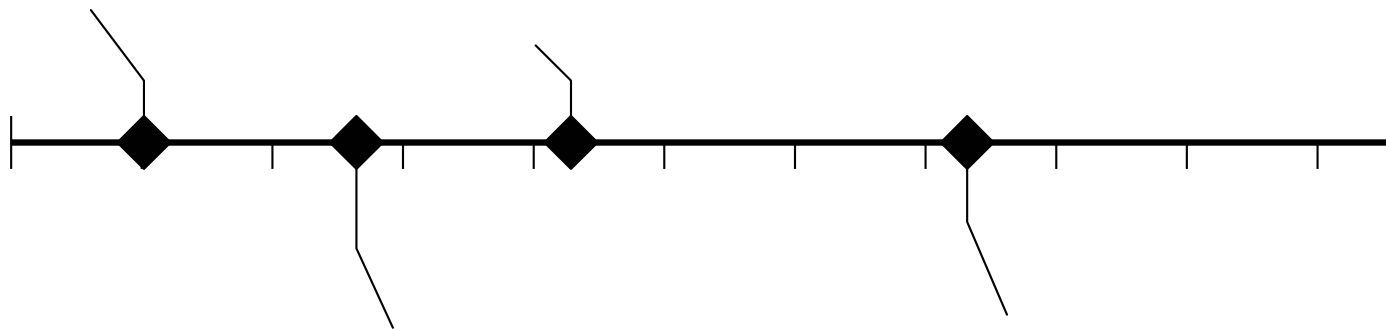
# Gantt Chart tạo bằng Visio 2000

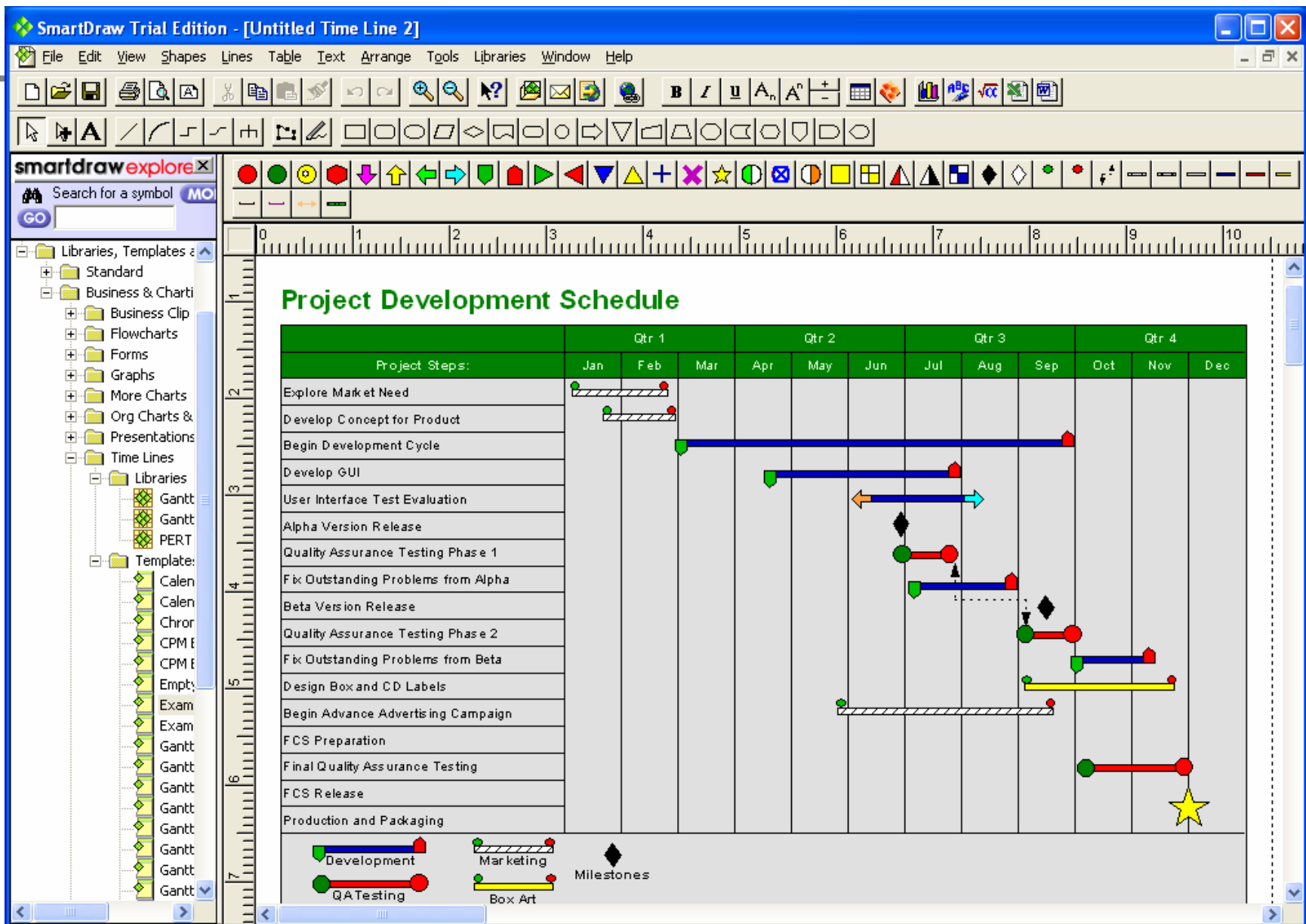
ID	Task Name	Start	End	Duration	Feb 22 2004						Feb 29 2004						Mar 7 2004				
					23	24	25	26	27	28	29	1	2	3	4	5	6	7	8	9	10
1	Gặp gỡ khách hàng	2/23/2004	2/24/2004	2d	■																
2	Phân tích yêu cầu	2/25/2004	2/26/2004	2d		■															
3	Đặc tả yêu cầu	2/27/2004	2/27/2004	1d					■												
4	Phân tích hệ thống	3/1/2004	3/3/2004	3d							■										
5	Thiết kế tổng thể	3/4/2004	3/10/2004	5d									■								
6																					



# Timeline tạo bằng Visio 2000

---





# Tổng kết: Quản lý dự án

---

- Người quản lý cần có kinh nghiệm, phải cứng rắn
  - Được trang bị kiến thức về quản lý dự án PM
  - Có thâm niên trong việc phát triển phần mềm
    - Đã từng tham gia các công đoạn của quá trình phát triển phần mềm
    - Kinh nghiệm đóng vai trò then chốt
    - Có trình độ quản lý tốt
    - Làm việc có kế hoạch
    - ...
- Cần có các độ đo
- Phải được lập tài liệu
- Luôn cần xem xét, điều chỉnh

---

Question?