


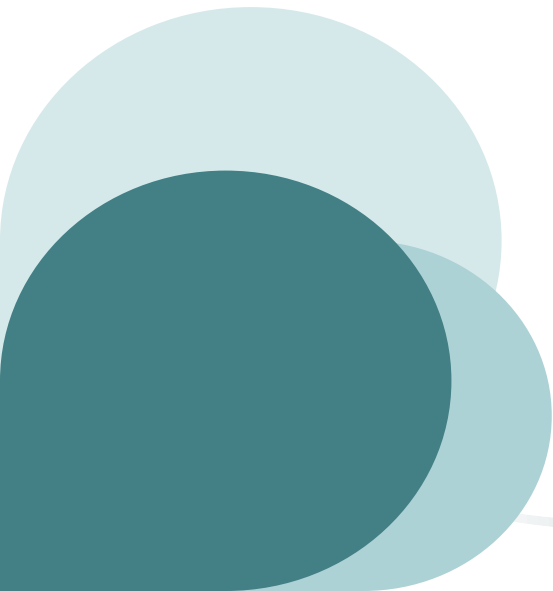


PHẦN II: QUY TRÌNH THIẾT KẾ HỆ TƯƠNG TÁC

- 
- I. Giới thiệu chung
 - II. Đặc tả yêu cầu và phân tích nhiệm vụ
 - III. Thiết kế tương tác người dùng máy tính
 - IV. Kiểm thử tính tiện dụng và đánh giá hệ thống
 - V. Quản lý hệ thống tương tác



CHƯƠNG I: GIỚI THIỆU CHUNG

- 
1. Thiết kế là gì ?
 2. Thế nào là một thiết kế tốt và một thiết kế tồi ?
 3. Các nguyên tắc cho tính dùng được (usability principles)
 4. Quy trình thiết kế phần mềm
 5. Thiết kế tương tác là gì ?
 6. Quy trình thiết kế hệ tương tác
 7. Các mô thức thiết kế (paradigms)



1. Thiết kế là gì

- “Design is achieving goals with constraints”
- Goals:
 - Mục đích thiết kế để tạo ra sản phẩm
 - Sản phẩm để cho đối tượng nào
 - Tại sao họ lại muốn sử dụng nó
- Constraints
 - Phải sử dụng các thiết bị gì ?
 - Các chuẩn mà ta phải tuân theo là gì ?
 - Giá thành
 - Thời gian phát triển
 - Ảnh hưởng đến sức khỏe, an toàn
- Trade-off
 - Lựa chọn giữa mục đích và các ràng buộc để làm tốt nhất nhiệm vụ đặt ra



2. Thiết kế tồi có mặt ở khắp mọi nơi

- Nghĩ rằng thiết kế tồi chỉ xuất hiện trong giao tiếp với máy tính là không đúng. Hãy nghĩ lại !
- Hầu hết các vấn đề về tính hữu dụng đều chỉ bật ra khi bạn đi du lịch đến một nước lạ.
- Hãy tưởng tượng bạn đi du lịch đến một đất nước và bạn gặp phải những điều sau:

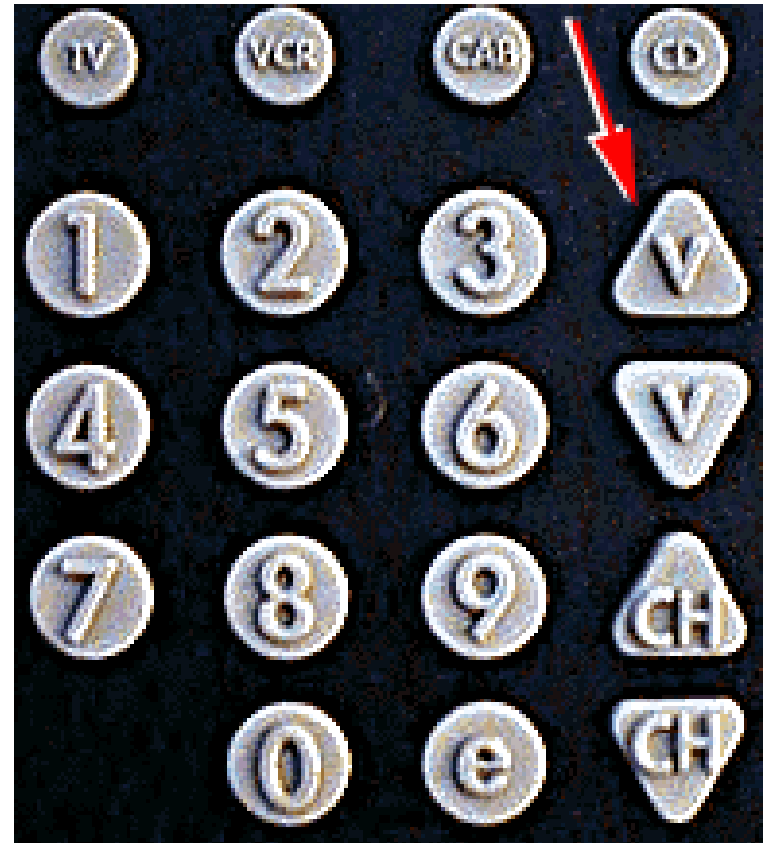
Trong một phòng nghỉ ở Hà Lan

- Thiết kế tồi ở đâu ?
- Đề xuất cải thiện ?



Giảm âm thanh TV của bạn

- Thiết kế tồi ở đâu ?
-
- Đề xuất cải thiện ?



Robinet ? Vặn theo chiều nào đây ?

- Thiết kế tồi ở đâu ?
- Đề xuất cải thiện ?



