



1



2

## Nội dung

- ❖ Các quy trình thiết kế giao diện
  - ✓ Mô hình thác nước (waterfall model)
  - ✓ Mô hình lặp (iterative design)
  - ✓ Mô hình xoắn ốc (spiral model)
  - ✓ **Mô hình thiết kế hướng người dùng (User-centered design)**
- ❖ Các nguyên tắc và quy luật trong thiết kế giao diện



3

## Giới thiệu

- ❖ Thiết kế giao diện là một bộ phận của phát triển phần mềm
- ❖ Quy trình thiết kế phụ thuộc vào quy trình phát triển
- ❖ Quy trình thiết kế được tích hợp với quy trình phát triển
- ❖ Thiết kế phần giao diện được thực hiện ở nhiều giai đoạn khác nhau của dự án
  - ✓ Không chỉ ở giai đoạn ban đầu

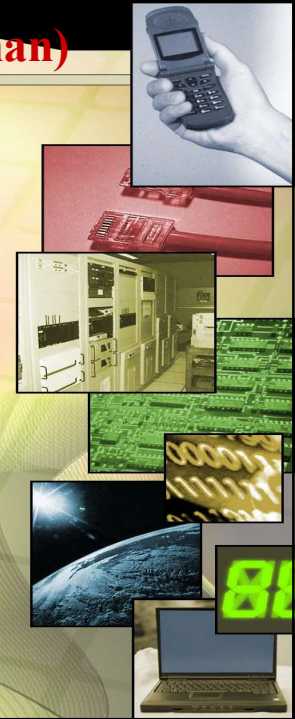


4



## Vòng đời của hệ thống tương tác (Shneiderman)

- 1) Thu thập thông tin
- 2) Xác định yêu cầu và ngữ nghĩa
- 3) Xác định cấu trúc và trang bị hỗ trợ
- 4) Đặc tả các thiết bị
- 5) Phát triển phần mềm
- 6) Tích hợp hệ thống và triển khai đến người dùng
- 7) Phát triển cộng đồng người dùng
- 8) Chuẩn bị kế hoạch cải tiến trong tương lai

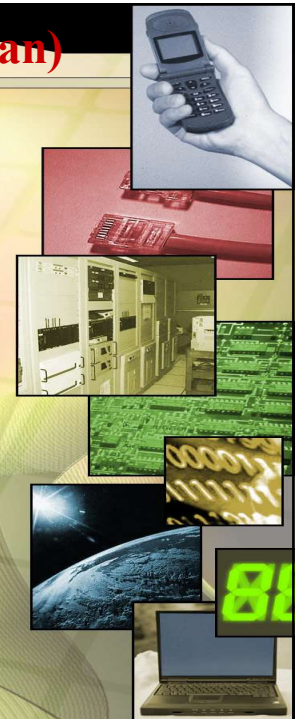


5

## Vòng đời của hệ thống tương tác (Shneiderman)

### 1. Thu thập thông tin

- ✓ Tổ chức nhóm thiết kế
- ✓ Cần có sự tham gia của quản lý và khách hàng
- ✓ Tiến hành phỏng vấn người dùng
- ✓ Gửi bảng câu hỏi khảo sát đến người dùng
- ✓ Ước lượng chi phí phát triển, huấn luyện, sử dụng và bảo trì
- ✓ Lập kế hoạch với các mốc thời gian và các đợt review rõ ràng, tường minh



6

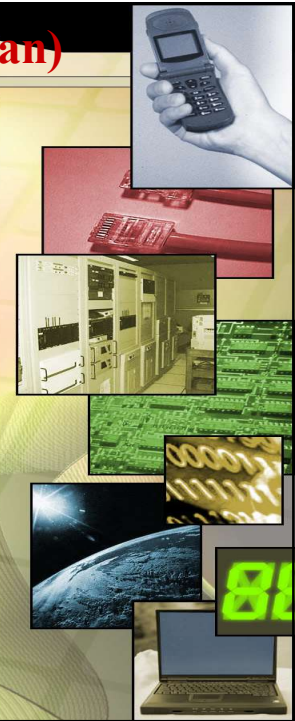
## Vòng đời của hệ thống tương tác (Shneiderman)

