MÔ HÌNH HOÁ HƯỚNG ĐỐI TƯỢNG THEO USE-CASE

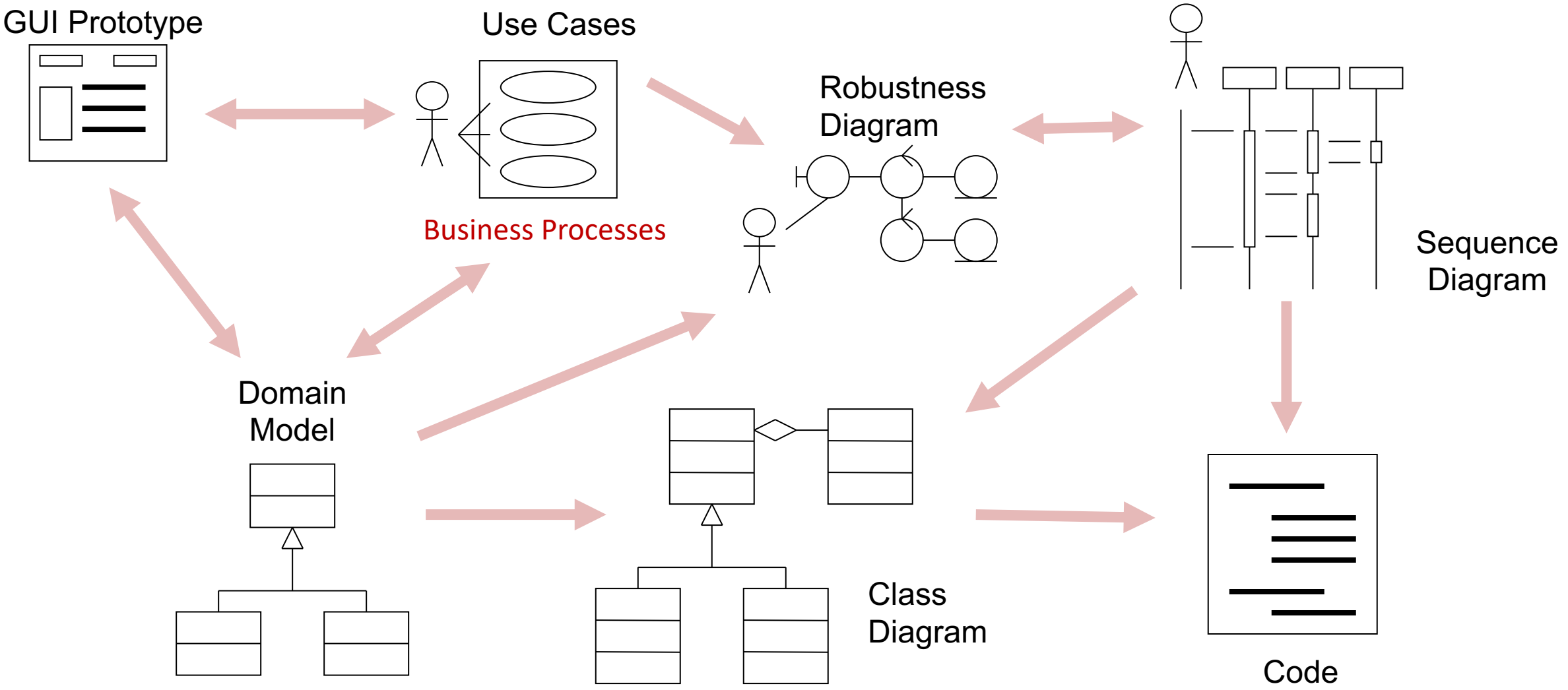
PGS.TS. HOÀNG HỮU HẠNH

FMM@PTIT

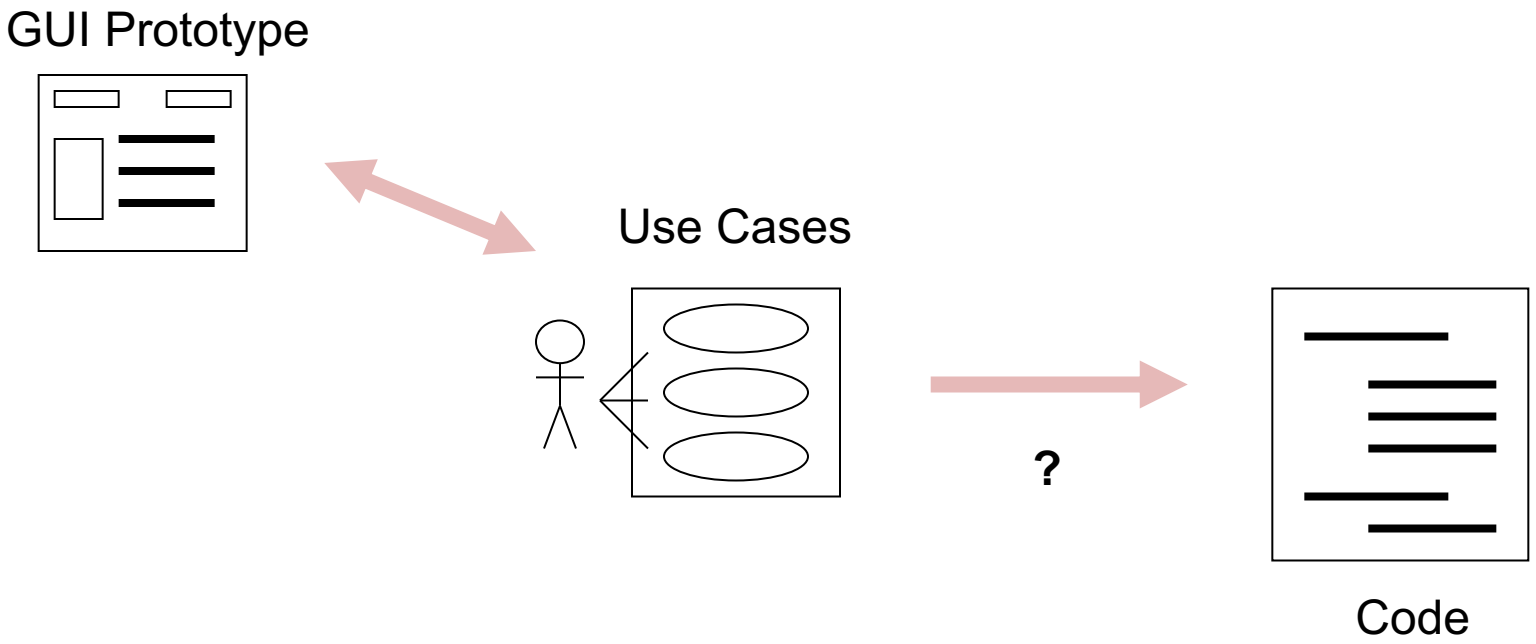
Kiến trúc đi theo Quy trình

- Một tổ chức thì biết rõ quy trình (nghiệp vụ) của họ là cái gì
  - Do đó, họ có thể lặp lại sự thành công của họ
- Các quy trình phát triển phần mềm là lớn và phức tạp, tuy nhiên việc áp dụng quy trình trong phát triển phần mềm là quan trọng
  - Dự án phần mềm càng lớn, thì tính chính chống và hình thức càng cần thiết
- Quy trình phát triển phần mềm thường đã được xác định, và các điểm xuất phát là từ yêu cầu người sử dụng
- Mô hình UC sẽ đóng vài trò quan trọng và dẫn dắt trong quy trình phát triển phần mềm

Lộ trình - Roadmap



Cách nào để từ các UC đến Code?



Hãy thử làm phép quay lui

Để đến bước "Code"…

- Chúng ta cần **một tập hoàn chỉnh các lớp**, với các thuộc tính được xác định, các phương thức được định nghĩa
  - Vậy chúng ta cần các **Biểu đồ Lớp mức thiết kế** (design-level class diagrams)
- Trước khi chúng ta có các phương thức, chúng ta cần **xác định các hành vi trong các lớp**
  - Vì vậy chúng ta cần **các biểu đồ tuần tự** (sequence diagrams) cho các use-case

… Tiếp tục quay lui

