

1. UI là gì.

- Là thuật ngữ viết tắt của User Interface có nghĩa là giao diện người dùng. - Là những gì người dùng dễ nhìn thấy nhất sẽ là UI như: màu sắc, bố cục, phông chữ, hình ảnh... - Trong thiết kế thì UI đóng vai trò là yếu tố truyền tải thông điệp từ người thiết kế, nhà cung cấp dịch vụ, sản phẩm đến người dùng.

2. UX là gì.

- Là thuật ngữ viết tắt của User Experience có nghĩa là trải nghiệm người dùng. - Là những đánh giá của người dùng khi sử dụng sản phẩm như: người dùng có thấy web/app của bạn có dễ dùng, có thân thiện hay không, bố cục web/app có hợp lý, bắt mắt hay không.

3. Quy trình thiết kế giao diện người dùng:

1. **Yêu cầu chức năng thu thập:** Bước này bao gồm lắp ráp một danh sách các chức năng cần thiết để thực hiện các mục tiêu của dự án và nhu cầu tiềm năng của người dùng. Thông thường, giai đoạn này bắt đầu ngay sau phiên khám phá với khách hàng.
2. **Phân tích người dùng và nhiệm vụ:** Đây là nghiên cứu về cách người dùng tiềm năng sẽ thực hiện các nhiệm vụ mà thiết kế phải hỗ trợ. Giai đoạn này được kết nối với nghiên cứu người dùng, chúng tôi thực hiện như một phần của quy trình thiết kế UX.
3. **Kiến trúc thông tin (IA):** Quá trình này bao gồm sự phát triển của quá trình và luồng thông tin của hệ thống. Trong giai đoạn này, chúng tôi chọn phong cách tương tác UI, mẫu thiết kế và kỹ thuật trực quan. Nhiều kỹ thuật thiết kế UI tôi đã mô tả trước đó được hình thành trong giai đoạn Kiến trúc thông tin.
4. **Tạo mẫu:** Giai đoạn này bao gồm sự phát triển của các nguyên mẫu, khung dây, mockup, nguyên mẫu giấy hoặc màn hình tương tác đơn giản.
5. **Kiểm tra khả năng sử dụng:** Kiểm tra khả năng sử dụng có thể được sử dụng để đánh giá các nguyên mẫu hoặc thông số kỹ thuật cho hệ thống, thường không thể được kiểm tra trên người dùng. Phương pháp kiểm tra khả năng sử dụng là một hướng đi nhận thức, đánh giá heuristic và hướng dẫn đa nguyên.
6. **Kiểm tra khả năng sử dụng:** Kiểm tra thiết kế giao diện người dùng cho phép hiểu được sự tiếp nhận của thiết kế từ quan điểm của người xem. Thông thường, trong quá trình kiểm tra khả năng sử dụng, người dùng được yêu cầu hoàn thành các nhiệm vụ để xem nơi họ gặp phải vấn đề và gặp phải sự nhầm lẫn.

7. **Thiết kế GUI (Giao diện người dùng đồ họa):** Đây là một cái nhìn và cảm nhận thực tế của thiết kế giao diện người dùng đồ họa cuối cùng. Ở giai đoạn này, chúng tôi quyết định giao tiếp bằng hình ảnh và giải quyết vấn đề thông qua việc sử dụng kiểu chữ, nhiếp ảnh và minh họa.
8. **Bảo trì phần mềm:** Bảo trì thường xuyên đi sau khi triển khai để sửa lỗi phần mềm, thay đổi tính năng hoặc nâng cấp hoàn toàn hệ thống.

4. Hãy trình bày và giải thích ngắn gọn ý nghĩa của 8 nguyên tắc vàng trong thiết kế giao diện (8 golden rules of interface design) của Shneiderman.

1. Strive for consistency (đảm bảo tính thống nhất).

- Trong những hành động tương tự nhau cần có sự nhất quán.
- Sử dụng từ ngữ, ký hiệu, hành động,... một cách thống nhất trong tất cả các trang và phù hợp với hệ thống chung..
- Sử dụng đồng bộ tất cả các yếu tố hiển thị.
- Ví dụ 1: màu đỏ dùng để cảnh báo nguy hiểm và màu xanh để thông báo trạng thái hoạt động tốt.
- Ví dụ 2: kí hiệu cụ thể của nút bấm sẽ luôn thực hiện một hành động duy nhất trong cả hệ thống.

2. Enable frequent users to use shortcuts (cho phép người dùng sử dụng phím tắt).

- Cho phép người dùng chi phối và cá nhân hóa những thao tác thường xuyên trên ứng dụng, sản phẩm. Sử dụng các phím tắt sẽ giúp người dùng thao tác nhanh hơn, nâng cao hiệu quả sử dụng hệ thống.
- Những từ viết tắt, phím chức năng, lệnh ẩn hay những tài nguyên mở rộng sẽ đặc biệt hữu ích cho nhóm người dùng chuyên gia.
- Ví dụ 1: Ctrl + N: mở một cửa sổ mới.
- Ví dụ 2: Window + D: ẩn tất cả các cửa sổ.

