

Kỹ nghệ phần mềm

Software Engineering

Đại học Kinh doanh và Công nghệ Hà Nội

Khoa CNTT

GV: Đào Thị Phụng

Email: phuongdt102@gmail.com

Page fb: [facebook.com/it.hubt](https://www.facebook.com/it.hubt)

Phone: 0946.866.817

Bài 2: Kỹ nghệ phần mềm



Nội dung

- Định nghĩa và khái niệm
- Lịch sử phát triển
- Các mô hình tiến trình

TÀI LIỆU THAM KHẢO



1. Nguyễn Văn Vy, Nguyễn Việt Hà. *Giáo trình kỹ nghệ phần mềm*. Nhà xuất bản Đại học Quốc gia Hà nội, 2008
2. Grady Booch, James Rumbaugh, Ivar Jacobson. *The Unified Modeling language User Guid*. Addison-Wesley, 1998.
3. M. Ould. *Managing Software Quality and Business Risk*, John Wiley and Sons, 1999.
4. Roger S.Pressman, *Software Engineering, a Practitioner's Approach*. Fifth Edition, McGraw Hill, 2001.
5. Ian Sommerville, *Software Engineering*. Sixth Edition, Addison-Wasley, 2001.
6. Nguyễn Văn Vy. *Phân tích thiết kế hệ thống thông tin hiện đại. Hướng cấu trúc và hướng đối tượng*, NXB Thống kê, 2002, Hà Nội.

Các định nghĩa kỹ nghệ phần mềm



Kỹ nghệ phần mềm – *Software Engineering:SE*

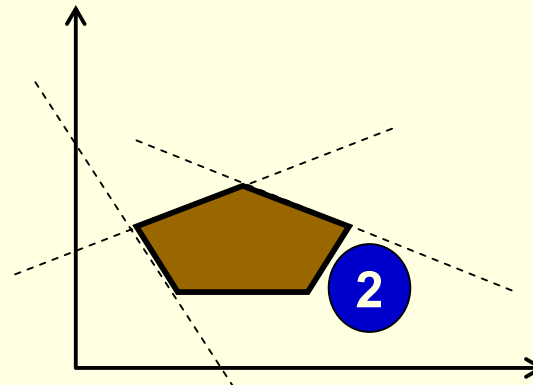
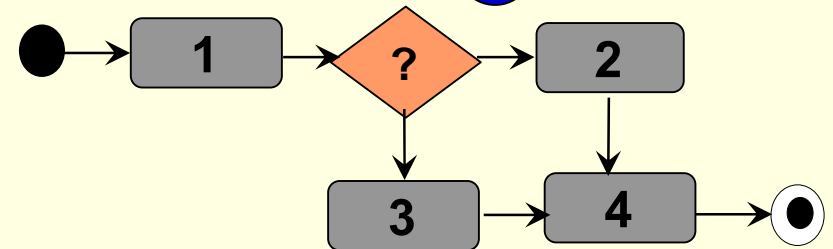
- SE là **thiết lập và sử dụng các nguyên lý công nghệ đúng đắn** để được phần mềm 1 cách kinh tế, vừa tin cậy vừa làm việc hiệu quả trên các máy thực (**Bauer [1969]**)
- SE là **nguyên lý kỹ nghệ** liên quan đến tất cả các mặt **lý thuyết, phương pháp và công cụ** của phần mềm (**Sommerville [1995]**)

Định nghĩa SE

- SE là bộ môn tích hợp cả **qui trình, các phương pháp, các công cụ** để phát triển phần mềm máy tính (**Pressman [1995]**)

- Ba mặt cơ bản của kỹ nghệ phần mềm là:

1. Quy trình/thủ tục
2. Phương pháp
3. Công cụ



Lịch sử của kỹ nghệ phần mềm



- Đề xướng, hình thành (70s)
Các phương pháp lập trình và cấu trúc dữ liệu
 - ◆ Khái niệm về tính môđun
 - ◆ Khái niệm sơ đồ khối, lập trình top-down
 - ◆ Lập trình có cấu trúc (Dijkstra)
 - ◆ phương pháp chia môđun một chương trình
 - ◆ Trừu tượng hóa dữ liệu (Liskov)

Lịch sử của kỹ nghệ phần mềm



■ Tăng trưởng (*nửa đầu 80s*)

- Xuất hiện các phương pháp phát triển hệ thống
 - công nghệ CSDL (*mô hình quan hệ*)
 - phân tích, thiết kế hướng cấu trúc (*biểu đồ luồng,..*)
- Các bộ công cụ phát triển
 - công cụ trợ giúp phân tích, thiết kế
 - bộ khởi tạo chương trình, kiểm thử
 - các ngôn ngữ bậc cao
- Bắt đầu quan tâm đến quản lý
 - Các độ đo phần mềm
 - Quản lý theo thống kê

Lịch sử của kỹ nghệ phần mềm



■ Phát triển (*từ giữa 80s*)

- Hoàn thiện công nghệ cấu trúc, ra đời công nghệ đối tượng
 - Nhiều mô hình hướng cấu trúc triển khai, chuẩn hóa
 - CASE hoàn thiện, đạt mức tự động hóa cao
 - Ngôn ngữ thế hệ 4 ra đời (LIPS, PROLOG,..)
 - Công nghệ hướng đối tượng bắt đầu phát triển:
 - Quy trình RUP, UML, Các công cụ đầy đủ (ROSE, JIBULDER,..)
 - Sử dụng lại chiếm vị trí quan trọng trong phát triển

Lịch sử của kỹ nghệ phần mềm



■ Phát triển (từ giữa 80s)

➤ Phát triển công nghệ đối tượng

- ◆ Kho dữ liệu, CSDL đối tượng, đa phương tiện
- ◆ Định hướng sử dụng lại: *thành phần, mẫu, Framework*
- ◆ Công nghệ Web: *web services*

➤ Phát triển các mô hình quản lý

- ◆ Chuẩn quản lý được công nhận (CMM, IS9000-03)
- ◆ Nhiều mô hình tổ chức làm phần mềm được đề xuất
- ◆ Nhiều công cụ trợ giúp quản lý dự án hoàn thiện

Các yếu tố cơ bản của - SE



- Tổng hợp: Kỹ nghệ phần mềm
 - ◆ Là một quá trình kỹ nghệ tích hợp:
 - Thủ tục (*procedures*)
 - Phương pháp (*methods*)
 - Công cụ (*tools*)
 - ◆ Nhằm tạo ra phần mềm hiệu quả, với các giới hạn cho trước

Các thủ tục - *Procedures*



- Qui trình phát triển và quản lý:
 - Xác định trình tự thực hiện các công việc
 - Xác định các tài liệu, sản phẩm cần bàn giao, và cách thức thực hiện
 - Định các mốc thời gian (*milestones*) và sản phẩm đưa ra (*theo các chuẩn*)
 - Có thể ở mức chung cho nhiều dự án
 - Hay cụ thể hoá cho 1 dự án cụ thể

Các phương pháp - *methods*



- Cách làm cụ thể để xây dựng phần mềm
- Mỗi công đoạn có phương pháp riêng:
 - **Phân tích** (*xác định, đặc tả yêu cầu*)
 - **Thiết kế** (*đặc tả kiến trúc, giao diện, dữ liệu, thủ tục*)
 - **Lập trình** (*cấu trúc, hướng đối tượng*)
 - **Kiểm thử** (*hộp đen, hộp trắng, áp lực, hồi quy, luồng sợi*)
 - **Quản lý dự án** (*PERT, GANTT, COCOMO*)

Các phương pháp - *methods*







■ Nội dung phương pháp thường bao gồm:

- Các phân tử mô hình: mô tả khái niệm
- Các ký pháp: đặc tả phần tử
- Các quy tắc: liên kết các phân tử mô hình
- Quy trình xây dựng: trình tự tạo 1 mô hình
- Lời khuyên, cách dùng: cho mô hình tốt và dùng nó

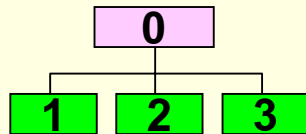
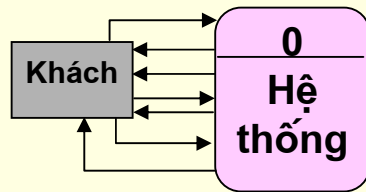
Các phương pháp *methods*