- Trước khi chúng ta thực hiện các **biểu đồ tuần tự**, ta cần biết các đối tượng nào là liên quan đến các use-case nào
  - Chúng ta có được các thông tin này từ các **biểu đồ phân tích đầy đủ** (robustness)
  - Biểu đồ này có được bang các **phân tích đầy đủ các lớp** / đối tượng theo các use-case
  - Đây không phải là phần của tiêu chuẩn UML, nhưng rất hiệu quả

… Tiếp tục quay lui

- Trước khi chúng ta có thể làm phân tích đầy đủ, chúng ta cần có một mô hình miền (domain model)
  - Chúng ta cần làm điều này để đảm bảo rằng chúng ta hiểu các quan điểm của người sử dụng về nhận thức các đối tượng trong bài toán đề ra

- Vậy thì, giờ chúng ta thực hiện công việc trở lại từng bước…

Bước 1: Yêu cầu

- Xác định các đối tượng trong miền của thế giới thực (bài toán) và các quan hệ giữa chúng
  - Chúng ta có: Domain Model
- Phác thảo một Giao diện đồ hoạ NSD (GUI Prototype/Sketch)
- Xác định các use-case
- Mô hình hoá nghiệp vụ (hoạt động UC) bằng Activity Diagram
- *Milestone 1: requirements review*

Bước 2: Phân tích

- Tạo các biểu đồ use-case và viết các bản mô tả use-case
- Thực hiện phân tích đầy đủ (robustness analysis)
  - Xác định các đối tượng sử dụng trong mỗi use-case
  - Cập nhật lại domain-model
- *Milestone 2: preliminary design review*