3. Offer informative feedback (cung cấp các thông tin phản hồi).

- Hiển thị trạng thái của hệ thống.
- Thông báo cho người dùng biết về những gì đang diễn ra theo một cách thích hợp và đúng thời điểm.
- Đảm bảo mọi hành động của người dùng đều được phản hồi lại cho họ biết chuyện gì đang xảy ra.
- Ví dụ 1: các nút bấm đổi màu khi người dùng nhấn vào.
- Ví dụ 2: khi copy file, download,... sẽ có thanh hiển thị hoạt động của công việc.

4. Design dialogs to yield closure (thiết kế các hộp thoại để mang lại sự khép kín).

- Chuỗi hành động nên được xếp chung nhóm với thứ tự: ban đầu, ở giữa và kết thúc. Các thông tin phản hồi ở mỗi nhóm khi hoàn thành được cung cấp cho người dùng để họ nhận ra và đi tới nhóm hành động kế tiếp.
- Ví dụ 1: hộp thoại thông báo hiện lên khi thực hiện xong tác vụ, khi có lỗi xảy ra.

- Ví dụ 2: hộp thoại yêu cầu sự xác nhận như xin quyền truy cập internet cho ứng dụng, quyền nhớ mật khẩu.

5. Offer error prevention and simple error handling (đề phòng và xử lý lỗi dễ dàng).

- Hệ thống cần loại bỏ những lỗi dễ xảy ra cho người dùng.
- Kiểm tra và thông báo giúp người dùng tránh gặp phải lỗi đơn giản.
- Hệ thống phải có chức năng phát hiện và đưa ra giải pháp đơn giản, dễ hiểu giúp người dùng sử dụng xử lý lỗi.
- Ví dụ 1: khi người dùng đăng kí mật khẩu luôn có thông báo về độ dài, những kí tự bắt buộc.
- Ví dụ 2: khi người dùng đăng kí mật khẩu, nếu mật khẩu chưa đủ điều kiện hệ thống sẽ hiển thị các gợi ý về quy tắc đặt mật khẩu cho người dùng.
- Ví dụ 3: khi copy file nếu file bị trùng sẽ có hộp thoại thông báo với nội dung file bị trùng và có các phương án cho người dùng lựa chọn để giải quyết vấn đề.

6. Permit easy reversal of actions (cho phép đảo ngược hành động dễ dàng).

- Trở lại hành động trước đó giúp người dùng sửa lỗi sai và lấy lại những thông tin cần thiết.
- Ví dụ 1: trong word có chức năng Undo và Redo để quay lại và tiếp tục các thao tác.
- Ví dụ 2: trong các trình soạn thảo luôn có phím tắt Ctrl + Z để quay lại thao tác trước đó.

7. Support internal locus of control (con người luôn làm chủ).

- Con người luôn đóng vai trò làm chủ hệ thống, quản lý mọi vấn đề và tùy chỉnh hệ thống theo ý của mình.
- Để người dùng làm chủ hệ thống, còn hệ thống thì đưa ra các gợi ý.
- Ví dụ 1: khi người dùng thoát ra khỏi hệ thống khi chưa lưu thì hệ thống sẽ hiển thị hộp thoại gợi ý người dùng lưu, không lưu hoặc hủy thao tác, nhờ đó người dùng không bị mất dữ liệu.
- Ví dụ 2: Task manager của window cho phép người dùng buộc dừng chương trình.

8. Reduce short-term memory load (giảm tải bộ nhớ ngắn hạn).

- Đơn giản hóa mọi thứ.
- Hạn chế việc ghi nhớ của người dùng bằng cách thiết kế vật thể, hành động, những chỉ dẫn một cách rõ ràng.
- Sử dụng các biểu tượng, màu sắc thống nhất giúp người dùng cũ nhanh chóng tìm được chức năng cần thiết.
- Ví dụ 1: nút bật tắt chế độ yên lặng trên điện thoại iphone.
- Ví dụ 2: phím window trên bàn phím máy tính.
- Ví dụ 3: phím mũi tên biểu thị cho undo và redo trên các thanh trạng thái.

5. Hãy phân biệt kiểu giao diện dòng lệnh (command-based) và giao diện đồ họa người dùng (graphical user interface).

- Tổng quan:

1. Giao người dùng là thuật ngữ được sử dụng để chỉ định cách người dùng tương tác với thiết bị điện tử, đặc biệt là máy tính. Giao diện dòng lệnh và giao diện đồ họa là các loại giao diện người dùng khác nhau.
2. Để thực hiện một thao tác trên hệ thống giao diện dòng lệnh, người ta phải viết một lệnh. Mặt khác, trong giao diện đồ họa người dùng đã được cung cấp các công cụ hỗ trợ trực quan bao gồm hình ảnh và biểu tượng, tạo điều kiện cho người dùng thực hiện trực tiếp một tác vụ.
3. Các hệ thống giao diện dòng lệnh yêu cầu chuyên môn về các lệnh để thực hiện tác vụ, trong khi giao diện đồ họa không yêu cầu chuyên môn, nó cũng có thể được vận hành bởi những người dùng mới làm quen.

