**Câu hỏi ôn tập môn giao diện người sử dụng**

**Chương 1. Giới thiệu chung về giao diện người máy**

1. **Các khái niệm căn bản về HCI**

HCI

* **Giao tiếp người–máy tính** ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *Human–computer interaction*, viết tắt **HCI**) bao gồm công việc nghiên cứu, hoạch định, thiết kế và khai thác sự giao tiếp giữa con người ([người dùng](https://vi.wikipedia.org/w/index.php?title=Ng%C6%B0%E1%BB%9Di_d%C3%B9ng_(%C4%91i%E1%BB%87n_to%C3%A1n)&action=edit&redlink=1)) và máy tính.

1. **Tính khả dụng – usability**

* Gồm 2 thành phần chính:
* Chức năng: cung cấp đầy đủ đặc tính cho người sử dụng
* Dễ sử dụng: Cung cấp phần mềm dễ học và dễ sử dụng
* Lợi ích của tính khả dụng:
* Tiết kiệm thời gian
* Đáp ứng được nhiều đối tượng người dùng
* Người sd có thể sử dụng nhiều hệ thống máy tính hơn

1. **Các vấn đề liên quan tới đến tính khả dụng**

* Biểu hiện gây khó khăn đến người sử dụng từ các thành phần giao diện
* Vấn đề tính khả dụng là một kiểu đặt biệt của khiếm khuyết hệ thống

Vd: Một số kiểu khiếm khuyết: - Program error <lỗi chương trình>

-Thiếu chức năng

- Vấn đề liên quan đến tính dễ sử dụng

1. **Kiểm thử tính khả dụng**

* Think- aloud test: Kiểm tra suy nghĩ
* Cho nsd thực hiện một công việc nào đó, sau đó đặt câu hỏi cho nsd, tại sao lại làm nv? Nếu giải thích đc=> đặt yêu cầu
* Thi hành hệ thống thật
* Các kiểu kiểm thử:
* Bản mẫu đơn giản
* Kiểm tra theo nhóm
* Lập kế hoạch kiểm thử:
* Xác định người kiểm thử
* Xác định các nhiệm vụ kiểm thử
* Nghiên cứu hệ thống: Tự nghiên cứu hệ thống kiểm thử
* Tiến hành kiểm thử:
* Nêu mục đích kiểm thử
* Triển khai nhiệm vụ kiểm thử
* Quan sát, lắng nghe và ghi chép
* Đặt các câu hỏi phỏng vấn
* Trợ giúp người sử dụng
* Báo cáo:
* Sau khi kiểm thử, ghi lại danh sách vấn đề nảy sinh
* Báo cáo trong vòng 12h

1. **Đánh giá tính khả dụng**

* Đánh giá được thực hiện bởi 1 chuyên gia về tính khả dụng được gọi là heuristic
* Đánh giá được thực hiện bởi 1 người sử dụng: đánh giá người sử dụng

1. **Các yêu cầu và độ đo về tính khả dụng**

* Yếu tố quyết định tính khả dụng:
* Chức năng
* Dễ sử dụng
* Dễ bảo trì
* Tính đúng đắn
* Các phép đo đạc:
* Task time (thời gian thực hiện): là phương pháp đo truyền thống về tính khả dụng (đo thời gian để thực hiện các nhiệm vụ)
* Số lượng các vấn đề: Sử dụng kiểm thử phỏng vấn và đếm các vấn đề nảy sinh
* Số lần gõ phím (keystroke counts):
* Tổng hợp ý kiến: bản chất là thực hiện phỏng vấn
* Mức độ hiểu biết (score for understanding)
* Mức dộ tuân thủ theo hướng dẫn

**Chương 2: Qui trình thiết kế**

1. Khái niệm Bản mẫu giao diện

* là một mẫu thử nghiệm của giao diện người dùng. Ví dụ, các mẫu thử nghiệm có thể tự biểu hiện như:
* phác thảo giấy hoặc hình ảnh
* bitmap từ một công cụ vẽ
* một nguyên mẫu thực thi tương tác; ví dụ, trong Microsoft®Visual Basic

1. **Qui trình phát triển giao diện**

* Phân tích người sử dụng
  + Thông tin phân tích người sử dụng cần trình bày để người thiết kế hiểu để xác định các yêu cầu
  + Kỹ thuật phân tích:

+Phân tích công việc (Task analys): Xác định chính xác các bước cần phải làm

+Phỏng vấn: Thiết kế sẵn format, phỏng vấn theo nhóm

+Quan sát người sd: Liên quan đến quan sát viên, đặt được câu hỏi trong quá trình phỏng vấn

* Xây dựng bản mẫu
  + Mục đích: Cho phép người sử dụng có kinh nghiệm trực tiếp với giao diện
  + Việc xác định bản mẫu thực hiện 2 giai đoạn:

+ Dùng bản mẫu trên giấy

+ Sử dụng bản mẫu tự động

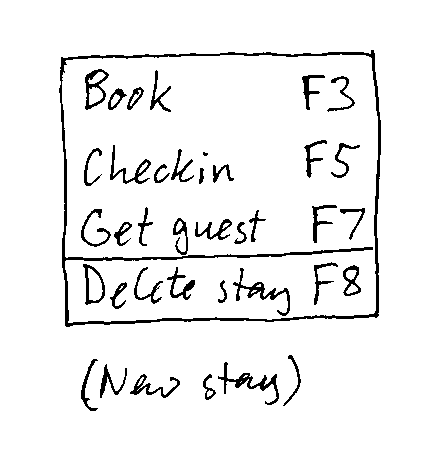
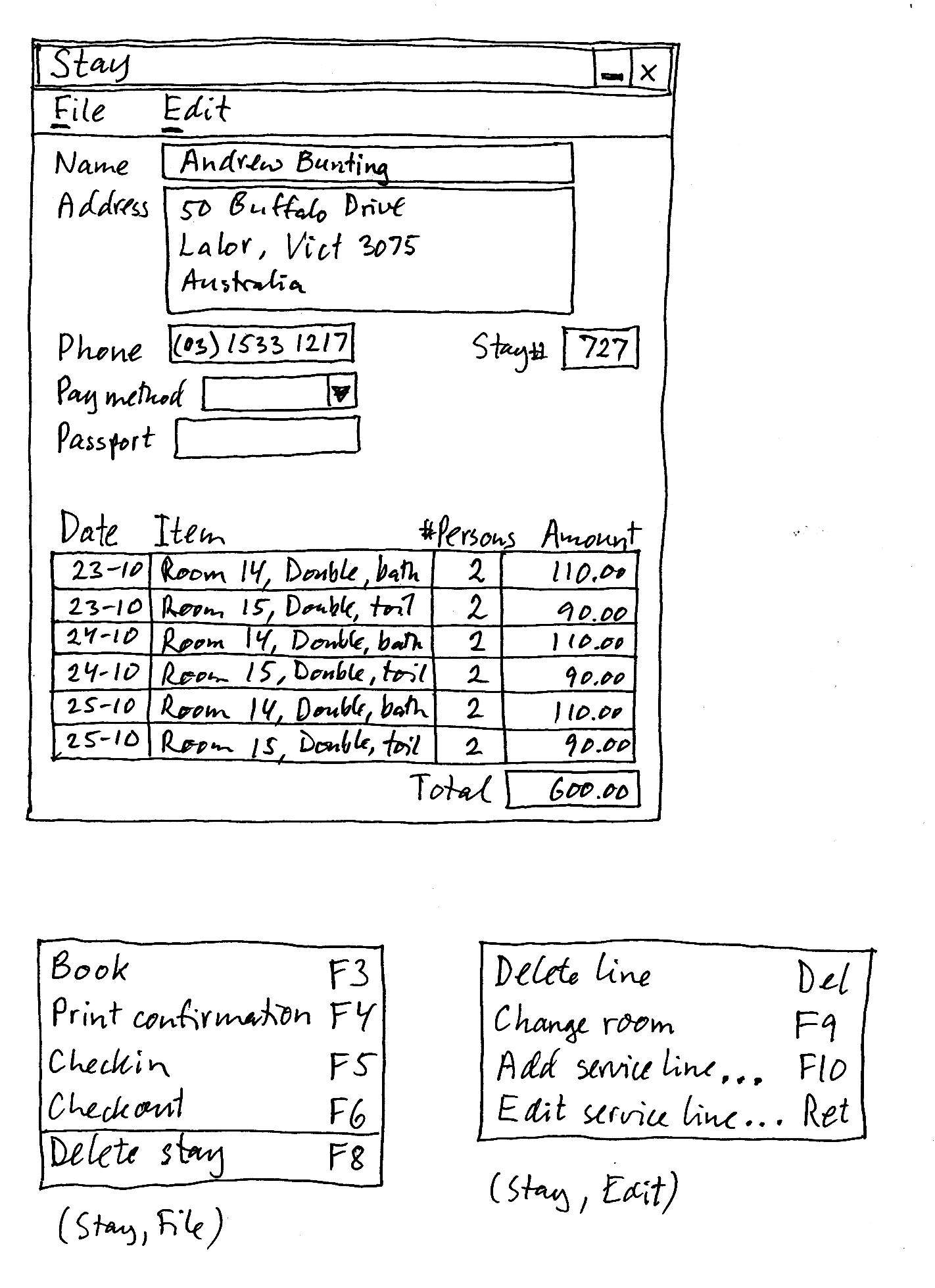
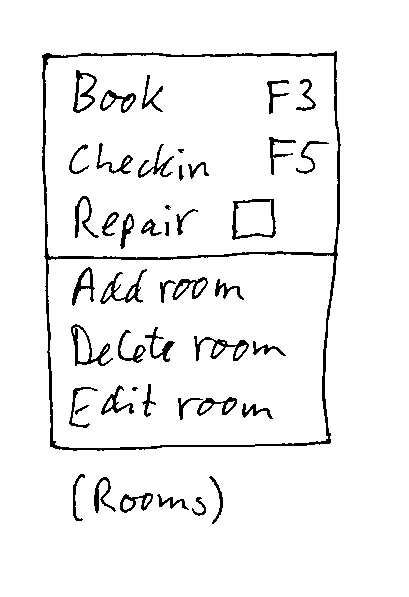
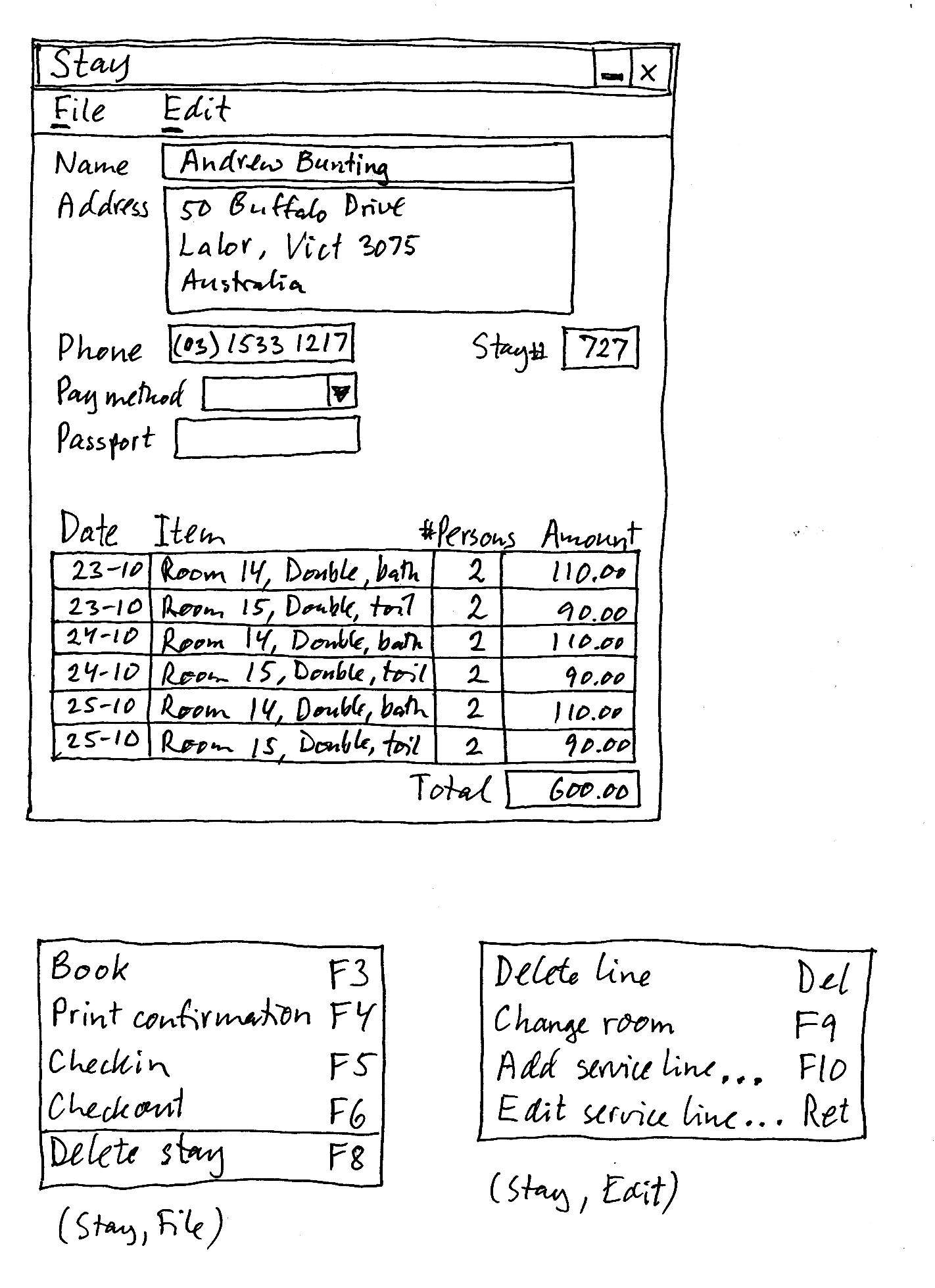
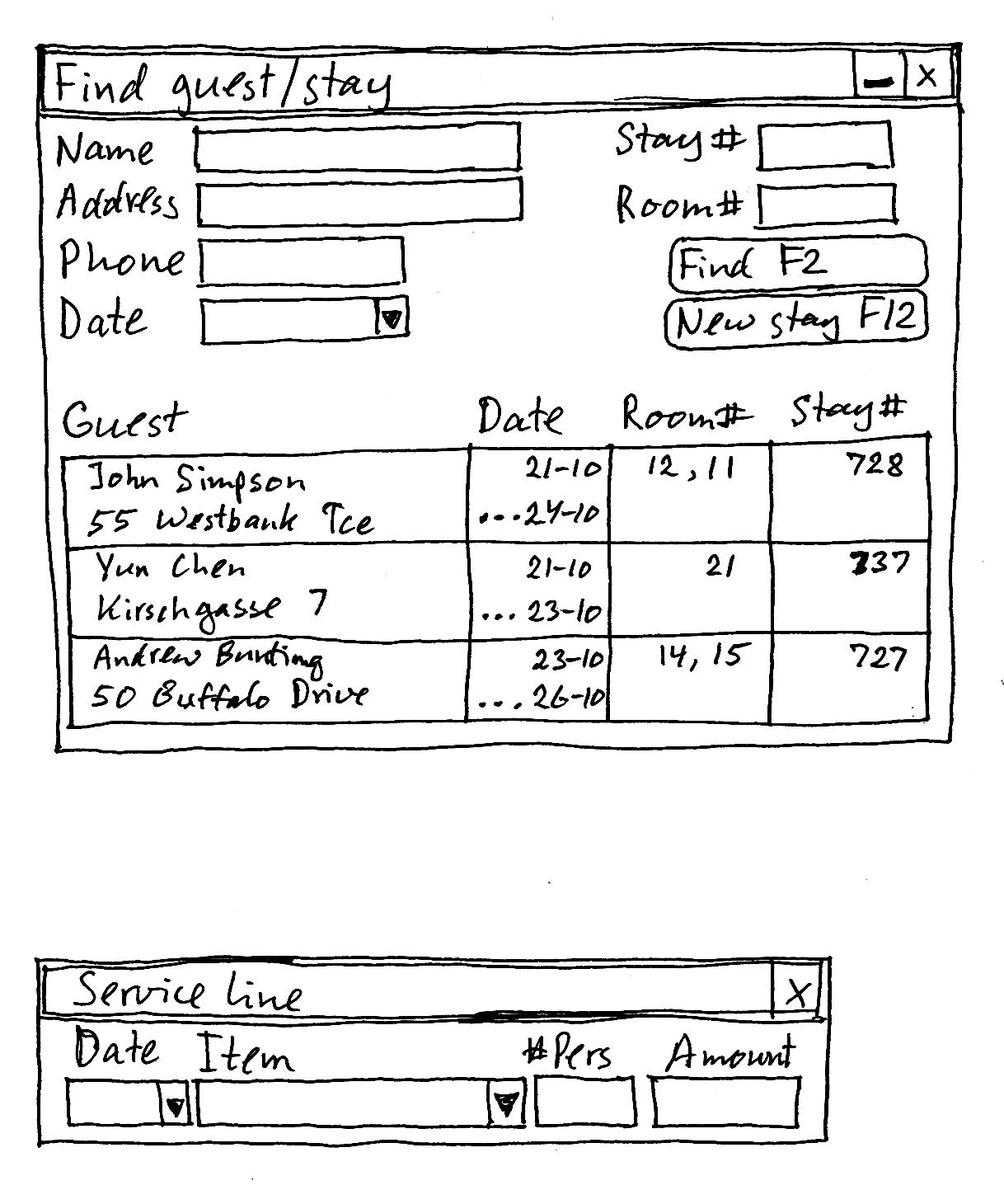
* + Một số kỹ thuật liên quan đến xây dựng bản mẫu:

+ Dựa trên kịch bản

+ Lập trình trực quan

+ Dựa trên internet

* Kiểm thử và đánh giá: Bản chất là kiểm tra tinh khả dụng thông qua bản mẫu
  1. Ví dụ minh họa về bản mẫu phần mềm quản lý khách sạn



2. **Các vấn đề thường gặp trong thiết kế giao diện**
   1. Tương tác
   2. Biểu diễn thông tin/dữ liệu:

* Người sd quan tâm đến độ chính xác của thông tin hay quan tâm đến mqh của dữ liệu
* Sự thay đổi thông tin gồm các cách thức và tốc độ
* Sự thay đổi của hệ thống có yêu cầu sự phúc đáp của người sử dụng hay không?
* Kiểm tra xem có cách thức biểu diễn tương tác trực tiếp không? (bao gồm cả thao tác với thông tin)

1. **Phân loại bản mẫu**

* Bản mẫu = tay: Dùng bút vẽ trên giấy
* Bản mẫu = công cụ
* Bản mẫu chức năng:
  + Các nút bấm/ menu đã được lập trình (giải quyết tronng trường hợp có nhiều màn hình)
  + Xác định thứ tự ưu tiên
  + Thực hiện kiểm thử tính khả dụng cho các màn hình, quan trọng đó, sau đó thiết kế các màn hình còn lại

**Chương 3 : Biểu diễn dữ liệu giao diện**

1. **Biểu diễn thông tin**

* Bản chất: Trình diễn thông tin lên hệ thống theo 2 cách trực tiếp hoặc gián tiếp (đồ họa)
* Phân loại thông tin:
  + Thông tin tĩnh: Khởi tạo khi bắt đầu phiên làm việc, không thay đổi, có thể là số hoặc văn bản
  + Thông tin động: Có thể thay đổi trong quá trình làm việc, các thay đổi đều được thông báo với người sử dụng, có thể là số hoặc văn bản

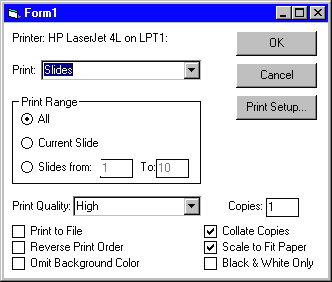
1. **Một số yếu tố ảnh hưởng tới trình bày thông tin**

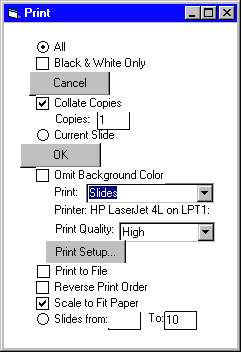
* Người dùng quan tâm tới chính xác của thông tin hay quan tâm tới mối quan hệ của dữ liệu
* Sự thay đổi của thông tin: Cách thức, tốc độ
* Sự thay đổi của hệ thống có yêu cầu sự phúc đáp của người dùng hay không
* Kiểm tra có cách thức biểu diễn thông tin trực tiếp hay không (bao gồm cả thao tác với thông tin)
* Kiểu thông tin sẽ trình bày là gì (text, số)

1. **Các luật biểu diễn thông tin**

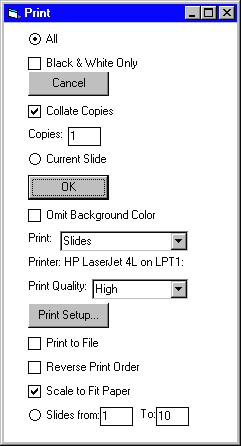
* Mục đích của luật biểu diễn thông tin: Cần phải biểu diễn nhiều thành phần dữ liệu trên một giao diện 🡪 Cần bố trí hợp lý các thành phần. Có các luật sau:
  + Luật lân cận: Tất cả các biểu diễn đồ họa gần nhau được coi như một đối tượng
  + Luật đóng: Tất cả các thành phần bên trong một dây truyền khép kín được xem là một đối tượng
  + Luật liên tục: Các thành phần biểu diễn trên một đường trơn được coi như một đối tượng
  + Luật đường thẳng là sự mở rộng của luật liên tục

Ví dụ về các luật biểu diễn:





**C**



**B**

**A**

* Ở hình A: Sắp xếp thông tin theo luật lân cận và luật đóng. Những thành phần gần gũi, tương đồng với nhau được sắp xếp cạnh nhau, tạo thành 1 khối.
* Ở hình B: Sắp xếp thông tin theo luật đường thẳng (căn trái). Mặc dù nó tuân theo luật nhưng nó hoàn toàn khó hiểu và kém logic. Ở hình B đã không sử dụng luật tương tự
* Ở hình C: Tất cả các văn bản được sắp xếp theo logic bảng chữ cái.

1. **Các đối tượng văn bản**

* Chiều dài của dòng văn bản
* Chiều dài của dòng là yếu tố ảnh hưởng đến việc đọc văn bản
* Quan trọng trong việc theo dõi văn bản (Nếu dòng quá dài ta sẽ rất dễ bị nhầm dòng khi quay lại dòng mới
* Dòng quá dài dẫn đến hiện tượng nhảy dòng
* Khi biểu diễn chiều dài của văn bản, không nên để chiều dài dòng quá dài
* Màu sắc văn bản
* Ngày nay, người ta có thể sử dụng màu để thể hiện sự tương tự.
* Nên sử dụng các màu đơn giản
* Màu sắc có thể sử dụng như một chiều thông tin phụ (Trợ giúp quá trính phân tích thông tin phức tạp)
* Một số gợi ý khi sử dụng màu:
  + Việc sử dụng màu phải thống nhất
  + Sử dụng việc thay đổi màu sắc để thông báo thay đổi hệ thống
  + Có thể sử dụng mã màu (Chú ý việc sử dụng phải có nghĩa và thống nhất)
  + Chú ý đến việc sử dụng cặp màu
  + Sự tương phản màu sắc
  + Mức độ đậm nhạt
  + Tạo sự tương phản bằng chuyển động
  + Sử dụng ở mức độ vừa phải, hài hòa, tránh lạm dụng
  + Muốn tạo ra tương phản, đầu tiên phải chọn màu nền từ đó mới quyết định màu tương phản