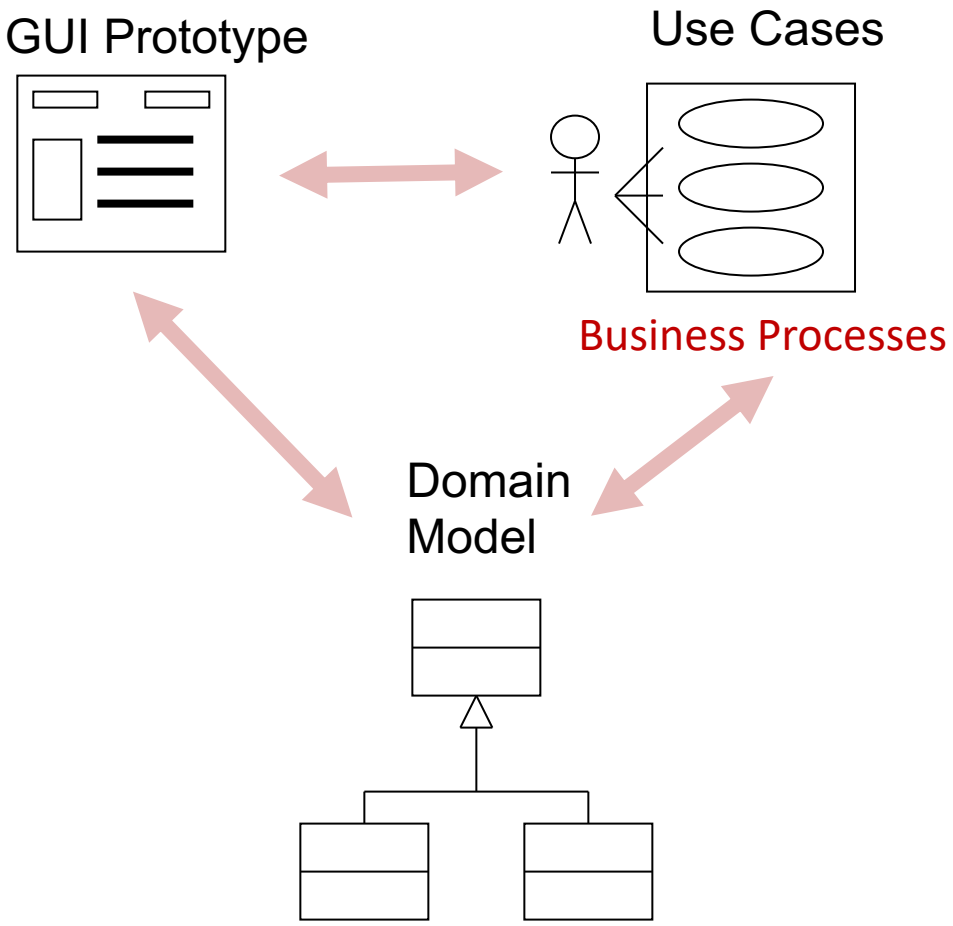
Bước 3: Thiết kế

- Xác định các thông điệp gửi giữa các đối tượng
- Hoàn thành việc chuyển domain model thành Biểu đồ lớp
  - Bổ sung và hoàn thiện: visibility, kiểu các thuộc tính, phương thức,...
  - Quyết định các kiểu quan hệ (Liên kết, Thành phần, …)
- *Milestone 3: detailed design review*

Bước 4: Hoàn thiện

- Viết các và kiểm thử đơn vị
- Viết mã / lập trình
- Thực hiện kiểm thử tích hợp, và kiểm thử chấp thuận
- *Milestone 4: deployment*

Xem lại … Yêu cầu



- Còn được gọi là pha ”phát hiện"
  - Phát hiện cái mà KH muốn
  - Phát hiện hệ thống sẽ được xây dựng
- Mục tiêu là tạo một bộ thuật ngữ hoàn thiện