- Biểu đồ so sánh sự khác nhau:

Cơ sở để so sánh	Giao diện dòng lệnh	Giao diện đồ họa
Căn bản	Cho phép người dùng giao tiếp với hệ thống thông qua các lệnh	Cho phép người dùng tương tác với hệ thống bằng cách sử dụng đồ họa bao gồm hình ảnh, biểu tượng,...
Thiết bị được sử dụng	Bàn phím	Bàn phím và chuột
Dễ thực hiện nhiệm vụ	Khó để thực hiện một hoạt động và đòi hỏi chuyên môn	Dễ dàng thực hiện các nhiệm vụ và không đòi hỏi chuyên môn
Độ chính xác	Cao	Thấp
Mềm dẻo	Trực tiếp	Linh hoạt hơn
Xuất hiện	Không thể thay đổi	Thay đổi tùy chỉnh có thể được sử dụng
Tốc độ	Nhanh	Chậm
Tích hợp và mở rộng	Phạm vi cả tiến tiềm năng	Bị ràng buộc
Tiêu thụ bộ nhớ	Thấp	Cao

- Sự khác biệt chính giữa giao diện dòng lệnh và giao diện đồ họa.

1. Giao diện dòng lệnh cho phép người dùng nhập lệnh thủ công để thực hiện các thao tác mong muốn trong khi giao diện đồ họa cung cấp hình ảnh để tương tác vs hệ điều hành như nút, biểu tượng,...
2. Dễ dàng để thực hiện một tác vụ trong giao diện đồ họa và tốt cho người mới bắt đầu, mặt khác giao diện dòng lệnh cần chuyên môn về các lệnh và cú pháp.
3. Hệ thống giao diện đồ họa yêu cầu chuột và bàn phím trong khi giao diện dòng lệnh chỉ cần bàn phím để hoạt động.
4. Giao diện dòng lệnh không thể thay đổi và không linh hoạt còn giao diện đồ họa có thể tùy chỉnh và linh hoạt hơn.

5. Giao diện dòng lệnh có độ chính xác, tốc độ và tiêu tốn ít tài nguyên hệ thống hơn giao diện đồ họa.

- **Giao diện dòng lệnh.**

1. Là những giao diện thông thường được sử dụng rộng rãi trong những năm 1980. Cho phép người dùng viết các lệnh trong cửa sổ terminal hoặc console để giao tiếp với hệ điều hành, là nơi người dùng phản hồi bằng cách viết lệnh và nhận phản hồi của hệ thống. Người dùng phải gõ lệnh hoặc huấn luyện lệnh để thực hiện một tác vụ. Giao diện dòng lệnh chính xác hơn giao diện đồ họa nhưng nó đòi hỏi phải thành thạo các lệnh và cú pháp.
2. Hạn chế của giao diện dòng lệnh.
 - a. Chỉ phù hợp với người nó một cách thường xuyên và có thể ghi nhớ phạm vi của các lệnh và tùy chọn.
 - b. Sai lầm có thể dẫn đến hỗn loạn hoàn toàn.
 - c. Các lệnh không bao giờ có thể trực quan.
 - d. Nó không phù hợp cho đồ họa tương tác.

- **Giao diện đồ họa.**

1. Sử dụng đồ họa để cho phép người dùng giao tiếp với hệ điều hành hoặc ứng dụng. Nó cung cấp các cửa sổ, thanh cuộn, nút, trình hướng dẫn, hình ảnh biểu tượng, các biểu tượng khác. Để tạo thuận lợi cho người dùng. Là một giao diện dễ sử dụng cho người mới làm quen. Nó trực quan, dễ học và giảm tải nhận thức. Người dùng không cần phải nhớ các dòng lệnh.
2. Hạn chế của giao diện đồ họa.
 - a. Thiếu độ chính xác.
 - b. Phân tích nhân rộng và retracing các bước khó khăn.
 - c. Khó thiết kế.

6. Hãy trình bày sự phát triển của giao diện thiết bị di động và một số vấn đề thiết kế liên quan.

- **Sự phát triển của giao diện thiết bị di động.**

1. Quá khứ:

- a. trước đây, nhiều ứng dụng di động ban đầu tập trung vào phong cách Skeuomorphism (phong cách tạo ra một cảm giác quen thuộc bằng cách mô phỏng các vật liệu). Ví dụ một vài ứng dụng phát hành trên Apple store như là Apple Newsstand được thiết kế giống như một kệ tạp chí, hoặc ứng dụng Notes được thiết kế giống như một quyển ghi chú với bìa da.
- b. Những thiết kế này tạo ra những giao diện quen thuộc như đời sống thực.
- c. Những ứng dụng hàng đầu vào 2008 gồm những cái tên quen thuộc như Texas Hold'em, Facebook, Shazam, tuy nhiên phiên

bản 2008 so với phiên bản ngày nay hoàn toàn khác biệt. Ví dụ như Facebook phiên bản 2008 đem lại sử dụng thiết kế 3D đem lại cho những người dùng mới làm quen với việc chạm cảm ứng, một trải nghiệm giống đời sống thực, như là bấm một cái nút.