Ví dụ 1: Thư thoại (voice mail)

- Bạn có một chuyến công tác và phải nghỉ trong khách sạn. Tuy nhiên bạn vừa phát hiện ra mình đã để quên điện thoại di động ở nhà và phải dùng điện thoại của khách sạn để thực hiện các trao đổi công việc. Khách sạn có một hệ thống thư thoại trong mỗi phòng.
- Để xem bạn có tin nhắn, bạn nhấn vào một phím và nghe tiếng chuông. Nếu có 3 tiếng “Beep, Beep, Beep” nghĩa là có tin nhắn.



Ví dụ 1: Thư thoại (voice mail)

- Để nghe tin nhắn, bạn phải tuân theo các chỉ dẫn sau:
 - “1. Nhấn nút 491”.
Bạn làm như vậy và hệ thống trả lời:
“Chào mừng bạn đến với trung tâm tin nhắn của khách sạn Sunrise. Vui lòng nhấn số phòng mà bạn muốn để lại tin nhắn.”
Bạn nhấn số phòng và chờ nghe làm thế nào để thu lại tin nhắn nhưng không có thêm chỉ dẫn nào nữa.
Bạn nhìn vào hướng dẫn và thấy:
 - “2. Bấm số phòng của bạn, tiếp theo bấm dấu #”.
Bạn làm như vậy và hệ thống trả lời:
“Bạn đã vào hòm thoại của phòng 106. Để nhắn tin, vui lòng gõ vào password”



Ví dụ 1: Thư thoại (voice mail)

- Bạn không biết password là gì, bạn nghĩ đó là số phòng của bạn, nhưng khi bạn bấm thì không có tín hiệu gì!!!
Bạn muốn từ bỏ việc kiểm tra tin nhắn và gọi lễ tân.
- Lễ tân giải thích về cách thức thu và nghe tin nhắn:
“Quý khách phải nhập vào đúng thời điểm số phòng và số mở rộng của điện thoại. Tiếp theo đó, quý khách phải làm 6 thao tác để truy nhập vào hộp thoại và 5 thao tác để nhắn thoại”.
- Quá bức, bạn đi ra và mua một chiếc điện thoại mới



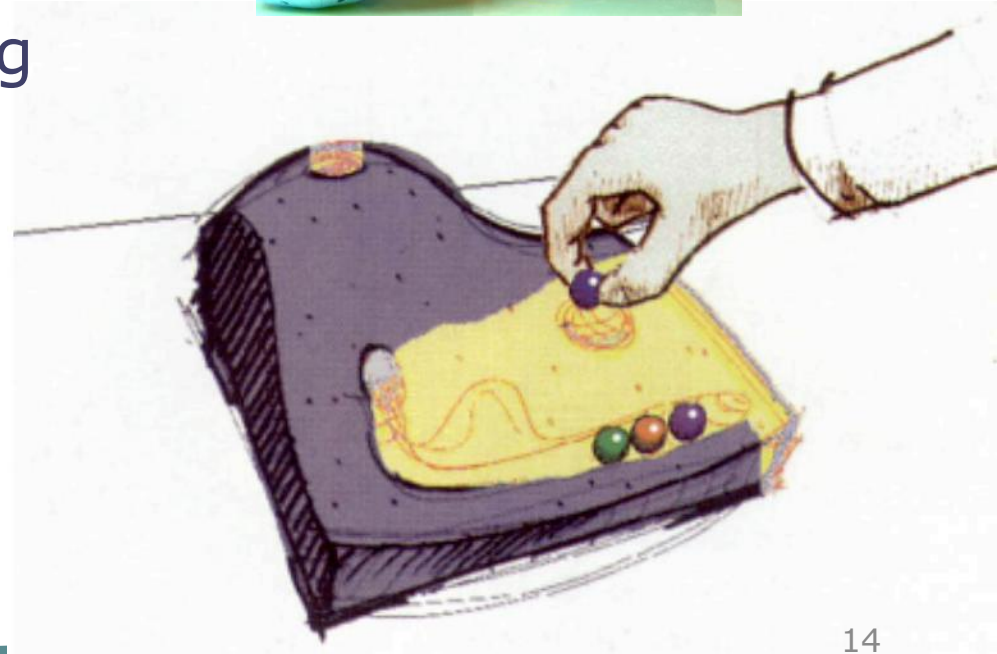
Đâu là vấn đề của hệ thống thư thoại?



