UIT-HCM

GIỚI THIỆU TỔNG QUAN

GV. THS. TRẦN ANH DŨNG

GIỚI THIỆU MÔN HỌC

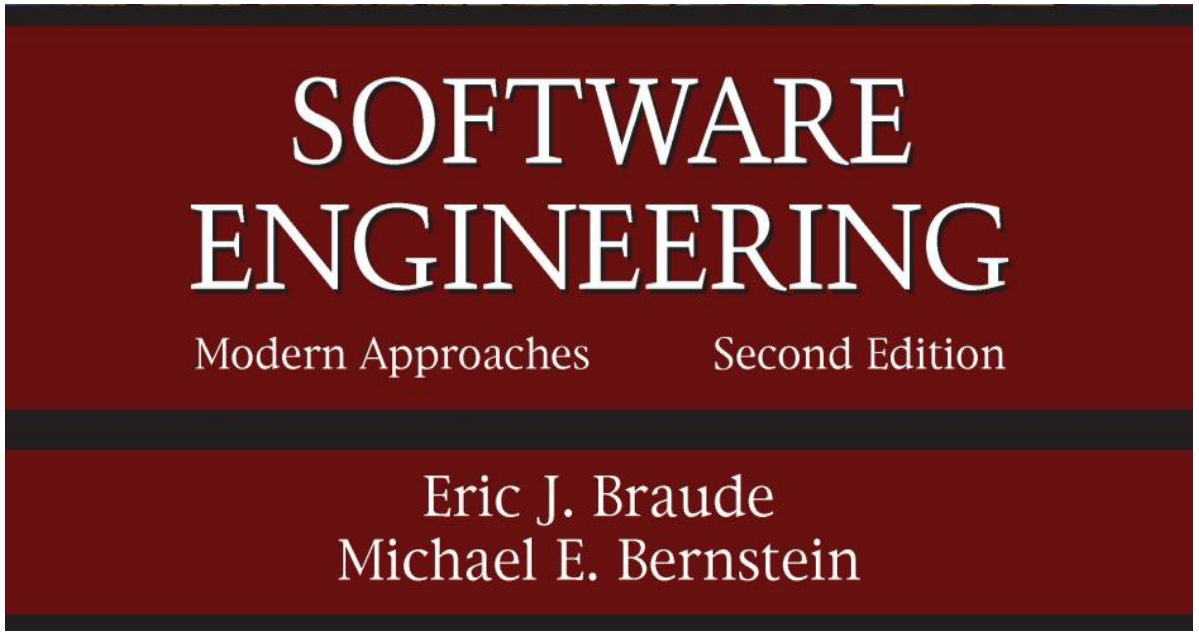
- Tên môn: Công nghệ phần mềm chuyên sâu
- 3 tín chỉ LT + 1 tín chỉ thực hành (HT2)
- Thông tin giảng viên:
  - ThS. Trần Anh Dũng
  - Email: dungta@uit.edu.vn

GIỚI THIỆU MÔN HỌC

- Đánh giá môn học:
  - Quá trình (Bài tập + Seminar + Đồ án) : 50%
  - Cuối kỳ (thi lý thuyết) : 50%