2. Hiện tại:

- a. Thiết kế ứng dụng trong những năm gần đây chuyển từ Skeuomorphism sang flat design (thiết kế phẳng). Thiết kế phẳng đề cập đến hai yếu tố, ưu tiên chức năng và trải nghiệm người dùng. Thiết kế phẳng tập trung vào sự tối giản, gọn gàng, giao diện tươi sáng.
- b. So sánh ứng dụng Facebook ban đầu và hiện tại, những nút 3D được thay thế bởi những hình ảnh phẳng và chữ thì được giữ ở mức tối thiểu.
- c. 2014, Google phát hành hướng dẫn thiết kế của họ, thiết kế Material. Thiết kế Material đem lại thiết kế chiều sâu (những cái mà đã bị bỏ khỏi thiết kế phẳng) như là hoạt ảnh chuyển động, hiệu ứng ánh sáng, hiệu ứng đổ bóng.

3. Tương lai:

- a. Trong khi thiết kế phẳng vẫn đang thống trị, sự xuất hiện của các kích thước màn hình mới và các thiết bị đeo thông minh sẽ làm cho xu hướng thiết kế liên tục được tinh chỉnh. Thách thức đặt ra là làm thế nào để tối giản thiết kế những vẫn giữ được đầy đủ tính năng. Sự ra đời 3D touch của Apple trên iPhone 6s có thể là một cách để giải quyết vấn đề này.
- b. Có thể chỉ là dự đoán, nhưng một điều ta có thể nói là tương lai của thiết kế giao diện thiết bị di động sẽ đem lại những thách thức mới và những giải pháp vô cùng sáng tạo.

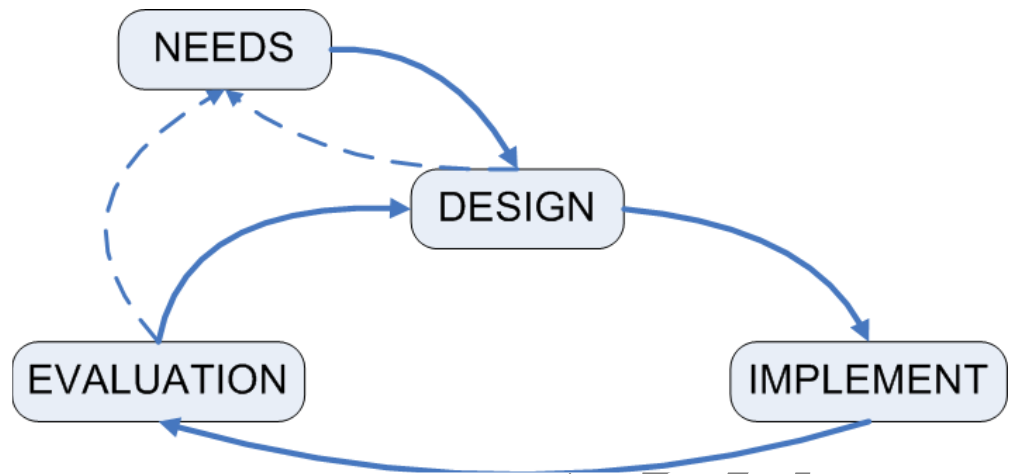
- Một số vấn đề thiết kế liên quan:

1. Thiết kế một ứng dụng ngày nay là một quá trình rất khó khăn. Thị trường ứng dụng không chỉ đạt đến điểm bão hòa mà còn phải đối mặt với sự cạnh tranh khốc liệt khi ra mắt ứng dụng. Vì thế xây dựng giao diện người dùng là yếu tố chính giúp ứng dụng nổi bật.

7. Hãy trình bày quá trình thiết kế tương tác và bốn hoạt động cơ bản trong quá trình này.

- Quá trình thiết kế tương tác:

1. Needs: nhu cầu.
2. Design: thiết kế.
3. Implement: thực hiện.
4. Evaluation: đánh giá.



- **Bốn hoạt động cơ bản:**

1. Xác định nhu cầu và thiết lập các yêu cầu (Identify needs & establishing requirements):

- a. Một yêu cầu là một nhu cầu mà sản phẩm phải đáp ứng.
- b. Thiết lập những gì cần thiết cho sản phẩm là điều cơ bản để đảm bảo rằng sự tương tác phù hợp nhất với người dùng, cả về những gì họ cần làm và cách họ trải nghiệm tương tác.
- c. Các yêu cầu sẽ phụ thuộc vào đặc điểm của người dùng, các hành động mà người dùng sẽ thực hiện khi sử dụng sản phẩm và môi trường mà người dùng tương tác với sản phẩm.
- d. Xác định người dùng: đặt ra những câu hỏi:
 - i. Ai mua sản phẩm.
 - ii. Ai trực tiếp sử dụng sản phẩm.
 - iii. Ai nhận output từ sản phẩm.
- e. Xác định nhu cầu: đặt ra những câu hỏi:
 - i. Người dùng cần gì.
 - ii. Hiện tại người dùng đang làm việc đó như thế nào.
 - iii. Người dùng trông chờ gì ở sản phẩm.
 - ➔ Liệu có cách nào tốt hơn để hoàn thành công việc của người dùng không?
- f. Xác định nhu cầu bằng cách thu thập thông tin:
 - i. Bảng hỏi.
 - ii. Phỏng vấn.
 - iii. Quan sát.
 - iv. Đọc tài liệu.
 - v. Nghiên cứu các sản phẩm tương tự.
- g. Phân tích nhu cầu để thiết lập yêu cầu:

- i. Các hàm.
- ii. Dữ liệu.
- iii. Ngữ cảnh sử dụng.
- iv. Người dùng.