Mô hình (p.pháp) luồng dữ liệu

Khái niệm	Ký pháp	Quy tắc	Quy trình
tác nhân		<ul style="list-style-type: none"> - Tiến trình I là duy nhất - Các luồng vào 1 tiến trình phải # các luồng ra 	<ul style="list-style-type: none"> ▪ Vẽ luồng dữ liệu mức 0 từ mô hình nghiệp vụ
tiến trình		<ul style="list-style-type: none"> - Các luồng vào 1 tiến trình đủ để tạo ra các luồng ra 	<ul style="list-style-type: none"> ▪ Vẽ biểu đồ mức i: từ mỗi tiến trình mức i-1 chưa là cơ sở
kho dữ liệu		<ul style="list-style-type: none"> - Không có các luồng dữ liệu sau: <ul style="list-style-type: none"> <input type="checkbox"/> Kho → kho <input type="checkbox"/> Kho → tác nhân • ... 	
luồng dữ liệu			

Ví dụ: Biểu đồ luồng dữ liệu mức 0 bài toán trông gửi xe

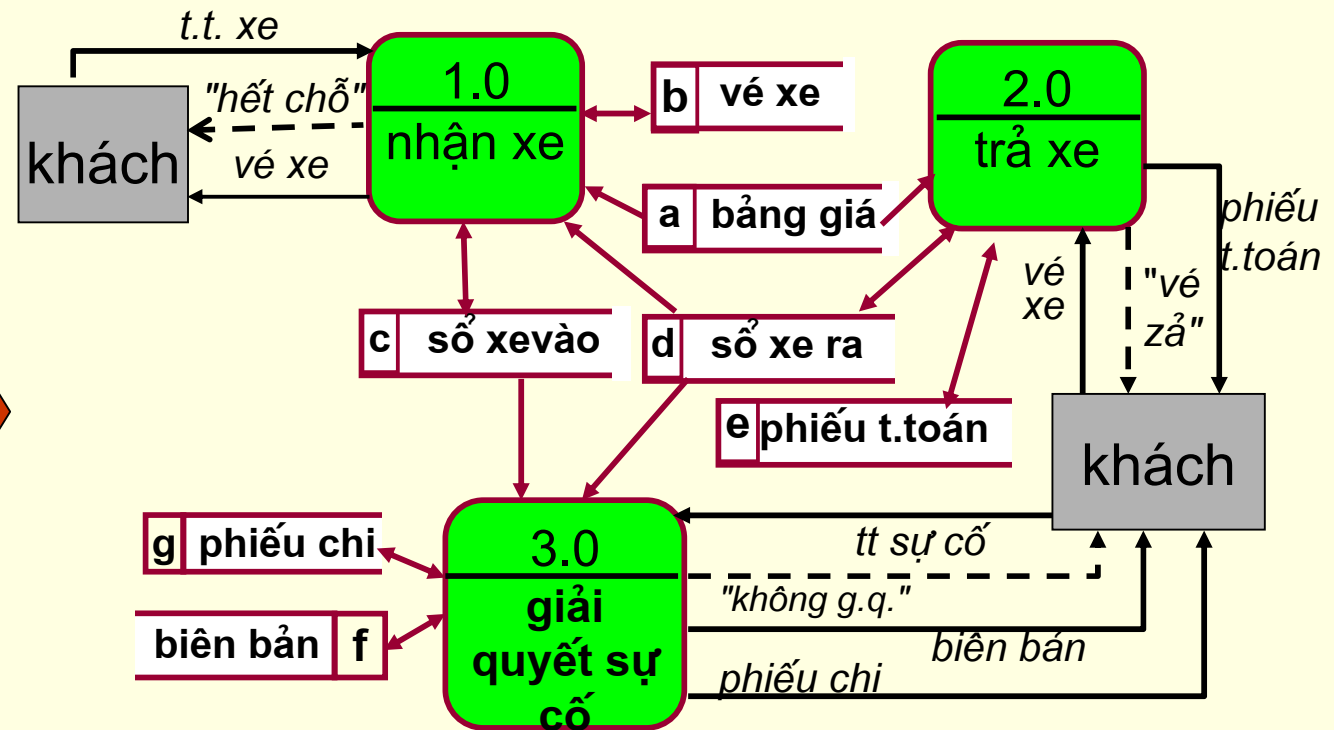


DS thực thể

a.	XXXX
b.	XXXX
c.	XXXX
d.	XXXX
e.	XXXX
f.	XXXX

	a	b	c	d	e	f
1	R	C	U	R		
2	R	R		U	C	
3			R	R		C

Mô hình nghiệp vụ



nét đỏ là phần thay thế hay thêm vào

Các công cụ - *tools*

- Trợ giúp tự động/bán tự động phương pháp
- **Computer Aided Software Engineering** □ **CASE**
các công cụ trợ giúp các công đoạn khác nhau
tiến trình phát triển phần mềm
- Các ngôn ngữ lập trình

Ví dụ:

- ◆ công cụ sinh giao diện (*C Builder, ...*)
- ◆ hỗ trợ phân tích, thiết kế (*Rwin, Modeler*
Oracle Designer, Rational Rose,...)
- ◆ trợ giúp lập trình: compiler, debugger
- ◆ trợ giúp quản lý: *project management*

Các công cụ - *tools*



Resource Flow Properties - Resource Flow_1 (Resourc...

Process	Resource
Package item	Stock

General | Condition | Data | Notes | Rules | Version Info

Name: Resource Flow_1

Code: Resource_Flow_1

Comment:

Process: Package item

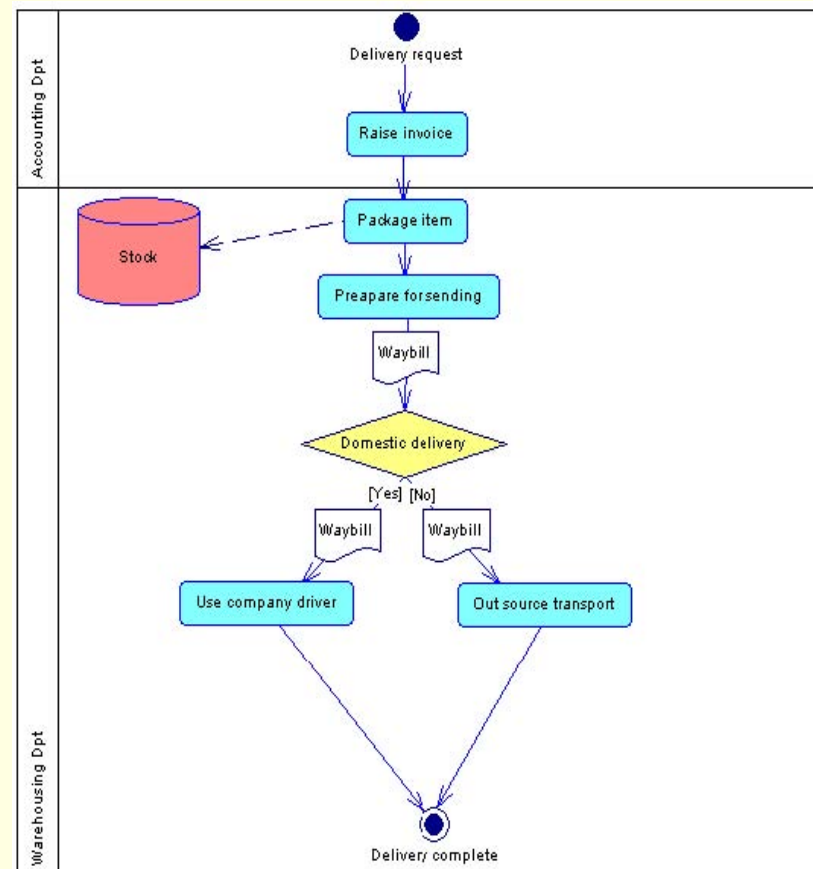
Resource: Stock

Stereotype:

Message format: <Undefined>

Access mode: ☐ Create ☐ Read ☒ Update ☐ Delete

OK Cancel Apply Help



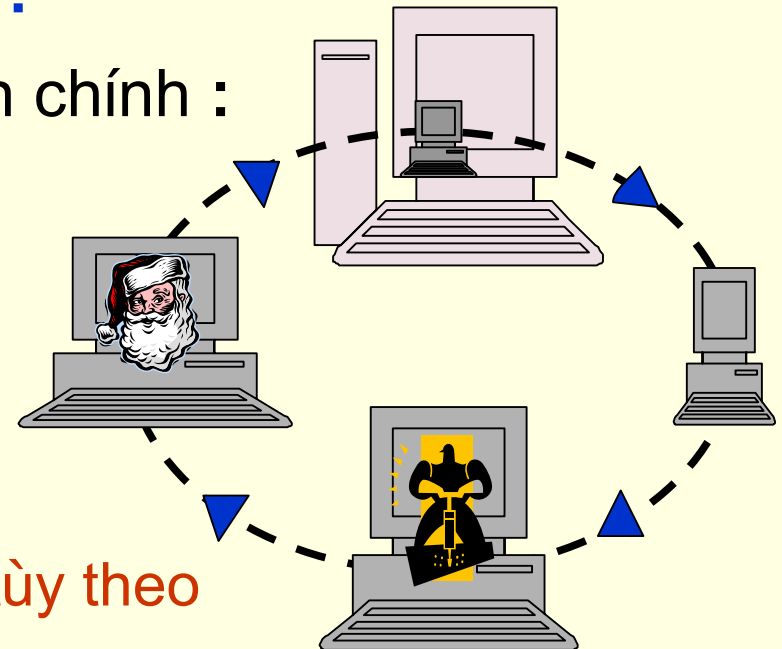
Biểu đồ mô hình

Vòng đời phát triển hệ thống phần mềm

(*systems development life cycle - SDCL*)

- Là các hoạt động **từ khi được đặt hàng, phát triển, sử dụng đến khi bị loại bỏ nó**
- **Vòng đời** chia thành các giai đoạn chính :
 - **xác định yêu cầu,**
 - **triển khai,**
 - **kiểm thử,**
 - **Vận hành, bảo trì và lặp lại**

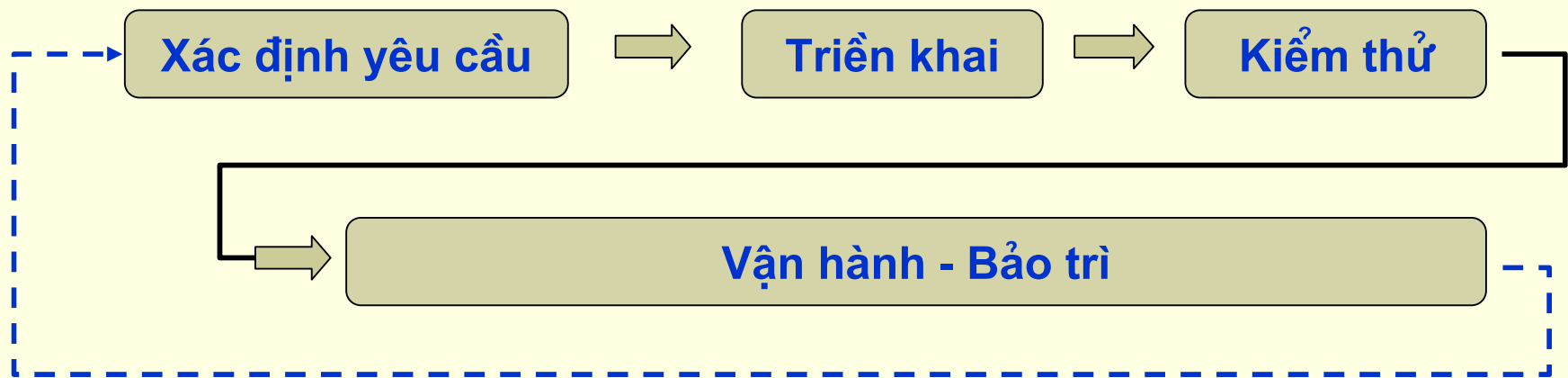
Phạm vi, thứ tự, nội dung khác nhau tùy theo từng sản phẩm và dự án



Vòng đời phát triển phần mềm



- Tùy mô hình áp dụng, phân chia các pha, các bước có thể khác nhau : từ 3 đến 20 bước



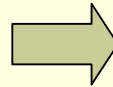
Các pha chính của vòng đời

Các bước chung nhất phát triển



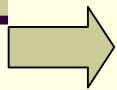
■ Xác định yêu cầu

- phân tích hệ thống
- lập kế hoạch
- phân tích yêu cầu
- đặc tả yêu cầu



■ Phát triển

- Thiết kế
- Mã hoá
- Kiểm thử
- Làm tài liệu



■ Tiến hóa

- sửa lỗi
- thích nghi
- nâng cao
- bổ sung



Xác định yêu cầu

Xác định: *Hệ thống làm gì?*

Những ràng buộc gì nó cần tuân thủ?

■ Phân tích hệ thống

- ◆ vai trò phần mềm trong hệ thống
- ◆ phân tích & chọn phương án khả thi

■ Lập kế hoạch

ước lượng, lập lịch, tổ chức, bước đi, sản phẩm

■ Phân tích yêu cầu

- các yêu cầu cụ thể (chức năng, ràng buộc)
- đặc tả yêu cầu (kiến trúc, giao diện, xử lý, dữ liệu)

Phát triển



Tạo ra phần mềm như thế nào?

■ Thiết kế (*design*)

dịch yêu cầu thành bản thiết kế HT như nó tồn tại
(*môi trường, kiến trúc, dữ liệu, xử lý, giao diện*)

■ Mã hóa (*coding*)

Chuyển thiết kế thành chương trình (*của 1 ngôn ngữ*)

■ Kiểm thử (*testing*)

sửa lỗi, hoàn thiện chương trình (*kiểm thử đơn vị, tích hợp, hệ thống, chấp nhận*), làm tài liệu

Vận hành, bảo trì



Hoàn thiện hệ thống sau khi đưa vào hoạt động?

- **Sửa lỗi:** để vận hành thông suốt
- **Thích nghi:** với môi trường (kỹ thuật, nghiệp vụ) đã thay đổi để hoạt động hiệu quả
- **Nâng cao:** Hoàn thiện chức năng, phát triển dự phòng
- **Thêm mới:** thêm các chức năng mới

Tiến trình phần mềm



- **Tiến trình phần mềm (software process)** là một tập các hoạt động có cấu trúc nhằm phát triển và tiến hóa 1 phần mềm
- Một tiến trình cụ thể phải trả lời được các câu hỏi: *làm gì? khi nào? ai làm? như thế nào? bằng gì? ở đâu? kết quả? tiêu chí đánh giá?*

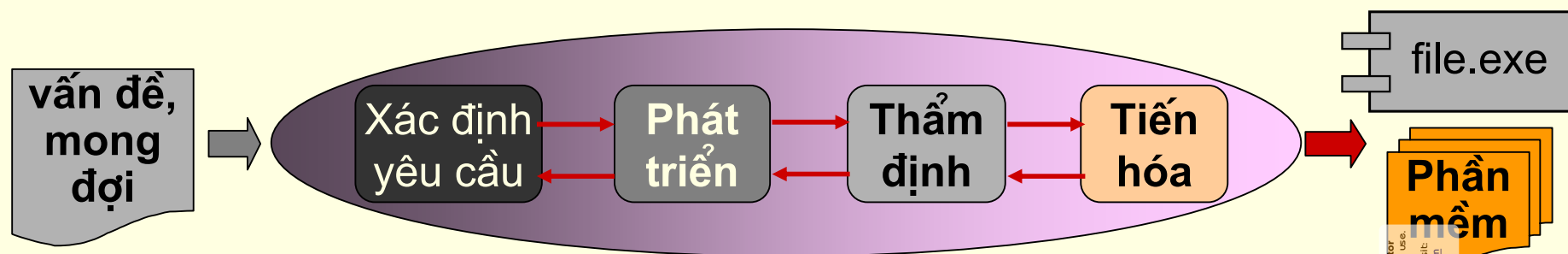
Đặc trưng

- ◆ Gắn với mỗi dự án
- ◆ Có cấu trúc xác định (*công việc, trình tự, công cụ, phương pháp*)
- ◆ Sản phẩm cuối cùng là phần mềm bàn giao