Hãy kể một số ví dụ về thiết kế tồi

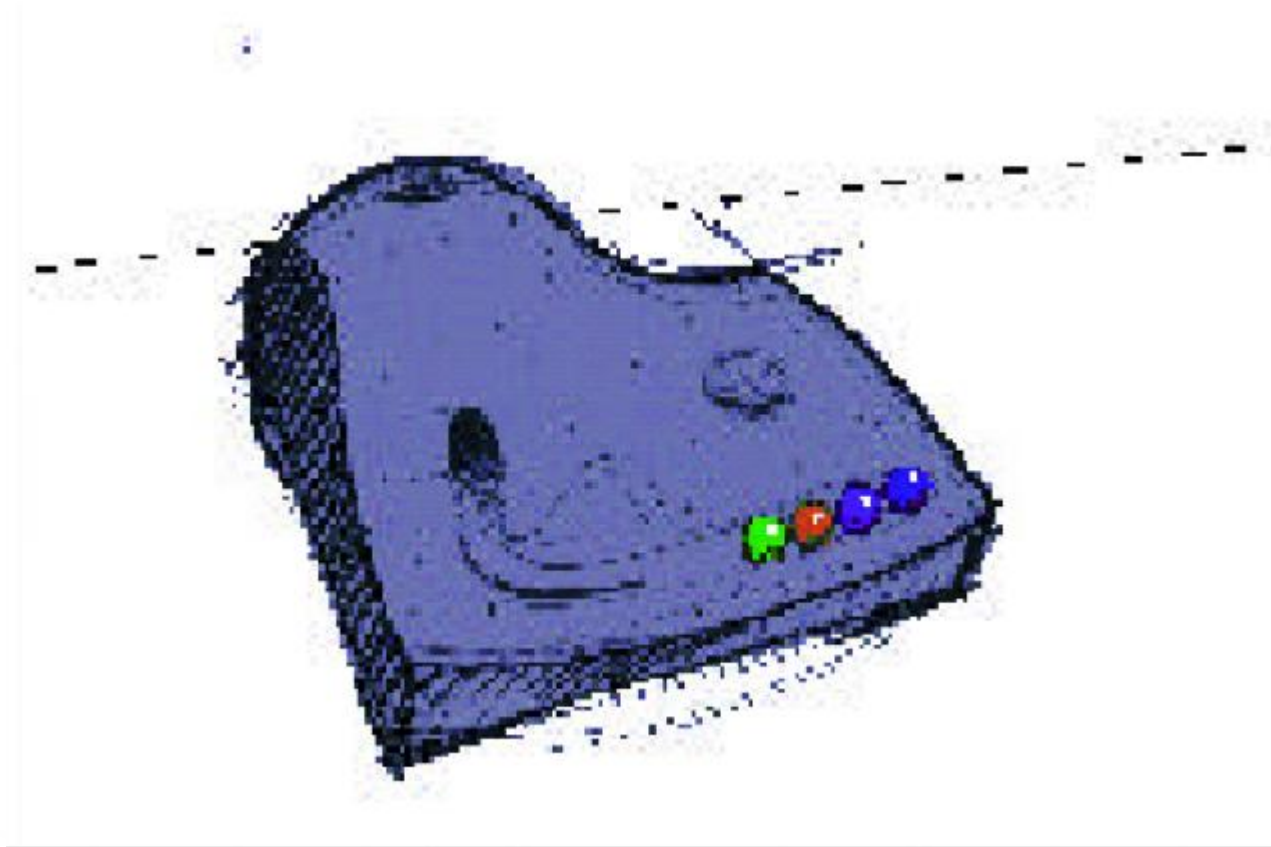
Ví dụ 2: Marble answering machine

- Do Durrell Bipshop – cựu sinh viên cao học trường Royal College of Art in London – thiết kế
- Là một ví dụ đầu tiên về thiết kế tương tác trong đó nó có sự kết nối hai thế giới vật lý và thế giới SỐ