→ Đây là bước dễ xảy ra những sai lầm.

2. Phát triển các thiết kế thay thế (Develop alternative designs that meet those requirements):

- a. Đưa ra các thiết kế thay thế cho phép các nhà thiết kế khám phá các cách diễn giải khác nhau và đáp ứng các yêu cầu cho một sản phẩm tương tác cụ thể.
- b. Ý tưởng thiết kế nên được dựa trên các nguyên tắc thiết kế cơ bản từ những hiểu biết chúng ta thu được về cách hoạt động của suy nghĩ và cơ thể chúng ta.

3. Xây dựng bản mẫu thiết kế (Building interactive versions of the designs):

- a. Một khi các nhà thiết kế tương tác đã xác định được một số ý tưởng khả thi, họ cần tìm ra ý tưởng nào có tiềm năng đáp ứng tốt nhất cho người dùng, cho hoạt động và môi trường của họ.
- b. Để làm điều này, các nhà thiết kế cần phải chọn ra các ý tưởng thiết kế hứa hẹn nhất để tạo ra một mô hình bản mẫu đầu tiên (thường là thô) để họ có thể thử chúng.
- c. Tạo bản mẫu còn có thể dùng để khám phá các khía cạnh khác nhau của một thiết kế.

4. Đánh giá những gì đang được xây dựng trong suốt quá trình (Evaluating what is being built throughout the process):

- a. Cho phép nhà thiết kế tiếp cận những hạn chế của một sản phẩm.
- b. Tìm ra mức độ nào mà bản mẫu đáp ứng so với những yêu cầu đã được xác định.
- c. Xác định những yêu cầu chưa xuất hiện.
- d. Thiết lập những thay đổi cần thiết để đáp ứng những yêu cầu đó.
- e. Các bước:
 - i. Lên kế hoạch – huấn luyện: chuẩn bị đánh giá.
 - ii. Thực hiện: làm đánh giá.
 - iii. Tổng hợp: ghi nhận kết quả.
 - iv. Thảo luận: công bố và sử dụng kết quả.

8. Thiết kế tương tác (interaction design) là gì? Tầm quan trọng của thiết kế giao diện người dùng cho phần mềm?

- Giao diện người dùng là mọi thứ được thiết kế thành một thiết bị hoặc phần mềm để cho phép con người giao tiếp với công nghệ.
- **Thiết kế tương tác (interaction design):** Tạo các giao diện hấp dẫn với các hành vi suy nghĩ tốt. Sự giao tiếp đơn giản, tường minh giữa người dùng và công nghệ. Thiết kế tương tác sẽ đảm nhiệm các chức năng cho phép tương tác giữa mọi người và giao diện người dùng cá nhân hoặc trên nhiều giao diện (hoặc hệ thống).
- **Tầm quan trọng của thiết kế giao diện người dùng:**
 1. Không quan trọng là ý tưởng của bạn tuyệt vời như thế nào nếu như ứng dụng trông mất thẩm mỹ và không thể sử dụng một cách dễ dàng.
 2. Nếu phần mềm của bạn có giao diện kém chất lượng thì người dùng cũng quan ngại về nội dung, thông tin mà phần mềm của bạn mang lại. Vì vậy đây cũng là một trong những tiêu chí mà các công ty luôn coi trọng trong việc thiết kế phần mềm cho họ.
 3. Nếu bạn đang phát triển một ứng dụng web hoặc ứng dụng trên thiết bị di động, một trong những điều quan trọng bạn cần làm đúng là đáng vẻ và cảm giác mà ứng dụng của bạn mang lại. Với ví dụ rằng nếu lĩnh vực của bạn là Thương mại điện tử, một ứng dụng có thiết kế kém sẽ khiến bạn mất nhiều khách hàng tiềm năng.

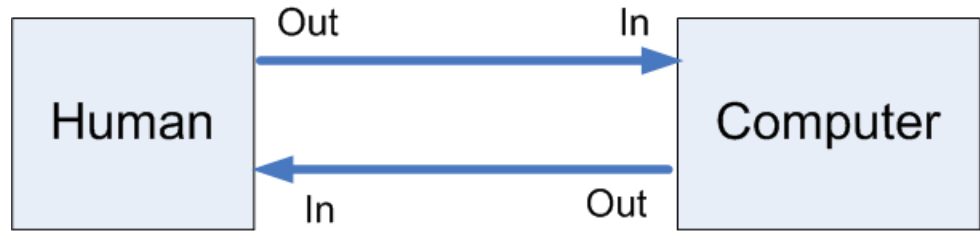
9. Trình bày khái niệm về tính tiện lợi (usability) của một giao diện phần mềm? Cho biết cụ thể bốn (4) yêu cầu sau tính tiện lợi: tính hiệu suất (effective), tính hiệu năng (efficient), tính dễ học (learnable) và tính dễ nhớ (memorable). Cho ví dụ minh họa cụ thể: tiêu cực hoặc tích cực.

