

Nhập môn Công nghệ phần mềm

Topic 7 Thiết kế giao diện người dùng

Giao diện người dùng

- ❑ Các giao diện người dùng cần được thiết kế sao cho phù hợp với các kỹ năng, kinh nghiệm và mong đợi của người dùng hệ thống.
 - ❑ Các người dùng hệ thống thường đánh giá một hệ thống thông qua giao diện hơn là thông qua tính năng của nó.
 - ❑ Một giao diện được thiết kế kém có thể làm cho người dùng tạo nên các lỗi nghiêm trọng.
 - ❑ Thiết kế giao diện người dùng kém cũng là một lý do chính gây ra nhiều hệ thống phần mềm không bao giờ được sử dụng.
-

Các nhân tố con người

- ☐ Không nhớ được lâu, nhiều: một người thông thường có thể nhớ đồng thời 7 thông tin khác nhau.
 - ☐ Con người có thể mắc lỗi: khi mắc lỗi, hệ thống có thể chạy sai. Nếu các thông báo lỗi không phù hợp có thể tạo áp lực cho người dùng và họ sẽ mắc nhiều lỗi hơn.
 - ☐ Người này khác người kia.
 - ☐ Mỗi người có cách tương tác hệ thống khác nhau: có người thích hình ảnh, người thích dạng văn bản...
-

Một số nguyên tắc khi thiết kế giao diện

- ❑ Thiết kế giao diện cần quan tâm đến nhu cầu, kinh nghiệm và khả năng của người sẽ dùng hệ thống.
 - ❑ Các nhà thiết kế cần hiểu được giới hạn về khả năng vật lý và trí óc của con người.
 - ❑ Các nguyên tắc áp dụng cho thiết kế giao diện được nêu ra ở đây không có nghĩa là chúng có thể áp dụng cho bất kỳ thiết kế giao diện nào.
-

Một số nguyên tắc thiết kế giao diện

- ❑ **Quen thuộc:** giao diện cần sử dụng khái niệm và từ ngữ quen thuộc theo kinh nghiệm của người sẽ dùng hệ thống.
 - ❑ **Nhất quán:** giao diện cần nhất quán
 - ❑ **Giảm thiểu các bất ngờ:** người dùng không bị bất ngờ bởi những hành động của hệ thống.
 - ❑ **Có thể phục hồi:** hệ thống cho phép người dùng thực hiện lại hay sửa chữa các lỗi có thể mắc phải
 - ❑ **Hướng dẫn:** hệ thống cung cấp đầy đủ các thông tin hướng dẫn người dùng theo ngữ cảnh tương ứng.
 - ❑ **Đa dạng người dùng:** cung cấp khả năng tương tác khác nhau cho các nhóm người dùng khác nhau.
-

Một số vấn đề về thiết kế giao diện

- Có 2 vấn đề cần quan tâm khi thiết kế giao diện người dùng:
 - Làm thế nào thông tin từ người dùng cung cấp cho hệ thống?
 - Làm thế nào thông tin của hệ thống thể hiện cho người dùng.
 - Tương tác người dùng và thể hiện thông tin được tích hợp thông qua một framework thống nhất.
-

