



Chương 2

Quản lý đề án phần mềm

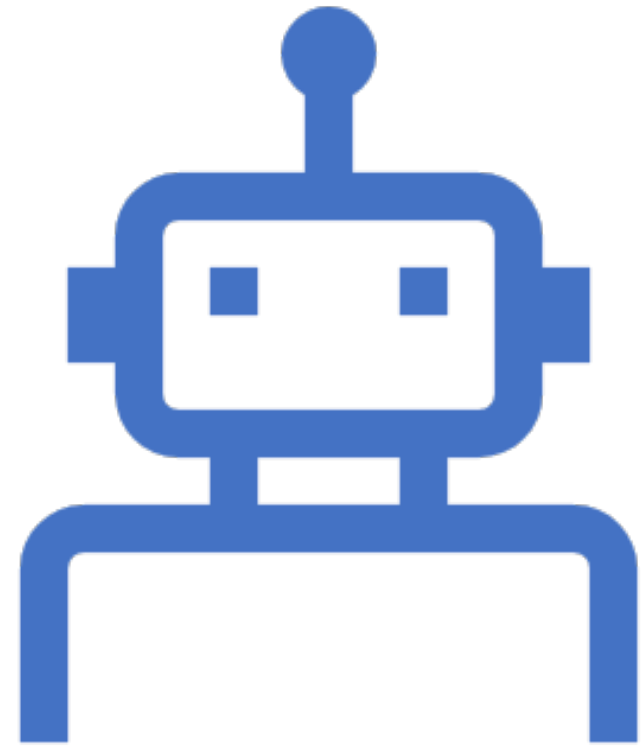
Giảng viên: TS. NGUYỄN THÀNH HUY

Nội dung chương 2

Cái nhìn chung về công nghệ phần mềm

Các mô hình, tiến trình phát triển phần mềm

Cái nhìn chung về công nghệ phần mềm



Vì sao cần quản lý dự án phần mềm?


- Phát triển phần mềm hiện đại làm theo teamworks
- Các dự án phần mềm đòi hỏi nhiều nguồn nhân lực với chuyên môn khác nhau
- Bảo đảm tính chuyên nghiệp trong phát triển dự án phần mềm:
 - Bảo đảm lịch trình của dự án
 - Điều phối và khai thác tối đa nguồn nhân lực hiện có
 - Bảo đảm chất lượng của sản phẩm
 - Khả năng khắc phục các sự cố xảy ra khách quan
- Dự án càng lớn càng cần có sự quản lý chặt chẽ và đồng bộ

Nhân tố con người

- Nhân tố con người đóng 1 vai trò hết sức quan trọng trong CNPM được gọi là “mô hình tính trưởng thành về khả năng quản lý con người” (people management capability maturity model: PM-CMM).
- Mô hình QL phần mềm được phát triển để làm tăng tính sẵn sàng của tổ chức phần mềm trong việc đảm trách những ứng dụng phức tạp ngày càng gia tăng bằng cách:
 - Thu hút -> Phát triển -> Thúc đẩy -> Triển khai -> Giữ lại những tài năng cần thiết để cải tiến khả năng phát triển phần mềm.

Những người tham gia trong dự án phần mềm

- Chia làm 5 loại:
 - **A1. Senior Managers:** định nghĩa ra những chiến lược kinh doanh và có những ảnh hưởng đáng kể đến dự án.
 - **A2. Project (Technical) Managers:** lập kế hoạch, thúc đẩy tinh thần làm việc, tổ chức và kiểm soát những thành viên làm công việc phần mềm.
 - **A3. Practitioners:** vận dụng những kỹ năng kỹ thuật của mình để làm sản phẩm hay ứng dụng.
 - **A4. Customers:** đưa ra những yêu cầu phần mềm và những stakeholders khác.
 - **A5. End-users:** sử dụng phần mềm khi phần mềm đưa ra sử dụng.



