

# CTT010

## Kỹ thuật phần mềm



KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# Khái niệm: Công nghệ phần mềm - 1

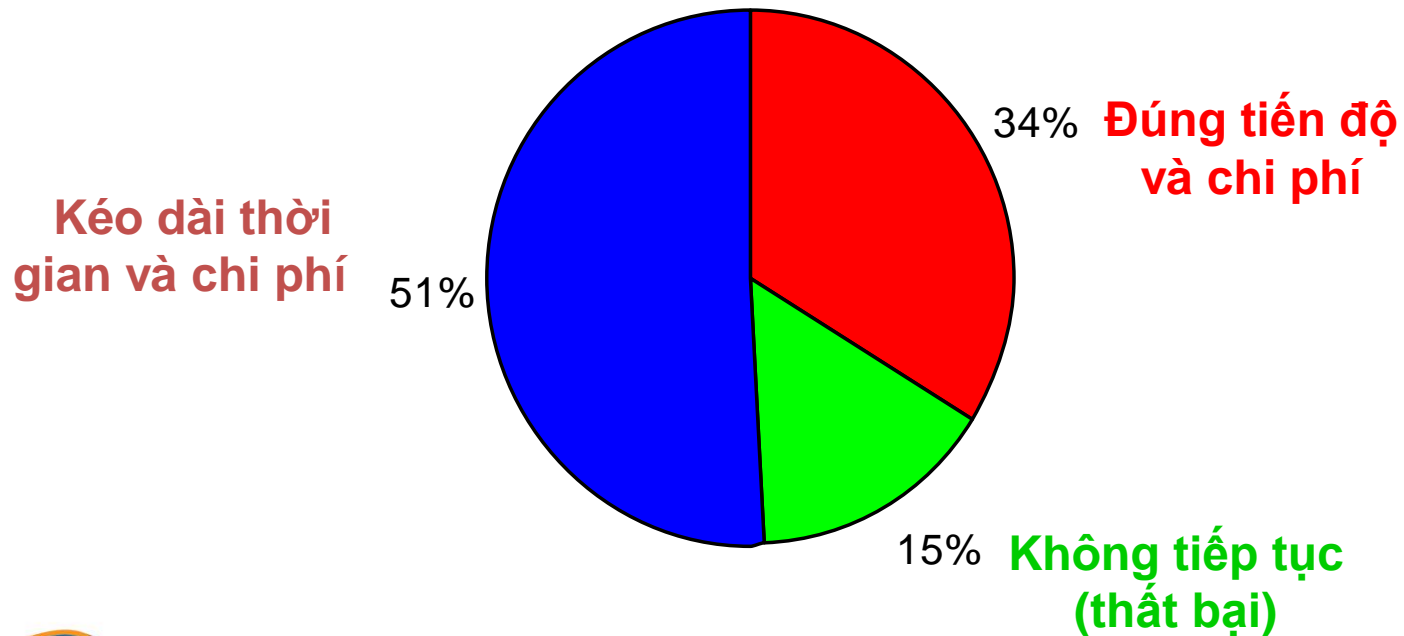
- Lĩnh vực liên quan đến tất cả các khía cạnh phát triển phần mềm chuyên nghiệp
- Mục tiêu: phát triển phần mềm chuyên nghiệp
  - ▣ Chi phí thấp
  - ▣ Chất lượng cao
  - ▣ Đúng tiến độ (thời gian)
  - ▣ Thoả mãn yêu cầu của khách hàng, người dùng

# Tầm quan trọng của CNPM

- ☐ Xã hội hiện đại không thể thiếu phần mềm
- ☐ Nền kinh tế phụ thuộc vào phần mềm
- ☐ CNPM giúp phát triển phần mềm nhanh hơn, chi phí thấp hơn, thời gian ngắn hơn, và đáp ứng yêu cầu tốt hơn

# Thách thức trong CNPM - 1

- Theo Standish Group CHAOS Report 2003: tỉ lệ các dự án phần mềm thành công thấp



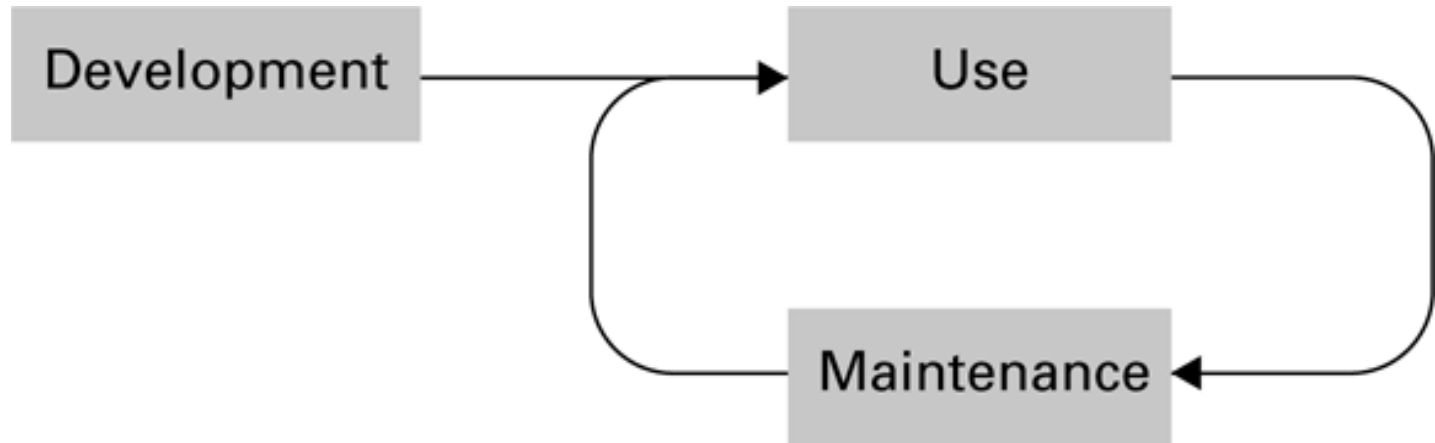
# Thách thức trong CNPM - 2

- Phần mềm khác với phần cứng
  - PM không sờ được
  - PM linh động, dễ chỉnh sửa
  - Phụ thuộc vào con người
- PM đa dạng
  - Môi trường
  - Hệ điều hành, thiết bị
  - Chức năng, cách thức phát triển