### 2. Xác định yêu cầu và ngữ nghĩa

- ✓ Xác định các yêu cầu phần mềm
- ✓ Xác định các luồng công việc
- ✓ Tạo các đối tượng và hành động của các tác vụ

### 3. Xác định cấu trúc và trang bị hỗ trợ

- ✓ Xem xét các dạng hiển thị khác nhau
- ✓ Thiết kế kiểu phản hồi cho các hành động
- ✓ Duyệt, đánh giá và hiệu chỉnh đặc tả cho thiết kế
- ✓ Đánh giá bản prototype, pilot tests (thử nghiệm)



7

## Vòng đời của hệ thống tương tác (Shneiderman)

### 4. Đặc tả các thiết bị

- ✓ Chọn các thiết bị phần cứng, phần mềm
- ✓ Chọn các thiết bị âm thanh, hình ảnh và các thiết bị ngoại vi khác
- ✓ Xem xét môi trường làm việc: độ ồn, ánh sáng, không gian
- ✓ Tiếp tục thực hiện pilot tests và hiệu chỉnh thiết kế

### 5. Phát triển phần mềm

- ✓ Sử dụng các công cụ phát triển phù hợp
- ✓ Cài đặt
- ✓ Kiểm thử đơn vị (unit testing)



8

## Vòng đời của hệ thống tương tác (Shneiderman)

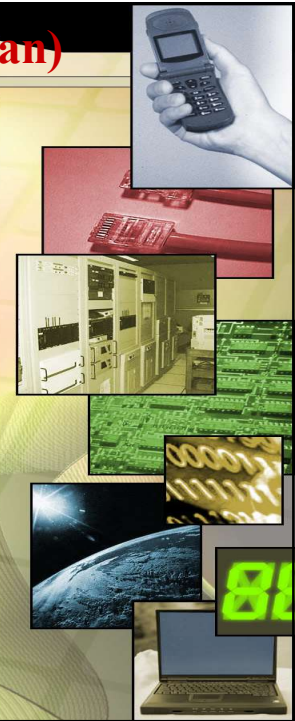
### 6. Tích hợp hệ thống và triển khai đến người dùng

- ✓ Đảm bảo sự tham gia của khách hàng ở mỗi giai đoạn
- ✓ Thực hiện kiểm thử chấp nhận và hiệu chỉnh hệ thống
- ✓ Viết tài liệu và hướng dẫn người dùng sử dụng hệ thống

### 7. Phát triển (nurture) cộng đồng người dùng

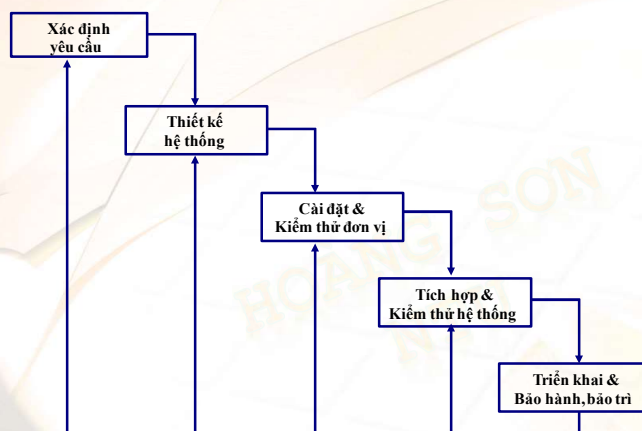
- ✓ Hỗ trợ người dùng
- ✓ Giám sát quá trình sử dụng và đưa ra đánh giá