1. **Định dạng biểu diễn dữ liệu**

* Thông qua trực quan hóa nhận biết dc

1. **Chuyên môn hóa dữ liệu**

* Giúp người dùng hiểu biết về dữ liệu
* Lượng lớn thông tin 🡪 chuyên môn hóa

Chương 4: Model for UID (Mô hình cho thiết kế giao diện người sử dụng)

1. **Yếu tố con người trong thiết kế giao diện**

* Khả năng nhớ của con người là giới hạn
  + Trung bình là nhớ 7 hạng mục thông tin
  + Nếu chúng ta đưa quá nhiều thông tin cho người dùng 🡪 người dùng sẽ quên mất
* Con người có thể mắc lỗi
* Các cá nhân đều có các điểm khác nhau
* Người sử dụng có sở thích tương tác khác nhau
* Người thiết kế cần xem xét đến kinh nghiệm của người sử dụng
* Người thiết kế cần xem xét đến sự giới hạn của con người

1. **Nguyên tắc thiết kế**

* Thân thiện với người sử dụng (user familiarity)
* Thống nhất (consistency)
* Giảm thiểu sự bất ngờ (minimal surprise)
* Recoverability (khả năng khôi phục)
* Hướng dẫn người sử dụng (user guidance)
* Đa dạng

1. **Mô hình trí tuệ**

* Mô hình trí tuệ mô tả trực quan hóa dữ liệu và cách thức vận hành hệ thống
* Mô hình trí tuệ của hệ thống thông thường không khớp hoàn toàn với hệ thống thực
* Xem xét, khai thác mô hình trí tuệ
* Bài học 2:
* Các mô hình được xây dựng một cách vô thức
* Phần lớn người sử dụng không thể giải thích chi tiết nhận thức của họ về hệ thống
* Tuy nhiên, người dùng sẽ có phản ứng
* Bài học 3:
* Mô hình trí tuệ có thể sẽ không được sử dụng
* Nhận thức của người sử dụng về thế giới thực phụ thuộc khá nhiều vào kinh nghiệm của người sử dụng
* Mô tả cách thức thực hiện: Nói cách khác chuyển đổi mô hình ứng dụng thành mô hình dữ liệu và mô hình chức năng

2. Mức tên miền (domain level)

- Khi thiết kế một hệ thống, thực hiện phân tích tên miền.

Phân tích miền là một phần của hầu hết các phương pháp tiếp cận phát triển, và không phải cái gì đặc biệt cho thiết kế người dùng.

* Mô tả công việc. Là một phần của việc phân tích miền, chúng tôi nghiên cứu những gì người dùng thực hiện và xác định các nhiệm vụ chủ yếu của họ.
* Data models: Trong hầu hết các trường hợp, điều quan trọng là làm cho một số loại mô hình dữ liệu.
* Nó mô tả dữ liệu trong một cách đó là hoàn toàn độc lập của nó như thế nào sẽ được hiển thị trên màn hình và làm thế nào nó sẽ được lưu trữ trong máy tính.

**Data**

* Dữ liệu bền vững: được lưu trữ trên máy tính lâu dài
* Dữ liệu hội thoại: Chỉ tồn tại trong thời gian ngắn và khi chương trình tắt thì dữ liệu hội thoại cũng sẽ mất
* Thông tin trợ giúp: Xuất hiện trên màn hình, đi cùng giao diện
* Chức năng:
  + Bản chất là thực hiện các lệnh
  + Chức năng nhập liệu

Thao tác trực tiếp dữ liệu, giúp nhận được dữ liệu trên màn hình

* + Chức năng ngữ nghĩa: làm công việc quan trọng, liên quan đến tính toán
  + Chức năng tổng hợp: Là chức năng bao gồm nhiều chức năng con
  + Chức năng tìm kiếm: Tìm một số dữ liệu đặt biệt trong số dữ liệu đã có
  + Chức năng di chuyển (Điều hướng) di chuyển dữ liệu trên một màn hình, dùng các nút bấm

Chương 5: Phân tích yêu cầu giao diện

1. **Lý do cần quan tâm đến phân tích yêu cầu**

* Cần giao diện tốt
* Hiểu hoàn toàn miền ứng dụng
* Tìm hiểu về thế giới thực của bài toán
* Cần trả lời 4 câu hỏi:
  + What
  + Why: Tại sao phải cần có hệ thống mới
  + Who: ai là người sử dụng
  + Which: Dữ liệu liên quan là gì
* Thể hiện nhìn nhận của chúng ta về hệ thống mới và thường được tài liệu hóa : mô tả chung, mô tả yêu cầu, mô tả công việc cần phải làm…
* Tài liệu cần thiết : Mô hình dữ liệu, mô tả công việc cần làm

1. **Mô hình dữ liệu và mô tả dữ liệu**

* Rất quan trọng đó là cách thức trình diễn dữ liệu trên màn hình, tuy nhiên chúng ta cần xác định dữ liệu ấy được mô tả như thế nào (xác định dữ liệu khác với trình diễn dữ liệu)
* Mô hình dữ liệu rất phù hợp với người thiết kế và phát triển, không phù hợp với người sử dụng
* Bản mô tả công việc giải thích chi tiết công việc của người sử dụng
* Bản mô tả công việc giải thích chi tiết công việc của người sử dụng
* Có nhiều cách mô tả công việc
* Các quy tắc mô tả Chức năng
* Quy tắc 1 : Thông thường 1 công việc từ khi bắt đầu đến khi kết thúc thường không có ngắt
* Quy tắc 2 : Phiên
  + Các công việc nhỏ, có mối quan hệ với nhau thì thường thực hiện trong cùng một phiên mà không có ngắt nên được gom nhóm và mô tả chung (Chỉ dùng một bản mô tả)
  + Công việc là đóng nhưng khi chúng ta gom nhóm trong cùng 1 phiên thường thường được xem như 1 công việc 🡪 Loại trừ đi những công việc quá nhỏ
* Quy tắc 3 : Quy tắc về mức độ của miền ứng dụng
  + Mô tả người sử dụng làm gì, máy tính cần làm gì cùng với nhau
* Quy tắc 4 : Không lập trình
  + Không đ quá sâu vào chi tiết cách thức thực hiện 🡪 mất sự sáng sủa
* Hoạt cảnh
* Các hoạt cảnh có tính chất thu hút, cung caaspo cho chúng ta hiểu biết về vùng làm việc. Tuy nhiên các hoạt cảnh không mô tả hết các công việc
* Use case
* Thường là sử dụng UML
* Hoạt cảnh là trường hợp đặc biệt của ca sử dụng
* Ca sử dụng là một hoạt cảnh thể hiện rõ máy tính làm những gì và người sử dụng làm gì. Có nhiều hoạt cảnh mô tả ca sử dụng (text, ngôn ngữ tự nhiên, sơ đồ…. )

Chương 6: Cửa sổ ảo

YC: Cung cấp những hiểu biết về phương pháp thiết kế cửa sổ ảo.

1. Phương phap thiết kế

* Chọn ra nhiệm vụ hay chức năng quan trọng và được thực hiện thường xuyên.
* Tập hợp các dữ liệu vào một cửa sổ ảo, và phác thảo nội dung của mỗi cửa sổ.
* Chọn các công việc tiếp theo và tưởng tưởng các dữ liệu người dùng muốn thấy để thực hiện nó.
* Nếu công việc này tái sử dụng dữ liệu được trong cửa sổ ảo đã được quy hoạch , xem xét việc tái sử dụng các cửa sổ ảo.
* Nếu nhiệm vụ sử dụng dữ liệu bổ sung hợp lý liên quan đến một cửa sổ ảo đã lên kế hoạch, xem xét việc mở rộng cửa sổ này. Nếu không, xác định cửa sổ ảo mới với các dữ liệu bổ sung.
* Tiếp theo sau bước 4: thực hiện các bước tiếp theo cho đến khi tất cả các nhiệm vụ được hỗ trợ đúng đắn.
* Khi phác thảo các nội dung của một cửa sổ ảo, một số nhà thiết kế chỉ lập danh sách các lĩnh vực trước mắt, những người khác làm nhiều hơn hoặc ít thành phác thảo đồ họa.
* Cho các nhiệm vụ, đối với nhiều bước, nhiều loại dữ liệu, ta cần xử lí chúng từng bước từng bước một như thể chúng là các nhiệm vụ riêng biệt.