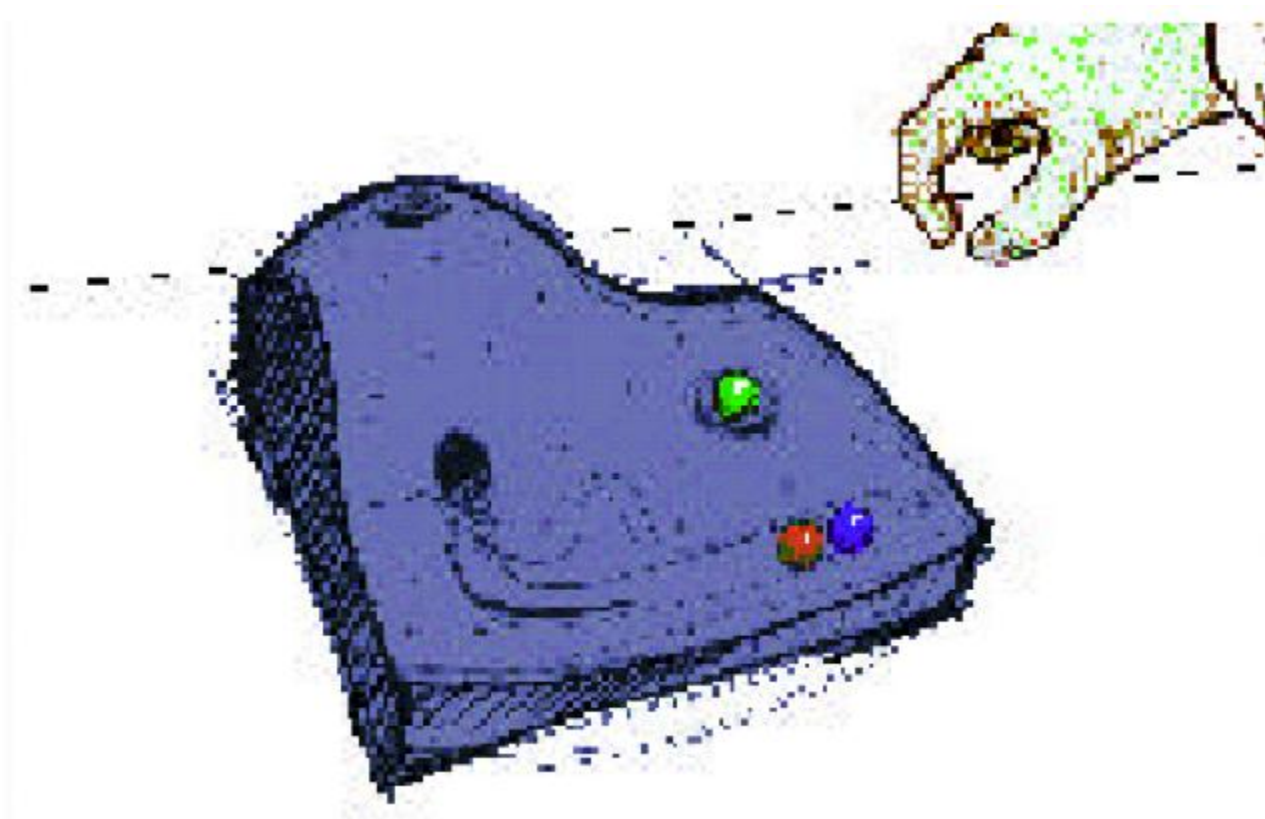
Ví dụ 2: Marble answering machine

- Các tin nhắn được lưu trữ



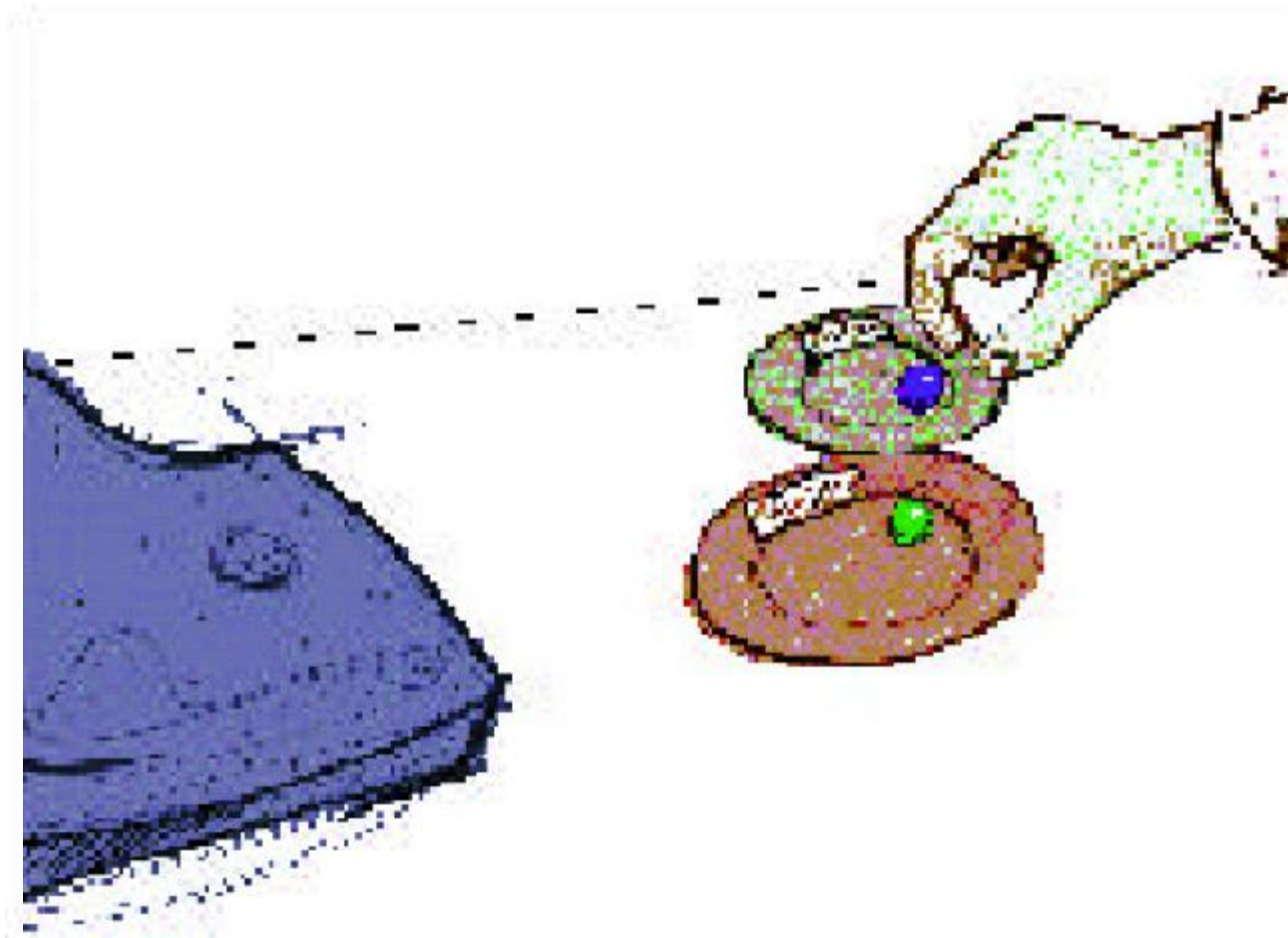
Ví dụ 2: Marble answering machine

- Người dùng có thể tùy ý lựa chọn tin nhắn để nghe theo thứ tự bất kỳ



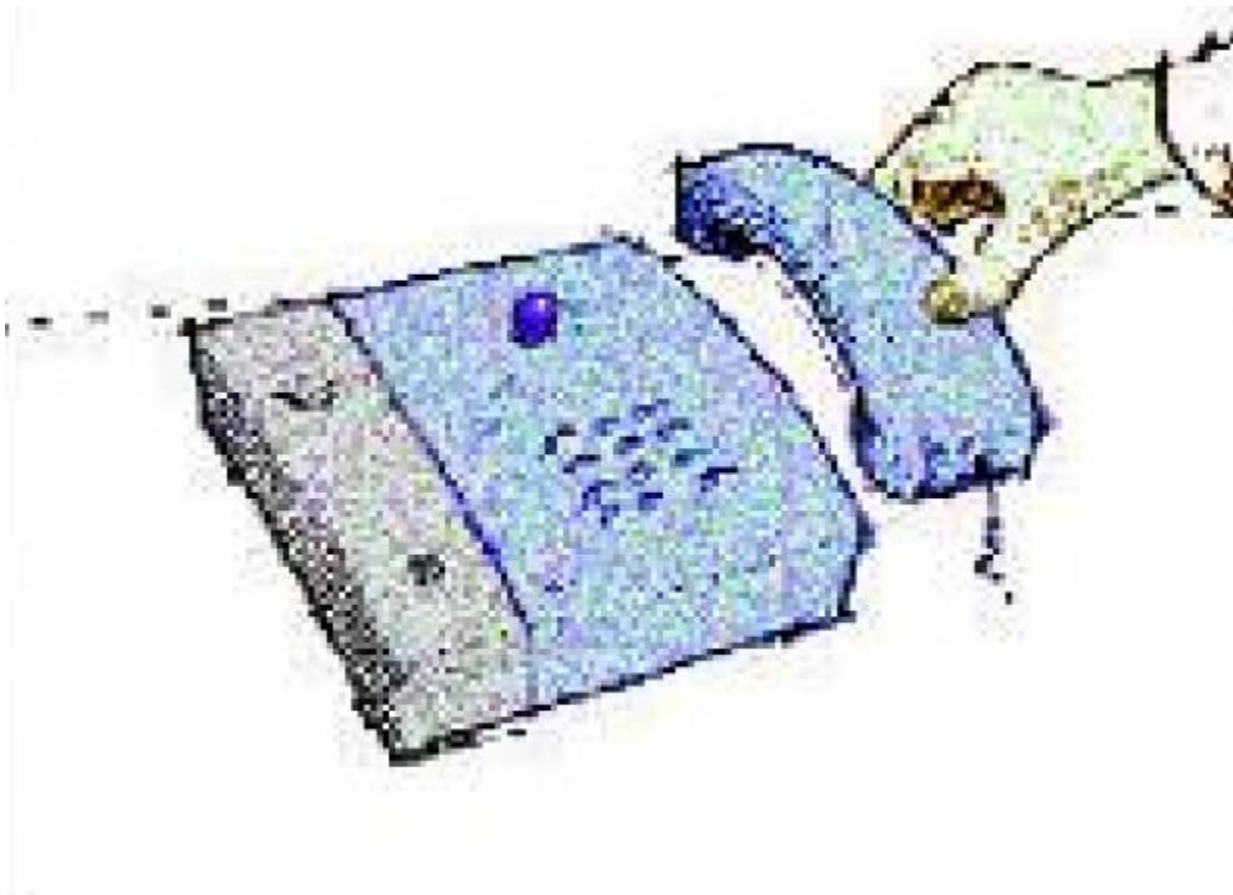
Ví dụ 2: Marble answering machine

- Người dùng phân loại các tin nhắn gửi tới họ.



Ví dụ 2: Marble answering machine

- Người dùng gọi lại người để lại lời nhắn





Marble answering machine có gì tốt ?



3. Tính dùng được

**Một thiết kế tốt cần đảm bảo
tính dùng được**



Nguyên tắc của tính dùng được

- [Nielson-1993]
 - Tính hiệu quả
 - Tính dễ học
 - Tính dễ nhớ
 - Tính dự đoán lỗi
 - Đáp ứng tính chủ quan