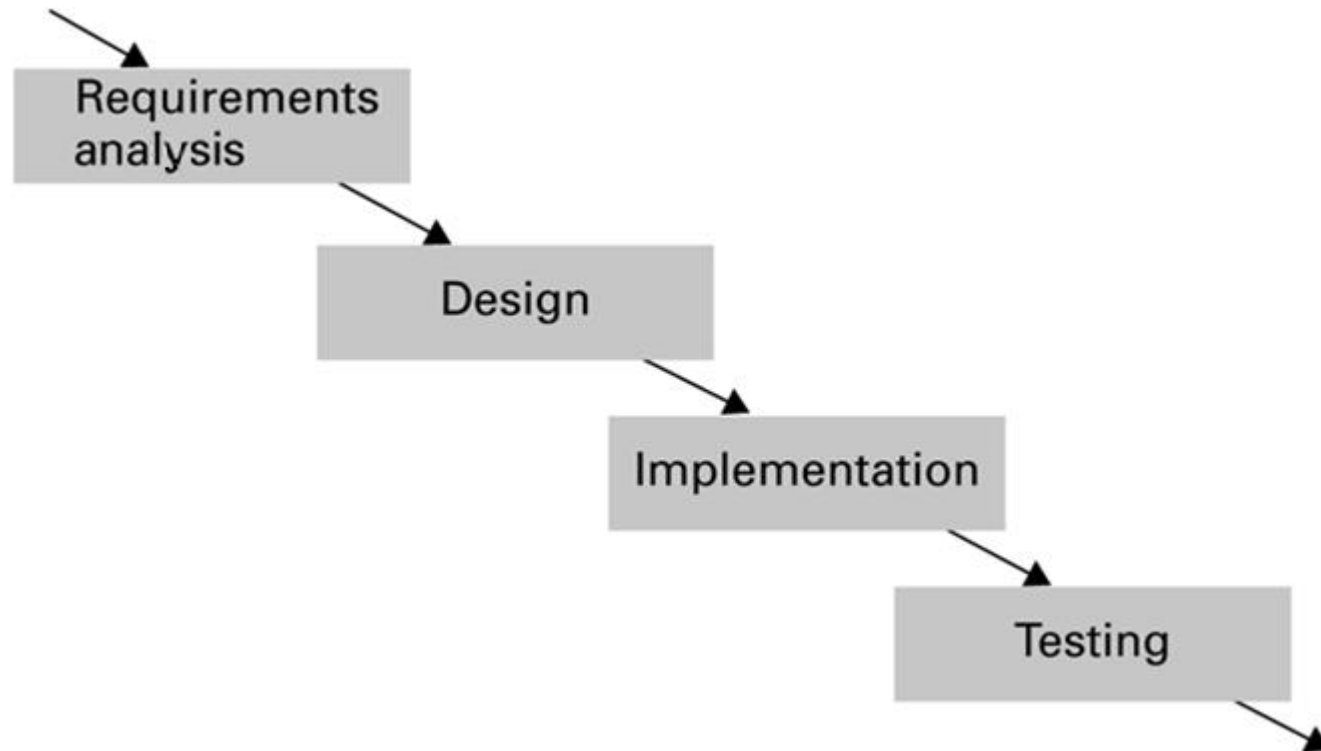
# Một số lĩnh vực

- ☐ Lập kế hoạch dự án
- ☐ Quản lý dự án
- ☐ Sưu liệu (Documentation)
- ☐ Tạo mô hình mẫu (Prototyping) và mô phỏng (simulation)
- ☐ Quy trình
- ☐ Thiết kế giao diện
- ☐ Lập trình
- ☐ ...

# Vòng đời phần mềm



# Giai đoạn phát triển trong vòng đời phần mềm





# Giai đoạn phân tích

- ☐ Lấy yêu cầu
- ☐ Đặc tả yêu cầu
  - ☒ Hướng về ứng dụng
- ☐ Tài liệu về phân tích yêu cầu phần mềm

# Giai đoạn thiết kế

- ☐ Thiết kế giao diện
- ☐ Thiết kế chức năng
- ☐ Thiết kế dữ liệu
- ☐ ....

# Giai đoạn cài đặt

- ☐ Cài đặt hệ thống từ tài liệu thiết kế
  - ☐ Cài đặt chương trình
  - ☐ Cài đặt dữ liệu

# Giai đoạn kiểm thử

- ☐ Validation testing
  - ☐ Xác nhận rằng hệ thống đáp ứng các yêu cầu đặt ra
- ☐ Defect testing
  - ☐ Tìm lỗi

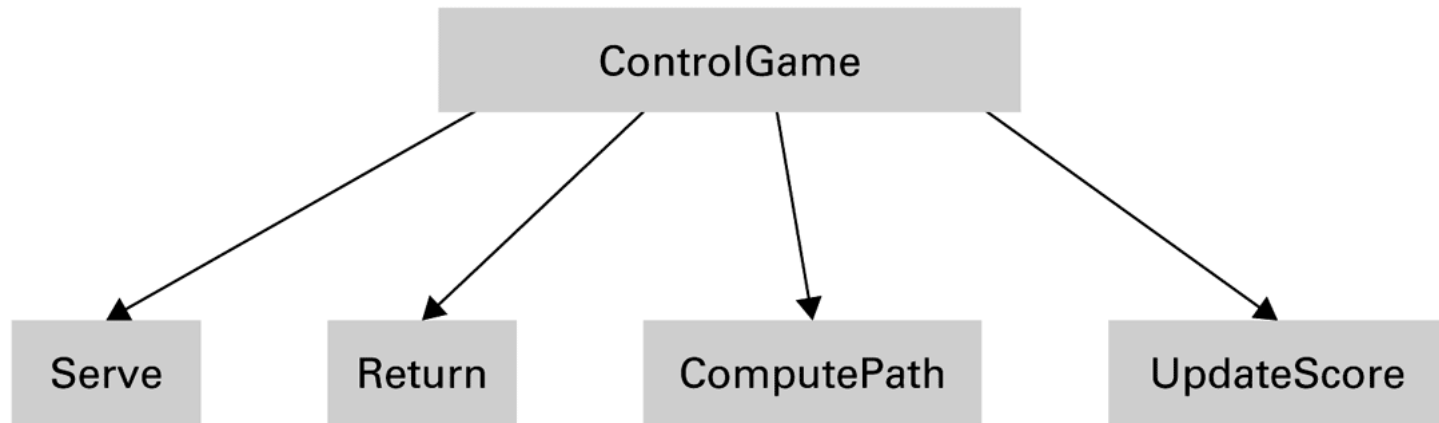
# Quy trình phát triển phần mềm

- ☐ Mô hình thác nước (Waterfall Model)
- ☐ Mô hình tăng trưởng (Incremental Model)
  - ☒ Prototyping (Evolutionary vs. Throwaway)
- ☐ Phát triển phần mềm mã nguồn mở (Open-source Development)
- ☐ Lập trình cực độ (Extreme Programming)

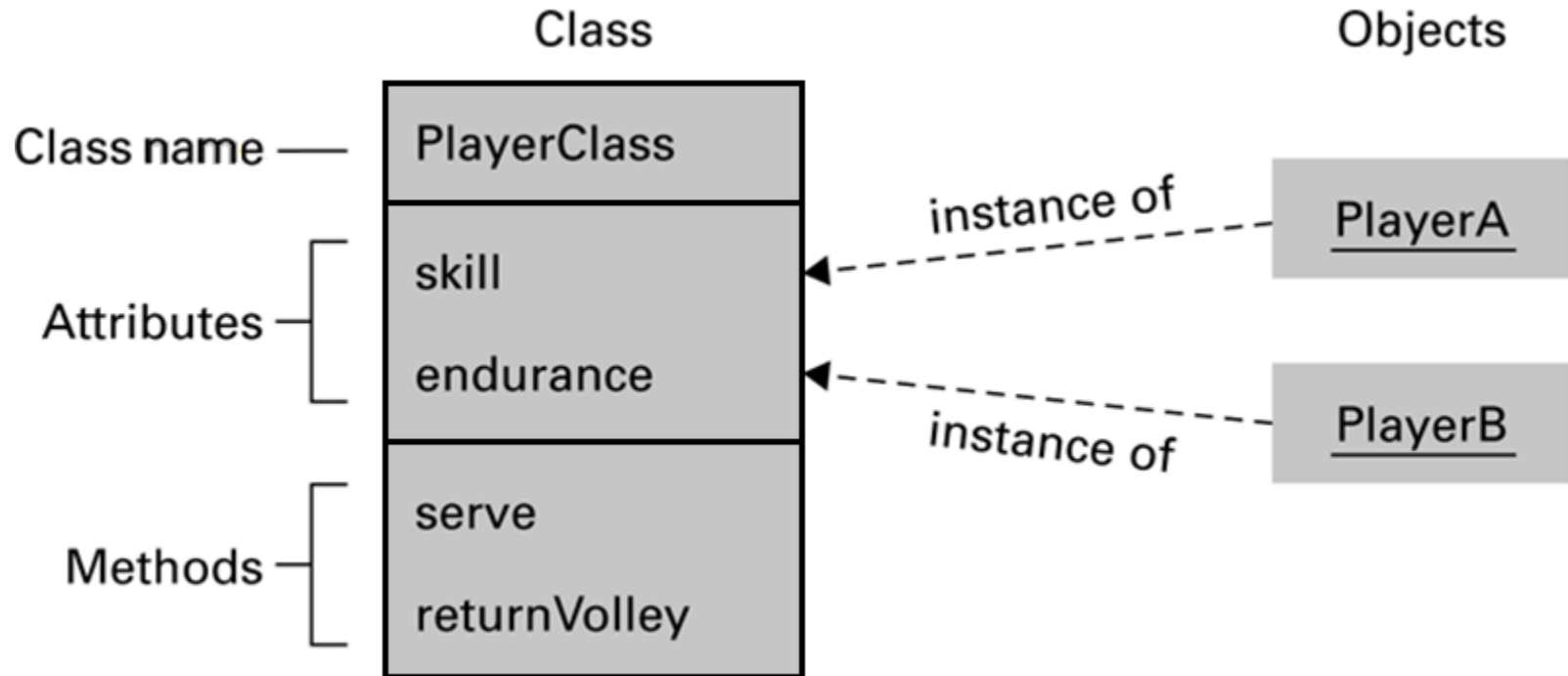
# Phân chia phần mềm thành các phần (**Modularity**)