1. Quy tắc thiết kế

* Các mẫu cửa sổ: Giữ tổng số mẫu cửa sổ nhỏ nhất có thể. Hãy thử sử dụng lại các mẫu cửa sổ qua các nhiệm vụ.
* Rất ít trường hợp một cửa sổ cho một nhiệm vụ: Đối với mỗi công việc, người dùng phải truy cập vào vài cửa sổ(Điều này hỗ trợ cho công việc)
* Dữ liệu trong một cử sổ là suy nhất : tránh người dùng có thể nhìn thấy các mục dữ liệu giống nhau trong một số trường hợp cửa sổ.
* Bắt nguồn từ 1 đối tượng: một cửa sổ ảo thường được bắt nguồn từ một đối tượng duy nhất. Nó cho thấy dữ liệu về đối tượng này và đối tượng được liên quan.
* Một cửa sổ ảo có thể hơi lớn hơn so với màn hình vật lý có sẵn, nhưng không lớn hơn nhiều. Ở giai đoạn sau của thiết kế, chúng tôi có thể cắt cửa sổ ảo vào một vài màn hình nhỏ hơn hoặc sử dụng thanh cuộn để tạo ra ảo tưởng rằng đây là một màn hình lớn.
* Các cửa sổ ảo phải cung cấp tổng quan của nhiều dữ liệu, cho dù người dùng có thể cần chỉ là một phần của những dữ liệu này.
* Tất cả các dữ liệu. Thường thì tất cả dữ liệu từ các mô hình dữ liệu phải được nhìn thấy và điều chỉnh được thông qua một số cửa sổ ảo. Nếu không chúng ta có thể thiếu một cửa sổ ảo và có lẽ cũng là một nhiệm vụ.

1. Cơ sở thiết kế đồ họa

Kế hoạch cửa sổ ảo có thể nhìn khá thuyết phục, và các nhà phát triển có thể kết luận rằng họ có thể đến với các bài trình bày đồ họa khi họ thiết kế các màn hình chính thức.

Về sau khi họ làm như vậy và cố gắng để điền vào một số dữ liệu thực tế, họ nhận ra rằng những phác thảo không hoạt động.

Có thể không có đủ không gian cho dữ liệu thực tế, người dùng không hiểu những gì các cửa sổ hiển thị, hoặc người dùng không có được cái nhìn tổng quan cần thiết.

Kết quả: quay trở lại và thiết kế lại rất nhiều - hoặc bỏ qua khả năng sử dụng.

=> Vì lý do này, điều quan trọng là làm cho một thiết kế đồ họa cẩn thận trong việc kết nối với kế hoạch.

4. Vai trò của cửa sổ ảo trong phát triển

- Sản phẩm công việc(work product): bốn sản phẩm công việc quan trọng trong thiết kế giao diện người dùng:

+ Các mô hình dữ liệu và mô tả dữ liệu liên quan (từ điển dữ liệu)

+ Mô tả công việc

+ Các cửa sổ ảo

+ Danh sách các lỗi thiết kế hoặc những thứ cần tìm hiểu

* Trình tự thiết kế: Cho chúng ta biết nên làm công việc nào trước, Cho đến bây giờ, chúng ta vẫn luôn có thứ tự công việc đầu tiên là mô hình dữ liệu, mô tả công việc và sau đó là làm cửa sổ ảo. Trong một số trường hợp, trình tự này không sao, nhưng với một số nhà thiết kế, họ lại thực hiện thiết kế cửa sổ ảo trước mô hình dữ liệu, và thiết kế.

Chương 7: Thiết kế chức năng dựa trên VW

1. Các use cases mô tả hành động và feedback từ các chức năng phần mềm
2. Chức năng của Undo

* Chức năng chuyển hướng: Trong hầu hết các trường hợp, bạn không cần phải có chức năng undo riêng để lùi lại. Chức năng chuyển hướng này giúp cho người sử dụng có thể quay trở lại bước đầu tiên.
* Chức năng tìm kiếm: Chức năg tìm kiếm là một dạng của chức năng chuyển hướng thường được điều khiển bằng một số luồng dữ liệu.

+ thông thường, Undo đã được xây dựng vào menu

+ Từ tiêu chuẩn của tìm kiếm là đơn giản và dễ nhớ, cho nên không cần Undo cho những sự thay đổi của người dùng

+ Tuy nhiên, với các tiêu chí phức tạp, cho các trường hợp địa chỉ Web dài , cần thiết cho nhu cầu sử dụng Undo .

1. Sự cần thiết của chức năng Undo trên các UI

* Xác định các chức năng cho Undo : Nhìn tất cả các chức năng và xác định cái cần được đảo ngược.
* Hỗ trợ trượt: Cung cấp một nút Undo để hoàn tác các chức năng mới nhất.
* Hỗ trợ thăm dò: Một cách tốt để tìm hiểu một hệ thống là để khám phá nó - hãy thử những điều khác nhau và xem những gì sẽ xảy ra. Các nút undo giúp người sử dụng thoát ra khỏi mớ hỗn độn. Nếu có nguy hiểm, nó có thể là cần thiết để có một chế độ, nơi người dùng có thể làm bất cứ điều gì, nhưng chức năng nguy hiểm không có bất kỳ ảnh hưởng đến môi trường xung quanh.
* Hỗ trợ sửa chữa sai lầm: Cho phép người dùng lựa chọn một thực thể và lùi lại thay đổi nó

1. Chuyển đổi cửa sổ ảo thành các màn hình giao diện : kích thước màn hình

Cho đến bây giờ chúng ta đã giả định rằng có một màn hình lớn ('bàn'), nơi chúng ta có thể thấy tất cả các cửa sổ ảo cùng một lúc.

Ở giai đoạn tiếp theo của thiết kế chúng ta có để đóng gói tất cả mọi thứ vào màn hình vật lý hạn chế.

Nếu chúng ta đã làm theo nguyên tắc thiết kế của việc giữ các cửa sổ ảo gần với những gì là chất có thể trên màn hình,

chúng ta chỉ cần làm một số 'trimming' của mỗi cửa sổ ảo.

Tuy nhiên, chúng ta có thể cố gắng giữ nhiều cửa sổ ảo trên màn hình cùng một lúc để hỗ trợ công việc tốt hơn.

Điều này có thể được thực hiện theo nhiều cách khác nhau tùy thuộc vào nền tảng hệ thống của chúng ta chạy.

# CHƯƠNG 8: BẢN MẪU VÀ GIẢI QUYẾT KHIẾM KHUYẾT

## Bản mẫu hệ thống

Ở giai đoạn này, để làm cho 1 bản mẫu tốt:

- Các cửa sổ ảo có sự xuất hiện đồ họa của các trang.

- Danh sách chức năng trình bày cho các chức năng cho từng trang và sự xuất hiện về đồ họa của chúng.

- Các thông số kỹ thuật cho thấy những gì mỗi chức năng phải làm.

## Quy trình thiết kế

**-** Xác định kiểu bản mẫu: bản mẫu vẽ bằng tay, vẽ bằng công cụ, bản mẫu màn hình, bản mẫu chức năng.

Thông thường bản mẫu bằng tay là lựa chọn tốt nhất.

- Xác định biểu diễn đồ họa cho mỗi trang: lấy thông tin từ các cửa sổ ảo, thông tin biểu diễn các chức năng.

- Nếu bản mẫu thử nghiệm là rỗng => điền các trường dữ liệu cần thiết và kiểm thử khả năng sử dụng.

Nếu bản mẫu có chức năng, lấy dữ liệu từ csdl và thử nghiệm.

- Xây dựng menu thiết yếu: trình bày trên giấy, đầu vào là sự trình bày về các chức năng.

- Xây dựng các hộp thoại thông báo lỗi: đầu vào là các chức năng. Chúng sẽ cho biết lỗi có thể xảy ra.

- Sử dụng các danh sách dạng combobox, listbox, các hộp thoại, các đoạn văn bản trợ giúp.