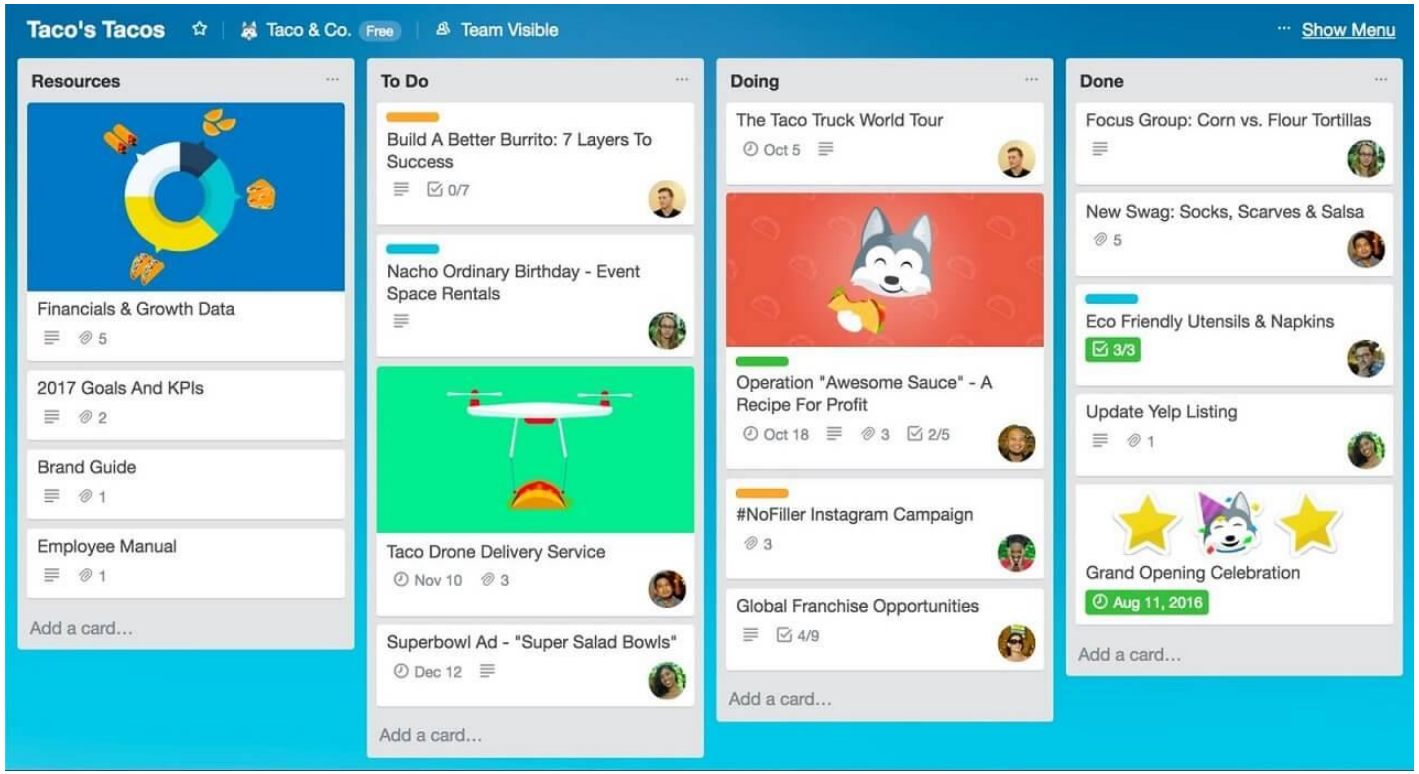
GIỚI THIỆU MÔN HỌC

- Seminar:



GIỚI THIỆU MÔN HỌC

- Đồ án:
  - Tuân thủ quy trình làm phần mềm
  - Sử dụng các công cụ để quản lý (tài liệu, source code,…)



TÀI LIỆU THAM KHẢO

1. Ian Sommerville, **Software Engineering**, 8th Ed., Pearson Education Limited, Essex, England and Addison-Wesley Publishers, Boston, MA, 2007.
2. Laplante, Phil. **Requirements Engineering for Software and Systems** (1st ed.). Redmond, WA: CRC Press. ISBN 1-42006-467-3, 2009.

NỘI DUNG

- Tổng quan
- Các khái niệm cơ bản
- Vai trò của Công nghệ phần mềm
- Những khó khăn của kỹ nghệ phần mềm

Tổng quan

- Những nước phát triển đều phụ thuộc chủ yếu vào các hệ thống phần mềm.
- Có nhiều hệ thống được kiểm soát bởi phần mềm.
- Xây dựng và bảo trì hệ thống phần mềm một cách hiệu quả là yêu cầu cần thiết đối với nền kinh tế toàn cầu và của từng quốc gia.

Các khái niệm cơ bản

- Phần mềm (Software) là gì?

Các khái niệm cơ bản

❖ Công nghệ phần mềm:

- **Theo Fritz Bauer (1969):** CNPM là sự thiết lập và sử dụng những nguyên tắc công nghệ hợp lý để đạt được những phần mềm có tính kinh tế mà đáng tin cậy và làm việc hiệu quả trên máy thực.
- **Theo Roger S. Pressman:** CNPM là bộ môn tích hợp cả các qui trình, các phương pháp, các công cụ để phát triển phần mềm máy tính

Các khái niệm cơ bản

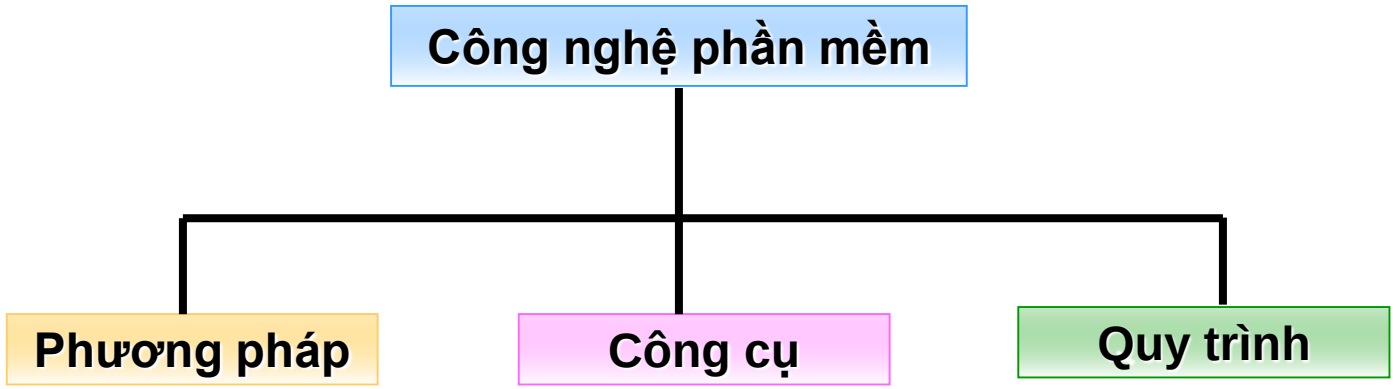
❖ Công nghệ phần mềm:

- **Theo Ian Sommerville:** CNPM là một lĩnh vực mà liên quan đến tất cả các khía cạnh của sản xuất phần mềm từ những giai đoạn đầu của đặc tả hệ thống đến bảo trì hệ thống sau khi nó đã được đưa vào sử dụng.

Các khái niệm cơ bản

❖ Công nghệ phần mềm:

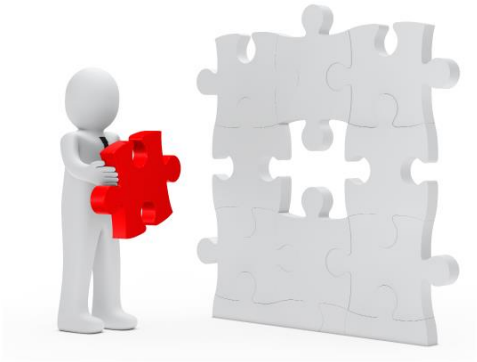
- Công nghệ phần mềm là ngành khoa học nghiên cứu về việc xây dựng các phần mềm có chất lượng cao trong thời gian và chi phí thực hiện hợp lý.