Tiến trình phần mềm

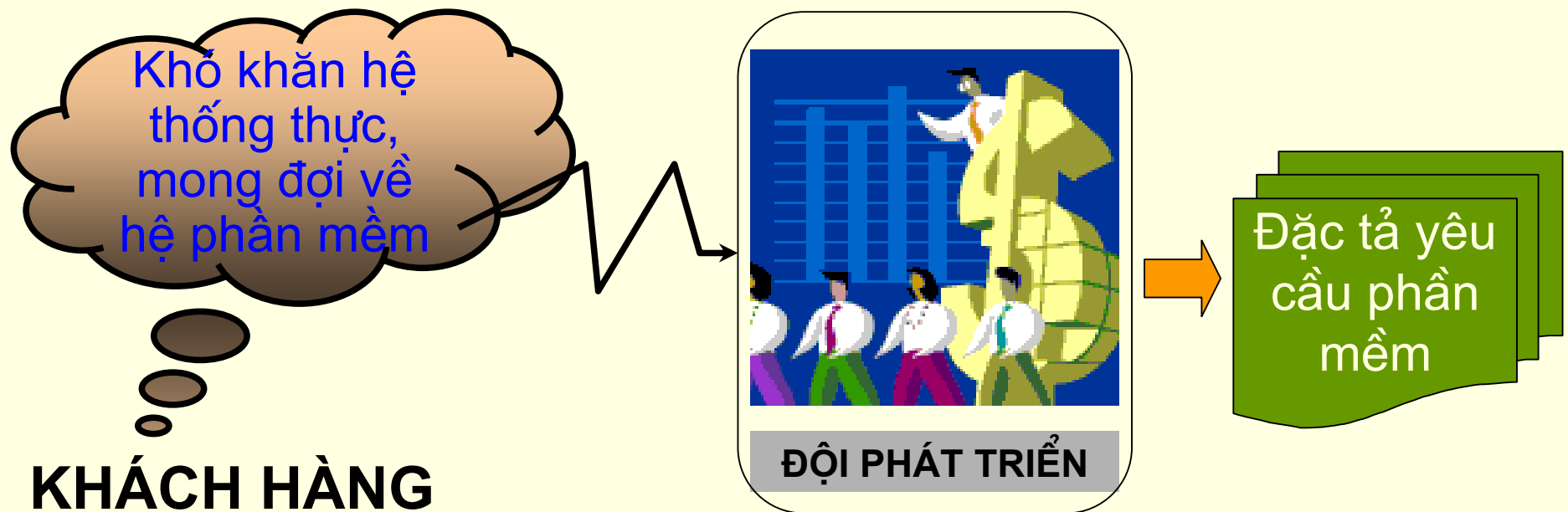
Các hoạt động chính của mọi tiến trình

- **Xác định yêu cầu:** định rõ yêu cầu sản phẩm
- **Phát triển:** tạo ra sản phẩm
- **Thẩm định:** phần mềm đáp ứng được yêu cầu không?
- **Tiến hóa phần mềm:** thay đổi nhằm đáp ứng yêu cầu thay đổi (người dùng, môi trường)



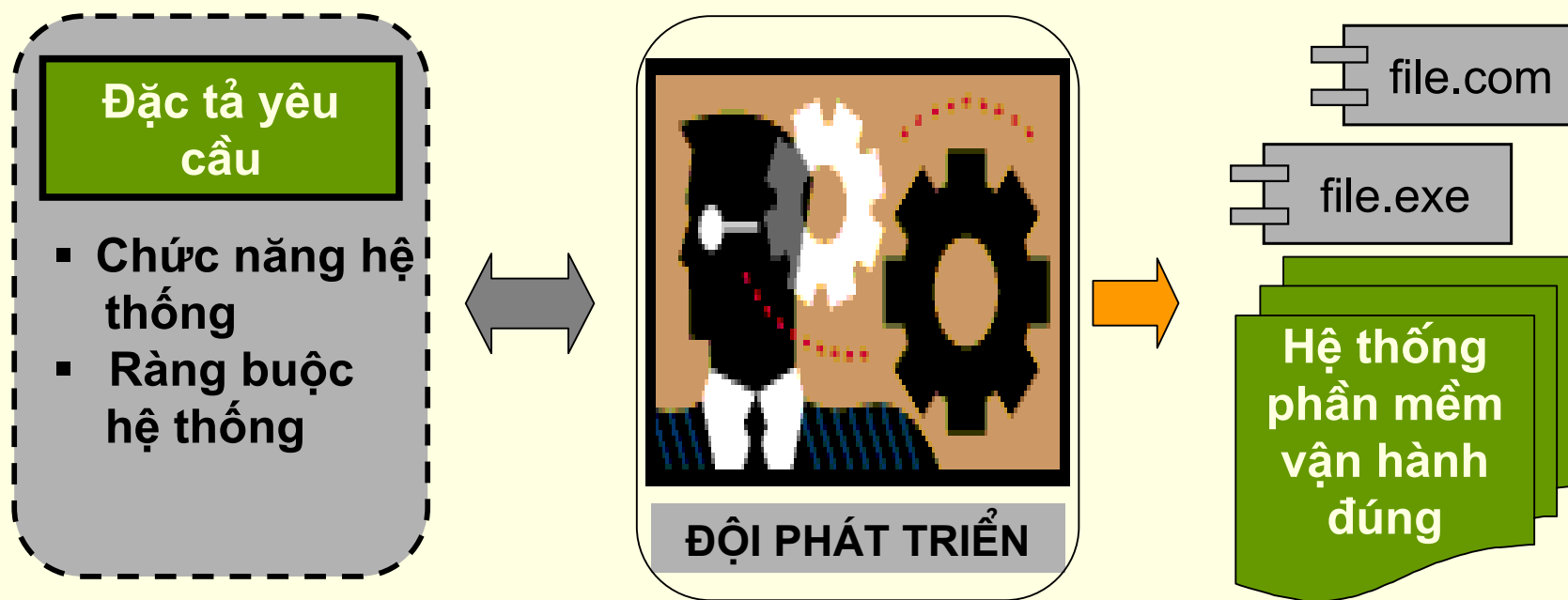
Tiến trình phần mềm

1. Xác định yêu cầu (*Requirements*)



Tiến trình phần mềm

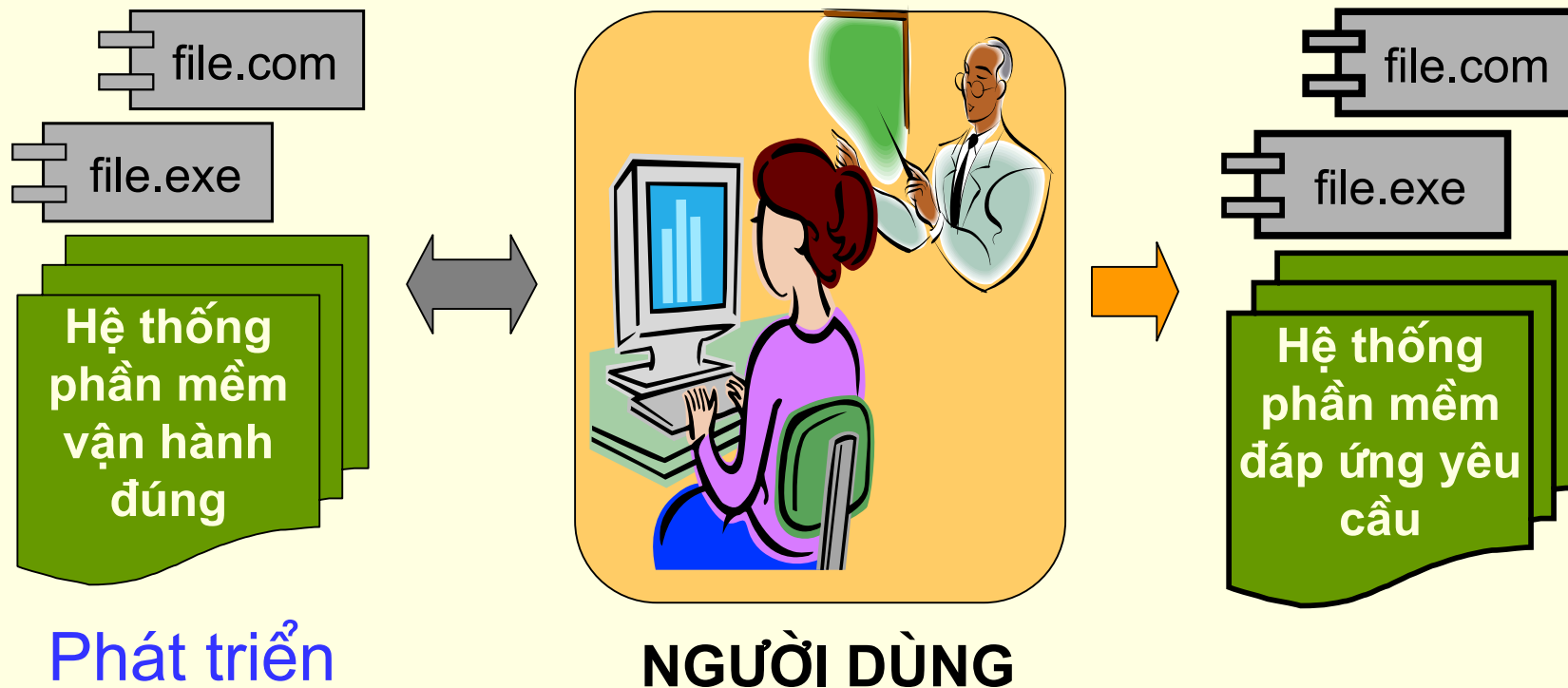
2. Phát triển phần mềm (*Development*)