Theo phương pháp cửa sổ ảo, chúng ta chỉ kiểm tra được tính dễ hiểu của cửa sổ ảo. Tiếp theo sẽ kiểm tra tính khả dụng của bản mẫu.

- Thực hiện đánh giá dựa trên kinh nghiệm của các bản mẫu để tìm ra các vấn đề ngầm.

- Kiểm tra tính khả dụng của bản mẫu: nếu các cửa sổ ảo đã được thử nghiệm về sự dễ hiểu và sửa đổi cho phù hợp, nhất là vấn đề về tính khả dụng. Ở giai đoạn này nên có các tên chức năng, các thông báo lỗi và các chi tiết khác.

Tuy nhiên nhiều vấn đề có thể là khó khăn.

-Tìm ra nguyên nhân của các khiếm khuyết về tính khả dụng, rà soát và chỉnh sửa lại bản mẫu.

Trong trường hợp đặc biệt có thể là cần thiết phải xây dựng các chức năng mẫu của một số phần giao diện người dùng.

- Kiểm tra tính khả dụng cho đến khi kết quả đáp ứng được các yêu cầu về tính khả dụng hoặc được coi là chấp nhận được. Nếu không giải quyết được thì cho vào danh sách khiếm khuyết chính thức.

## - Lập trình và kiểm thử hệ thống.

Sau khi kiểm tra tính khả dụng và sửa đổi bản mẫu,tiến hành lập trình và kiểm thử hệ thống.

Thủ tục thực hiện:

*a) Xây dựng chương trình:*

- Cơ sở để xây dựng chương trình là bản mẫu và đặc tả: Nếu bản mẫu được xây dựng bằng công cụ hoặc là bản mẫu chức năng => có thể sửa thành cửa sổ thực. Tuy nhiên không nên tái sử dụng chương trình riêng của mình, nó có thể không phù hợp với 1 phiên bản chuyên nghiệp và có cấu trúc.

Trong hầu hết các hệ thống, một phần lớn của hệ thống sẽ đối phó với các vấn đề kỹ thuật, cho sự tương tác với hệ thống máy tính khác. Phát triển các phần này thường tiến hành đồng thời với việc thiết kế giao diện người dùng.

Phần này thường phát triển cùng với việc phát triển giao diện.

*b) Kiểm thử hệ thống*

Mục đích là để tìm và sửa chữa lỗi cũng như khiếm khuyết của chương trình. Kiểm thử theo từng unit, sau đó tích hợp các phần đã kiểm thử lại với nhau.

*c) Khi hệ thống đã sẵn sàng thì kiểm thử thêm 1 lần nữa tính khả dụng của hệ thống. Mục đích là để tìm kiếm các khiếm khuyết về tính khả dụng còn lại, không phải là lỗi.*

Rà soát 1 lần nữa các khiếm khuyết liên quan đến giao diện.

*d) thực hiện các công việc tiếp theo*

Kiểm thử chấp nhận: người dùng có chấp nhận ko, dựa trên các yêu cầu; tài liệu hướng dẫn người sử dụng, huấn luyện đào tạo người sử dụng.

## Khiếm khuyết và cách sửa chữa

- Khi xây dựng 1 danh sách các khiếm khuyết, chúng ta nên theo dõi cả tình trạng của chúng.

- nó cũng nên được xem xét lại vào cuối của dự án để xem cách họ đã được phát hiện và làm thế nào họ chúng được gỡ bỏ.

*Quá trình xử lý khiếm khuyết:*

*a)* Xây dựng danh sách hoặc 1 csdl để ghi lại các khiếm khuyết. Nên làm điều đó sớm trong quá trình thiết kế.

+ mã khiếm khuyết (thường là 1 số tuần tự)

+ Mô tả khiếm khuyết

+ cách thức tìm ra

+ Phương pháp sửa chữa

+ trạng thái khiếm khuyết

Có thể bổ sung người tìm ra, ngày tìm ra, số giờ làm việc để đối phó với nó.

b) ghi nhận các khiếm khuyết trong quá trình thiết kế, kiểm thử tính khả dụng, lập trình, kiểm thử hệ thốn.

c) Khi một lỗi được sửa chữa, ghi lại cách chữa và thay đổi trạng thái Done.

d) định kì rà soát các khiếm khuyết và quyết định làm gì với những khiếm khuyết chưa được xử lý.

Các trạng thái khiếm khuyết:

+ đang chờ xử lý: các khiếm khuyết đã được ghi nhận nhưng chưa quyết định phải làm gì với nó.

+ Từ chối. Nó đã được quyết định không phải làm bất cứ điều gì về các khiếm khuyết, ví dụ vì nó là một vấn đề sai.

+ Bỏ qua. Mặc dù có một vấn đề, nó đã được quyết định không phải làm bất cứ điều gì về nó. Vấn đề là nhỏ so với chi phí để đối phó với nó. Hoặc chưa cần thiết được xử lý.

+ Training: Vấn đề là quan trọng, nhưng nó là khó khăn để đối phó với trong hệ thống. Người dùng cần được đào tạo để tránh các vấn đề, hoặc các tài liệu hướng dẫn người sử dụng sẽ giúp người sử dụng tránh ra.

+ Đã xử lý: Vấn đề đã được gỡ bỏ - các lỗi được sửa chữa.

+ Phiên bản x: nó sẽ được sữa chữa ở phiên bản thứ x.

+ trùng lặp: khiếm khuyết đã được thông báo rồi.

## Bổ sung tính năng cho phiên bản của sản phầm

Một số các khuyết tật không được phát hiện bởi các thiết kế hệ thống hoặc các kiểm thử tính khả dụng.

## Phân biệt khiếm khuyết và yêu cầu thay đổi

**-** Khiếm khuyết là điều mà hệ thống không đáp ứng được yêu cầu đã đặt ra.

- Yêu cầu thay đổi: yêu cầu mới từ phía người sử dụng.

## Vấn đề, nguyên nhân và giải pháp

Có rất nhiều văn bản về làm thế nào để tìm ra vấn đề khả năng sử dụng, nhưng không về làm thế nào để sửa chữa chúng.

- 1 số nguồn thông tin tìm ra nguyên nhân

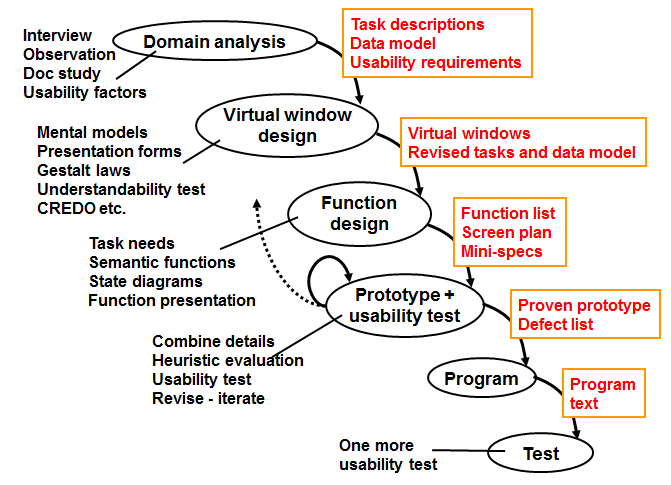
a) tương tác với người sử dụng

b) Yêu cầu người sử dụng lý do tại sao ông đã làm điều này, những gì ông mong đợi, vv

c) Dùng mắt để quan sát điều người sử dụng thực sự nhìn vào

d) Sử dụng kinh nghiệm từ người dùng khác về những gì mà người dùng thường làm.

Quá trình thiết kế giao diện:



Phân tích miền ứng dụng -> thiết kế cửa sổ ảo -> thiết kế chức năng -> xây dựng bản mẫu, kiểm thử tính khả dụng -> lậ trình -> kiểm thử

# CHƯƠNG 9: CÁC CHUẨN THIẾT KẾ

## Phương pháp tiếp cận

- Giả định rằng 1 tiêu chuẩn là 1 đặc tả.