  → từ điển dữ liệu

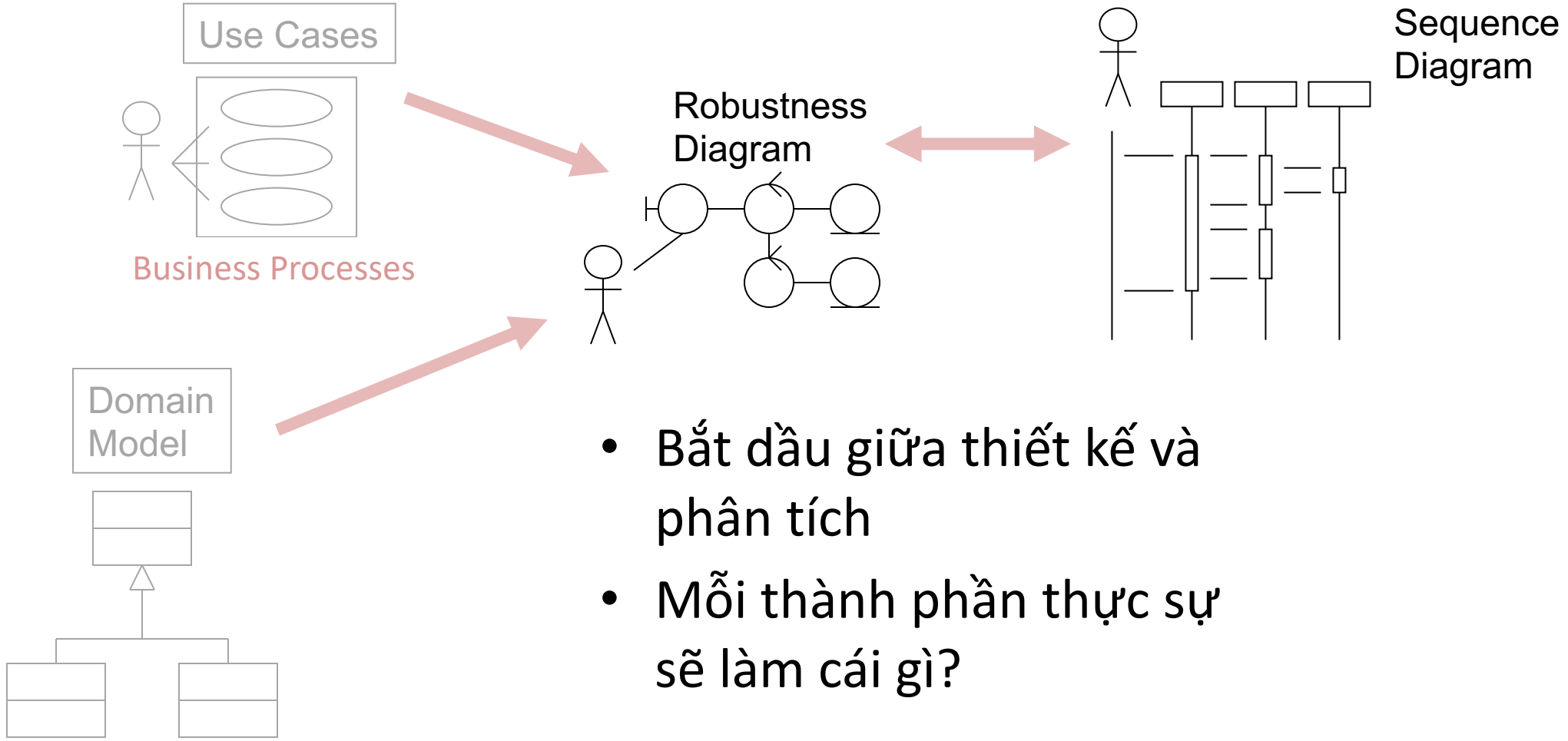
Tìm kiếm điều gì?

- Các mục từ lặp, đúp
- Đối tượng nào tương tác trực tiếp với đối tượng nào
- Đổi tượng nào tổng quát hoá cái nào
- Đối tượng nào ”sở hữu" hay ”chứa" cái nào
  - Hãy quan tâm đến xác định loại quan hệ sau (multiplicity)
  - Nhưng tạo ghi chú nếu quan hệ có thuộc tính (tính chất)

Một số lỗi trong Mô hình hoá miền

- Đừng gán các số lượng cho quan hệ (multiplicity) sớm
- Đừng cố thực hiện use-case, mô hình hoá miền, và phát thảo GUI một cách độc lập nhau
- Đừng lo lắng về các kiểu dữ liệu, các bảng CSDL quan hệ, .. tại giai đoạn này
- Đừng cố gắng ép các mẫu thiết kế (design patterns) lúc này

Xem lại … Phân tích



- Bắt dầu giữa thiết kế và phân tích
- Mỗi thành phần thực sự sẽ làm cái gì?

Mục đích Phân tích Robustness

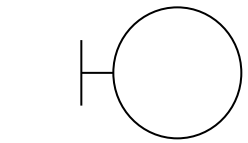
- Kiểm tra kỹ lưỡng các use-case
- Kiểm tra tính hoàn chỉnh: tất cả các case phát sinh đã được xem xét chưa?
- Tiếp tục phát hiện các đối tượng mới
- Bản thiết kế sơ bộ

… Ba kiểu Đối tượng

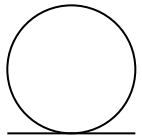
- Đối tượng **Biên (Boundary)** là những thứ mà tác nhân (ví dụ ngươi sử dụng) tương tác
  - Windows, dialogs, etc.
- Đối tượng **Thực thể (Entity)** là dữ kiệu tồn tại lâu dài
  - Các lớp dữ liệu được chuyển thành các bảng dữ liệu
- Đối tượng **Điều khiển (Control)** chứa trí tuệ của ứng dụng (các thuật toán, các phương thức xử lý,…)