- 3 tiêu chí chính:
 - Tính dễ học (Learnability)
 - Tính mềm dẻo (Flexibility)
 - Tính vững chắc (Robustness)



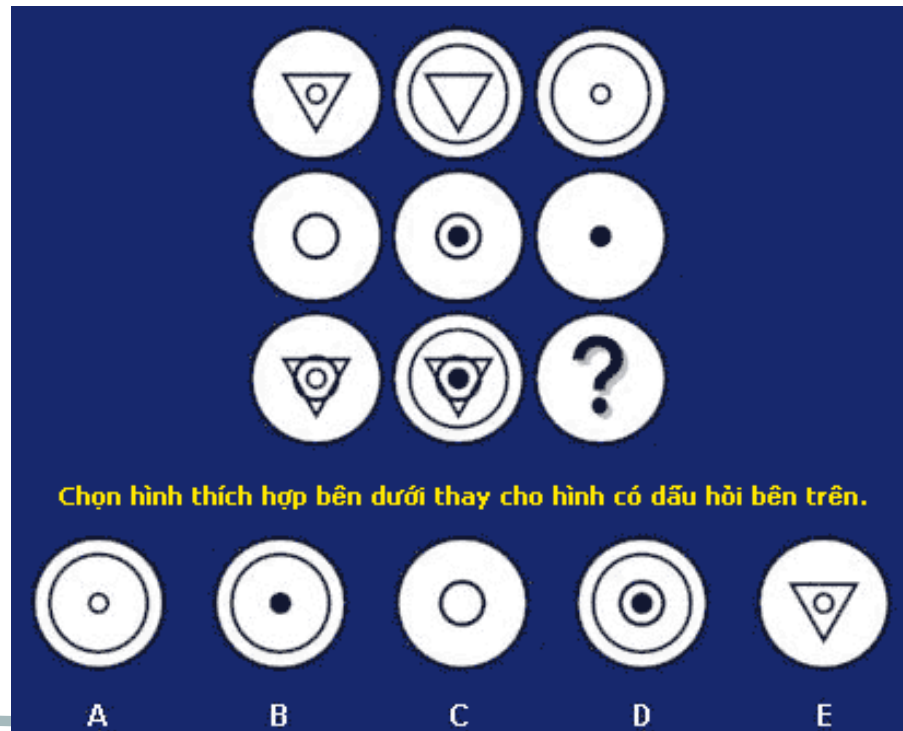
3.1. Dễ học

- Cho phép người dùng mới thiếu kinh nghiệm
 - Có thể sử dụng
 - Và sử dụng đạt hiệu quả tối đa
- Thể hiện ở các tính chất sau
 - Tính dự đoán
 - Tính tổng hợp
 - Tính khái quát
 - Tính nhất quán

Dự đoán

Dễ học

- Dựa vào tri thức đã biết để xác định các tương tác mới và dự báo kết quả
- Ví dụ 1:
- Ví dụ 2: trong thiết kế



Dễ học

- **Tính tổng hợp**

- Tính dự đoán chỉ tập trung vào khả năng **xác định** các động tác **tiếp theo** dựa vào cái đã qua
- Tính tổng hợp cho **phép hình thành mô hình về hành vi** của hệ thống từ một chuỗi các tương tác trước