- ☐ Theo các hàm chức năng riêng biệt (Function) – Lập trình hàm
  - ☐ Biểu đồ cấu trúc (Structure chart)
- ☐ Theo các đối tượng (Object) – Lập trình hướng đối tượng
  - ☐ Sơ đồ công tác (Collaboration diagrams)
- ☐ Theo các thành phần (Component)
  - ☐ Kiến trúc thành phần (Component architecture)

# Một biểu đồ cấu trúc

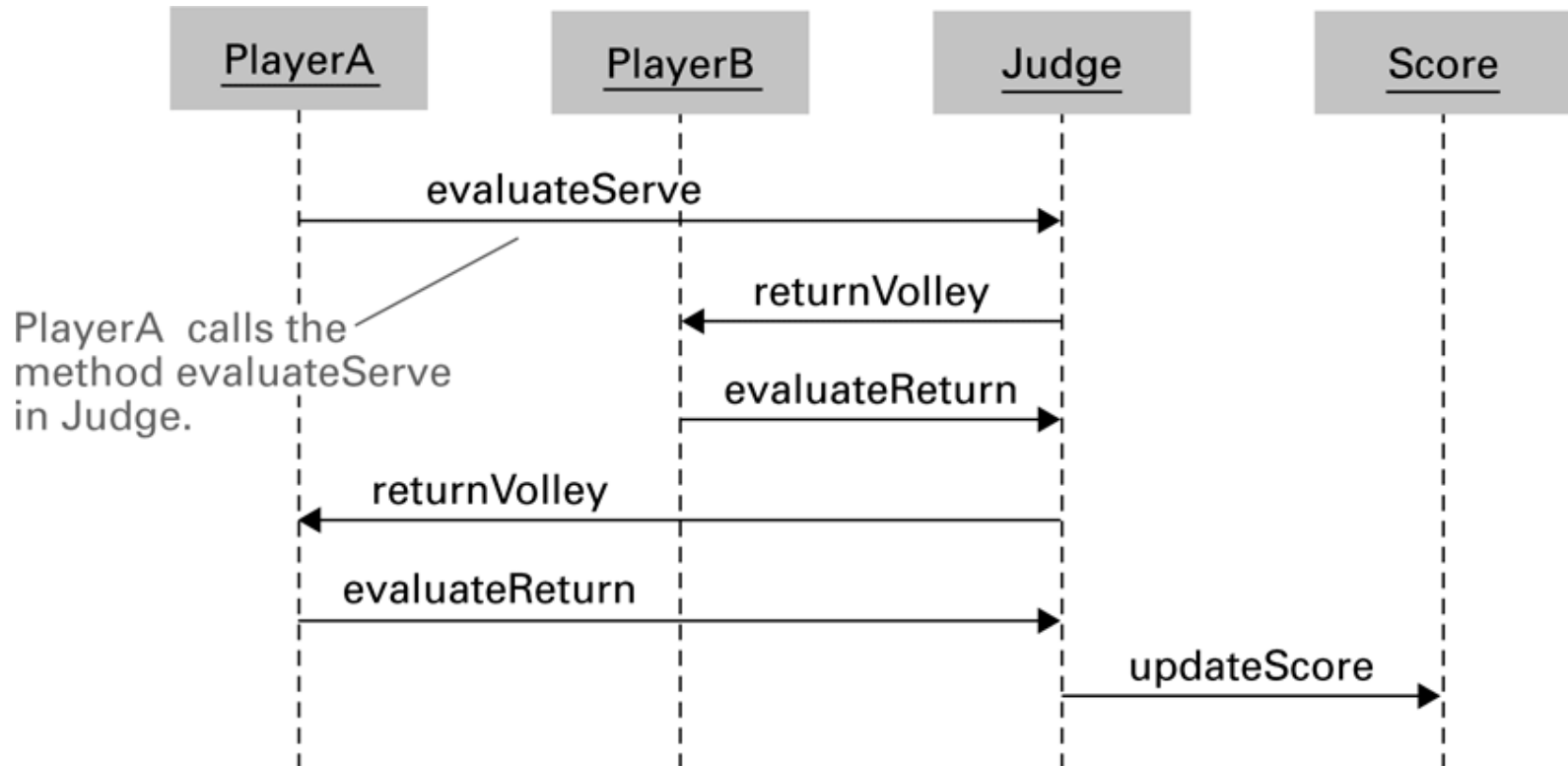


# Lớp PlayerClass và các thể hiện (đối tượng) của nó





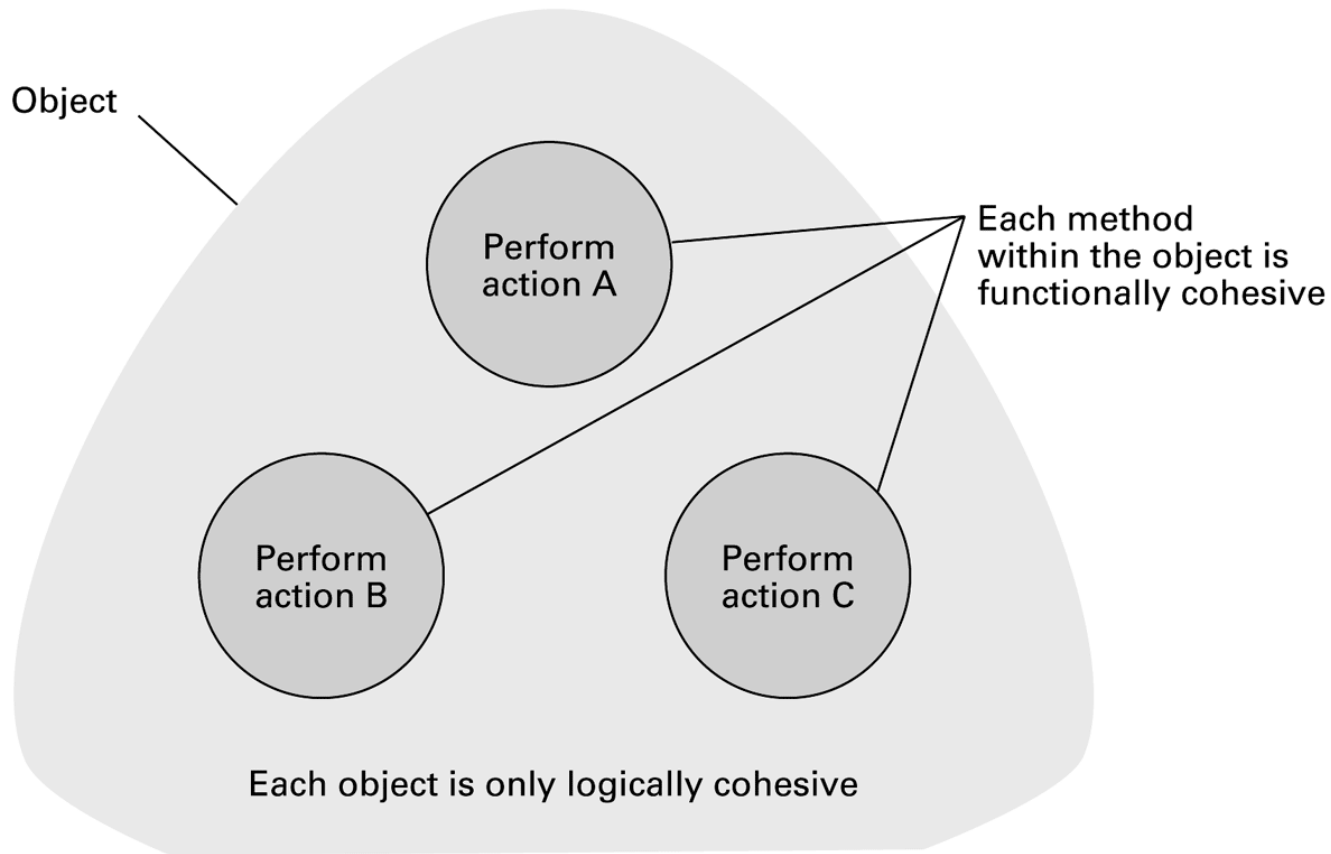
# Sự tương tác của các đối tượng để phục vụ cho người chơi A