Project manager (PM), Project leader (PL) và Team leader (TL)

- **Motivation** (tính thúc đẩy, động lực)
- **Organization** (tính tổ chức)
- **Ideas or innovation** (sáng kiến và sự cải tiến mới)
- **Problem solving** (giải quyết vấn đề)
- **Managerial identity** (đặc tính quản lý)
- **Achievement** (thành tích)
- **Influence and team building** (ảnh hưởng và xây dựng đội)

Đội phần mềm (software team)

- **Democratic decentralized (DD)** (phân quyền dân chủ)
- **Controlled decentralized (CD)** (phân quyền có kiểm soát)
- **Controlled centralized (CC)** (tập trung kiểm soát)

CHÚ Ý: Tùy theo từng dự án phần mềm ta có các team khác nhau.



Tiến trình phần mềm (software process)

Tổng quan về tiến trình phần mềm

- Tiến trình phần mềm là một tập hợp các hoạt động để sản xuất phần mềm theo từng giai đoạn. Những hoạt động này liên quan đến việc phát triển phần mềm từ những thứ lộn xộn vào trong ngôn ngữ lập trình chuẩn (ví dụ: JAVA, C++, C#)

Công việc của công nghệ phần mềm

- Phân tích hệ thống/vấn đề
- Xác định các yêu cầu
- Thiết kế phần mềm
- Viết phần mềm (coding)
- Kiểm tra và tích hợp hệ thống
- Cài đặt và chuyển giao phần mềm
- Lập tài liệu
- Bảo trì
- Quản lý chất lượng
- Huấn luyện
- Dự đoán tài nguyên
- Quản trị dự án

Tiến trình chung cho tất cả các Tiến trình

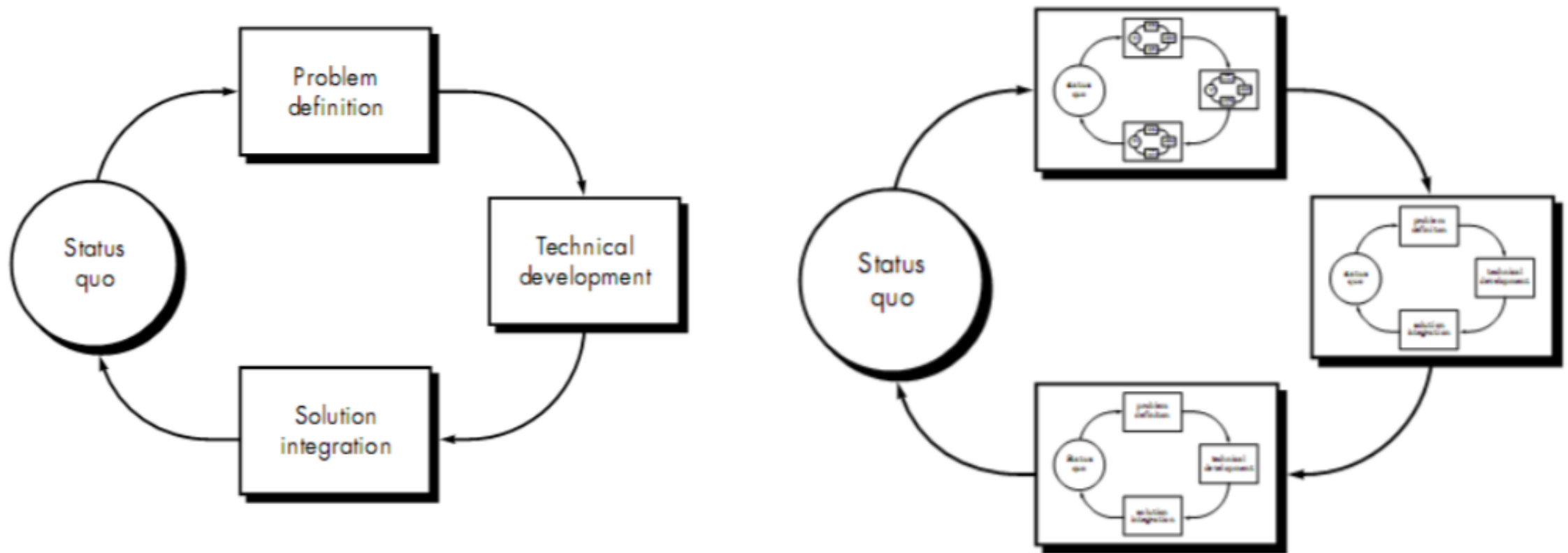
Đặc tả phần mềm (Software/problem specification): chức năng của phần mềm và những ràng buộc trong hoạt động của nó phải được định nghĩa.