### 8. Chuẩn bị kế hoạch cải tiến trong tương lai



9

## Mô hình thác nước (waterfall model)



👉 Bất lợi: khó xử lý với các thay đổi

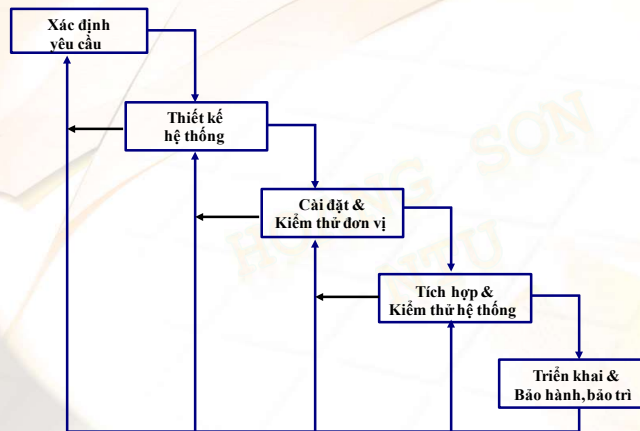


10



## Mô hình thác nước (waterfall model)

### ❖ Mô hình thác nước cải tiến



11

## Mô hình thác nước (waterfall model)

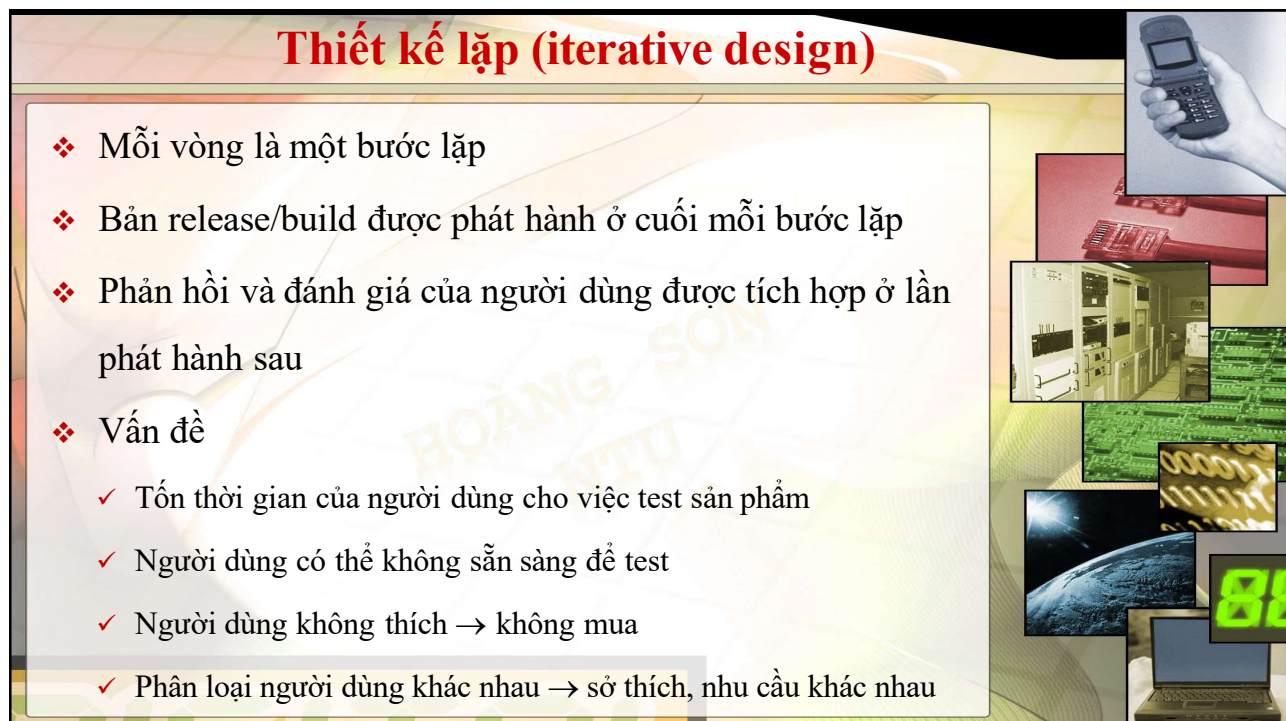
### ❖ Vấn đề của mô hình thác nước

- ✓ Người dùng chỉ tham gia vào quá trình đánh giá ở bước cuối cùng, kiểm thử chấp nhận (acceptance testing)
- ✓ Lỗi giao diện gây ra sự thay đổi trong yêu cầu và thiết kế
  - Lãng phí công sức giai đoạn ban đầu
- ✓ Chia dự án thành các giai đoạn riêng biệt một cách cứng nhắc
  - Khó đáp ứng tốt sự thay đổi yêu cầu từ phía khách hàng
- ✓ Chỉ tích hợp khi yêu cầu đã được hiểu rõ
  - Ít hệ thống phần mềm có được yêu cầu ổn định, rõ ràng