- Và, Vâng, đây chính là ***Model-View-Controller***…

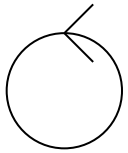
Vẽ các Biểu đồ Robustness



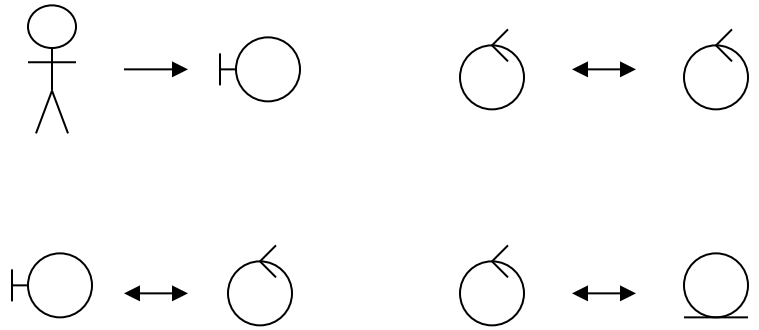
Boundary object



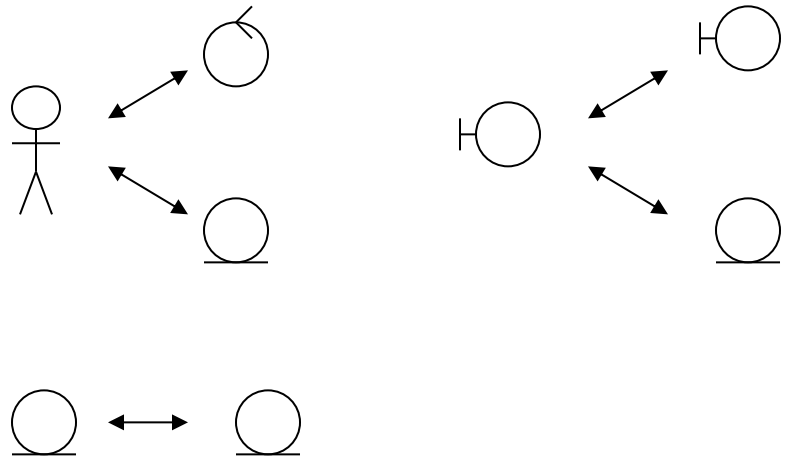
Entity object



Control object



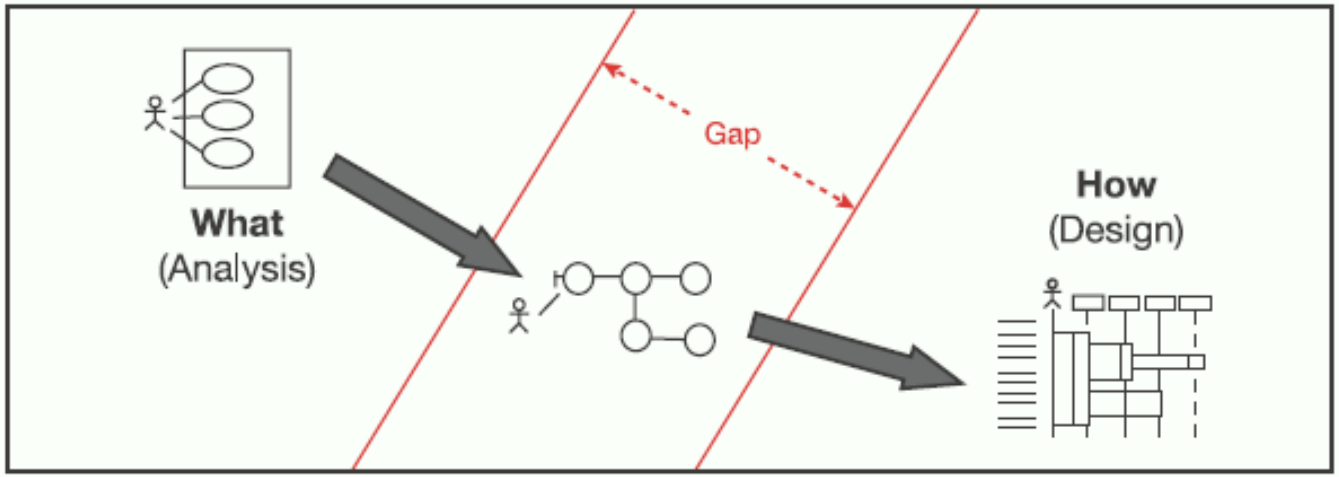
Allowed



Not Allowed

Phân tích Tương tác

- Phân tích robustness cho ta biết đối tượng cụ thể nào trao đổi nhau
- Phân tích tương tác cho ta biết chúng nói gì
- Giúp dịch các use-case sang các biểu đồ tuần tự
  - Nhưng, việc dịch này không thể thực hiện trực tiếp vì chúng ta không biết đối tượng nào ta có