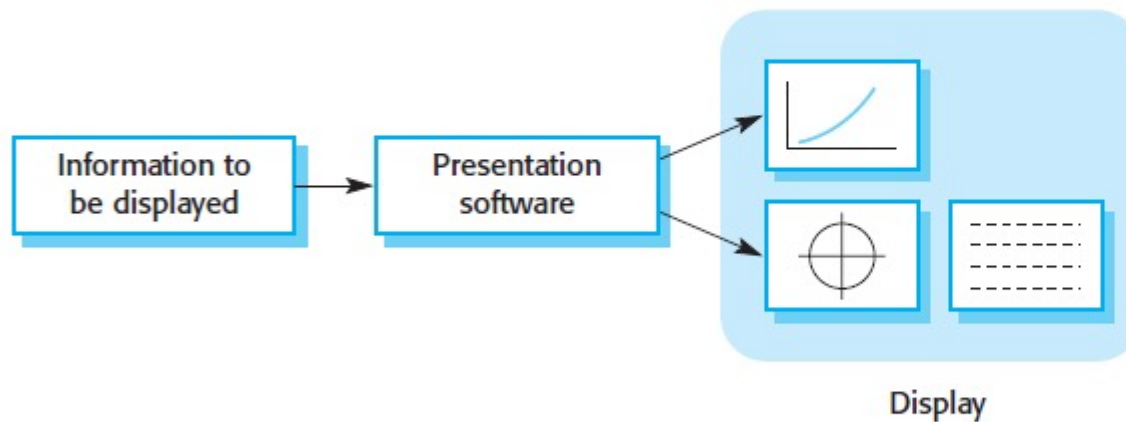
Một số kiểu tương tác

- ☐ Các tác động trực tiếp: games, CAD
 - ☐ Thông qua việc chọn trên menu: các hệ thống tổng quát
 - ☐ Điền form: banking, quản lý
 - ☐ Sử dụng lệnh command: hệ điều hành
 - ☐ Sử dụng ngôn ngữ tự nhiên: information retrieval system
 - ☐ ...
-

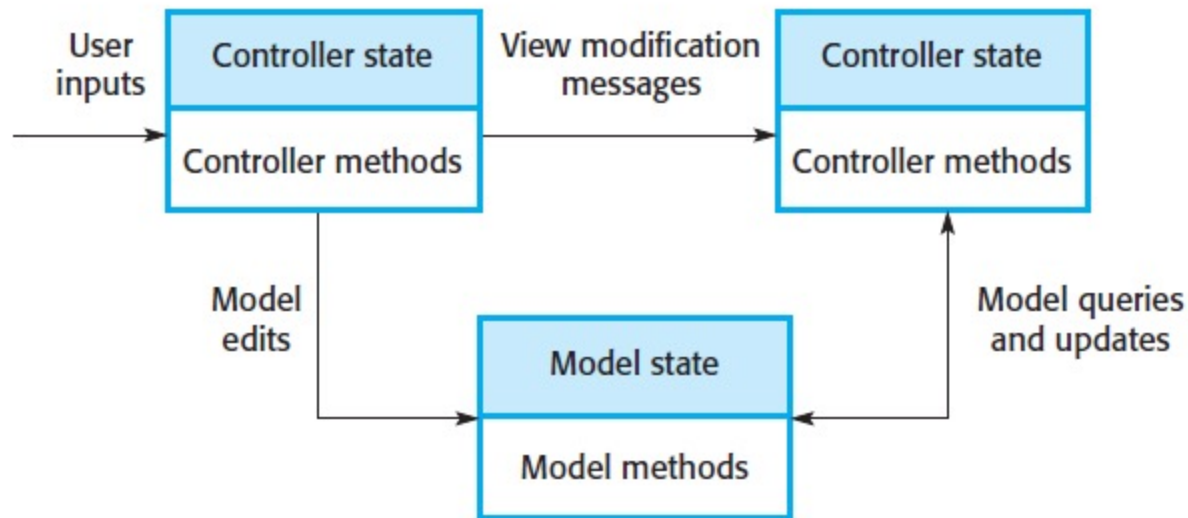
Thẻ hiện thông tin

- ❑ Thẻ hiện thông tin: hiển thị các thông tin của hệ thống cho người sử dụng.
 - ❑ Thông tin có thể thẻ hiện một cách trực tiếp (thông qua văn bản, con số...) hay được biến đổi thành các cách thẻ hiện khác (đồ thị, hình ảnh...)
 - ❑ Cách tiếp cận Model-View-Controller (MVC) là phương pháp hỗ trợ thẻ hiện thông tin bằng nhiều cách khác nhau.
-

Thẻ hiện thông tin



Model-View-Controller (MVC)