12

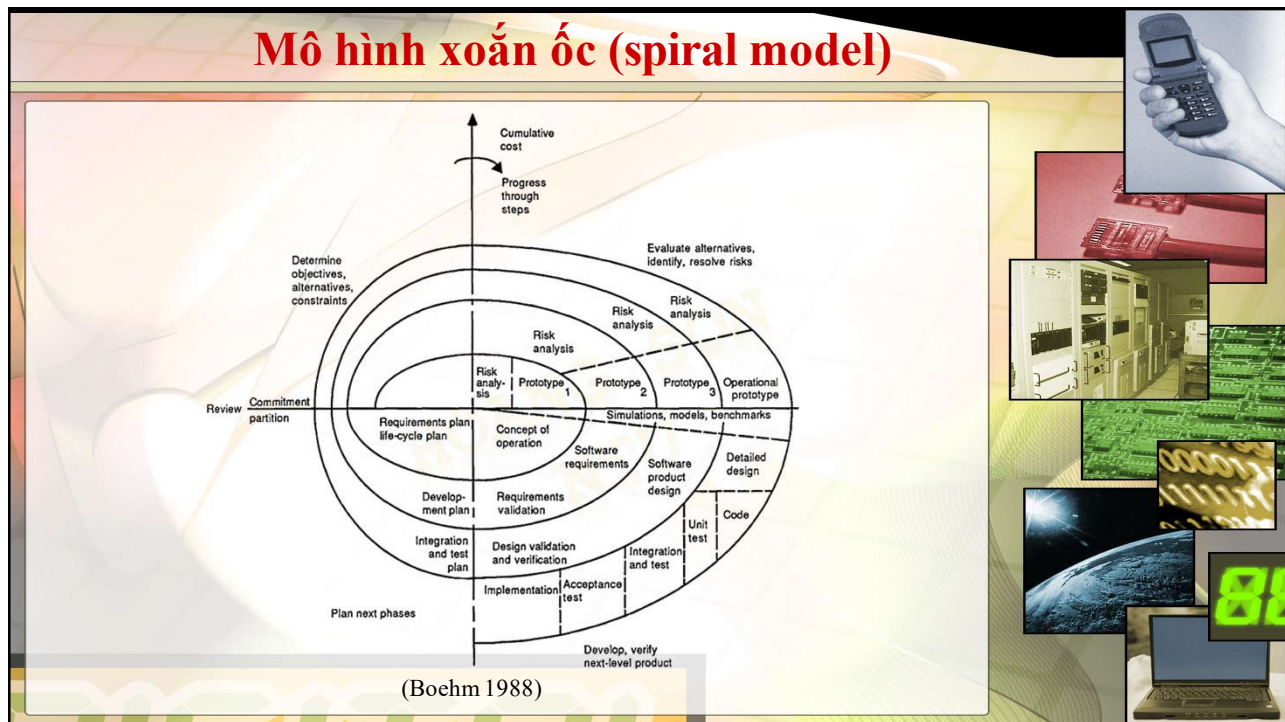


13



14

## Mô hình xoắn ốc (spiral model)



15

## Mô hình xoắn ốc (spiral model)

- ❖ Quy trình được thể hiện dưới dạng xoắn ốc thay vì một chuỗi các hoạt động liên tiếp
- ❖ Mỗi lần lặp trong mô hình xoắn ốc tương ứng với một pha trong quy trình
- ❖ Không tồn tại các pha cố định như đặc tả hay thiết kế
  - ✓ Các lần lặp trong mô hình xoắn ốc được chọn dựa trên yêu cầu
- ❖ Rủi ro định đánh giá và giải quyết trong suốt quy trình