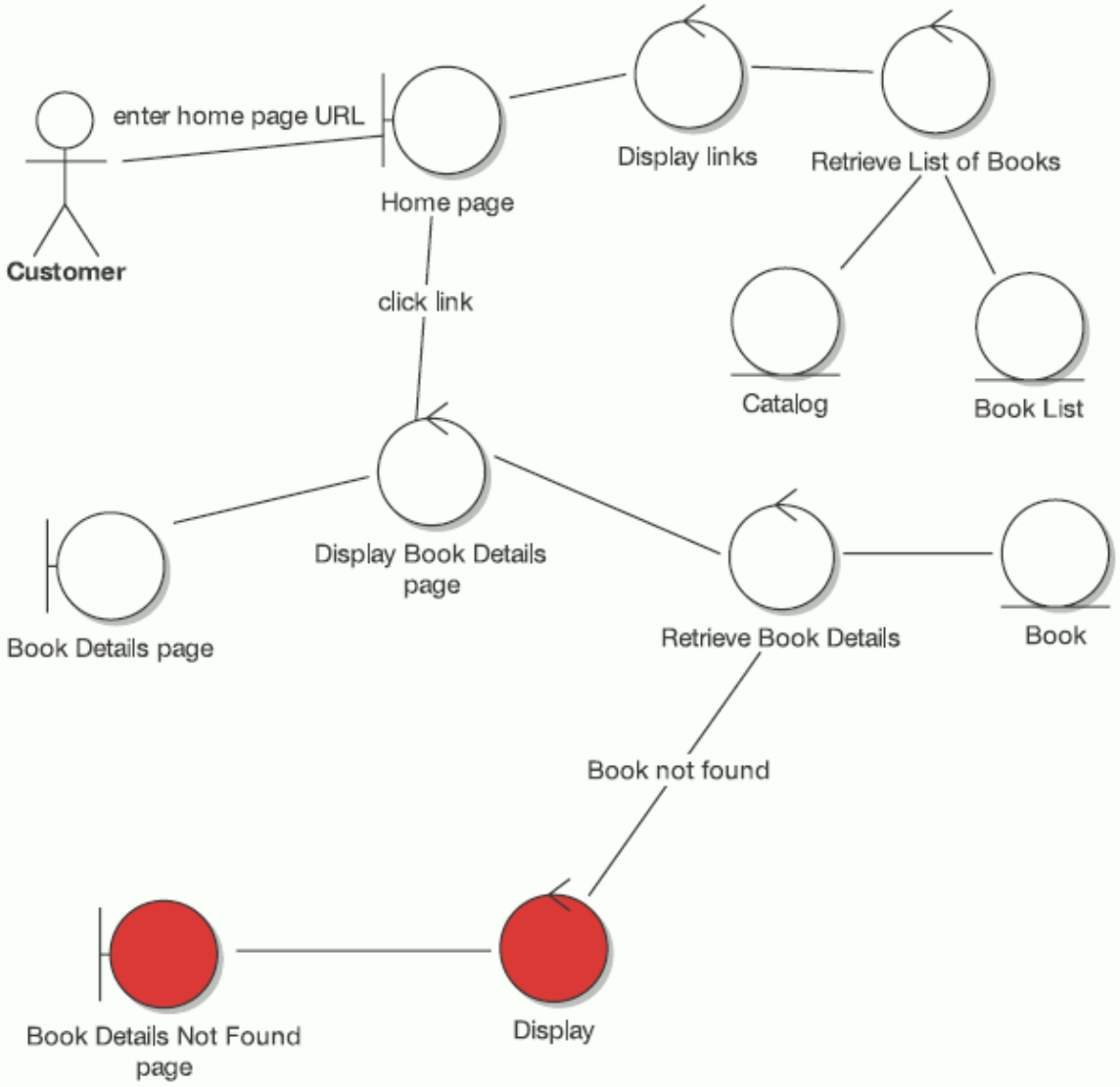


MULTIMEDIA

BASIC COURSE:
The Customer types in the URL for the bookstore's home page. The system displays a list of books from the Catalog on the home page, in the form of clickable links.

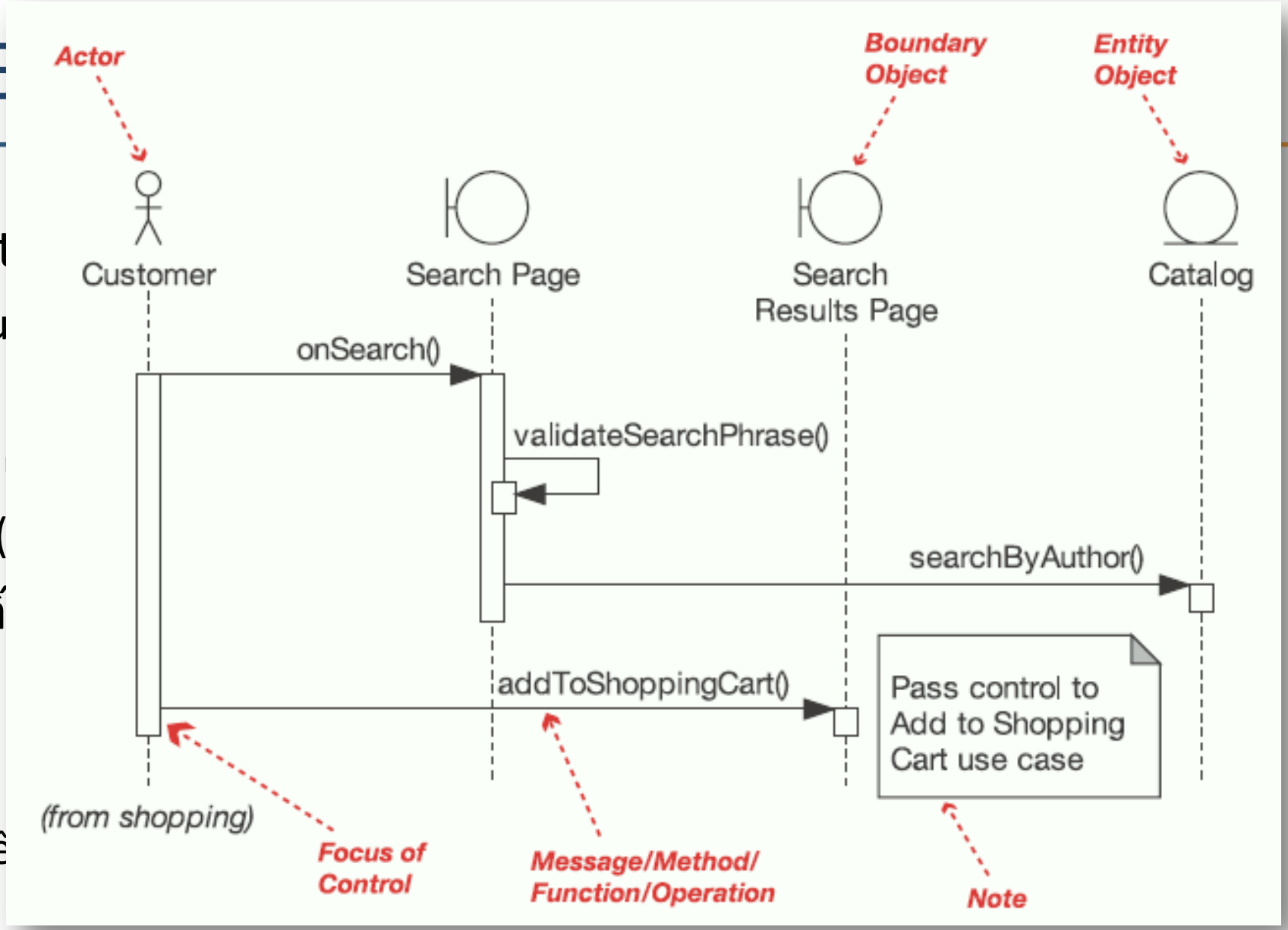
The Customer clicks a link on the home page, and the system retrieves the book details for the selected book and displays them on the Book Details page.