Xác định yêu cầu

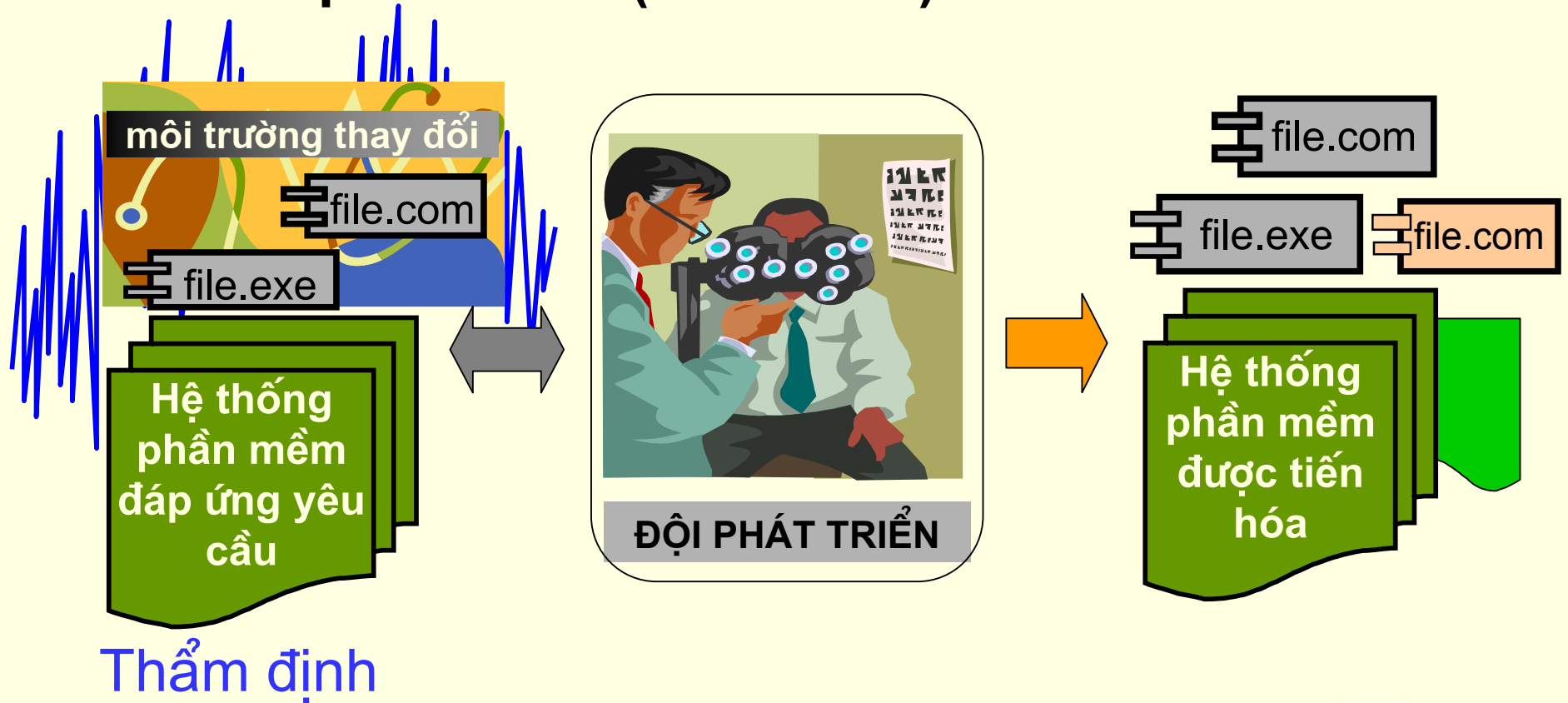
Tiến trình phần mềm

3. Thẩm định phần mềm (*Validation*)



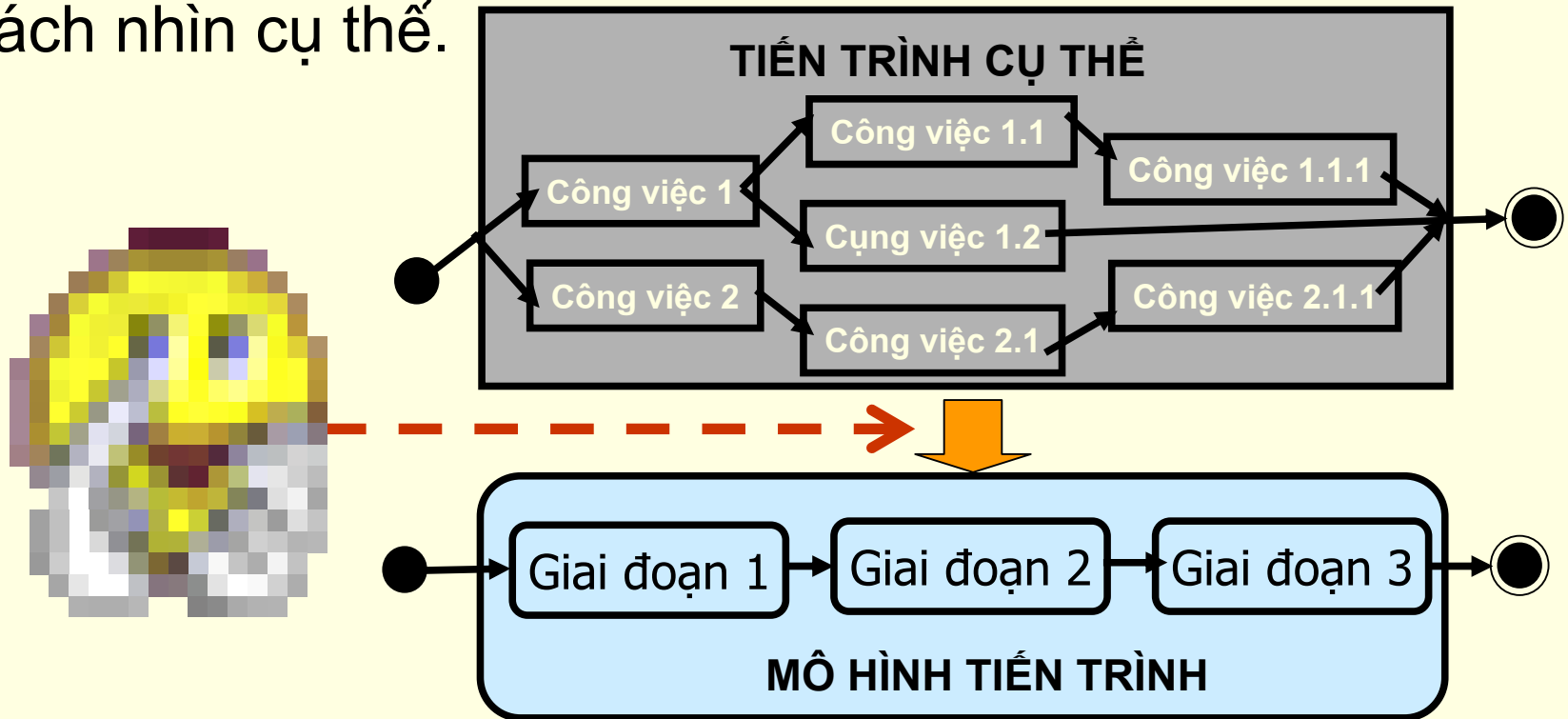
Tiến trình phần mềm

4. Tiến hóa phần mềm (*Evolution*)



Mô hình tiến trình phần mềm

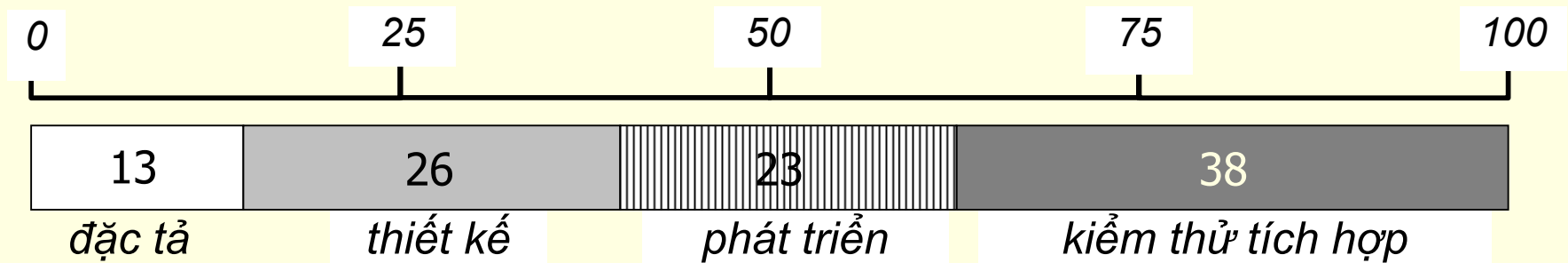
- **Mô hình tiến trình** (*software process model*) là cách biểu diễn trừu tượng tiến trình phần mềm theo cách nhìn cụ thể.



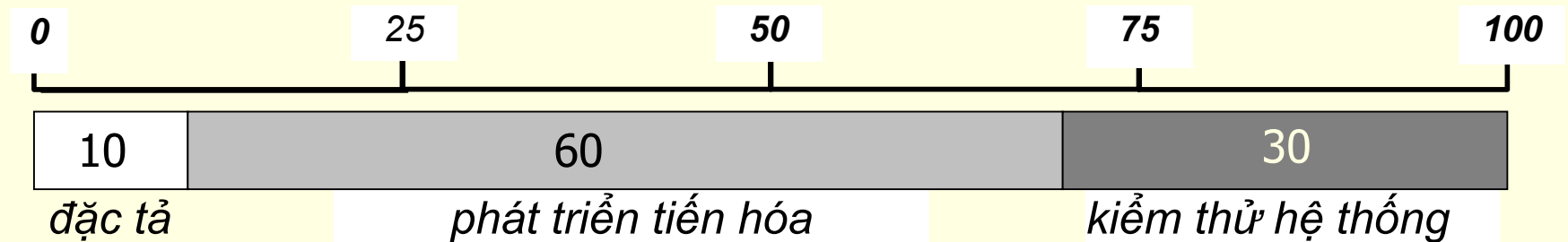
Chi phí của kỹ nghệ phần mềm

- **Chi phí kỹ nghệ phần mềm:** là các khoản chi liên quan đến toàn bộ sự phát triển
- **Chi phí phụ thuộc:**
 - Loại hệ thống (*đơn giản, phức tạp*)
 - Yêu cầu đặt ra (*nhiều, ít, cao, thấp*)
 - Mức độ hoàn thiện (*hiệu năng, độ tin cậy, an toàn,...*)
 - Năng lực của tổ chức (*nhân lực, công nghệ, công cụ, kỹ năng có được*)
 - Loại tiến trình sử dụng

Chi phí của kỹ nghệ phần mềm

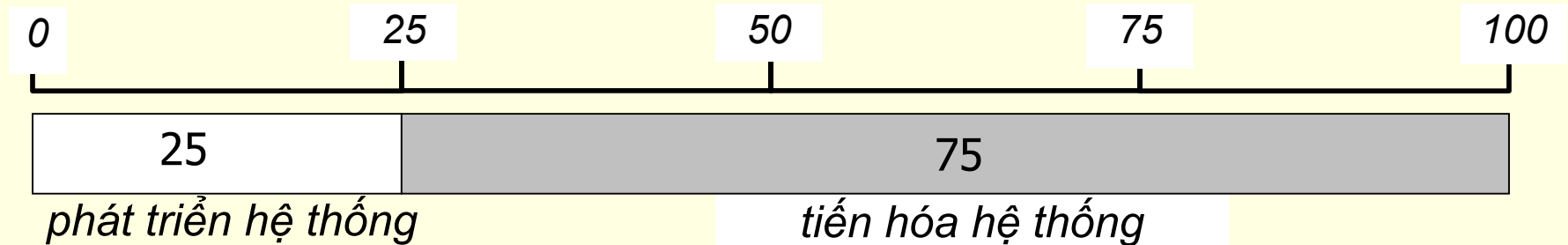


Phân bố chi phí phát triển (nói chung)

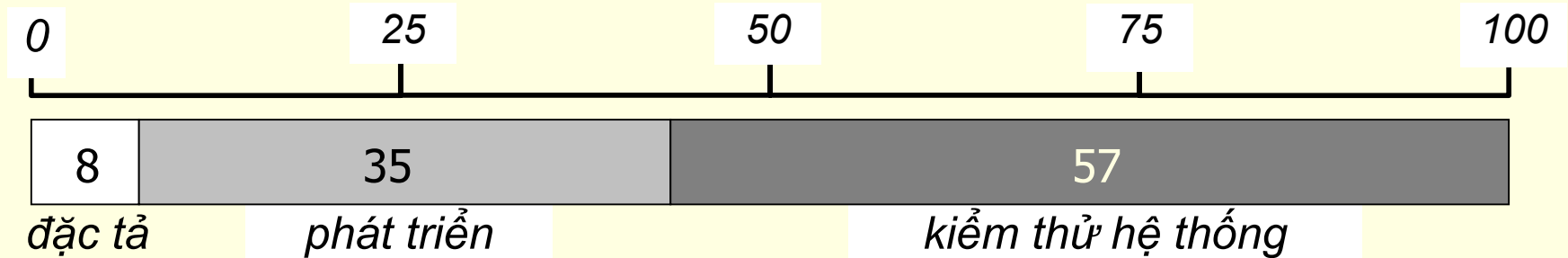


Chi phí phát triển tiến hóa

Chi phí của kỹ nghệ phần mềm



Chi phí hệ có vòng đời dài



chi phí phát triển sản phẩm trực tuyến

Quy mô phần mềm