- **Tính thân thiện**

- Được đo bởi **sự tương hỗ** giữa tri thức đang tồn tại và tri thức cần có để thực hiện tương tác có hiệu quả
- Ví dụ: trong công nghệ soạn thảo văn bản, có sự tương tự giữa bàn phím của máy chữ và máy tính





Dễ học

- Tính khái quát
 - Tính khái quát là tính chất cho phép người dùng mở rộng tri thức từ một ứng dụng cụ thể sang một tình huống tương tự để dự báo mô hình tương tác hoàn thiện hơn
 - Ví dụ: Trong một gói đồ họa: hình vuông là hình chữ nhật có ràng buộc; trong hệ thống window, thao tác cắt-dán áp dụng cho mọi đối tượng
- Tính nhất quán:
 - hành vi tương tự trong các tình huống tương tự



3.2. Mềm dẻo

- Đối thoại chủ động
- Đa luồng
- Di trú nhiệm vụ
- Thay thế
- Cá nhân hóa



Mềm dẻo

- Đối thoại chủ động:
 - Kiểu đối thoại ưu tiên hệ thống
 - Hệ thống chủ động mọi đối thoại
 - Người dùng đơn thuần đáp ứng các yêu cầu
 - Ví dụ: các hộp thoại ngăn cấm ND đưa thông tin trực tiếp vào
 - Kiểu đối thoại ưu tiên người dùng
 - ND hoàn toàn chủ động đối thoại
 - Tuy nhiên việc chủ động sẽ làm mất vết của các nhiệm vụ mà người dùng chủ động đề nghị mà chưa được hoàn thành

Mềm dẻo

- **Đa luồng**

- **Một luồng** là một phần của đối thoại có liên quan đến một nhiệm vụ nào đó
- **Đa luồng** cho phép hỗ trợ nhiều hơn một nhiệm vụ tại một thời điểm
- Có **hai đặc trưng** của đa luồng
 - Nhiều kênh có thể cho phép thực hiện một nhiệm vụ
 - Mở cửa sổ = nhấn kép chuột trên một biểu tượng; dùng phím nóng; nói « hãy mở cửa sổ »
 - Một biểu diễn đơn có thể tạo bằng cách trộn các kênh
 - Hệ thống cảnh báo = cửa sổ cảnh báo + sound (bipbip)



Mềm dẻo

- Di trú nhiệm vụ
 - Chuyển điều khiển thực hiện nhiệm vụ giữa người dùng và hệ thống
 - Ví dụ:
 - Kiểm tra lỗi chính tả của văn bản
 - Điều khiển bay trong buồng lái
- Thay thế
 - Tập các giá trị tương đương có thể thay thế cho nhau tùy theo ứng dụng
 - Ví dụ: trong lựa chọn về kích thước: cm, inch, point



Mềm dẻo

- Cá nhân hóa
 - Tính thay đổi tự động mà hệ thống muốn thực hiện dựa vào tri thức của nó về người dùng
 - Ví dụ: người dùng có thể thay đổi ảnh nền, tên, phím nóng mà mình thích / hay sử dụng



3.3. Vững chắc

- Tính quan sát (Observability)
- Tính khôi phục (Recoverability)
- Tính đáp ứng (Responsiveness)
- Tính tương hợp nhiệm vụ (Task conformance)



Vững chắc

- Tính quan sát (Observability)
 - Cho phép người dùng đánh giá được trạng thái bên trong của hệ thống nhờ biểu diễn cảm nhận được trên giao tiếp
- Tính khôi phục (Recoverability)
 - Người dùng gây lỗi và muốn sửa lỗi
 - Tính khôi phục là khả năng đạt tới đích mong muốn sau khi nhận ra một số lỗi trong các tương tác trước
 - Khôi phục kiểu tiến: Chấp nhận trạng thái hiện tại, đàm phán từ trạng thái đó tiến về trạng thái đích
 - Khôi phục kiểu lùi: thử bỏ đi ảnh hưởng của các tương tác trước và



Vững chắc

- Tính đáp ứng
 - Đo vận tốc giao tiếp giữa người dùng và máy tính
 - Thời gian đáp ứng là thời gian cần thiết để thay đổi trạng thái
 - Mong muốn: đáp ứng tức thì, tuy nhiên hệ thống phải có thông báo đã nhận được yêu cầu và đang xử lý
- Tính tương hợp nhiệm vụ
 - Hệ thống khi thiết kế đảm bảo đáp ứng đầy đủ các yêu cầu trong đặc tả
 - Tuy nhiên hệ thống nên có khả năng cho phép người dùng định nghĩa các nhiệm vụ mới



4. Các mô thức của tính dùng được

1. Mô thức phân chia thời gian (Time Sharing)
2. Các thiết bị hiển thị quan sát
3. Các công cụ lập trình
4. Máy tính cá nhân
5. Hệ thống Windows và giao diện WIMP
6. Cảnh trí (metaphor)
7. Điều khiển trực tiếp
8. Ngôn ngữ ngược với hành động
9. Siêu văn bản
10. Đa thể thức
11. WEB
12. Giao tiếp dựa vào tác tử

Tại sao cần nghiên cứu các mô thức ?