Thiết kế và thực thi phần mềm (Technical development): phần mềm đáp ứng đặc tả phải được tạo ra.

Tiến triển phần mềm (Solution Integration): phần mềm phải tiến triển để đáp ứng những nhu cầu thay đổi của khách hàng.

Chứng nhận phần mềm (Status Quo): phần mềm phải được chứng nhận để đảm bảo rằng nó làm những gì mà khách hàng muốn.

Vòng lặp 4 giai đoạn cơ bản

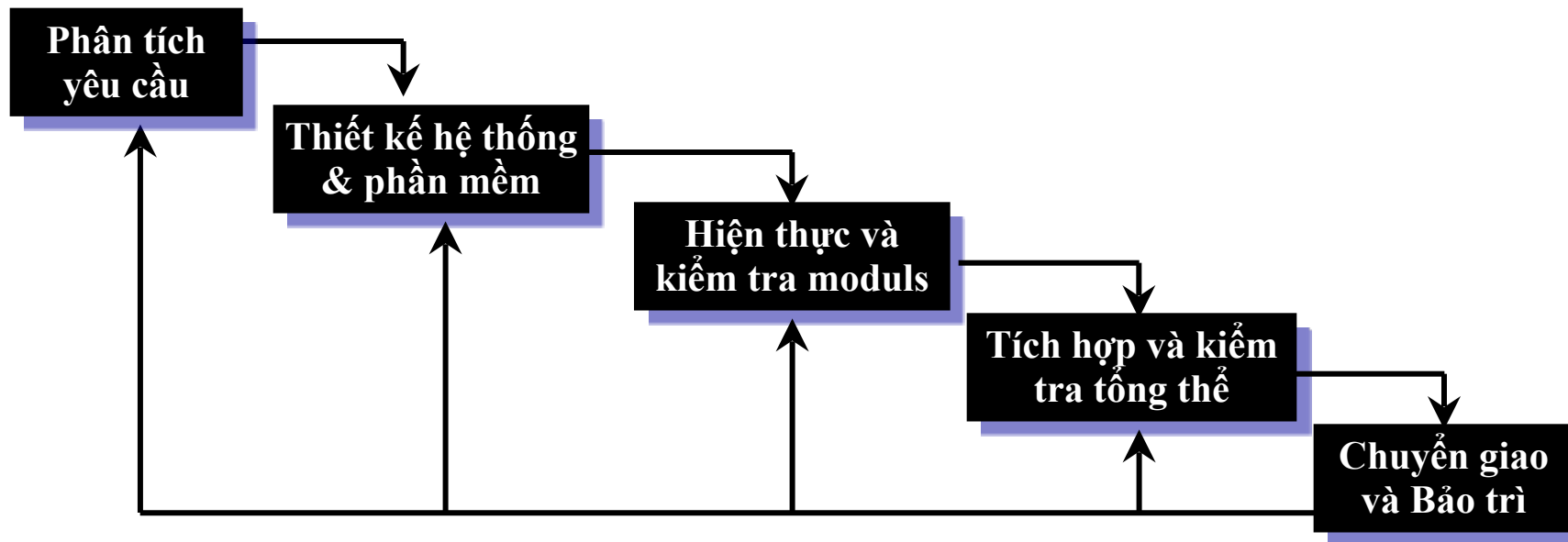


Các Tiến trình phần mềm phổ biến


- Mô hình thác nước (the waterfall model)
- Mô hình mẫu (the prototyping model)
- Mô hình phát triển ứng dụng nhanh RAD (the rapid application development model)
- Mô hình tiến hóa (evolutionary development model)
 - + Mô hình gia tăng (incremental model)
 - + Mô hình xoắn ốc (the spiral model)
 - + Mô hình xoắn ốc WINWIN (the WINWIN spiral model)
- **Mô hình Agile**