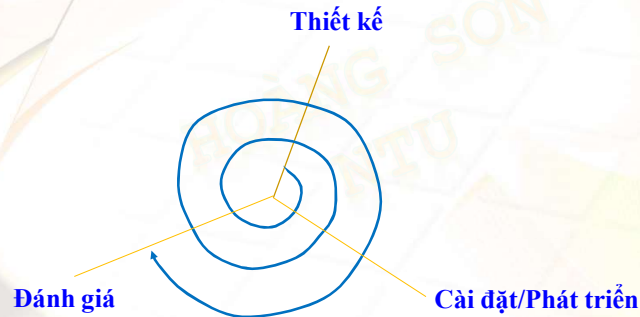
16



## Mô hình xoắn ốc (spiral model)

### ❖ Mô hình xoắn ốc cho thiết kế giao diện

Là một bước cải tiến của mô hình thiết kế lặp



17

## Mô hình xoắn ốc (spiral model)

### ❖ Mô hình xoắn ốc cho thiết kế giao diện – tt

- ✓ Các vòng lặp đầu sử dụng các prototype chi phí thấp
  - Prototype giấy
  - Bản phác thảo trên máy tính
  - Sử dụng các công cụ vẽ prototype nhanh
- ✓ Cung cấp nhiều prototype thay thế khác nhau
  - Parallel prototyping (thiết kế prototype song song)
- ✓ Các vòng lặp sau sẽ cải tiến tốt hơn
- ✓ Chỉ các bản release trưởng thành ở các vòng lặp sau mới được gửi chuyển giao cho người dùng

18

## Thiết kế hướng người dùng (user-centered design)

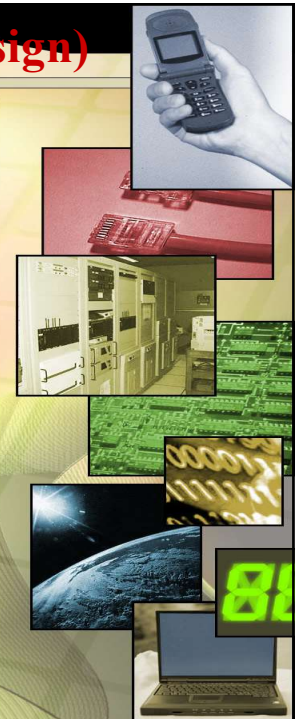
- ❖ Còn được gọi là *Participatory Design*
- ❖ Một dạng mô hình thiết kế xoắn ốc
- ❖ Tập trung vào người dùng và tác vụ
  - ✓ Phân tích người dùng: ai sử dụng hệ thống này
  - ✓ Phân tích tác vụ: người dùng cần làm gì trên hệ thống này
- ❖ Đặt người dùng vào quy trình phát triển
  - ✓ Người dùng đóng vai trò là người đánh giá, người tư vấn và đôi khi cũng là người thiết kế
- ❖ Liên tục đánh giá
  - ✓ Người dùng đánh giá các bản prototype và các bản release



19

## Thiết kế hướng người dùng (user-centered design)

- ❖ **Thuận lợi**
  - ✓ Có được thông tin chính xác và gợi ý hữu ích
  - ✓ Cơ hội để tranh luận giữa các lựa chọn thiết kế
  - ✓ Cho phép người dùng có cơ hội tác động đến thiết kế
- ❖ **Bất lợi tiềm ẩn**
  - ✓ Người dùng không phải lúc nào cũng sẵn sàng để tham gia vào quy trình phát triển
  - ✓ Thời gian của người dùng là quý báu
  - ✓ Người dùng không phải là người thiết kế giao diện
  - ✓ Người dùng có cái tôi mạnh mẽ và sở thích riêng
  - ✓ Người thiết kế giao diện tuân thủ quá mức vào sở thích của người dùng