- Các tiêu chuẩn này mang lại lợi ích trong nhiều lĩnh vực ví dụ:

Máy ATM có thể đọc thẻ tín dụng, và trình biên dịch mà có thể đọc ngôn ngữ lập trình. Một số tiêu chuẩn HCI cũng thuộc loại này: nhiều hướng dẫn thiết kế cung cấp một đặc điểm kỹ thuật chi tiết về bản chất của giao diện người dùng.

Mặc dù các chuẩn giao diện cung cấp lợi ích về tính nhất quán nhưng các tiêu chuẩn có thể lạc hậu do sự phát triển về công nghệ. Vì vậy hầu hết các công việc trên các tiêu chuẩn quốc tế cho HCI đã không nói về đặc điểm kỹ thuật chính xác, nhưng thay vào đó đã tập trung vào các nguyên tắc cần được áp dụng để tạo ra một giao diện, đáp ứng nhu cầu của người sử dụng và nhiệm vụ(, quan tâm nhiều tới nguyên lý hơn là thông tin trực tiếp.).

Các tiêu chuẩn này được chia ra làm 2 loại:

+ Từ trên xuống (top-down): xác định tính khả dụng là mục tiêu của sản phầm cần đáp ứng.

+ Từ dưới lên (Bottom-up): quan tâm nhiều hơn tới các khía cạnh của giao diện, đảm bảo hệ thống dễ sử dụng.

Quan điểm chất lượng bắt nguồn từ yếu tố con người, và các tiêu chuẩn của loại hình này được áp dụng trong bối cảnh rộng lớn của thiết kế và chất lượng các mục tiêu.

Quan điểm sản phẩm theo định hướng tập trung vào việc thiết kế các thuộc tính cụ thể, và liên quan chặt chẽ hơn với nhu cầu của các nhà thiết kế giao diện và vai trò của khả năng sử dụng trong công nghệ phần mềm.

## Tính khả dụng như là 1 tiêu chuẩn chất lượng

- Các tiêu chuẩn liên quan đến khả năng sử dụng như là một mục tiêu chất lượng cao, và khả năng sử dụng được xác định theo cách này trong ISO 9241-11:

- Tính khả dụng: mức độ mà một sản phẩm có thể được sử dụng bởi người dùng quy định để đạt được mục tiêu cụ thể, có hiệu quả, hiệu quả và sự hài lòng trong một bối cảnh cụ thể sử dụng.

- Tiêu chuẩn loại này có thể được sử dụng để hỗ trợ các hoạt động sau:

+ đặc điểm kỹ thuật về chất lượng và khả năng sử dụng các yêu cầu và đánh giá tổng thể đối với các yêu cầu này (ISO 9241-11 và tiêu chuẩn ISO / IEC 14.598-1)

+ kết hợp của khả năng sử dụng thành một hệ thống chất lượng (ISO 9241-11)

+ kết hợp của khả năng sử dụng vào quá trình thiết kế (ISO / IEC 13.407).

## Tiêu chuẩn sản phầm theo định hướng

Tiêu chuẩn mà đối phó với khả năng sử dụng trong các điều khoản của các thuộc tính đó phải được thiết kế thành một sản phẩm phần mềm để làm cho nó dễ dàng để sử dụng:

ISO 9241: Yêu cầu Ergonomics cho công việc văn phòng với thiết bị đầu cuối hiển thị hình ảnh: Phần 10, 12-17: thiết kế đối thoại

ISO / IEC 10.741-1 tương tác đối thoại - Điều khiển con trỏ để chỉnh sửa văn bản

ISO / IEC 11581 biểu tượng Icon và chức năng

ISO / IEC 9126 đánh giá sản phẩm phần mềm - đặc điểm chất lượng và hướng dẫn sử dụng của chúng

## Các tiêu chuẩn này có thể được sử dụng trong các cách sau:

Để xác định các chi tiết của sự xuất hiện và hành vi của giao diện người dùng

Để cung cấp các hướng dẫn chi tiết về thiết kế giao diện người dùng

Để cung cấp các tiêu chuẩn đánh giá của giao diện người dùng

Tuy nhiên các thuộc tính mà một sản phẩm đòi hỏi cho khả năng sử dụng phụ thuộc vào bản chất của người sử dụng, nhiệm vụ và môi trường.

Một sản phẩm không có khả năng sử dụng nội tại, chỉ có một khả năng được sử dụng trong một bối cảnh cụ thể.

ISO 9241-11 có thể được sử dụng để giúp hiểu rõ bối cảnh trong đó các thuộc tính cụ thể có thể được yêu cầu.

## Các loại

Tiêu chuẩn liên quan đến khả năng sử dụng có thể được phân loại như là chủ yếu quan tâm:

Việc sử dụng của sản phẩm (tính hiệu quả, hiệu quả và sự hài lòng trong một bối cảnh cụ thể sử dụng).

Giao diện người dùng và tương tác.

Quá trình này được sử dụng để phát triển sản phẩm.

Khả năng của một tổ chức để áp dụng sử dụng làm trung tâm thiết kế.

## Phát triển của chuẩn ISO

tiêu chuẩn quốc tế cho HCI được phát triển dưới sự bảo trợ của Tổ chức Quốc tế về Tiêu chuẩn hóa (ISO) và Ủy ban kỹ thuật điện quốc tế (IEC).

ISO và IEC bao gồm các cơ quan tiêu chuẩn quốc gia từ các nước thành viên.

Các công việc kỹ thuật diễn ra trong các nhóm công tác của các chuyên gia, đề cử bởi các ủy ban tiêu chuẩn quốc gia.

Các tiêu chuẩn được phát triển trong một thời gian vài năm, và trong giai đoạn đầu các tài liệu được công bố có thể thay đổi đáng kể từ phiên bản lên phiên bản cho đến khi được sự đồng thuận.

Theo tiêu chuẩn trở nên trưởng thành hơn, từ giai đoạn ban Dự thảo trở đi, bỏ phiếu chính thức diễn ra bằng cách tham gia các tổ chức thành viên quốc gia.

Tình trạng của tài liệu ISO và IEC được tóm tắt trong tiêu đề của tiêu chuẩn.

**Sử dụng trong bối cảnh và phương thức kiểm thử**

### *ISO 9241-11*: Hướng dẫn về khả năng sử dụng (1998)

Tiêu chuẩn này (đó là một phần của tiêu chuẩn ISO 9241 loạt mô tả trong mục 2.1) đưa ra định nghĩa về khả năng sử dụng được sử dụng trong tiêu chuẩn ergonomic liên quan sau:

Tính khả dụng: mức độ mà một sản phẩm có thể được sử dụng bởi người dùng quy định để đạt được mục tiêu cụ thể, có hiệu quả, hiệu quả và sự hài lòng trong một bối cảnh cụ thể sử dụng.

## ISO / IEC 9126-1:

Công nghệ phần mềm - Chất lượng sản phẩm - Phần 1: Mô hình chất lượng (2001)

Mô tả sáu loại chất lượng phần mềm có liên quan trong quá trình phát triển sản phẩm bao gồm chất lượng sử dụng (tương tự như định nghĩa về khả năng sử dụng trong tiêu chuẩn ISO 9241-11), với khả năng sử dụng định nghĩa hẹp hơn như dễ sử dụng

## Phương pháp IEC DTR 19.764

Hướng dẫn, và tiêu chí tham khảo 1,3 ISO cho khả năng thích ứng văn hóa và ngôn ngữ trong các sản phẩm công nghệ thông tin (2003)

Định nghĩa một phương pháp và một danh sách kiểm tra hướng dẫn để đánh giá khả năng thích ứng văn hóa trong phần mềm, phần cứng và các sản phẩm CNTT khác.

## Hệ thống chất lượng và tiêu chuẩn ISO 9001

Đối phó với khả năng sử dụng như là một phần của một hệ thống chất lượng cho thiết kế và phát triển sản phẩm, như quy định trong tiêu chuẩn ISO 9001, liên quan đến việc xác định hệ thống các yêu cầu về khả năng sử dụng, bao gồm cả các biện pháp khả năng sử dụng và mô tả có kiểm chứng trên bối cảnh sử dụng.

Những cung cấp các mục tiêu thiết kế mà có thể là cơ sở để xác minh của thiết kế kết quả.