ALTERNATE COURSES:
Book Not Found: The system displays a Book Details Not Found page.



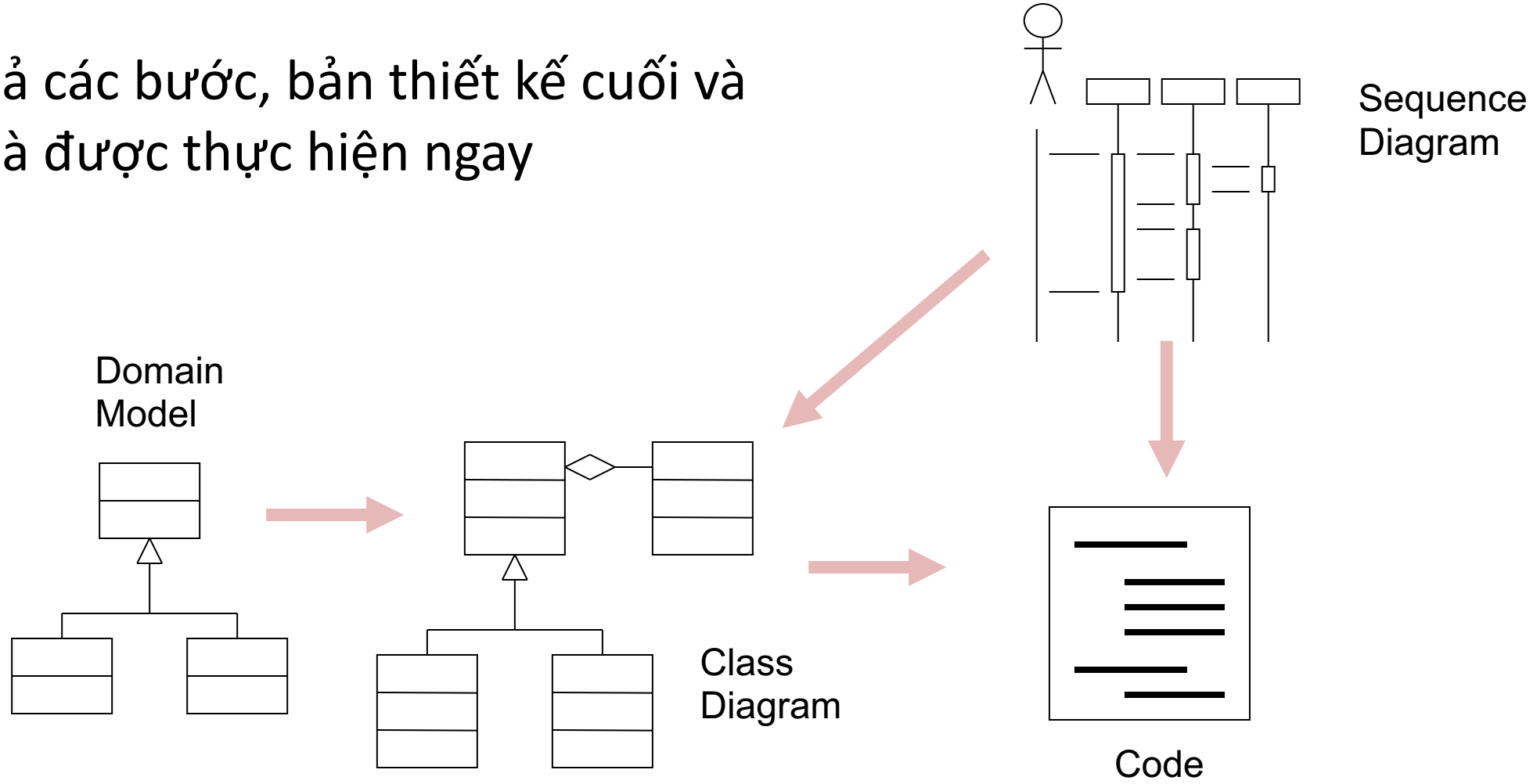
Ví dụ:
Phân tích Tương tác:
UC → Robustness