Thể hiện thông tin

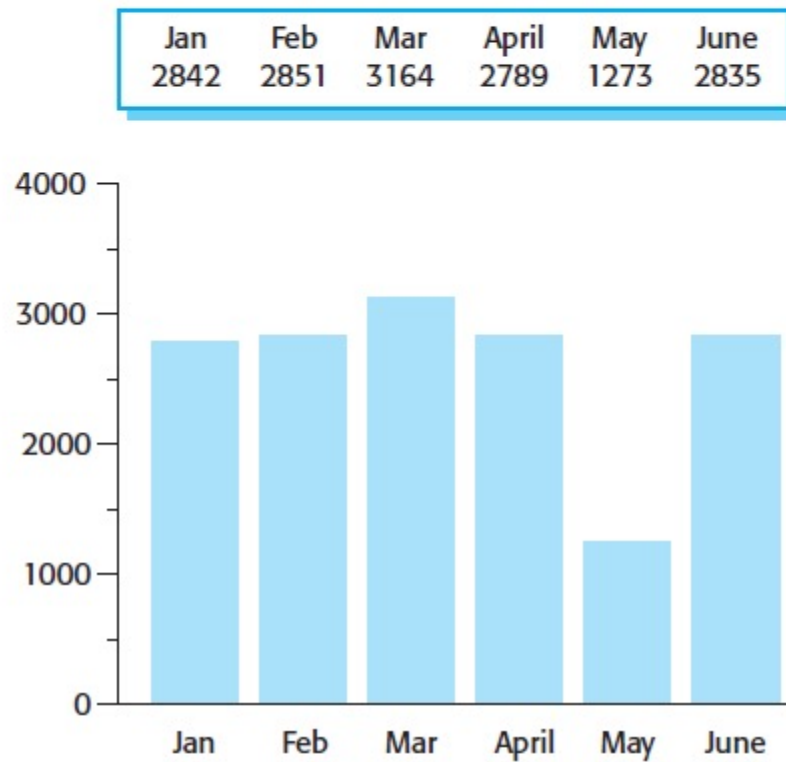
Có 2 loại thông tin:

- ☐ Thông tin tĩnh: thông tin không thay đổi trong session.
 - ☐ Thông tin động: có sự thay đổi và người dùng thấy được các thay đổi này.
-

Các yếu tố quan tâm

- ☐ Người dùng quan tâm đến sự chính xác của thông tin hay mối quan hệ giữa các dữ liệu?
 - ☐ Các giá trị thay đổi nhanh chóng ra sao? Các thay đổi có cần phải thể hiện ngay lập tức không?
 - ☐ Người dùng có cần phản hồi gì khi có sự thay đổi không?
 - ☐ Người dùng có cần can thiệp trực tiếp vào các thông tin ngay trên giao diện không?
 - ☐ Thông tin là văn bản hay số? Các giá trị có quan trọng không?
-

Một cách thể hiện thông tin



Thẻ hiện theo digital hay analog?

☐ Thẻ hiện theo số:

- Ít chiếm không gian
- Giá trị được thể hiện chính xác.

☐ Thẻ hiện analog:

- Dễ có khái niệm sơ sơ về giá trị khi thoát nhìn qua
 - Có thể thể hiện các giá trị mang tính tương đối
 - Dễ thấy hơn các giá trị bất thường
-

Dùng màu sắc

- ❑ Màu sắc tạo thêm một chiều nữa cho việc thể hiện và có thể giúp người dùng hiểu được các cấu trúc thông tin phức tạp.
 - ❑ Màu sắc có thể được dùng để nhấn mạnh các trường hợp ngoại lệ.
 - ❑ Một số lỗi hay mắc phải khi dùng màu sắc:
 - Sử dụng màu sắc để cung cấp ngữ nghĩa
 - Lạm dụng quá nhiều màu sắc khi thể hiện
-

Một số lời khuyên khi sử dụng màu

- ❑ Giới hạn số lượng màu. Chẳng hạn không nên dùng quá 4-5 màu cho một màn hình và không quá 7 màu cho toàn bộ hệ thống.
 - ❑ Sử dụng sự thay đổi về màu sắc để cho thấy sự thay đổi về trạng thái của hệ thống.
 - ❑ Dùng mã màu sắc để hỗ trợ người dùng
 - ❑ Dùng mã màu sắc một cách nhất quán. Chẳng hạn dùng màu đỏ để thông báo lỗi thì luôn dùng như vậy.
 - ❑ Cần trọng khi sử dụng các cặp màu. Một số cặp màu gây khó khăn khi đọc ví dụ đỏ và xanh biển.
-

Các câu thông báo lỗi

- ❑ Thiết kế các câu thông báo lỗi là rất quan trọng cho một hệ thống. Nếu hệ thống có các câu báo lỗi không được thiết kế tốt có thể dẫn đến việc người dùng không thích sử dụng hệ thống.
 - ❑ Các câu thông báo cần lịch sự, cô đọng, nhất quán và có tính xây dựng.
 - ❑ Cần xem xét đối tượng người dùng (kinh nghiệm và nền tảng của họ) khi thiết kế các câu thông báo lỗi.
-

Thông báo lỗi

- ❑ Nếu được, khi thông báo lỗi, hệ thống cần gợi ý hay giúp người dùng sửa lỗi.
 - ❑ Thông báo lỗi có thể có mối liên hệ trực tiếp đến hệ thống hướng dẫn ngay đúng với vấn đề đang bị lỗi (context-sensitive help system)
-

Ví dụ

Please type the patient name in the box then click on OK

Patient name

MacDonald, R.