## ISO 9001

xác định những gì là cần thiết cho một hệ thống chất lượng.

Một hệ thống chất lượng là một tập tài liệu về thủ tục nhằm đảm bảo rằng sản phẩm sẽ đáp ứng các yêu cầu ghi nhận ban đầu.

Một hệ thống chất lượng là một mong muốn (mặc dù chưa đủ) điều kiện để đạt được chất lượng của sản phẩm cuối cùng.

## ISO 9241-11

mô tả cách các tiện ích của một sản phẩm có thể được xác định, tài liệu và xác nhận như là một phần của một hệ thống chất lượng mà phù hợp với tiêu chuẩn ISO 9001.

Các bối cảnh chung của việc sử dụng cần được xác định, yêu cầu khả năng sử dụng nên được chỉ định, vấn đề khả năng sử dụng nên được theo dõi quá trình phát triển và khả năng sử dụng đạt được nên được đánh giá.

## 1.4 ISO / IEC TR 9126-4:

Công nghệ phần mềm - Chất lượng sản phẩm - Phần 4: Chất lượng số liệu sử dụng (2004)

Có những ví dụ về số liệu cho hiệu quả, năng suất, an toàn và sự hài lòng.

## 1.5 ISO 20.282:

Dễ hoạt động của các sản phẩm hàng ngày Dễ hoạt động liên quan đến khả năng sử dụng giao diện người dùng của sản phẩm hàng ngày.

Phần 1: Bối cảnh sử dụng và người sử dụng đặc tính (DIS: 2004)

Giải thích làm thế nào để xác định các khía cạnh có liên quan trong bối cảnh sử dụng và mô tả làm thế nào để xác định các đặc điểm đó gây ra biến đổi bên trong dân số người sử dụng dự định.

Phần 2: Phương pháp thử (CD TS: 2005)

Chỉ định một phương pháp thử nghiệm để đo lường sự dễ dàng trong hoạt động của bộ công lập và sử dụng các sản phẩm và các sản phẩm tiêu dùng hàng ngày.

## Công nghiệp chung 25.062 Format ISO:

Format nghiệp chung cho Báo cáo kiểm tra khả năng sử dụng (2006)

Chỉ định một định dạng để lưu văn bản báo cáo kiểm tra khả năng sử dụng tổng kết để sử dụng trong các tình huống hợp đồng, và ban đầu được xuất bản như ANSI / NCITS 354 (Bevan et al, 2002).

Dự thảo Format nghiệp chung cho yêu cầu khả năng sử dụng (2005)

Chỉ định một định dạng để lưu văn bản yêu cầu khả năng sử dụng tổng kết để hỗ trợ thông tin liên lạc đầu trong phát triển, và được kỳ vọng sẽ trở thành một tiêu chuẩn ISO.

## Giao diện phần mềm và tương tác

Các tiêu chuẩn này có thể được sử dụng để hỗ trợ phát triển giao diện người dùng theo các cách sau:

Để xác định các chi tiết của sự xuất hiện và hành vi của giao diện người dùng.

ISO 14.915 và 61.997 IEC chứa các khuyến nghị cho các giao diện đa phương tiện truyền thông.

Nhiều hướng dẫn cụ thể có thể được tìm thấy cho các biểu tượng trong ISO / IEC 11581, PDA trong ISO / IEC 18.021 và điều khiển con trỏ trong ISO / IEC 10741.

Để cung cấp các hướng dẫn chi tiết về thiết kế giao diện người dùng (ISO 9241 phần 12-17).

Để cung cấp các tiêu chuẩn đánh giá của giao diện người dùng (ISO / IEC 9126 phần 2 và 3).

Tuy nhiên các thuộc tính mà một sản phẩm đòi hỏi cho khả năng sử dụng phụ thuộc vào bản chất của người sử dụng, nhiệm vụ và môi trường. ISO 9241-11 có thể được sử dụng để giúp hiểu rõ bối cảnh trong đó các thuộc tính cụ thể có thể được yêu cầu.

sản phẩm có thể sử dụng có thể được thiết kế bằng cách kết hợp các tính năng sản phẩm và các thuộc tính được biết đến với lợi ích người sử dụng trong hoàn cảnh cụ thể sử dụng.

## ISO 9241: Yêu cầu Ergonomic cho công việc văn phòng với thiết bị đầu cuối hiển thị hình ảnh

Phần 10: nguyên tắc đối thoại (1996)

Có nguyên tắc chung ergonomic áp dụng cho việc thiết kế các cuộc đối thoại giữa con người và hệ thống thông tin: phù hợp cho công việc, phù hợp cho việc học tập, phù hợp cho individualisation, phù hợp với mong đợi của người sử dụng, tự descriptiveness, khả năng kiểm soát, và khả năng chịu lỗi.

Phần 12: Trình bày các thông tin (1998)

Bao gồm hướng dẫn về cách để đại diện cho thông tin phức tạp sử dụng chữ số và đồ họa / mã tượng trưng, bố trí màn hình, và thiết kế cũng như việc sử dụng các cửa sổ.

  Phần 13: hướng dẫn sử dụng (1998)

Cung cấp các khuyến nghị cho việc thiết kế và đánh giá các thuộc tính người dùng hướng dẫn của giao diện người dùng của phần mềm bao gồm nhắc nhở, phản hồi, giúp đỡ và quản lý hộ lỗi, on-line.

Phần 14: Menu thoại (1997)

Cung cấp các khuyến nghị cho việc thiết kế các menu được sử dụng trong các cuộc đối thoại với người sử dụng máy tính, bao gồm cả cấu trúc menu, chuyển hướng, lựa chọn tùy chọn và thực hiện, và trình bày menu.

Phần 15: đối thoại Command (1997)

Cung cấp các khuyến nghị cho việc thiết kế của ngôn ngữ lệnh được sử dụng trong các cuộc đối thoại với người sử dụng máy tính, bao gồm cả cấu trúc lệnh ngôn ngữ và cú pháp, đại diện lệnh, đầu vào và đầu ra cân nhắc, và thông tin phản hồi và giúp đỡ.

Phần 16: cuộc đối thoại trực tiếp thao tác (1999)

Cung cấp các khuyến nghị cho việc thiết kế các cuộc đối thoại trực tiếp thao tác, và bao gồm các thao tác của các đối tượng, và các thiết kế của ẩn dụ, đối tượng và thuộc tính.

Phần 17: Mẫu điền đối thoại (1998)

Cung cấp các khuyến nghị cho việc thiết kế các cuộc đối thoại mẫu điền, bao gồm cấu trúc hình thức và cân nhắc đầu ra, cân nhắc đầu vào, và hình thức chuyển hướng.

## ISO / IEC 9126: Công nghệ phần mềm - Chất lượng sản phẩm

ISO / IEC 9126-1 (mô tả trong phần 1.2) xác định khả năng sử dụng về tính dễ hiểu, learnability, khả năng hoạt động và hấp dẫn.

Phần 2 và 3 bao gồm các ví dụ về số liệu cho các đặc điểm.

Phần 2: số liệu bên ngoài (2003)

Mô tả số liệu mà có thể được sử dụng để xác định hoặc đánh giá hành vi của phần mềm khi điều hành bởi người sử dụng.

Phần 3: số liệu nội bộ (2003)

Mô tả số liệu mà có thể được sử dụng để tạo ra các yêu cầu mà mô tả các thuộc tính tĩnh của giao diện có thể được đánh giá qua sự kiểm tra mà không có hoạt động phần mềm.

## các ký hiệu biểu tượng và chức năng

ISO / IEC 11581 biểu tượng Icon và chức năng

Phần 1: Các biểu tượng - chung (2000)

Phần này chứa một khuôn khổ cho sự phát triển và thiết kế các biểu tượng, bao gồm các yêu cầu chung và khuyến nghị áp dụng cho tất cả các biểu tượng.

Phần 2: biểu tượng Object (2000)

Phần 3: biểu tượng con trỏ (2000)

Phần 4: biểu tượng Control (CD: 1999)

Phần 5: Công cụ biểu tượng (2004)

Phần 6: biểu tượng hành động (1999)