Mô hình WaterFall

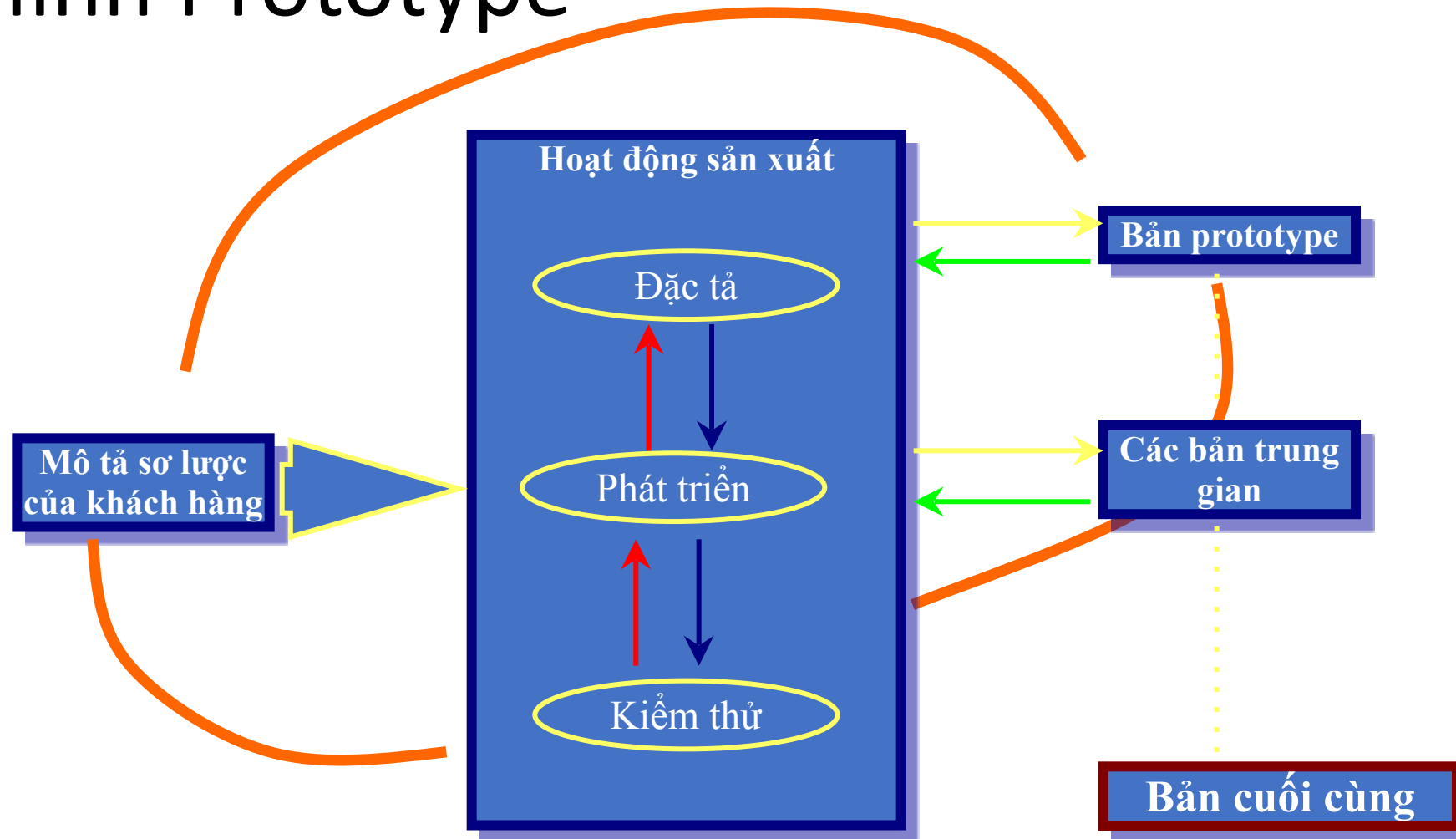
- Mô hình phát triển phần mềm đầu tiên
- Các công việc tiếp nối nhau một cách tuần tự
- Đặt nền móng cho các phương pháp phân tích, thiết kế, kiểm tra...