# Tính ghép nối (Coupling) và tính kết dính (Cohesion)

- Thiết kế phần mềm cần đạt được:
  - ▣ Các thành phần cần ghép nối với nhau lỏng lẻo.
    - Ghép nối càng lỏng lẻo thì càng dễ thích nghi, phù hợp với hoàn cảnh mới.
  - ▣ Các bộ phận trong một thành phần phải kết dính với nhau

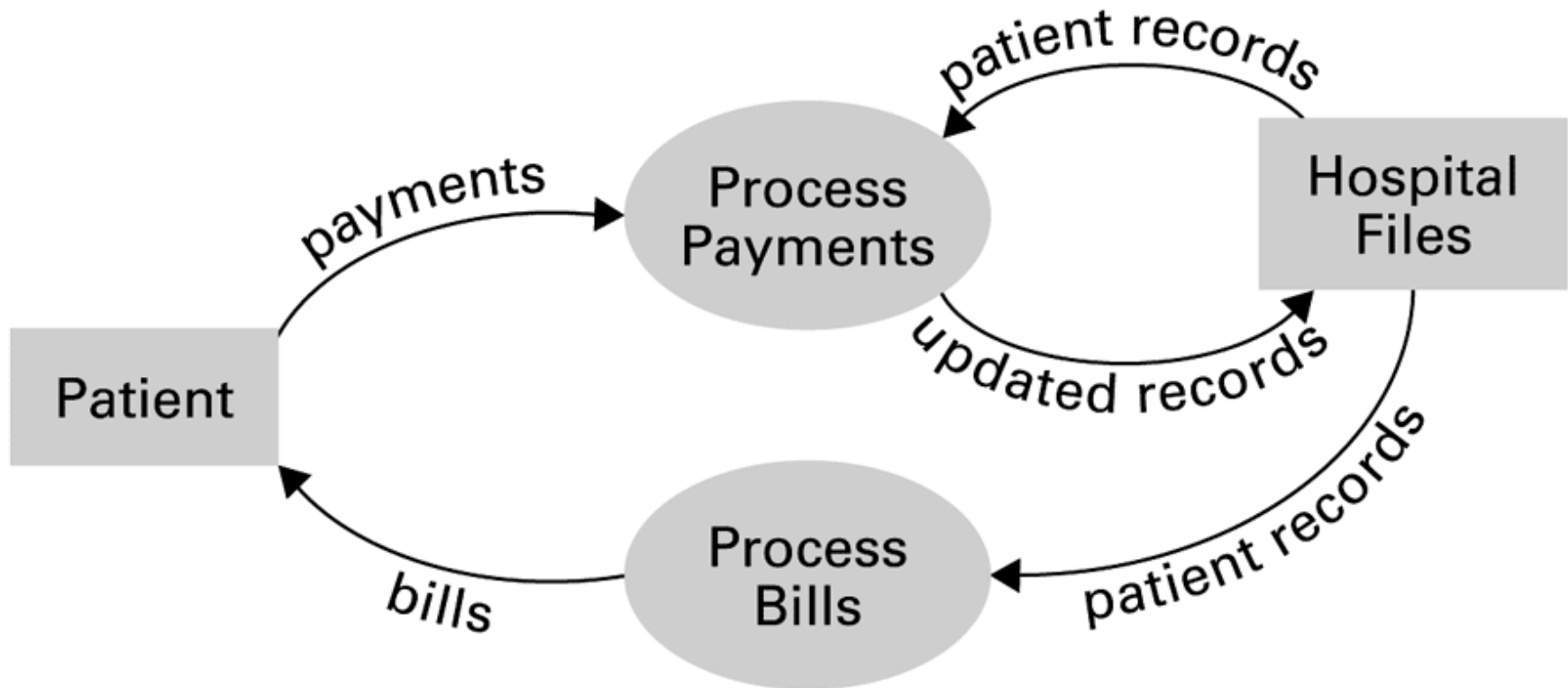
# Tính kết dính (Cohesion) trong trong một đối tượng



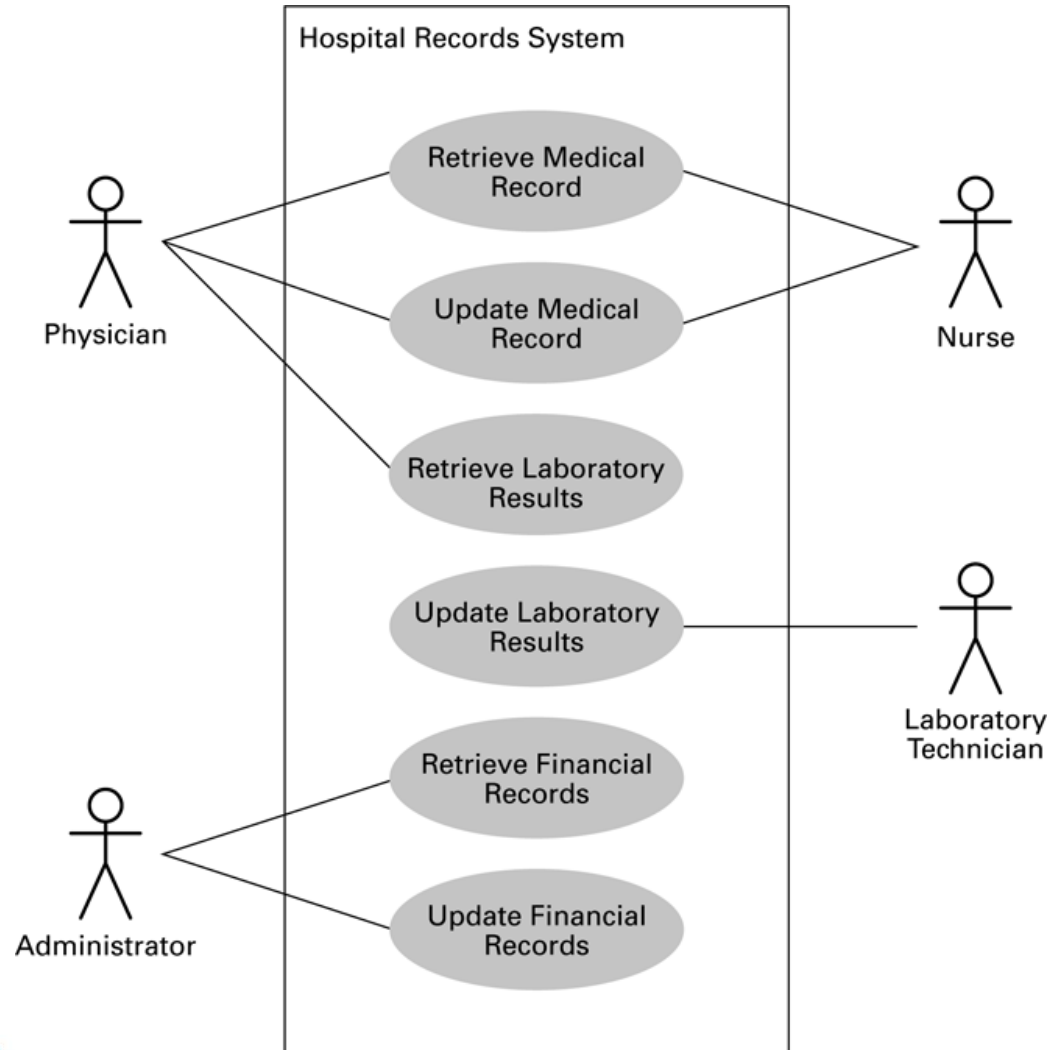
# Một số mô hình trong phát triển phần mềm