- Liên quan đến việc thiết kế các hệ thống tương tác
 - Các hệ thống phải được thiết kế thế nào để đảm bảo tính dùng được
 - Tính dùng được được đánh giá như thế nào ? Có đo được hay không ?
- Các hệ tương tác đã được thiết kế thành công cho phép cung cấp mô thức cho việc phát triển các hệ tương tác trong tương lai
- Xây dựng các hệ tương tác “tiện dụng hơn” các hệ tương tác đã và đang tồn tại
- Thảo luận về các vấn đề liên quan nhằm cải tiến thiết kế các hệ tương tác

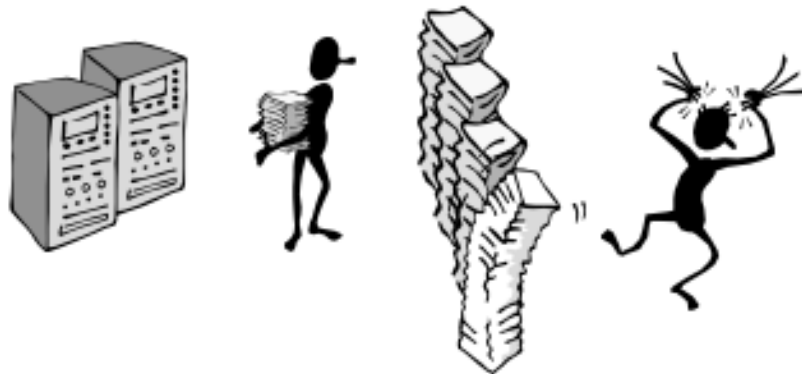


Về nhà

- Sinh viên về tự đọc sách “Tương tác người máy ”
 - Nhà xuất bản KHKT của thầy Lương Mạnh Bá
- Trong mỗi mô thức đánh giá
 - Ưu điểm
 - Nhược điểm
 - Sự tiến bộ của mô thức sau so với mô thức trước

4.1. Xử lý theo lô

- Giai đoạn: 1950s – 1960s
- Xử lý theo lô: tập lệnh được lưu trên bìa hay băng giấy sau để nạp vào máy tính
- Nhược điểm:
 - Tại một thời điểm, chỉ chạy một chương trình của một người sử dụng
 - Khó sử dụng, vướng, cồng kềnh, không dự báo được



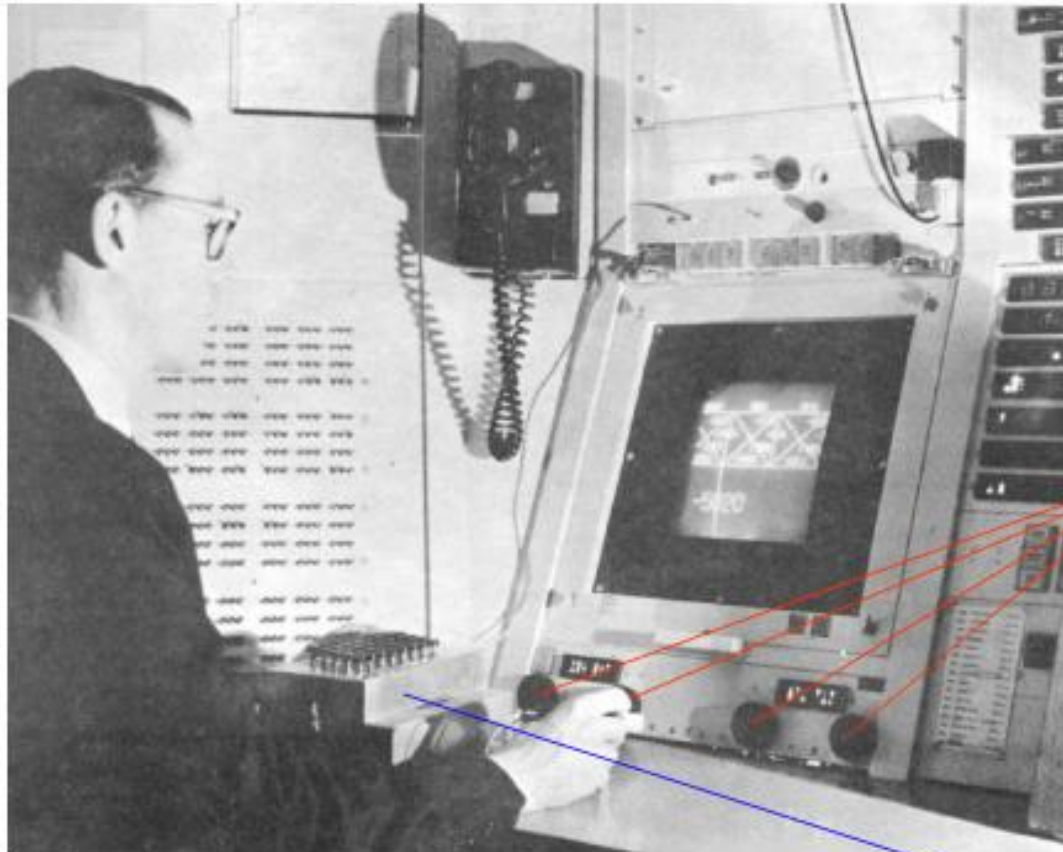
4.2. Phân chia thời gian (time sharing)

- Xuất hiện vào những năm 1960s
 - IBM 360, SDS 940, PDP-10
- Cho phép chia sẻ tài nguyên tính toán cho nhiều người sử dụng (Multi-users)
 - Cơ chế lập lịch cho phép các nhiệm vụ sẽ được thi hành
 - Lưu và chuyển đổi trạng thái người sử dụng và trạng thái chương trình của họ
 - Gán time slots cho mỗi công việc: Job1: 0-5; Job 2: 5-10; Job 3: 10-15
- Ưu điểm: so với xử lý theo lô (batch processing)
 - Gia tăng lượng thông tin giữa người dùng và máy tính
 - Cho phép người dùng tích cực hơn và có tương tác kịp thời