OK Cancel

System-oriented error message

?

Error #27

Invalid patient id

OK Cancel

User-oriented error message

R. MacDonald is not a registered patient

Click on Patients for a list of patients

Click on Retry to re-input the patient's name

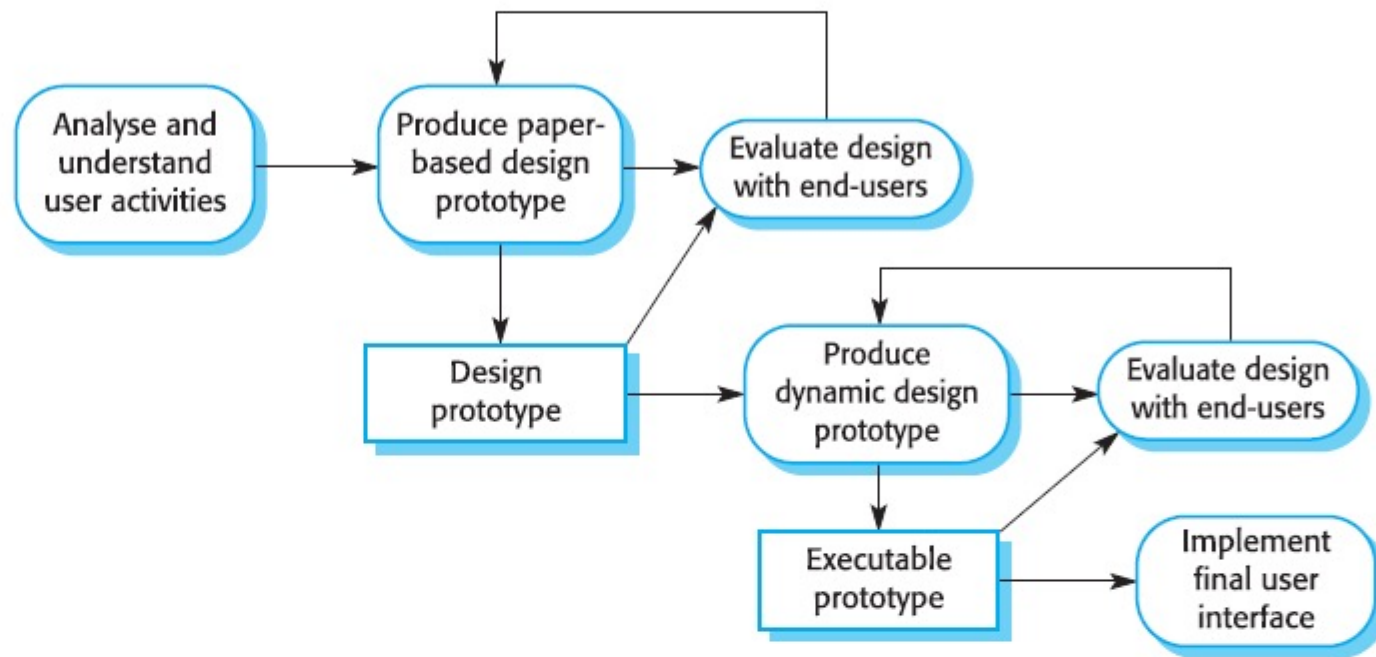
Click on Help for more information

Patients Help Retry Cancel

Tiến trình thiết kế giao diện

- ❑ Thiết kế giao diện người dùng là tiến trình lặp đi lặp lại với sự kết hợp giữa người thiết kế và người dùng.
 - ❑ Có 3 hoạt động chính của tiến trình
 - Phân tích người dùng: hiểu người dùng muốn làm gì với hệ thống
 - Prototype hệ thống: tạo ra một loạt các prototype của hệ thống để thực nghiệm
 - Đánh giá giao diện: thử nghiệm chúng với người dùng.
-