- Kích cỡ phần mềm ảnh hưởng lớn đến chi phí
- Kích cỡ thường đo bằng số dòng lệnh
- Kích cỡ phần mềm phụ thuộc:
 - Bài toán
 - Thiết kế
 - Ngôn ngữ lập trình (& trình độ người lập)
- Với cùng thiết kế, dùng ngôn ngữ khác nhau cho kích cỡ khác nhau

Số đo ngôn ngữ

- Mỗi ngôn ngữ có 1 năng lực thể hiện
- Chọn ngôn ngữ tốt sẽ giảm kích cỡ

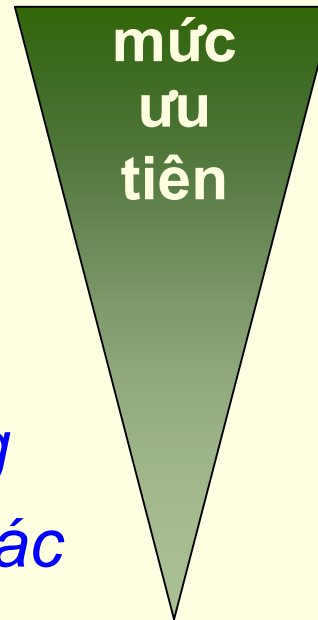
Ngôn ngữ	Dòng lệnh/chức năng
Assembly	320
C	128
FORTRAN 77	105
COBOL 85	91
Ada 83	71
C++	56
Ada 95	55
Java	55
Visual Basic	35
SQL	2

Phần mềm kỹ nghệ tốt



- Phần mềm kỹ nghệ tốt được xét theo quan điểm nhà phát triển
- Thuộc tính phần mềm cần có:

- *Tính bảo trì được*
- *Tính đáng tin cậy*
- *Tính hiệu quả*
- *Tính tiện dụng*
- *Giá cả phải chăng*
- *Một số tiêu chí khác*



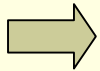
Tính bảo trì được



"Phần mềm luôn có yêu cầu sửa đổi"

Để sửa đổi (bảo trì) được phần mềm cần:

- ◆ **Kiến trúc tốt:** kết dính chặt, ghép nối lỏng (dễ đọc, dễ sửa, dễ phát triển, ảnh hưởng cục bộ)
- ◆ cài đặt bằng **ngôn ngữ bậc cao** (dễ đọc, viết nhanh)
- ◆ **tài liệu đầy đủ, tốt** (dễ theo dõi, dễ hiểu, có cơ sở sửa,..)