- Tính tiện lợi (usability): là một trong những yếu tố quan trọng thuộc về UX (trải nghiệm người dùng), nó đánh giá cách sử dụng của người dùng vào công việc của họ và mức độ hài lòng với quá trình thực hiện công việc đó.
- Tính hiệu quả (effectiveness):
 1. Cung cấp đầy đủ các chức năng cho người dùng hoàn thành công việc của họ.
 2. Đảm bảo độ chính xác và hoàn chỉnh của công việc mà người dùng có thể thực hiện.
 3. Liên kết trực tiếp với nhu cầu của người dùng.
 4. Ví dụ: cố định số ký tự khung nhập thẻ tín dụng (credit card), nó sẽ chỉ nhận thẻ hợp lệ, giảm tỉ lệ người dùng nhập sai và giúp họ hoàn thành việc nhập một cách chính xác.
- Tính hiệu suất (efficiency):
 1. Giảm thiểu tốc độ người dùng hoàn thành công việc.
 2. Tạo điều kiện cho người dùng học cách sử dụng dễ dàng nhằm đạt năng suất cao.
 3. Các bước thực hiện công việc ở mức tối thiểu.

4. Ví dụ: cho phép sử dụng các phím tắt để giảm thiểu số lần thao tác - bạn sẽ thao tác nhanh hơn khi dùng Ctrl+C, Ctrl+V để copy và paste thay vì dùng chuột.
- Tính an toàn (safety):
 1. Phòng tránh người dùng mắc lỗi..
 2. Có cơ chế hoàn tác.
 3. Phải bảo mật thông tin người dùng.
 - Tính dễ học (Learnable):
 1. Cung cấp hướng dẫn sử dụng một cách toàn vẹn cho người dùng.
 2. Dễ sử dụng và dễ tiếp nhận.
 3. Theo quy tắc 10 phút.
 4. Ví dụ: thiết kế dễ học là những thiết kế gần gũi với đời thực, một cái nút ảo nên thiết kế giống một cái nút thật, người dùng biết khi ấn nút thì sẽ có những thông tin được đưa ra dựa trên nút mà người dùng đã ấn
 - Tính ghi nhớ (memorable):
 1. Các quy trình thực hiện công việc phải dễ nhớ.
 2. Nên làm các chuẩn chung phổ biến.
 3. Nên có chỉ dẫn tới những chức năng không được sử dụng thường xuyên.
 4. Ví dụ: thiết kế giao diện của iPhone 4 và iPhone 6 chỉ có vài thay đổi nhỏ, khiến người dùng có thể dễ dàng nâng cấp iPhone của họ mà không gặp khó khăn gì.
 - Sự hài lòng (satisfaction):
 1. Thiết kế có ảnh hưởng đến cảm xúc của người dùng.
 2. Phải có tính rõ ràng tránh gây khó hiểu cho người dùng, vì nếu khó hiểu người dùng sẽ cảm thấy khó sử dụng và dễ mất hài lòng.

10.Trình Trình bày yếu tố người – người sử dụng trong quá thiết kế giao diện phần mềm. Trình bày mô hình xử lý thông tin của con người trong ngành khoa học tương tác người máy.

- **Yếu tố con người trong quá trình thiết kế.**
 1. Con người (Human trong HCI – Human Computer Interaction).
 2. Là mục tiêu, mục đích cuối cùng của tất cả hệ thống.
 3. Thấu hiểu người dùng về:
 - a. Khả năng.
 - b. Thói quen.
 - c. Lỗi và nhầm lẫn.
 - d. Quá trình nhận biết.
 trong việc sử dụng hệ thống tương tác.
- **Mô hình xử lý thông tin của con người tron ngành khoa học “tương tác người máy”.**



	Human	Computer
Input	Nhập các lệnh yêu cầu máy tính xử lý thông qua bàn phím, chuột, micro...	Tiếp nhận lệnh từ người dùng yêu cầu. VD: Bàn phím, chuột, micro...
Output	Nhận được các thông tin, dữ liệu từ máy tính gửi ra thông qua màn hình, máy in, loa...	Đưa ra các thông tin, dữ liệu mà người dùng yêu cầu. VD: Màn hình, máy in, loa...
Memory	Đưa các thông tin nhận được vào bộ não.	Lưu trữ dữ liệu. VD: ROM, RAM, đĩa cứng...
Processor	Xử lý các thông tin mà bộ não nhận được.	Thực hiện các lệnh, xử lý mọi thông tin và dữ liệu. VD: CPU

- Human – Processing information system model (Mô hình xử lý thông tin của con người).
 1. Input/Output.
 2. Bộ nhớ/Trí nhớ (memory).
 3. Bộ xử lý (processor).

Input	Output
Nhìn, nghe, chạm (thị giác, thính giác, xúc giác)	Tay, v.v...

Nhìn: có tác dụng nhất với giao diện đồ họa người dùng (GUI).

Trí nhớ: trí nhớ tạm thời (sensory memory), trí nhớ ngắn hạn (working memory hay short-term memory), trí nhớ dài hạn (long-term memory).

- Attention: sự chú ý, rehearsal: sự gợi nhớ.

Sensory memories

Attention

Short-term memory or working memory

Rehearsal

Long-term memory

- Bộ xử lý:
 1. Lập luận.
 2. Huấn luyện kỹ năng.
 3. Xử lý lỗi.
 4. Luôn nhớ là: nhận biết luôn tốt hơn là phải nhớ lại (Recognition Over Recall/ Think -> Congnition).