Các khái niệm cơ bản

- Chất lượng phần mềm?

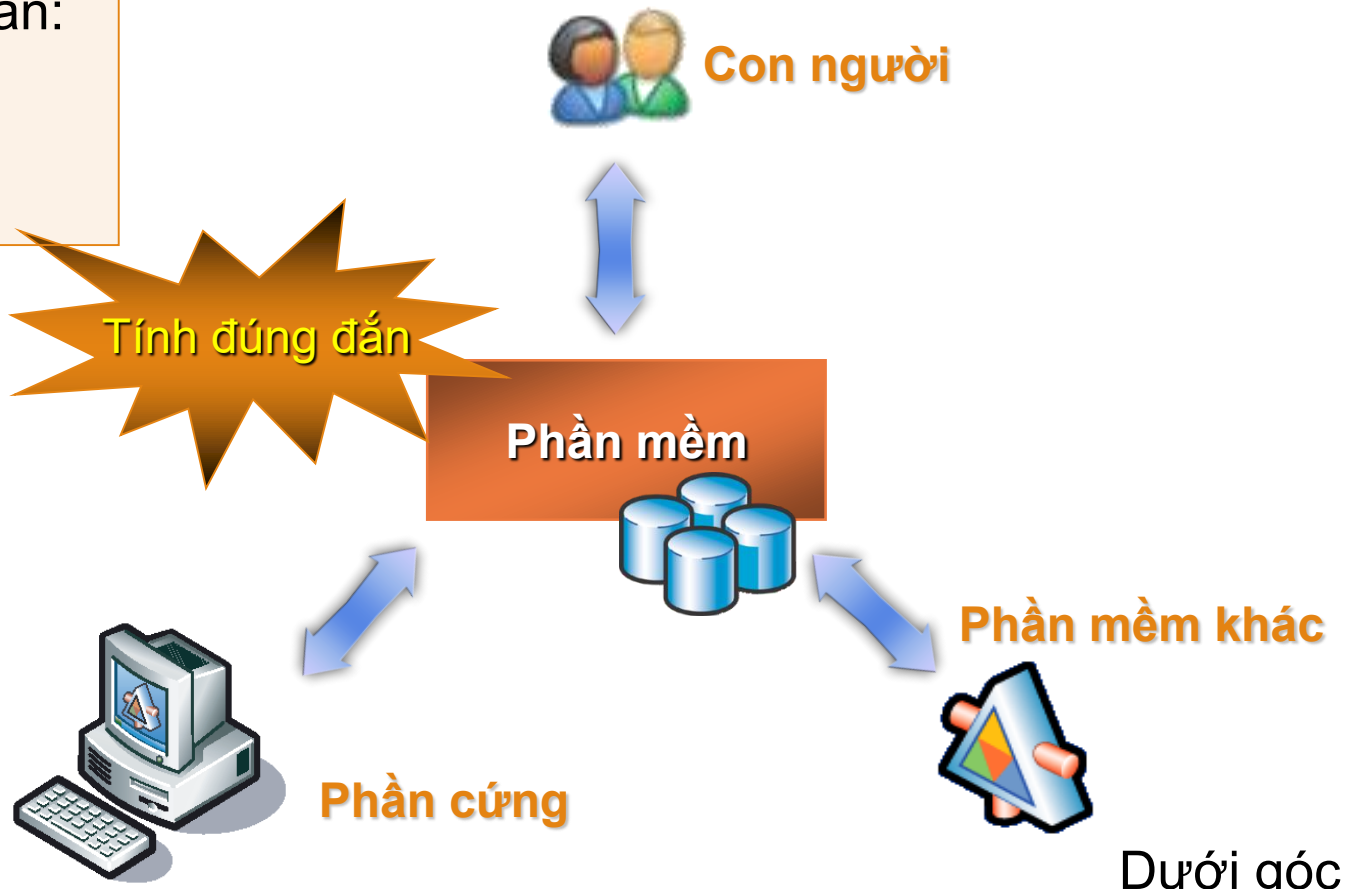
Bảo đảm chất lượng phần mềm **(QA)**



Chất lượng phần mềm

Tính đúng đắn:

- Đầy đủ
- Chính xác



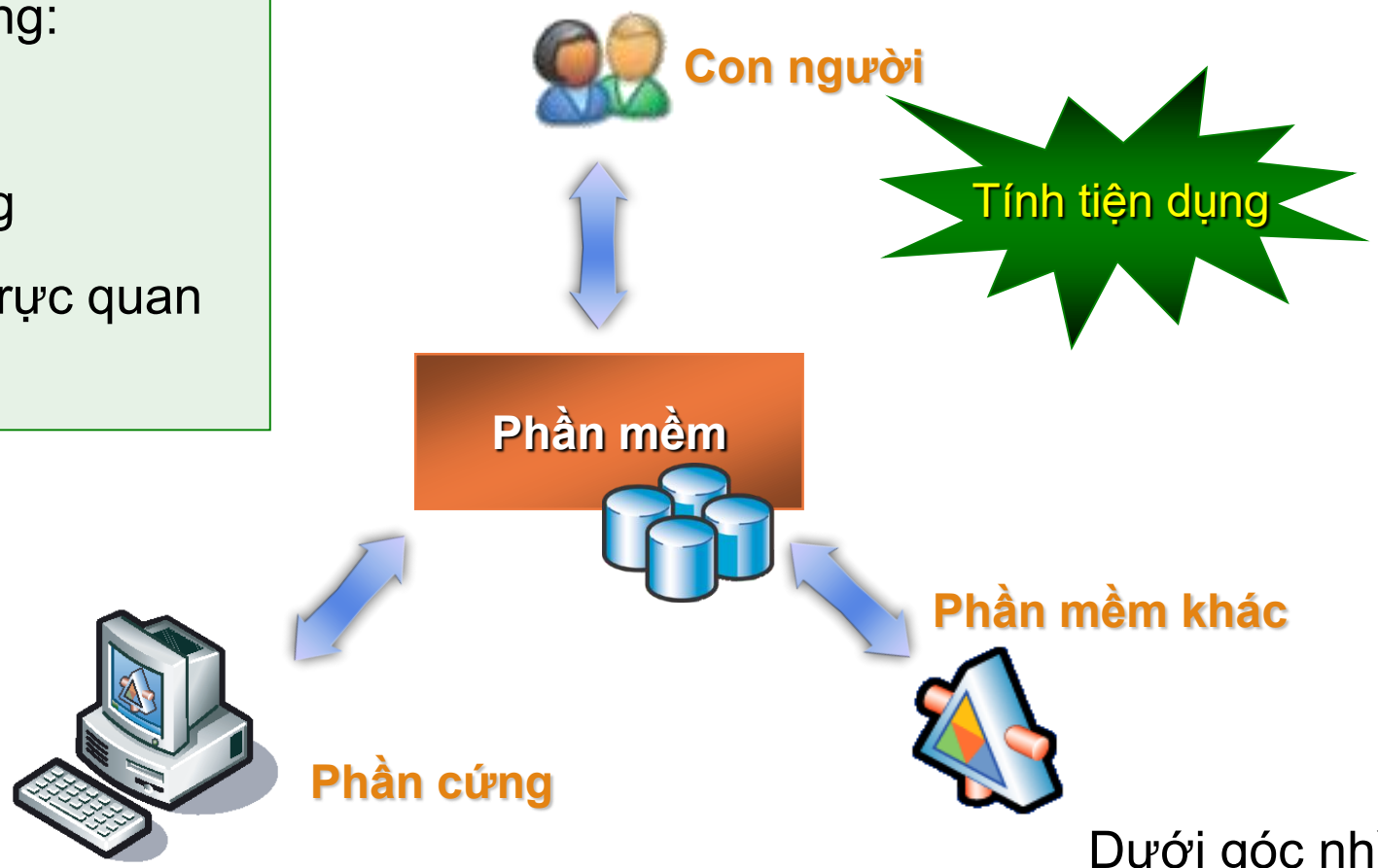
Dưới góc nhìn của Người sử dụng

Chất lượng phần mềm

Tính tiện dụng:

- Dễ học
- Dễ sử dụng
- Giao diện trực quan
- Tự nhiên

Tính đúng đắn

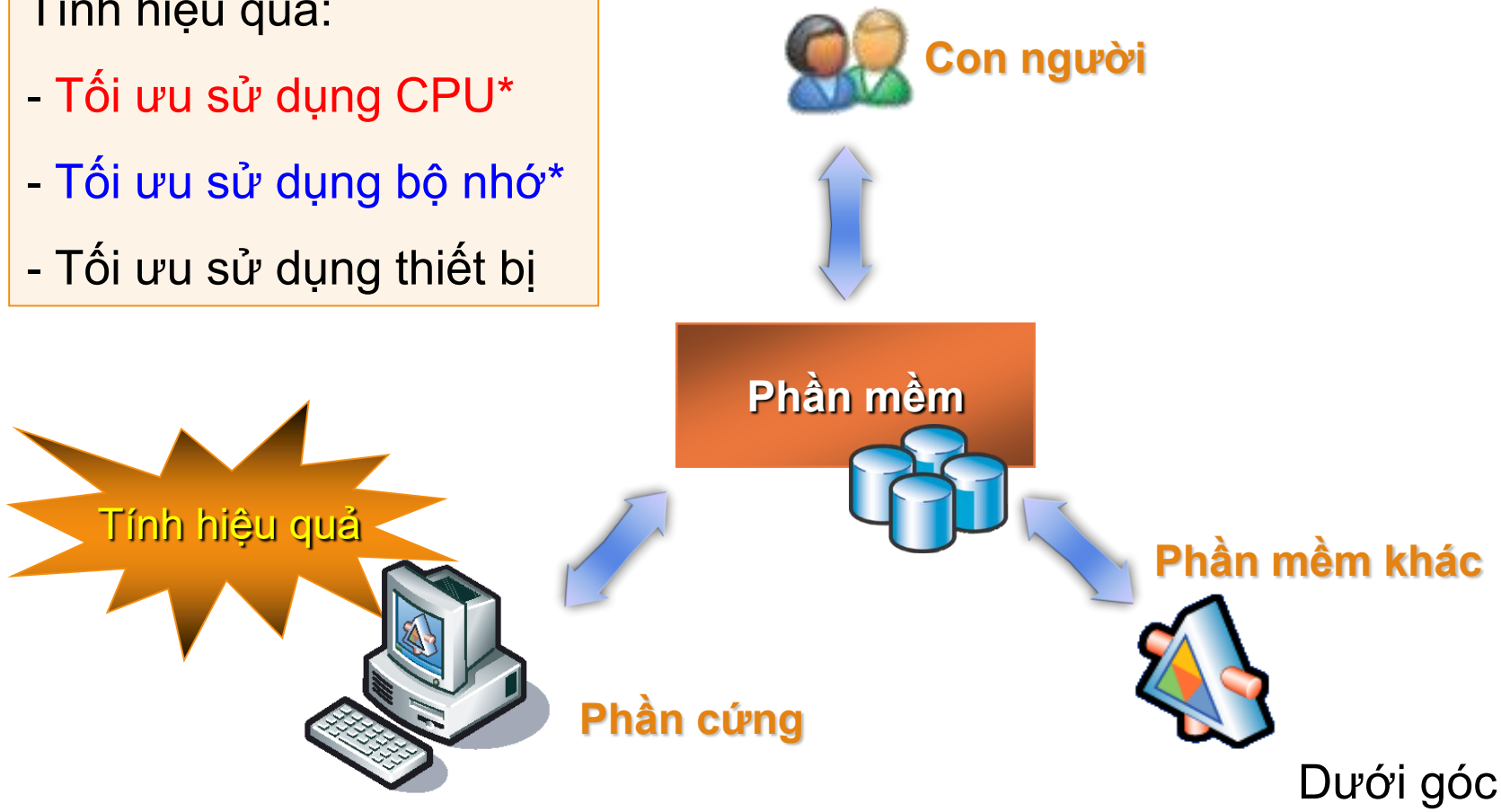


Dưới góc nhìn của Người sử dụng

Chất lượng phần mềm

Tính hiệu quả:

- Tối ưu sử dụng CPU*
- Tối ưu sử dụng bộ nhớ*
- Tối ưu sử dụng thiết bị



Tính đúng đắn

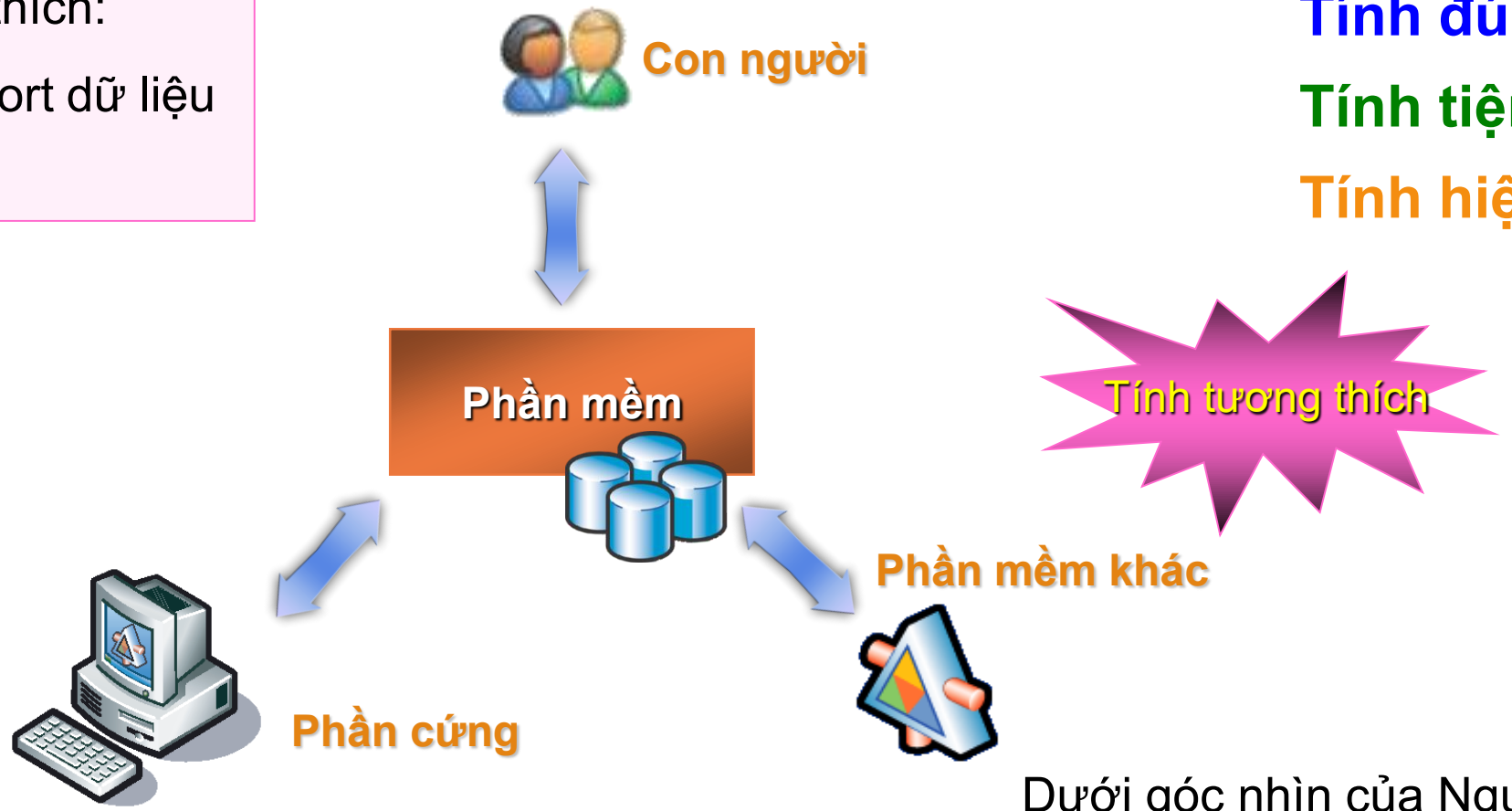
Tính tiện dụng

Dưới góc nhìn của Người sử dụng

Chất lượng phần mềm

Tính tương thích:

- Import/Export dữ liệu
- Tương tác



Tính đúng đắn

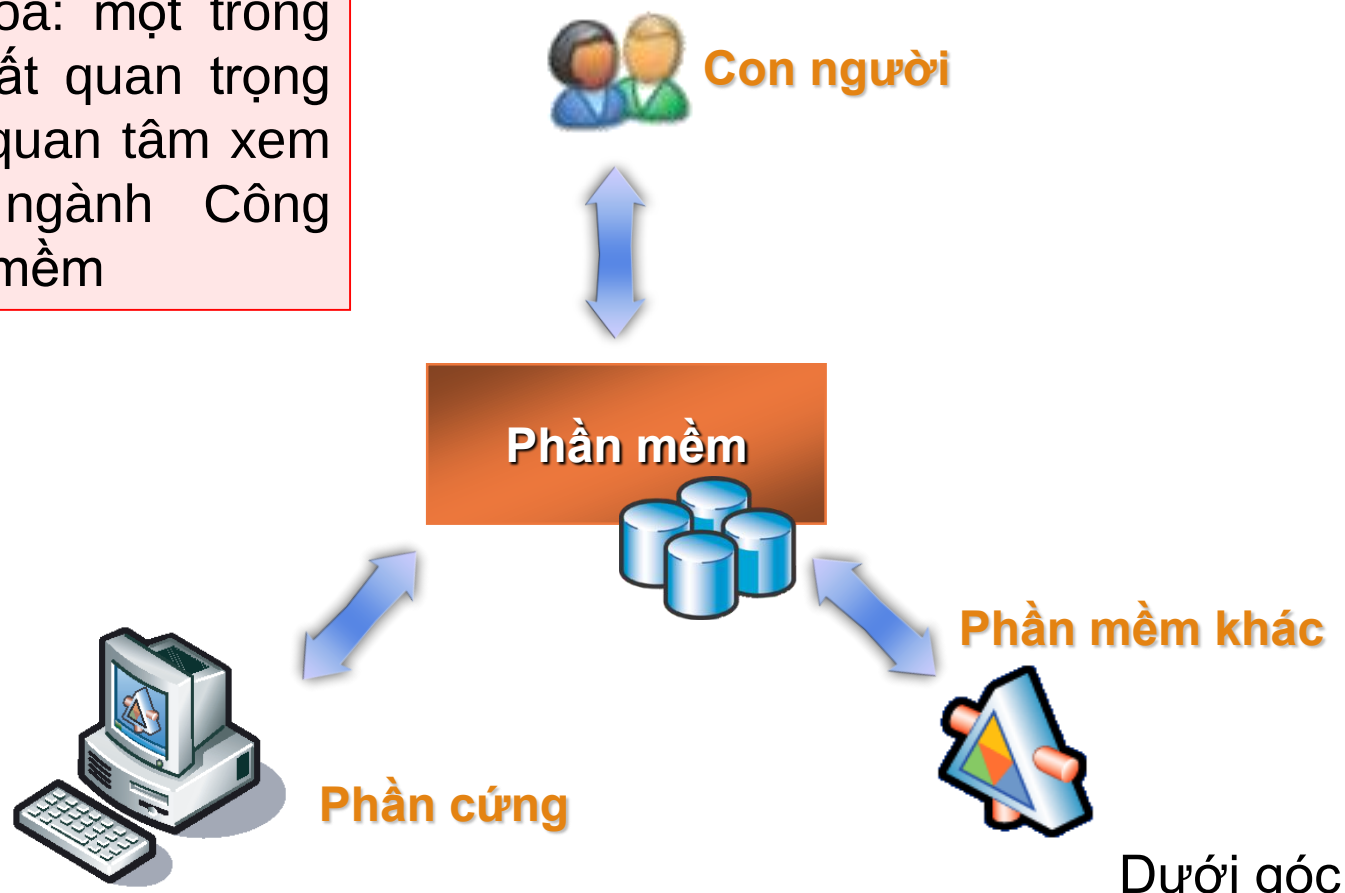
Tính tiện dụng

Tính hiệu quả

Dưới góc nhìn của Người sử dụng

Chất lượng phần mềm

Tính tiến hóa: một trong các tính chất quan trọng nhất được quan tâm xem xét trong ngành Công nghệ Phần mềm



Tính đúng đắn

Tính tiện dụng

Tính hiệu quả

Tính tương thích

Dưới góc nhìn của Người sử dụng

Chất lượng phần mềm

Tính tiến hóa

Tính đúng đắn

Tính tiện dụng

Tính hiệu quả

Tính tương thích

.....

Dưới góc nhìn của Người sử dụng

Chất lượng phần mềm

❖Tính dễ kiểm tra: việc kiểm tra các thành phần phù hợp với yêu cầu phần mềm là dễ dàng nhất có thể được

❖Tính dễ sửa lỗi: khi có sự không phù hợp (so với yêu cầu) trong quá trình kiểm tra một thành phần, việc phát hiện chính xác "vị trí lỗi" và sửa lỗi là nhanh nhất có thể được.