Tiến trình thiết kế giao diện



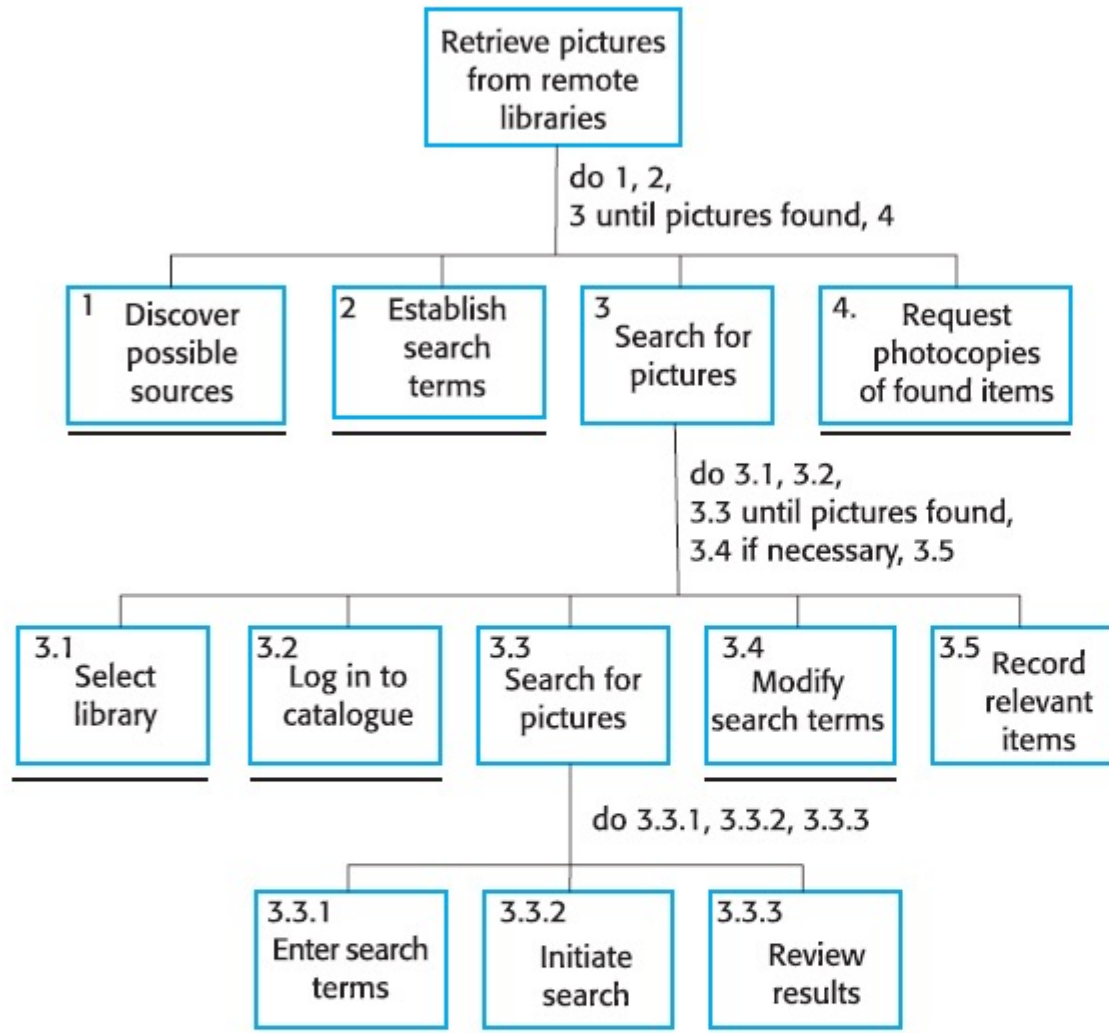
Phân tích người dùng

- ❑ Nếu không hiểu người dùng muốn làm gì với hệ thống thì chúng ta không thể thiết kế được giao diện người dùng hiệu quả
 - ❑ Các phân tích cần được thể hiện sao cho người dùng và người thiết kế hiểu được.
 - ❑ Các scenarios là một trong những cách được dùng để mô tả các phân tích
-

Một số kỹ thuật cho phân tích

- ❑ Phân tích công việc: mô hình các bước liên quan để hoàn tất công việc
 - ❑ Phỏng vấn và bảng câu hỏi: hỏi người dùng về các công việc họ làm
 - ❑ Ethnography: quan sát người dùng khi họ đang làm việc.
-

Phân tích công việc



Phỏng vấn

- ❑ Thiết kế các loại phỏng vấn bán-cấu-trúc dựa trên các câu hỏi mở
 - ❑ Người dùng cung cấp các thông tin mà họ nghĩ là cần thiết chứ không chỉ là các thông tin mà chúng ta cần thu thập
 - ❑ Phỏng vấn theo nhóm (focus group) giúp người dùng có thể thảo luận với nhau cách mà họ thực hiện công việc
-