11. 6 Nguyên tắc thiết kế (Principles for Design).

- **Visibility: tính rõ ràng.**
 1. **Giao diện giúp người dùng luôn hiểu:**
 - a. **Trạng thái hiện tại của hệ thống.**
 - b. **Hoạt động gì có thể được thực hiện**
 - c. **Sẽ làm gì sau khi bấm, thao tác. Kết quả sau khi bấm, thấy được nó và thấy dễ dàng.**
 2. Khả năng hiển thị cũng có thể được cung cấp thông qua việc sử dụng âm thanh hoặc thông qua cảm ứng với các kết cấu, hình dạng hoặc rung động khác nhau.
 3. Ví dụ 1: nút bật tắt sẽ thay đổi trạng thái khi bấm.
 4. Ví dụ 2: vô lăng xe được đặt ở vị trí có thể dễ dàng tìm thấy và sử dụng.
- **Afordance: gợi ý.**
 1. **Tập hợp các hoạt động và thủ tục có thể được thực hiện cho một đối tượng.**
 2. Là các thuộc tính mà mọi thứ có thể nhận biết và cách chúng liên quan đến cách sử dụng, mô tả những đặc tính giúp thực hiện chức năng của nó.
 3. Ví dụ 1: nút bánh răng thường để chỉ cài đặt.
 4. Ví dụ 2: tay cầm trên ô tô giúp chúng ta có thể sử dụng chính xác mà không cần phải suy nghĩ ngay cả khi lần đầu sử dụng.
 5. Ví dụ 3: điện thoại mở khóa bằng cách trượt luôn có mũi tên chỉ hướng trượt và dòng chữ "slide to unlock" giúp chúng ta có thể nhận biết được cách mở khóa điện thoại.

- **Containt: sự chặt chẽ.**

1. Là mô tả việc ràng buộc các hoạt động tương tác của người dùng trên hệ thống, để đơn giản hóa và dễ sửa dụng.
2. Hạn chế các hành động độc hại có thể được thực hiện (giúp ngăn người dùng chọn các tùy chọn không chính xác).
3. Giảm khả năng xảy ra lỗi.
4. Tập trung sự chú ý của người dùng vào nhiệm vụ cần thiết.
5. Ví dụ 1: việc đăng kí người dùng cần nhập đủ thông tin nếu thiếu sẽ có thông báo.
6. Ví dụ 2: trên thanh công cụ của word khi ta thực hiện một thao tác, các chức năng không cần thiết sẽ được ẩn đi và các chức năng cần thiết sẽ được in đậm để tập trung sự chú ý dùng vào nhiệm vụ cần làm giảm khả năng xảy ra lỗi.

- **Mapping: bản đồ.**

1. **Đối với thiết bị là mối liên hệ giữ thiết kế vật lý và chức năng của đối tượng.**
2. **Đối với thiết kế giao diện người dùng trên máy tính là ánh xạ giữa các điều khiển và hành động của chúng trên máy tính.**
3. Sự sắp xếp hợp lý và tự nhiên cho mối quan hệ giữa các điều khiển và chuyển động của chúng đến kết quả từ hành động đó vào thế giới. Chức năng thực sự của ánh xạ tự nhiên là giảm nhu cầu về bất kỳ thông tin nào từ bộ nhớ của người dùng để thực hiện một tác vụ.
4. Ví dụ 1: Mũi tên hướng lên luôn để chỉ việc tăng.

- **Feedback: phản hồi.**

1. **Sự phải hồi từ hệ thống gửi thông tin về những gì đang xảy ra giúp người dùng biết được đang làm gì.**
2. Thấy rõ được hành động nào đã được thực hiện, và phản hồi lại.
3. Khi bất cứ điều gì thay đổi, nó sẽ hiển thị, phản hồi liên tục và đồng bộ hóa với hành động của người dùng giúp người dùng có cảm giác được kiểm soát.
4. Ví dụ 1: Nhấp chuột vào nút sẽ đổi màu.
5. Ví dụ 2: phản hồi của âm thanh khi tắt/bật công tắc.
6. Ví dụ 3: đồng hồ đo tốc độ của xe máy sẽ quay khi xe chạy.

- **Consistency: tính nhất quán.**

1. Thiết kế giao diện sử dụng các yếu tố tương tự cho các nhiệm vụ tương tự để người dùng dễ dàng tương tác với các thành phần khác nhau bằng cách hiểu ban đầu.
2. Cho phép mọi người dùng nhanh chóng chuyển kiến thức trước sang bối cảnh mới và tập trung vào các nhiệm vụ có liên quan.
3. Ví dụ 1: kí hiệu của thể của nút bấm sẽ luôn thực hiện một hành động duy nhất hay thanh điều hướng cũng nên hiển thị có logic, hệ thống.

4. Ví dụ 2: Word, Excel, PowerPoint đều sử dụng cùng một thanh công cụ với các tùy chọn menu chính: Home, Insert, Design...