❖Tính dễ bảo trì: khi cần nâng cấp, cải tiến một thành phần (theo yêu cầu mới), việc cập nhật phần mềm là nhanh, chính xác nhất có thể được và đặc biệt là cố gắng hạn chế ảnh hưởng đến các thành phần khác

❖Tính tái sử dụng: các thành phần đã thực hiện có thể dùng lại trong các phần mềm cùng lớp (hoặc cùng lĩnh vực) với thời gian và công sức ít nhất có thể được

……

Dưới góc nhìn của Người lập trình

Vai trò của Công nghệ phần mềm

- Một ngành không thể thiếu trong lĩnh vực CNTT&TT
- Ngày càng có nhiều hệ thống khác nhau, trong mọi lĩnh vực, được kiểm soát bởi phần mềm
- Con người có thể sản xuất các hệ thống PM lớn và hữu ích là nhờ phần lớn vào sự phát triển của CNPM
- Mọi quốc gia phát triển đều phụ thuộc chủ yếu vào các hệ thống PM có chất lượng

Những khó khăn của Kỹ nghệ phần mềm

- Liệu có vấn đề trong việc phát triển PM?
  - Một số dự án thất bại
  - Những con số thống kê về các dự án PM
- Khủng hoảng PM
- Những khó khăn trong phát triển PM

Một số dự án thất bại

- AAS (FAA Advanced Automation System) (1989)
  - IBM phát triển (2.3 triệu dòng lệnh bằng Ada)
  - 1994: xây dựng lại từ đầu (vì đặc tả yêu cầu không đúng)
- C-17: 20M, cuối 80 →85 (lần thử đầu tiên 7/1990)
  - Gặp nhiều vấn đề khó về kỹ thuật, quá thời gian và kinh phí

Một số dự án thất bại

- Ariane 5 (June 04, 1996) nổ sau khi phóng (40s)
  - Do lỗi phần mềm điều khiển (chuyển 1 số thực 64bit →số nguyên 16bit)
- Head of AF Systems Command: “Phần mềm là nhược điểm của việc phát triển vũ khí”

Một số thống kê

- Sau khi khảo sát 8,000 dự án IT, Standish Group cho biết khoảng 30% bị hủy trước khi hoàn thành.
- Trung bình các dự án ở Mỹ bị hủy sau 1 năm tiến hành và tiêu tốn 200% kinh phí dự kiến (Capers Jones).
- Các dự án bị hủy chiếm khoảng 15% tổng kinh phí PM của Mỹ ($14 billion in 1993) (Capers Jones).

Một số thống kê

- 2/3 dự án được hoàn thành vượt quá thời gian và kinh phí dự kiến (Capers Jones) [bad estimates?].
- 2/3 dự án được hoàn thành là có độ tin cậy và chất lượng thấp trong một năm đầu triển khai (Jones).
- Tỷ lệ xảy ra lỗi của PM từ 0.5 đến 3.0 /1000 LOC (Bell Labs survey).

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.

Câu hỏi thảo luận

1. Bạn đã từng tham gia một dự án phần mềm mà nó chưa bao giờ kết thúc hoặc không được sử dụng?
2. Bạn có những ví dụ nào khác về thất bại của các dự án phần mềm?
3. Theo bạn, lý do chính dẫn đến thất bại là gì?
4. Kỹ sư phần mềm khó hơn hay Kỹ sư phần cứng?

Phát triển của Công nghệ phần mềm

- Giai đoạn 1 (1950 – giữa 1960)
  - Xử lý theo lô, xử lý tập trung, ít xử lý phân tán, ít sửa đổi phần mềm
- Giai đoạn 2 (từ giữa 1960 đến giữa 1970)
  - Hệ thống đa chương trình và đa người dùng
  - Bắt đầu cuộc “khủng hoảng” phần mềm

Phát triển của Công nghệ phần mềm

- Giai đoạn 3 (từ giữa 1970 đến giữa 1980)
  - Sự phát triển và sử dụng rộng rãi máy tính cá nhân
  - Sự phát triển của các công ty phần mềm
- Giai đoạn 4 (từ giữa 1980 đến nay)
  - Phần cứng ngày càng phát triển
  - Hệ thống phần mềm ngày càng đa dạng, phong phú, xử lý ngày càng phức tạp, công nghệ ngày càng phát triển…

Cuộc khủng hoảng phần mềm

- Số lượng các phần mềm tăng vọt (do sự phát triển của phần cứng: tăng khả năng, giá thành hạ).
- Phần mềm càng lớn sẽ kéo theo phức tạp hóa và tăng chi phí phát triển.
- Nhân lực chưa đáp ứng được nhu cầu phần mềm.