Ethnography: quan sát

- ❑ Quan sát người dùng làm việc từ bên ngoài và đặt các câu hỏi liên quan đến công việc của họ.
 - ❑ Phương pháp này rất hiệu quả vì có nhiều công việc mặc dù rất bình thường nhưng họ cảm thấy rất khó để mô tả và giải thích.
 - ❑ Phương pháp này giúp chúng ta hiểu rõ hơn các vai trò và ảnh hưởng của tổ chức, xã hội lên các công việc của người dùng.
-

Prototype giao diện người dùng

- ❑ Mục tiêu của prototype giúp người dùng có được cảm nhận trực tiếp với giao diện
 - ❑ Nếu không có sự cảm nhận trực tiếp thì rất khó để đánh giá mức độ tốt/xấu của giao diện
 - ❑ Prototype là tiến trình gồm 2 bước:
 - Bước đầu: có thể prototype trên giấy
 - Sau đó cần điều chỉnh và tăng độ tinh vi, tự động của prototype.
-

Prototype trên giấy

- ❑ Đi một lượt qua các scenario và phác thảo các giao diện.
 - ❑ Sử dụng kỹ thuật storyboard để thể hiện một loạt các tương tác của người dùng với hệ thống. Cách này thường thể hiện tốt nếu có nhóm người dùng.
 - ❑ Prototype trên giấy là một phương pháp rất hiệu quả để thấy được phản ứng của người sử dụng với giao diện được đề xuất.
-

Các kỹ thuật prototype khác

- ❑ Prototype dựa trên script: sử dụng công cụ, chẳng hạn Macromedia Director để phát triển các scripts và màn hình. Khi người dùng tương tác thì màn hình sẽ chuyển qua màn hình khác.
 - ❑ Visual programming: sử dụng các ngôn ngữ dành cho việc phát triển nhanh phần mềm, chẳng hạn Visual Basic.
 - ❑ Prototype dựa trên internet: sử dụng web browser và các script liên quan.
-

Đánh giá giao diện người dùng

- ❑ Cần thực hiện việc đánh giá giao diện người dùng để biết được chúng có hợp lý không.
 - ❑ Không cần phải đánh giá quá chi tiết và đầy đủ: tốn chi phí và không thực tế đối với nhiều hệ thống.
 - ❑ Theo đó, cách lý tưởng nhất là giao diện cần được đánh giá dựa trên các đặc tả về sử dụng. Tuy nhiên, các đặc tả này thường thường là không có.
-

Một số tính chất cần quan tâm

- ❑ Khả năng học nhanh (learnability): phải cần bao lâu thì một người sử dụng mới có thể sử dụng hệ thống một cách hợp lý và hiệu quả?
 - ❑ Tốc độ thực thi: hệ thống phản hồi tốt như thế nào với các tác động của người dùng?
 - ❑ Robustness: khả năng chịu lỗi của hệ thống ra sao nếu người dùng tương tác sai?
 - ❑ Khả năng phục hồi (recoverability): khả năng phục hồi khi người dùng bị lỗi.
 - ❑ Adaptability: hệ thống thể hiện gần gũi như thế nào với mô hình công việc trong thực tế?
-

Một số kỹ thuật đánh giá đơn giản

- ❑ Các bảng câu hỏi để đánh giá sự phản hồi của người dùng.
 - ❑ Hệ thống quay lại quá trình sử dụng hệ thống và đánh giá các đoạn phim quay.
 - ❑ Hệ thống mã nguồn thu giữ các thông tin về cách sử dụng và các lỗi người dùng.
 - ❑ Hệ thống mã nguồn trong phần mềm để thu lượm các thông tin phản hồi từ người dùng qua mạng.
-

Đối với các sản phẩm trên web?

☐ Tham khảo thêm sách:

- Don't make me think, 2nd edition, Steve Krug, 2006
- About Face 3: The Essentials of Interaction Design, Alan Cooper, 2007.
- Web Style Guide, Patrick J. Lynch, Sarah Horton
- Simply Brilliant: the Competitive Advantage of Common Sense: The Bestselling Guide to Getting Things Done, Fergus O'Connell.

☐ Sách tổng quát:

- The Design of Everyday Things, Don Norman.
-