- ☐ Sơ đồ luồng dữ liệu (Data Flow Diagram)
- ☐ Mô hình thực thể kết hợp (Entity-Relationship Diagram)
  - ☐ Mỗi quan hệ một-một (1-1)
  - ☐ Mỗi quan hệ một-nhiều (1-n)
  - ☐ Mỗi quan hệ nhiều-nhiều (n-n)
- ☐ Sơ đồ use-case
- ☐ Sơ đồ lớp
- ☐ ...

# Ví dụ sơ đồ luồng dữ liệu



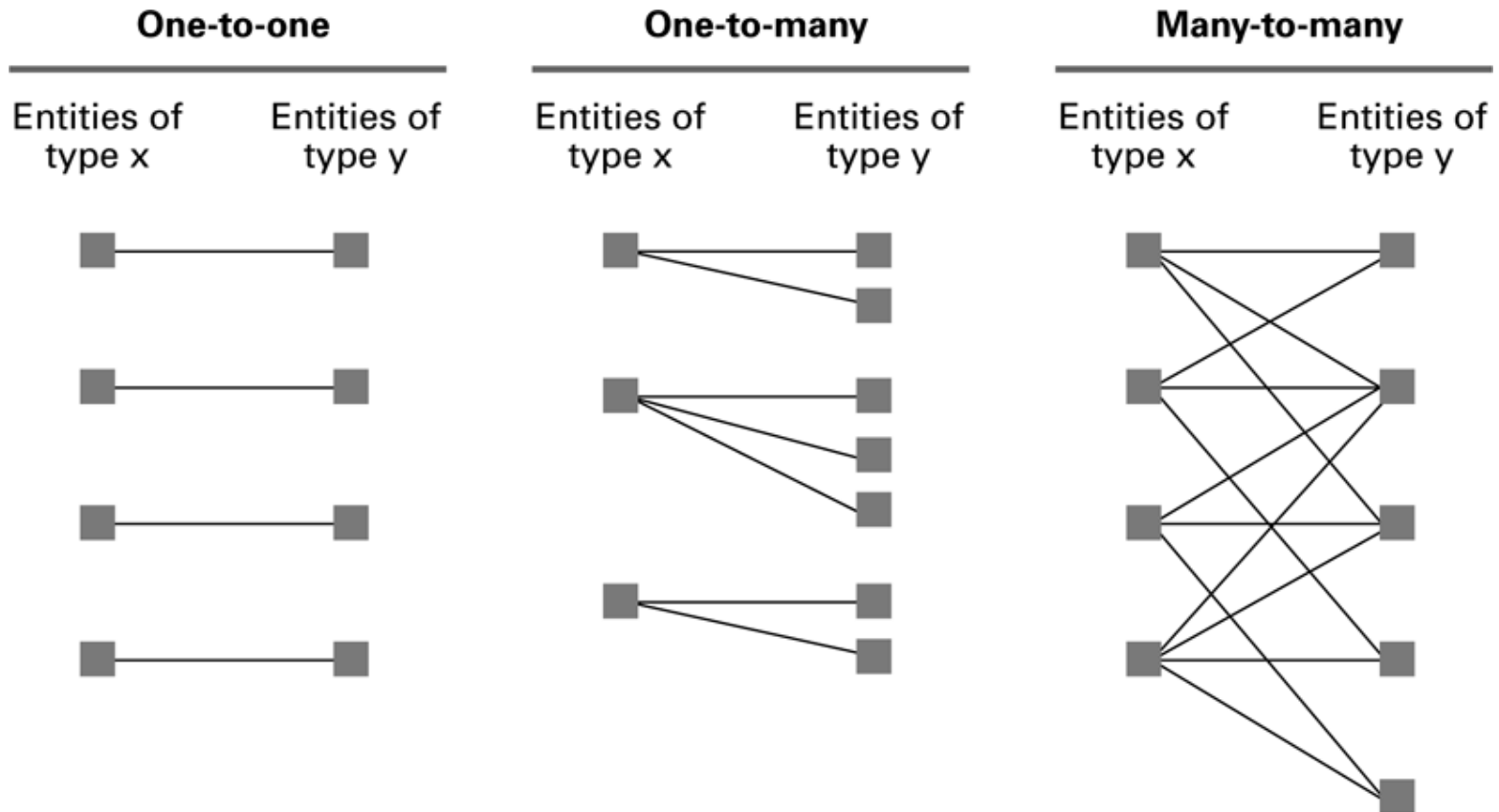
# Ví dụ về sơ đồ use-case



# Ví dụ về sơ đồ lớp

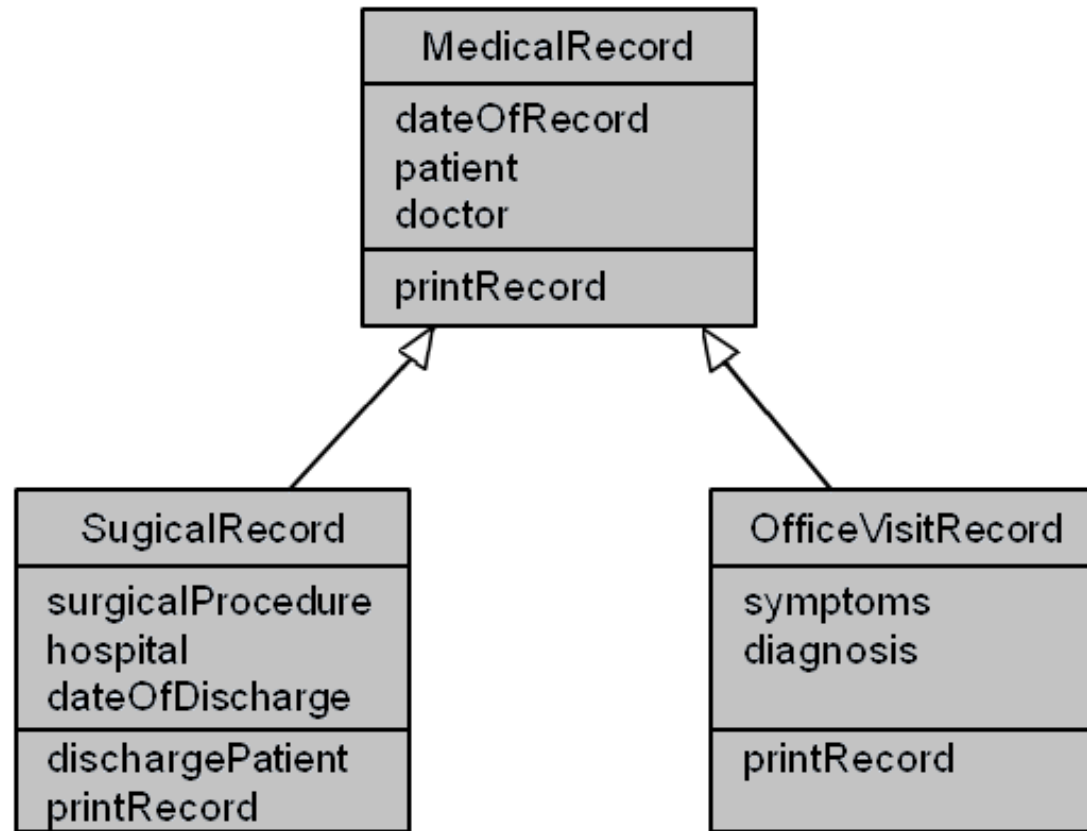


# Mối quan hệ giữa hai thực thể trong mô hình thực thể kết hợp

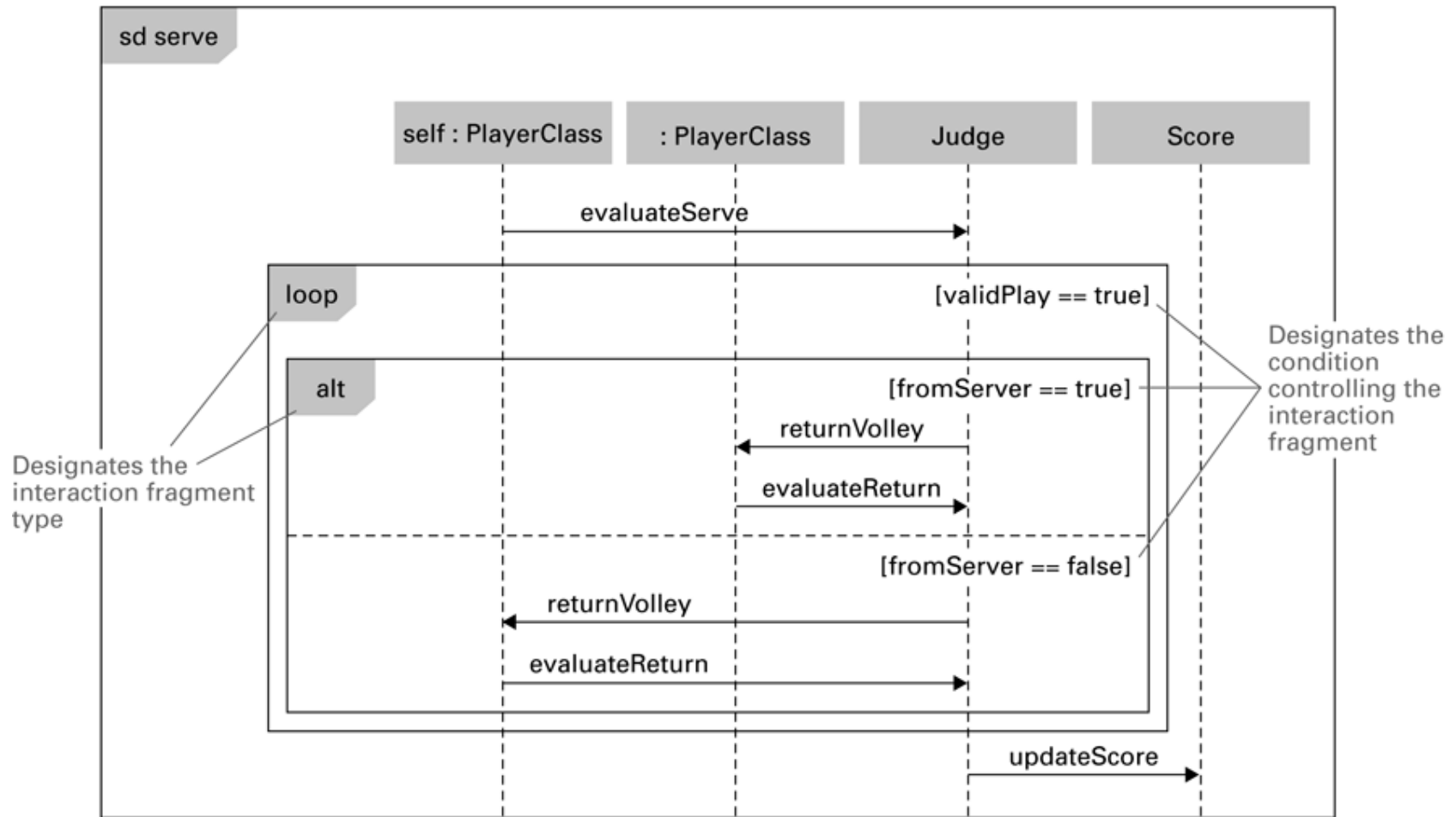




# Mối quan hệ tổng quát hóa trong sơ đồ lớp



# Một ví dụ về sơ đồ tuần tự



# Mẫu thiết kế (Design Pattern)

- ☐ Các “template” thiết kế tốt cho việc giải quyết một vấn đề
- ☐ Ví dụ:
  - ☐ Mẫu Adaptern: được sử dụng để làm một thành phần giao diện mô-đun mô-đun phù hợp với nhu cầu hiện tại

# Chiến lược kiểm thử phần mềm

- ☐ Glass-box testing
  - ☐ Pareto principle
  - ☐ Basis path testing
- ☐ Black-box testing
  - ☐ Boundary value analysis
  - ☐ Redundancy testing
  - ☐ Beta testing

# Tài liệu phần mềm (Documentation)

- ☐ Tài liệu cho hướng dẫn người dùng
  - ☐ Sách in
  - ☐ Online
- ☐ Tài liệu hệ thống
  - ☐ Mã nguồn
  - ☐ Tài liệu phân tích, thiết kế
- ☐ Tài liệu kỹ thuật
  - ☐ Hỗ trợ việc cài, cập nhật ... phần mềm