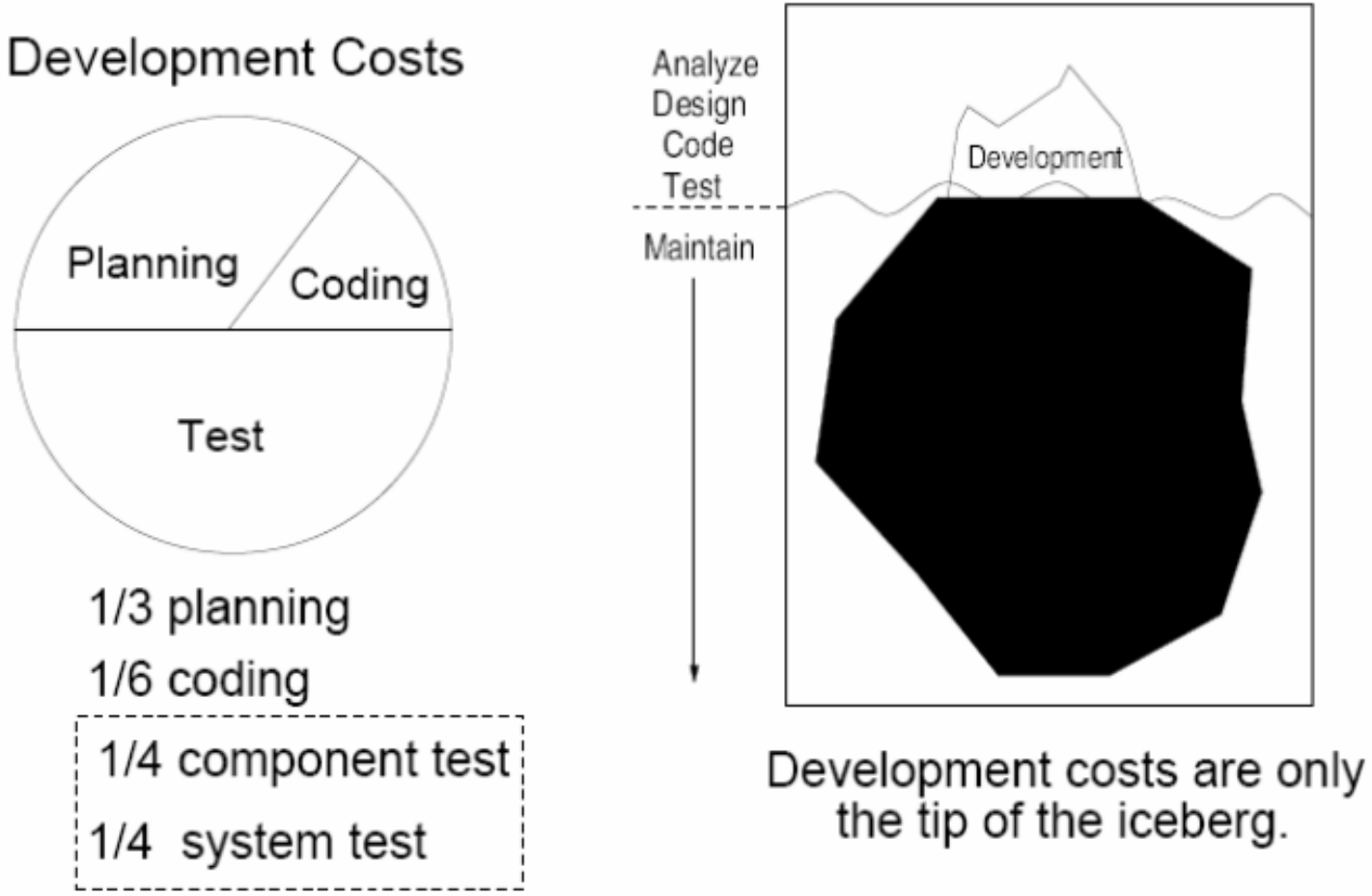
Cuộc khủng hoảng phần mềm

- Có quá nhiều khuyết điểm trong các phần mềm được dùng trong xã hội:
  - Thực hiện không đúng yêu cầu
  - Thời gian bảo trì nâng cấp quá lâu
  - Khó sử dụng, thực hiện chậm
  - Không chuyển đổi dữ liệu giữa các phần mềm
  - …

Cuộc khủng hoảng phần mềm

- Việc tăng vọt số lượng phần mềm là điều hợp lý và sẽ còn tiếp diễn.
- Các khuyết điểm của phần mềm có nguồn gốc chính từ phương pháp, cách thức và quy trình tiến hành xây dựng phần mềm.

Chi phí cho các pha



Những khó khăn trong phát triển PM

- Không có phương pháp mô tả rõ ràng định nghĩa yêu cầu của người dùng (khách hàng), sau khi bàn giao sản phẩm dễ phát sinh những trục trặc (troubles).
- Với những phần mềm quy mô lớn, tư liệu đặc tả đã cố định thời gian dài, do vậy khó đáp ứng nhu cầu thay đổi của người dùng một cách kịp thời trong thời gian đó.

Những khó khăn trong phát triển PM

- Nếu không có Phương pháp luận thiết kế nhất quán mà thiết kế theo cách riêng (của công ty, nhóm), thì sẽ dẫn đến suy giảm chất lượng phần mềm (do phụ thuộc quá nhiều vào con người).
- Nếu không có chuẩn về làm tư liệu quy trình sản xuất phần mềm, thì những đặc tả không rõ ràng sẽ làm giảm chất lượng phần mềm.

Những khó khăn trong phát triển PM

- Nếu không kiểm thử tính đúng đắn của phần mềm ở từng giai đoạn mà chỉ kiểm ở giai đoạn cuối và phát hiện ra lỗi, thì thường bàn giao sản phẩm không đúng hạn.
- Nếu coi trọng việc lập trình hơn khâu thiết kế thì thường dẫn đến làm giảm chất lượng phần mềm.

Những khó khăn trong phát triển PM

- Nếu coi thường việc tái sử dụng phần mềm (software reuse), thì năng suất lao động sẽ giảm.
- Phần lớn trong quy trình phát triển phần mềm có nhiều thao tác do con người thực hiện, do vậy năng suất lao động thường bị giảm.
- Không chứng minh được tính đúng đắn của phần mềm, do vậy độ tin cậy của phần mềm sẽ giảm.

Những khó khăn trong phát triển PM

- Chuẩn về một phần mềm tốt không thể đo được một cách định lượng, do vậy không thể đánh giá được một hệ thống đúng đắn hay không.
- Khi đầu tư nhân lực lớn vào bảo trì sẽ làm giảm hiệu suất lao động của nhân viên.
- Công việc bảo trì kéo dài làm giảm chất lượng của tư liệu và ảnh hưởng xấu đến những việc khác.

Những khó khăn trong phát triển PM

- Quản lý dự án lỏng lẻo kéo theo quản lý lịch trình cũng không rõ ràng.
- Không có tiêu chuẩn để ước lượng nhân lực và dự toán sẽ làm kéo dài thời hạn và vượt kinh phí của dự án.

Q & A