- ◆ Có tuổi thọ cao, phục vụ nhiều
- ◆ Chi phí bảo trì thấp → cho hiệu quả

Đánh tin cậy



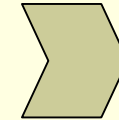
- Phần mềm có ít khiếm khuyết về:
 - ◆ lỗi lập trình
 - ◆ lỗi phân tích, thiết kế: sai, thiếu chức năng,
 - ◆ hoạt động không hiệu quả
- Đáp ứng được nhu cầu người sử dụng
 - ◆ Đảm bảo các chức năng cần
 - ◆ Ổn định, thời gian làm việc không lỗi lớn
 - ◆ Cho kết quả xác đáng

Tính hiệu quả



■ Không sử dụng lãng phí tài nguyên phần cứng

- ◆ Đòi hỏi bộ nhớ lớn
- ◆ Đòi hỏi tốc độ cao
- ◆ Chiếm không gian đĩa lớn,...



**Cần tối ưu
hợp lý**

Để tối ưu hóa cần:

- ◆ dùng ngôn ngữ bậc thấp
- ◆ truy cập trực tiếp đến thiết bị
- ◆ sử dụng thuật toán viết gọn

➔ **dễ dẫn tới khó bảo trì, giá thành cao.**

Tiện dụng, tinh vi



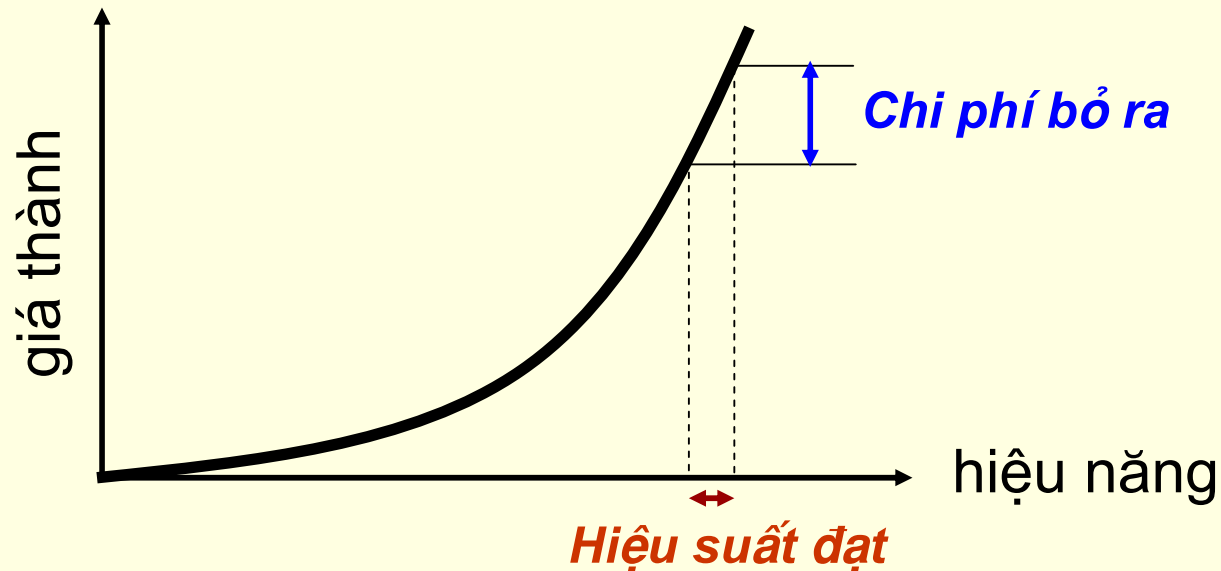
- Giao diện phù hợp với trình độ của người dùng
- Học nhanh, nhớ lâu, dễ thao tác
- Có đủ tài liệu & nhiều tiện ích, trợ giúp thông minh

➔ ***Giao diện quyết định thành công của sản phẩm***

Giá cả hợp lý

Cần cân đối, thỏa hiệp giữa các yêu cầu

- Khó tối ưu đồng thời các thuộc tính
giá cả, hiệu quả >>> <<< dễ bảo trì, sử dụng, sự tinh vi
- Chi phí cho tối ưu không tuyến tính (là hàm mũ)



Kỹ nghệ hệ thống (System Engineering)

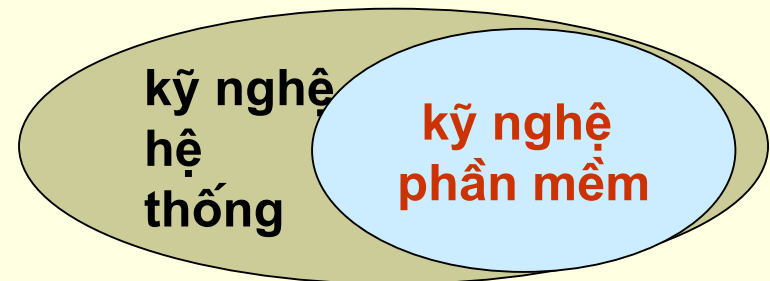


■ **Kỹ nghệ hệ thống** là toàn bộ công việc phát triển hệ thống dựa trên máy tính (*computer based system*):

- phần cứng
- phần mềm
- tổ chức, quản lý

■ **Kỹ nghệ hệ thống** bao gồm:

- đặc tả
- thiết kế
- triển khai
- tích hợp
- thẩm định
- bảo trì



Kỹ nghệ hệ thống



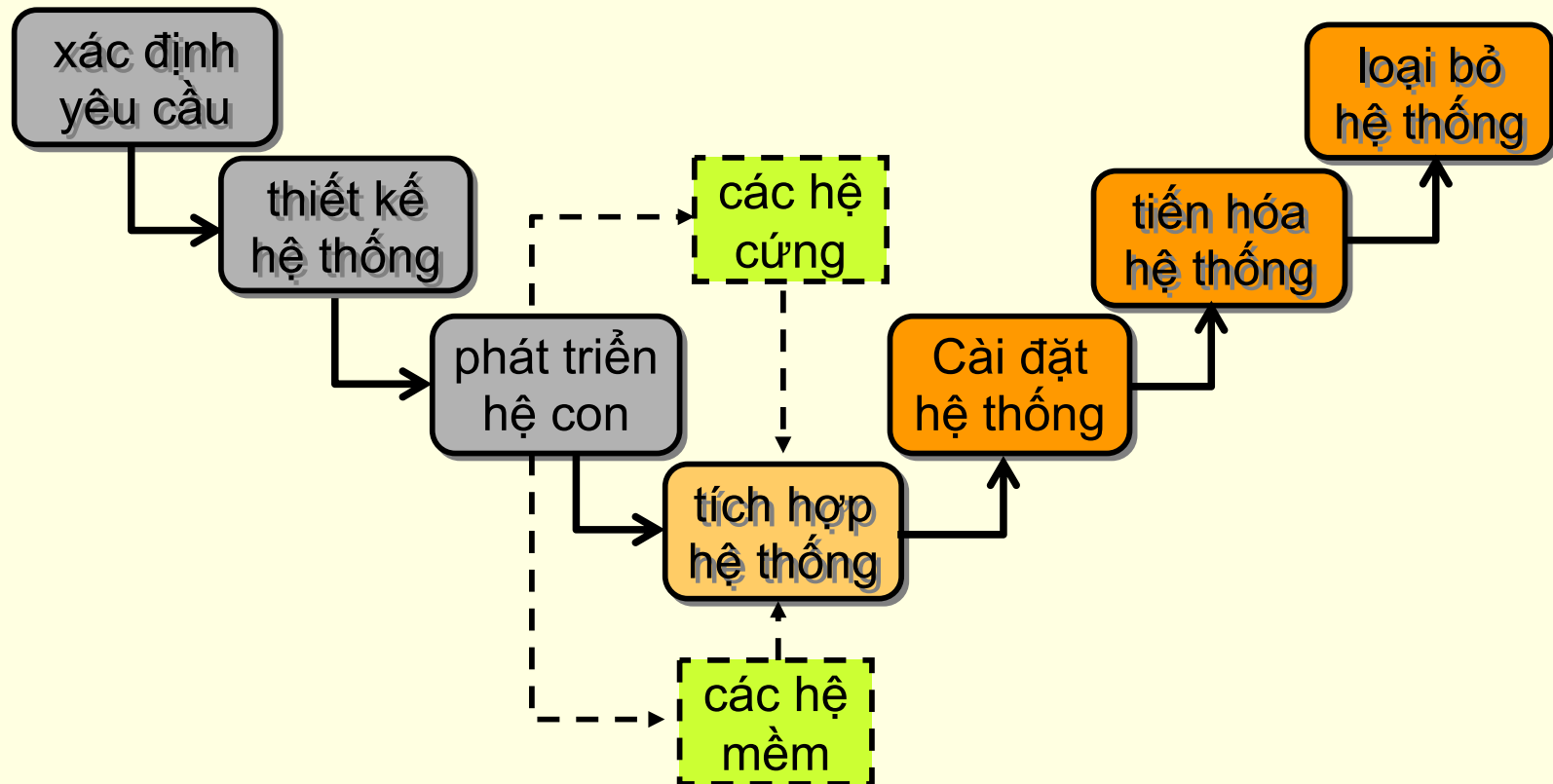
- Đặc tính nổi trội của hệ thống dựa trên máy tính
 - Thuộc tính chức năng: *điều khiển 1 hệ*
 - Thuộc tính phi chức năng: *độ tin cậy, an toàn*