# Quyền sở hữu phần mềm (Software Ownership)

## □ Copyright

- Cho phép phát hành sản phẩm trong khi vẫn giữ quyền sở hữu tài sản trí tuệ
  - Trong tất cả các công trình:
    - Thông số kỹ thuật
    - Mã nguồn
    - Sản phẩm cuối cùng

# Quyền sở hữu phần mềm (Software Ownership)

- Bản quyền phần mềm (Software License)
  - Thỏa thuận pháp lý cấp cho người dùng một số quyền nhất định mà không chuyển quyền sở hữu
- Bằng sáng chế (Patent)
  - Phải chứng minh tính mới, có thể sử dụng
  - Quá trình tốn kém và cần nhiều thời gian

# Tổng quan

- Bộ môn Công Nghệ Phần Mềm
  - ▣ Thành lập năm 1998
  - ▣ Địa chỉ: Phòng I 82, 227 Nguyễn Văn Cừ, Q5 TpHCM
  - ▣ Điện thoại: (028) 38 324 467 (ext: 802)





# Nguồn nhân lực

- Tổng số GV: 31 cán bộ công tác trong nước
  - 1 Phó Giáo sư
  - 5 Tiến sĩ
  - 24 Thạc sĩ
  - 1 học viên Cao học



**Trưởng bộ môn: TS. Nguyễn Văn Vũ**  
**Phó Trưởng bộ môn: TS. Nguyễn Thị Minh Tuyền**

# Mục tiêu đào tạo

- ☐ Sinh viên khi ra trường có khả năng tham gia phát triển phần mềm chuyên nghiệp
- ☐ Khả năng ***phân tích yêu cầu, thiết kế, kiểm thử và triển khai thực hiện*** các hệ thống phần mềm
- ☐ Khả năng ***sử dụng và tự tìm hiểu để sử dụng*** công cụ hỗ trợ phát triển phần mềm
- ☐ Kỹ năng *mềm*
- ☐ Khả năng ***tự tìm hiểu, nghiên cứu*** các công nghệ, phương pháp, quy trình mới trong lĩnh vực công nghệ phần mềm