9. 10 Usability Heuristics

- **1 Visibility of system status (hiển thị trạng thái hệ thống):**
 1. Khả năng hiển thị của trạng thái hệ thống đề cập đến trạng thái của hệ thống được truyền đạt đến người dùng như thế nào. Lý tưởng nhất là các hệ thống phải luôn thông báo cho người dùng về những gì đang diễn ra, thông qua phản hồi thích hợp trong thời gian hợp lý.
 2. Là hướng dẫn cơ bản nhất của thiết kế giao diện người dùng. Nó phục vụ để thông báo cho người dùng về tình trạng hiện tại và cho phép họ điều khiển sự tương tác theo đúng hướng, mà không lãng phí nỗ lực.
 3. Ví dụ 1: Điện thoại hoặc máy tính xách tay hiển thị thời lượng pin còn lại.
 4. Ví dụ 2: Ứng dụng email cho biết có bao nhiêu email chưa đọc.
- **2 Match between system and the real world (sự cân xứng giữa hệ thống ảo và thế giới thực):**
 1. Các hệ thống nên nói ngôn ngữ của người dùng bằng các từ, cụm từ và khái niệm quen thuộc thay vì các thuật ngữ hướng hệ thống. Các giao diện tuân theo các quy ước trong thế giới thực và làm cho thông tin xuất hiện theo thứ tự tự nhiên và hợp lý thể hiện sự đồng cảm và thừa nhận đối với người dùng.
 2. Ví dụ 1: các nút tăng giảm âm lượng trên tai nghe: chúng được định vị theo cách tương ứng với kết quả. Dấu cộng ở trên cùng làm tăng âm lượng và dấu trừ ở phía dưới làm giảm âm lượng.
- **3 User control and freedom (khả năng kiểm soát và sự tự do của người dùng):**
 1. Người dùng thường chọn nhầm chức năng hệ thống và sẽ cần một "lối thoát khẩn cấp" được đánh dấu rõ ràng để thoát khỏi trạng thái không mong muốn mà không phải trải qua một cuộc đối thoại mở rộng. Hỗ trợ hoàn tác và làm lại.
 2. Ví dụ 1: người dùng facebook có thể gỡ bỏ những bài đăng tùy ý.
 3. Ví dụ 2: tin nhắn flash của gmail (gợi ý chúng ta có thể khôi phục mail vừa xóa) khi chúng ta xóa mail.
- **4 Consistency and standards (sự nhất quán và các tiêu chuẩn):**
 1. Ví dụ 1: Màu sắc các cửa sổ trong ứng dụng luôn phải đồng bộ.
 2. Ví dụ 2: Nút "Send" cần phải được đồng bộ trong tất cả các trang.
- **5 Error prevention (phòng tránh lỗi):**
 1. Thiết kế cách kiểm tra và thông báo lỗi cho người dùng.

2. thiết kế cẩn thận ngăn chặn sự cố xảy ra ngay từ đầu. Hoặc loại bỏ các điều kiện dễ bị lỗi hoặc kiểm tra chúng và cung cấp cho người dùng một tùy chọn xác nhận trước khi họ cam kết hành động.
 3. Ví dụ 1: Trong quá trình gõ lệnh trên visual studio hệ thống sẽ kiểm tra cú pháp của các lệnh và thông báo nếu có lỗi.
- **6 Recognition rather than recall (nhận biết hơn là gợi ý):**
 1. Đề xuất cho người dùng một tập hợp các tùy chọn.
 2. Ví dụ 1: Khi nhập thông tin vào các loại form đăng kí hệ thống sẽ hiển thị các thông tin tương tự để người dùng có thể chọn nhanh.
 3. Ví dụ 2: google gợi ý kết từ khóa tìm kiếm mình muốn nhập.
 - **7 Flexibility and efficiency of use (sử dụng linh hoạt và hiệu quả):**
 1. cung cấp cho người dùng các cách để tăng tốc công việc của họ với hiệu quả và tính linh hoạt cao hơn.
 2. Ví dụ 1: Khi hiển thị các gợi ý luôn có nút bỏ qua tất cả để những người dùng đã biết có thể thao tác nhanh.
 3. Ví dụ 2: Mac OS cho phép người dùng tự do tạo các lệnh phím tắt và bàn phím tùy chỉnh.
 - **8 Aesthetic and minimalist design (thiết kế thẩm mỹ và đơn giản):**
 1. Các giao diện cần được xóa các yếu tố và nội dung không cần thiết không hỗ trợ các mục tiêu và nhiệm vụ của trang.
 2. Ví dụ 2: trang tìm kiếm của google rất đơn giản.
 3. Ví dụ 1: Logo của facebook rất đơn giản với chữ F.
 - **9 Help users recognize, diagnose, and recover from errors (giúp người dùng nhận biết, chẩn đoán, và khôi phục lỗi).**
 1. Kiểm tra cần phải được thực hiện nếu những lỗi đó đang được giải thích cho người dùng bằng ngôn ngữ dễ hiểu.
 2. Cần kiểm tra nếu xử lý ngoại lệ được thực hiện trên ứng dụng để có thể hiển thị các thông báo có liên quan cho người dùng.
 3. Ví dụ 1: Khi đăng kí tài khoản nếu người dùng nhập tên đã tồn tại hệ thống sẽ báo nhập sai tên, cho người dùng nhập lại.
 - **10 Help and documentation (trợ giúp và các tài liệu hướng dẫn):**
 1. Ví dụ 1: Các ứng dụng trên chplay luôn có phần mô tả thông tin và hướng dẫn.