20



## Các nguyên tắc và quy luật trong thiết kế giao diện

### ❖ Xác định mức độ kỹ năng của người dùng

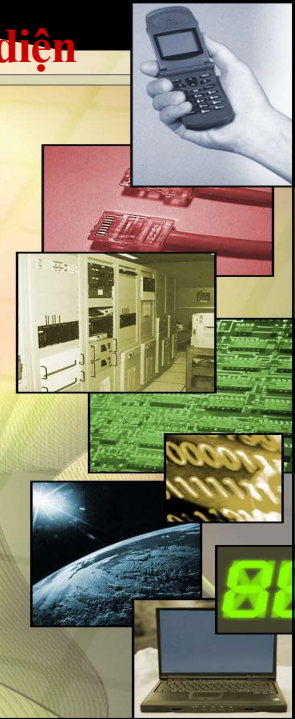
- ✓ Người dùng mới (novice/first-time users)
- ✓ Người dùng có hiểu biết nhất định (knowledgeable intermittent users)
- ✓ Người dùng là chuyên gia, sử dụng thường xuyên (experts and frequent users)

### ❖ Xác định các tác vụ

- ✓ Các tác vụ thường thực hiện
- ✓ Các tác vụ ít thực hiện hơn
- ✓ Các tác vụ rất ít/hiếm khi thực hiện

### ❖ Chọn kiểu tương tác phù hợp

- ✓ Thao tác trực tiếp
- ✓ Lựa chọn menu
- ✓ Điền dữ liệu vào form
- ✓ Ngôn ngữ dòng lệnh
- ✓ Ngôn ngữ tự nhiên



21

## Các nguyên tắc và quy luật trong thiết kế giao diện

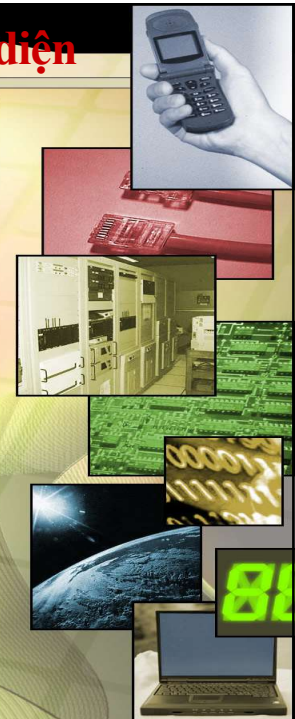
### ❖ Áp dụng 8 quy tắc vàng của Shneiderman khi thiết kế giao diện

#### ❖ Ngăn ngừa lỗi

- ✓ Thông báo lỗi cần đầy đủ thông tin và có tính xây dựng
- ✓ Tổ chức các màn hình và menu theo chức năng
- ✓ Cung cấp phản hồi về trạng thái hiện tại của giao diện (hình ảnh, âm thanh, haptic)
- ✓ Chỉ cho phép chức năng phù hợp (correct actions)
  - Ví dụ, tô xám menu đang được chọn
- ✓ Hoàn chỉnh chuỗi hành động (complete sequences)
  - Ví dụ, cửa sổ dạng wizard lúc nào cũng phải có nút Next và Finish

#### ❖ Tăng tính tự động nhưng vẫn đảm bảo sự điều khiển của người dùng

- ✓ Sử dụng auto suggestion
- ✓ Sử dụng auto completion
- ✓ Cho phép người dùng thay đổi



22

## 8 quy tắc vàng của Shneiderman (1987)

1. **Cố gắng đảm bảo tính nhất quán:** Các thông báo được sử dụng thống nhất ở menu, màn hình trợ giúp, popup và thống nhất trong các mệnh lệnh xuyên suốt trang web.
2. **Hỗ trợ tính khả dụng phổ quát (universal):** sử dụng phím tắt, lệnh ẩn...hỗ trợ người dùng giảm bớt tương tác.
3. **Cung cấp phản hồi với đầy đủ thông tin.**
4. **Thiết kế hộp thoại đóng:** Các chuỗi hành động nên được xếp chung nhóm với thứ tự: ban đầu, ở giữa và kết thúc.
5. **Ngăn ngừa lỗi, phục hồi nhanh**
6. **Cho phép hủy bỏ, quay ngược thao tác dễ dàng**
7. **Cho phép người dùng khả năng điều khiển giao diện:** hỗ trợ người sử dụng khởi tạo các hành động chứ không phải chỉ dùng để phản hồi.
8. **Giảm tải bộ nhớ:** thiết kế hiển thị đơn giản, chuyển động của cửa sổ cần được giảm bớt, những thứ cần học phải được chia nhỏ, các hành động cần có trình tự.

