1. Nhiều kiểu người dùng khác nhau: Thường thì một giao diện tỏ ra hợp lý với một nhóm này thì không phù hợp với các nhóm còn lại. Tốt hơn hết chúng ta nên cung cấp một số phương án thay thế ngay trong một giao diện; ví dụ, hầu hết Giao diện đồ họa người dùng (Graphic user interface) cung cấp các phím tắt. Những tính năng phụ thêm đó của giao diện sẽ giúp người dùng thông thường có được cách dùng phần mềm như một người dùng chuyên nghiệp.

2. Các kinh nghiệm để một giao diện người dùng tốt: có rất nhiều kinh nghiệm được đúc kết trong thiết kế giao diện. Kinh nghiệm quan trọng nhất là việc duy trì tính nhất quán xuyên suốt chương trình, nền tảng và các chương trình tương tự. Một kinh nghiệm khác đó là kiểm soát người dùng, giảm việc bắt người dùng động não, và làm các trạng thái của chương trình được hiển thị rõ ràng

3. Các chỉ dẫn về Giao diện người dùng: hầu hết nền tảng Giao diện đồ họa người dùng như Mac OS của Apple, Microsoft Windows, GNOME hoặc KDE, đều cung cấp các nguyên tắc chỉ dẫn chung trong thiết kế giao diện người dùng. Những nguyên tắc chỉ dẫn này chi tiết hơn rất nhiều phương pháp heuristics và cung cấp đầy đủ thông tin về các trình điểu khiển sẽ được sử dụng, các đối tượng nào của menu cần phải có, và nhiều chi tiết khác nữa. Tuân theo các chỉ dẫn này sẽ làm tất cả các chương trình có sự nhất quán với số còn lại.

4. Các vấn đề về đa văn hóa: Tạo ra một chương trình có tính hữu dụng cao tại nhiều nước, nhiều nền văn hóa, là một thách thức rất lớn. Quá trình tạo ra một phiên bản chương trình cho một nhóm người dùng dựa trên ngôn ngữ hoặc quốc gia, được gọi là bản địa hóa. Mỗi màu sắc, biểu tượng đều có ý nghĩa riêng khác tại từng nên văn hóa. Chuyển ngữ một thông điệp là một nhiệm vụ khó khăn. Trong nhiều trường hợp, bản địa hóa một phần mềm không chỉ đơn thuần là sự chuyển đổi về yếu tố ngôn ngữ, mà còn phải bao gồm các yếu tố địa phương của các nước sử dụng ngôn ngữ đó, bởi vì các dùng từ và câu có sự thay đổi từ chỗ này sang chỗ khác trong cùng 1 ngôn ngữ sử dụng. Thư viện lập trình có nhiệm vụ xử lý các vấn đề quốc tế hóa có sẵn rất nhiều. Vấn đề quốc tế hóa và bản địa hóa phần mềm ngày càng trở nên phù hợp với điều kiện sự toàn cầu hóa còn tiếp diễn. Việc thiết kế một phần mềm hữu dụng với nhiều người của nhiều nên văn hóa sẽ mở ra nhiều thị trường, và cũng yêu cầu một số hệ thống nhất định.

5. Sự liên tưởng: Rất nhiều giao diện người dùng thiết kế dựa trên tính nhận diện với các đối tượng đã biết trước. Phần lớn các trình quản lý tệp tin và các hệ điều hành tích hợp giao diện đồ họa người dùng đều thiết kế dựa vào sự liên tưởng đến các loại máy tính để bàn. Phần lớn các chương trình xử lý văn bản đều cố gắng thể hiện giống tài liệu giấy. Một sự liên tưởng thích hợp có thể thúc đẩy việc học để sử dụng chương trình và chuyển tải các kỹ năng thực tế. Tuy nhiên, trong một số tình huống, hệ thống thực khác với những sự liên tưởng sẵn có và người dùng cần nhận thức được những sự khác biệt đó.

6. Phần mềm đa nền tảng: Kỹ sư công nghệ phần mềm sử dụng các nền tảng phần mềm khác nhau. Phần lớn người dùng rất gắn bó với nền tảng của họ và sẽ không có ý định chuyển đổi nền tảng chỉ để chạy một chương trình khác. Trong nhiều trường hợp, phần mềm mới được phát triển cần chạy được trên nhiều nền tảng khác nhau và phải tích hợp tốt với từng nền tảng.

7. Khả năng truy cập: Phần mềm nên có tính dễ dùng, dễ truy cập, nhằm có số lượng người sử dụng nhiều nhất có thể. Một số người có thị lực không tốt và khả năng phần biệt các tông màu hạn chế, và một số khác không thể sử dụng loại chuột hay bàn phím bình thường.

8. Giao diện đa phương tiện: Đồ họa và văn bản không phải là cách duy nhất để cung cấp thông tin. Hiện nay người ta đã có thể sử dụng âm thanh trong một số trường hợp, hoặc phản hồi xúc giác để truyền đạt thông tin (cảm ứng). Thậm chí có cả thiết bị tạo mùi. Vậy, làm thế nào để phát huy tối đa lợi thế của các thiết bị đầu ra để làm cho một giao diện người dùng tốt hơn sẽ là thách thức lớn trong nhiều năm tới.