Mô hình WaterFall

- Bản chất của phát triển phần mềm là quá trình lặp đi lặp lại chứ không phải tuần tự
- Các bước thực chất không tách biệt hoàn toàn mà có chồng lấn và tham khảo lại
- Bắt buộc khách hàng đặc tả tất cả yêu cầu một cách chính xác và đầy đủ ngay từ ban đầu
- Khách hàng thường phải chờ đợi rất lâu để thấy được phiên bản đầu tiên của sản phẩm
- Tồn tại “delay” tích lũy trong nhóm làm việc  dự án thường bị trễ.
- Chỉ phù hợp cho dự án nhỏ, đơn giản.

Mô Hình Prototype



Mô Hình Prototype

- Prototype như là một cơ chế để nhận diện chính xác yêu cầu của khách hàng
 - Bản thân khách hàng chưa hiểu rõ yêu cầu của mình, cũng như các quy trình chưa được xác lập rõ ràng.
 - Khách hàng chưa hiểu rõ khả năng hỗ trợ của hệ thống máy tính
- Kích thích sự thích thú của người dùng với dự án
- Prototype có thể bị “throw-away” [?] Lãng phí
- Các process không được phân định rõ ràng
- Hệ thống thông thường có cấu trúc lỏng lẻo
- Cần có những kỹ năng đặc biệt trong quản lý và phát triển
- Khách hàng hối thúc nhà phát triển hoàn thành sản phẩm một khi thấy được các prototype đầu tiên

Mô Hình Prototype – Ứng dụng

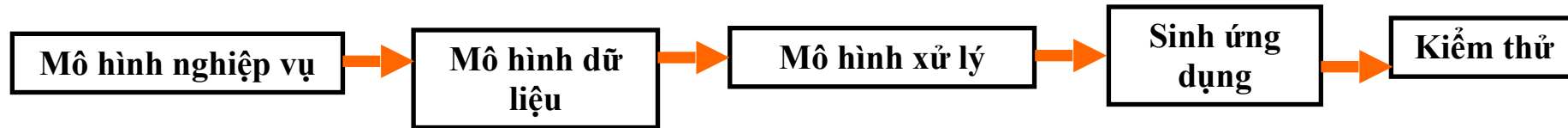
- Dùng cho các hệ thống nhỏ. Các chi phí khi thay đổi hệ thống là không quá lớn khi cần phải thay đổi sau khi thực hiện prototype
- Cần sự cấp bách về thời gian triển khai ngắn. Hệ thống cần được đưa vào ứng dụng từng phần trong khoảng thời gian nhất định.
- Trong trường hợp những hệ thống mà việc đặc tả các yêu cầu là rất khó và không rõ ràng ngay từ đầu.

Mô hình Xoắn Ốc



- Được thực hiện theo một chuỗi lặp kiểu xoắn ốc, mỗi lần lặp cải thiện sản phẩm
- Có phương pháp đánh giá rủi ro
- Có thể áp dụng prototype
- Mỗi lần lặp được cải thiện cho thích nghi với bản chất của đề án

Mô hình RAD



- Rapid Application Development là mô hình tuần tự tuyến tính có thời gian phát triển rất ngắn
- Sử dụng các thành phần có sẵn càng nhiều càng tốt
- Sử dụng công cụ lập trình ở dạng tự động sinh mã chứ không phải các ngôn ngữ truyền thống
- Phụ thuộc vào công nghệ phát triển có tính reusable cao.
- Phát triển hệ thống mẫu