# Các học phần chung chuyên ngành

- SV tích lũy ít nhất 5 học phần trong danh sách. [Xem chương trình](#)

## Nhập môn công nghệ phần mềm

Quản lý dự án phần mềm

Phân tích và quản lý yêu cầu phần mềm

Lập trình Windows

Phân tích và thiết kế phần mềm

Phát triển phần mềm cho thiết bị di động

Kiểm chứng phần mềm

Phát triển game

Phát triển ứng dụng web

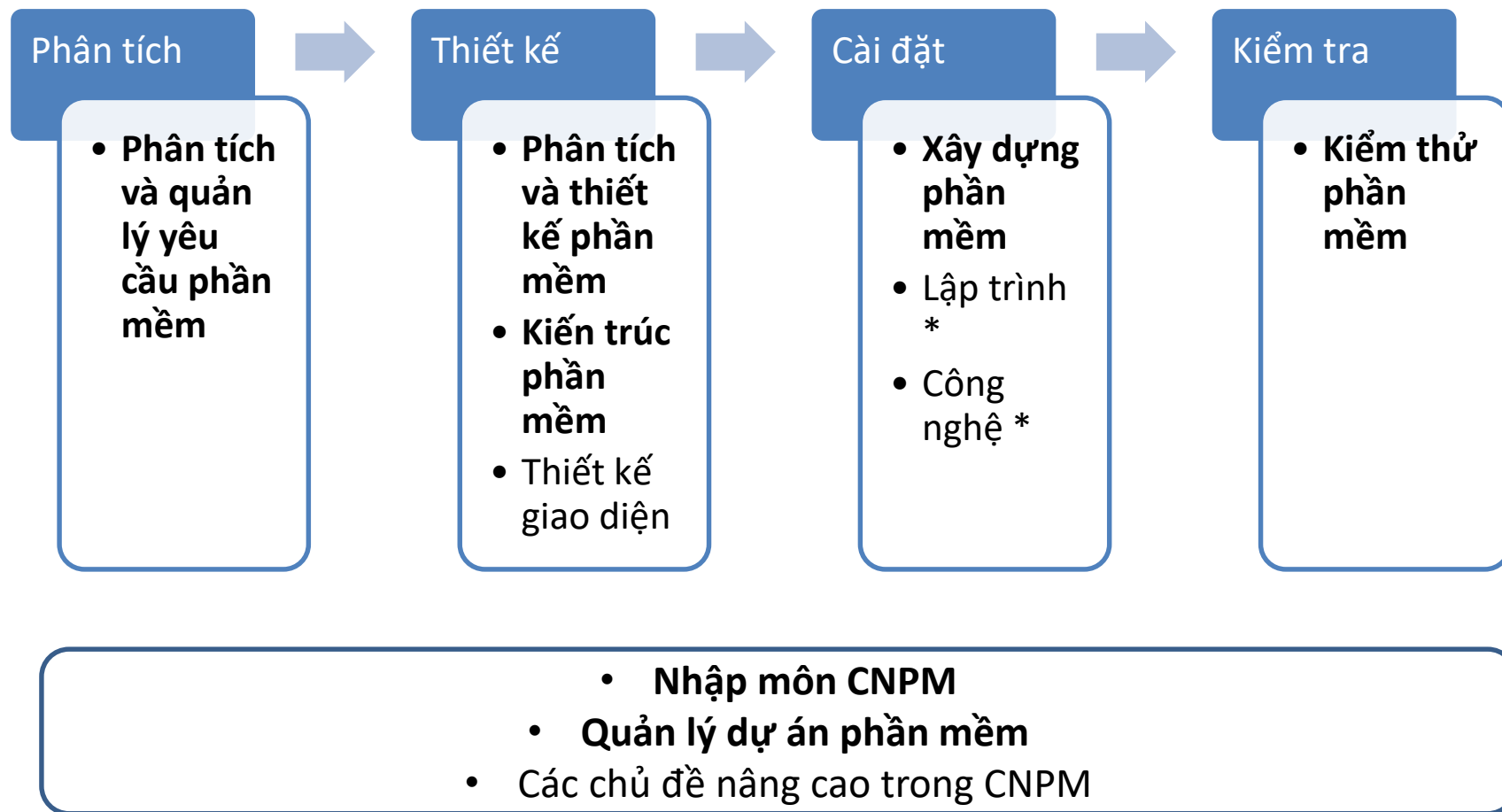
# Các học phần tự chọn ngành

□ SV tích lũy ít nhất 2 học phần trong danh sách

MÃ HỌC PHẦN	TÊN HỌC PHẦN
CSC10101	Kỹ năng mềm
CSC10102	Kiến tập nghề nghiệp
CSC10103	Khởi nghiệp
CSC10104	Quy hoạch tuyến tính
CSC10105	Nhập môn tư duy thuật toán
CSC10106	Thuật toán tổ hợp và ứng dụng
CSC10107	Thực tập thực tế
CSC13101	Các chủ đề nâng cao trong công nghệ PM
CSC13102	Lập trình ứng dụng Java
CSC13103	Công nghệ Java cho hệ thống phân tán
CSC13104	Công nghệ XML và ứng dụng

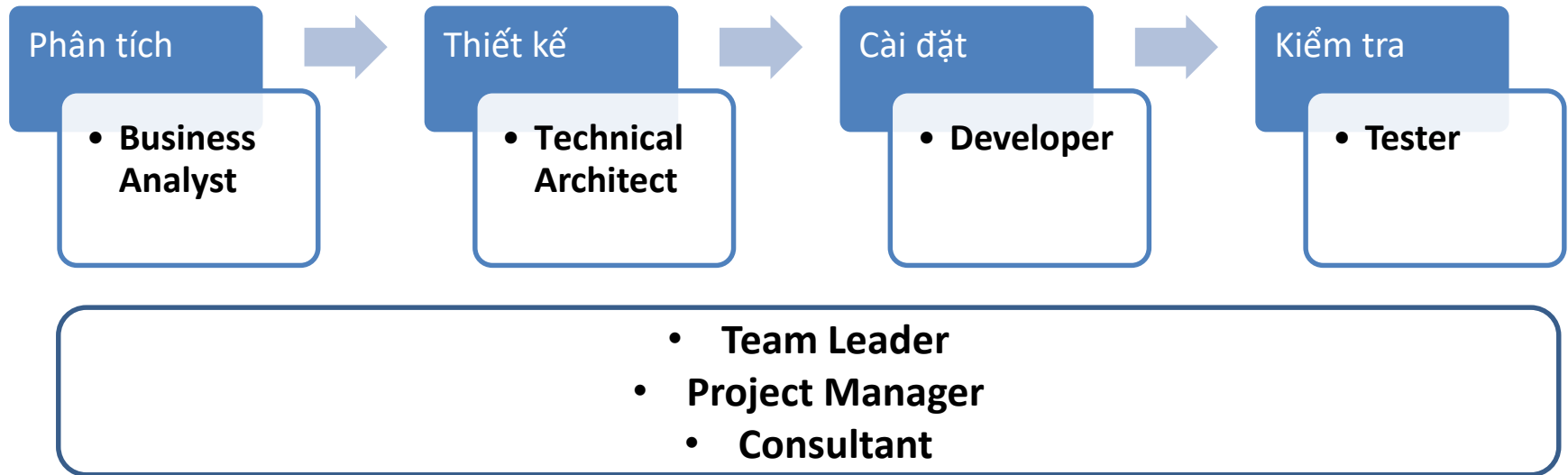
MÃ HỌC PHẦN	TÊN HỌC PHẦN
CSC13105	Đặc tả hình thức
CSC13106	Kiến trúc phần mềm
CSC13107	Mẫu thiết kế hướng đối tượng và ứng dụng
CSC13108	Mô hình hóa phần mềm
CSC13109	Phát triển phần mềm cho hệ thống nhúng
CSC13110	Phát triển phần mềm nguồn mở
CSC13111	Thanh tra mã nguồn
CSC13112	Thiết kế giao diện
CSC13113	Lập trình hướng đối tượng nâng cao