Mà:

- ♦ từng bộ phận không thể có được
 - ♦ quyết định đến toàn hệ thống
- Kỹ nghệ hệ thống liên quan đến:
 - áp dụng các nguyên lý tích hợp
 - giảm chi phí làm lại toàn hệ thống

Kỹ nghệ hệ thống

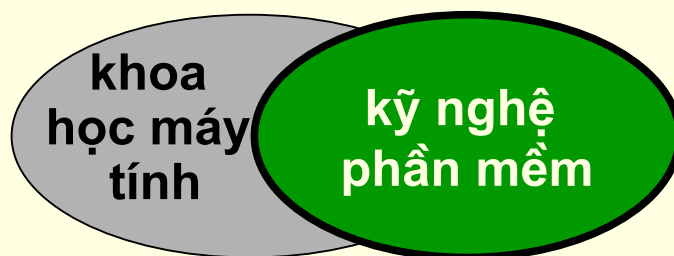
- Tiến trình kỹ nghệ hệ thống (*system engineering process*)



Khái niệm liên quan - Khoa học máy tính (Computer Science)

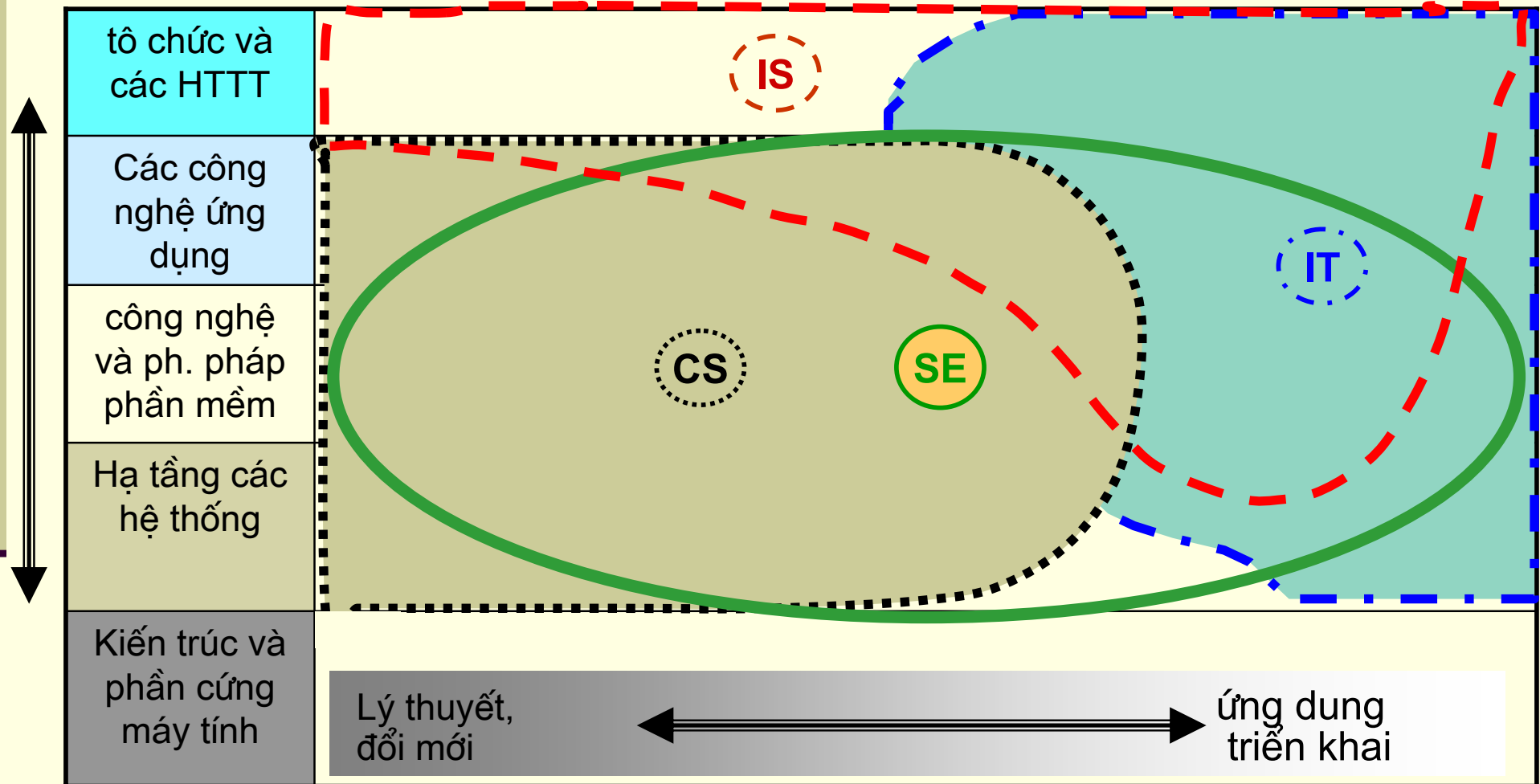


- **Khoa học máy tính** liên quan đến những vấn đề lý thuyết và nền tảng cho máy tính và phần mềm
- **Kỹ nghệ phần mềm** liên quan đến thực tiễn của việc phát triển và xuất ra các phần mềm hiệu quả
- ❖ **Lý thuyết khoa học máy tính còn chưa đủ cho các hoạt động của kỹ nghệ phần mềm**



Các lĩnh vực tính toán

(computing curricula 11/2004 -ACM, AIS, IEEE)



Phân biệt các lĩnh vực tính toán



- Mỗi lĩnh vực tính toán được giới hạn bằng một vùng với biên có màu sau:

◆ **CS** □ Khoa học máy tính:

◆ **IS** □ Hệ thống thông tin - - - -

◆ **IT** □ Công nghệ thông tin - . - . -

◆ **SE** □ Kỹ nghệ phần mềm —————

- **Kỹ nghệ phần mềm chiếm phần lớn ở trung tâm** -
Đó là lý do nó là ngành phức tạp và quan trọng

Câu hỏi ôn tập



- 1) Định nghĩa kỹ nghệ phần mềm -SE?
- 2) Các giai đoạn lịch sử phát triển SE? đặc trưng của mỗi giai đoạn (sản phẩm, phương pháp, công cụ, quản lý và tổ chức làm việc)?
- 3) Giải thích nội dung các yếu tố cơ bản trong SE? Lấy ví dụ minh họa về phương pháp, công cụ?
- 4) Tiến trình phần mềm là cái gì? Các bước chung nhất của tiến trình là gì?
- 5) Mô hình tiến trình là gì? Các mô hình có thể nhận được từ những cách nhìn nhận nào?

Câu hỏi ôn tập



- 6) Trình bày một số mô hình tiến trình cơ bản (nội dung, ưu, nhược, thích hợp khi nào)?
- 7) Chi phí của SE gồm những thành phần nào? nó phụ thuộc vào những cái gì?
- 8) Kỹ nghệ hệ thống là gì? Quan hệ với kỹ nghệ phần mềm ra sao?
- 9) Các thuộc tính của phần mềm được kỹ nghệ tốt?
- 10) Khoa học máy tính là gì? Quan hệ giữa nó và kỹ nghệ phần mềm như thế nào?

Câu hỏi và thảo luận