- Một biểu đồ tuần t
  - Tên và mô tả của u
  - Các đối tượng (sử
  - Các thông điệp (m
  - Tên phương thức (
- Các công cụ cao cấ
- Đây là lúc sử dụng
  - Nếu chúng ta quyê



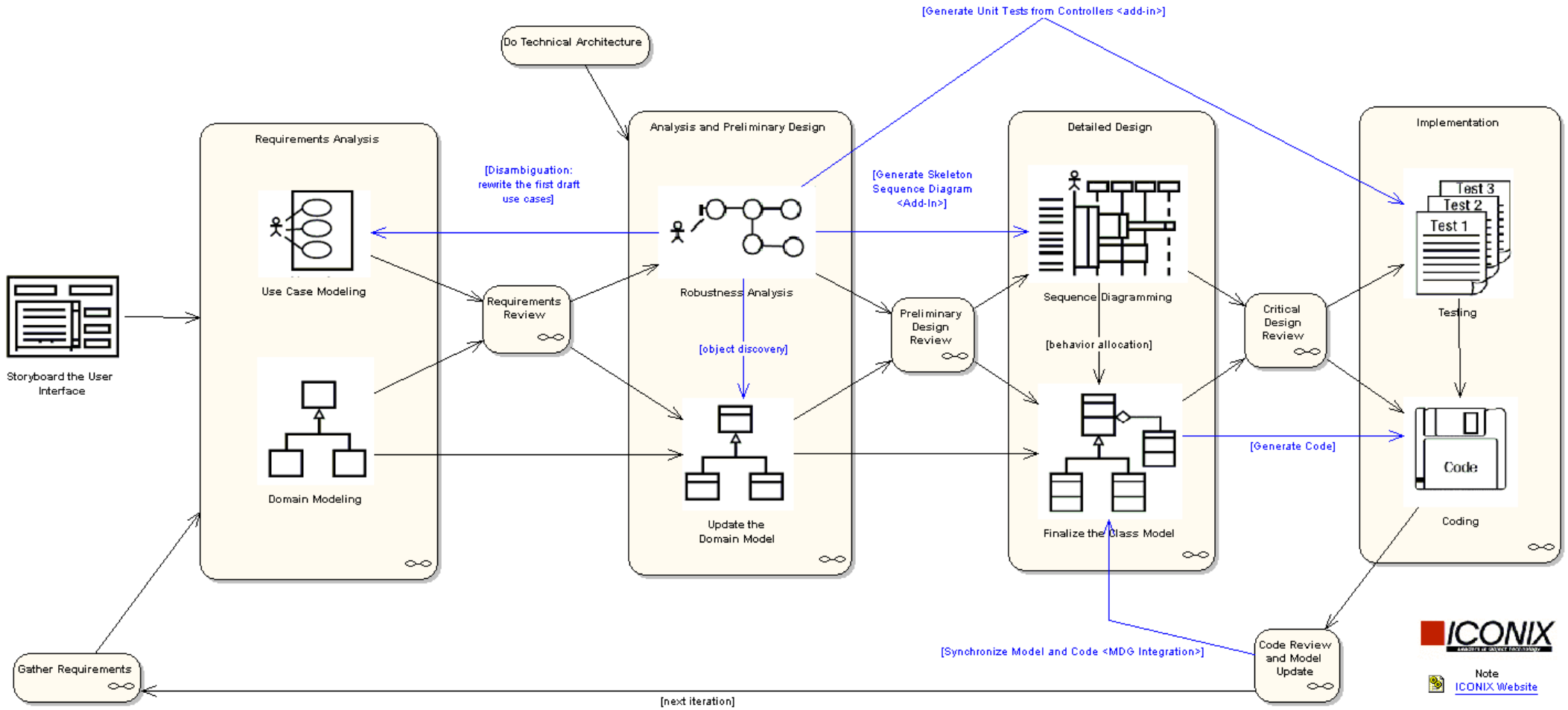
Các Lớp và viết Mã

- Sau tất cả các bước, bản thiết kế cuối và viết mã là được thực hiện ngay



Tóm lại

ICONIX is a “lite” version of Rational Unified Process (RUP) and its kin



MULTIMEDIA
Faculty of Multimedia

THẢO LUẬN

Q&A

XIN CẢM ƠN!

THẦY CÔ ĐÃ LẮNG NGHE

MÔ HÌNH HOÁ HƯỚNG ĐỐI TƯỢNG THEO USE-CASE

PGS.TS. HOÀNG HỮU HẠNH

FMM@PTIT