# Tương quan Học phần & CNPM

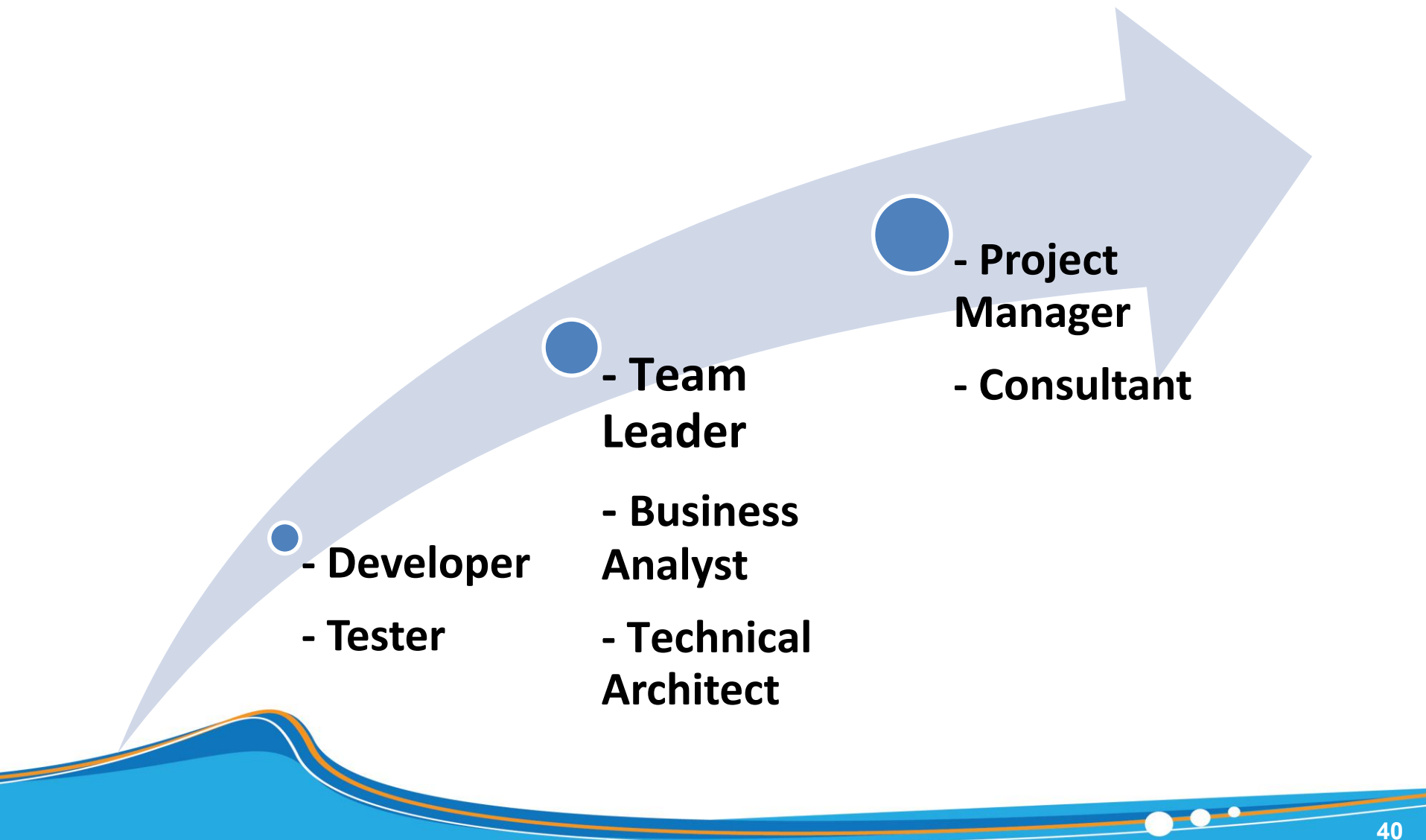




# Định hướng nghề nghiệp - 1



# Định hướng nghề nghiệp - 2





# Các kỹ năng cần thiết cho các vai trò

## ☐ Developer

- ☐ Kỹ năng lập trình
- ☐ Thuật toán
- ☐ Tư duy logic
- ☐ Tiếp cận công nghệ và ngôn ngữ mới
- ☐ tỉ mỉ và chính xác
- ☐ Đọc hiểu

# Các kỹ năng cần thiết cho các vai trò

## ☐ Backend developer

- ☐ Cấu trúc dữ liệu
- ☐ Logic
- ☐ Nghiệp vụ
- ☐ Ngôn ngữ như Java, Perl, Ruby, C#, JavaScript

## ☐ Frontend developer

- ☐ Giao diện
- ☐ Nghiệp vụ
- ☐ Ngôn ngữ như HTML, JavaScript, CSS, Java, ObjectC, Swift

# Các kỹ năng cần thiết cho các vai trò

## ☐ Tester

- ☐ Tỉ mỉ và chính xác
- ☐ Tư duy logic
- ☐ Tầm nhìn hệ thống
- ☐ Am hiểu nghiệp vụ khách hàng, người dùng
- ☐ Giao tiếp

## ☐ Business analyst

- ☐ Giao tiếp (đặc biệt là viết)
- ☐ Am hiểu và phân tích nghiệp vụ khách hàng, người dùng
- ☐ Tầm nhìn hệ thống

# Các kỹ năng cần thiết cho các vai trò

- Technical architect: như developer +
  - Tầm nhìn hệ thống
  - Am hiểu nghiệp vụ khách hàng, người dùng
  - Giao tiếp
- Project manager
  - Kỹ năng quản lý con người, thời gian, tài chính, v.v...
  - Tầm nhìn sâu và rộng (tầm mức hệ thống)
  - Giao tiếp, đàm phán
- Consultant
  - Như technical architect

# Hướng nghiên cứu ngành

- ❑ Ước lượng phần mềm (software estimation)
- ❑ Kiểm thử tự động (test automation)
- ❑ Khai phá dữ liệu phần mềm (mining software repositories)
- ❑ Dịch vụ phần mềm trên cloud (software cloud-based services)
- ❑ Giáo dục trong phần mềm (software engineering education)
- ❑ Kiểm chứng phần mềm (Verification and Validation)
- ❑ Tương tác người – máy (human computer interaction)

# Hướng nghiên cứu ngành - 2

- ❑ Thiết kế phần mềm, kiến trúc phần mềm, mẫu thiết kế phần mềm (Software design, software architecture and software design patterns)
- ❑ Phát triển phần mềm doanh nghiệp để bàn, web và di động (Engineering of desktop, web and mobile enterprise software)
- ❑ Phát triển ứng dụng đám mây đa người thuê (Cloud computing application development)
- ❑ Ứng dụng trí tuệ nhân tạo, học máy và xử lý ngôn ngữ tự nhiên cho phát triển phần mềm (Applying AI/ML/NLP to address software engineering problems)

# Thầy/Cô trưởng nhóm NC

- ☐ PGS TS Trần Minh Triết
- ☐ TS Đinh Bá Tiến
- ☐ TS Nguyễn Văn Vũ
- ☐ TS Nguyễn Thị Minh Tuyên
- ☐ TS Ngô Huy Biên