4.3. Thiết bị hiển thị quan sát (VDU)

- Giai đoạn: giữa những năm 1950s-
- Được ứng dụng đầu tiên trong quân sự
- Chương trình Sketchpad (Ivan Sutherland, 1962)
 - Lần đầu tiên cho phép biểu diễn thông tin dạng ảnh lên trên thiết bị
 - Cho phép người dùng sử dụng máy tính để tạo một cách nhanh chóng các mô hình trực quan, tinh vi trên màn hình hiển thị
- Ý nghĩa:
 - Máy tính được sử dụng để mở rộng khả năng của con người thông qua việc hiển thị và điều khiển các biểu diễn khác nhau của cùng một thông tin
 - Ý tưởng sáng tạo trong lịch sử phát triển máy tính

The first sketchpad [Ivan Sutherland, 1962]

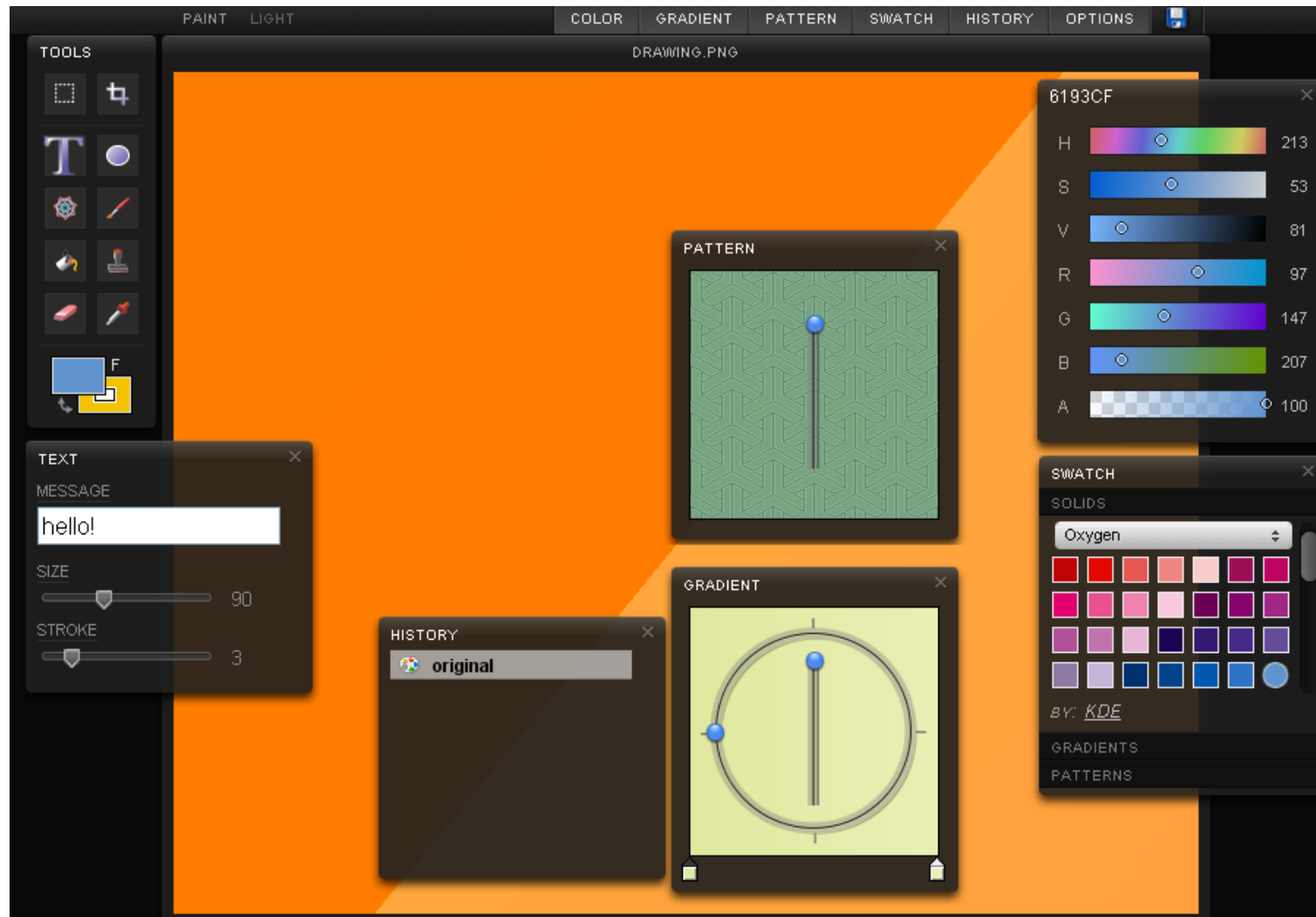


Sketchpad in Use

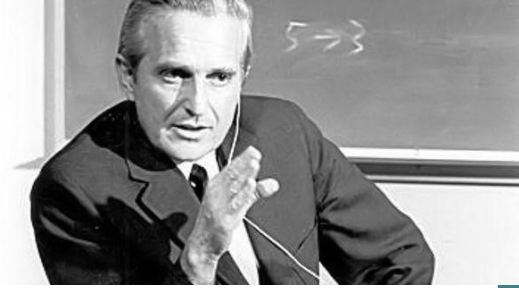
Control the size and position of the total picture seen on the display

Control the drawing function

<http://mugtug.com/sketchpad/>



4.4. Các bộ công cụ lập trì



- Douglas Engelbart's Augmenting Human Intellect, 1962
 - “The secret to producing computing equipment that aids human problem-solving ability is providing the right toolkit”
- Engelbart và các đồng sự đề xuất cần phát triển một bộ công cụ lập trình làm cơ sở cho việc phát triển các hệ thống phức tạp hơn
- Ưu điểm:
 - Các thành phần nhỏ khi được thiết kế tốt và được hiểu một cách thấu đáo sẽ cho phép tạo ra các công cụ lớn hơn
 - Một khi bộ công cụ lớn hơn được hiểu rõ, nó sẽ cho phép tạo ra các bộ công cụ lớn hơn nữa và cứ thế tiếp tục

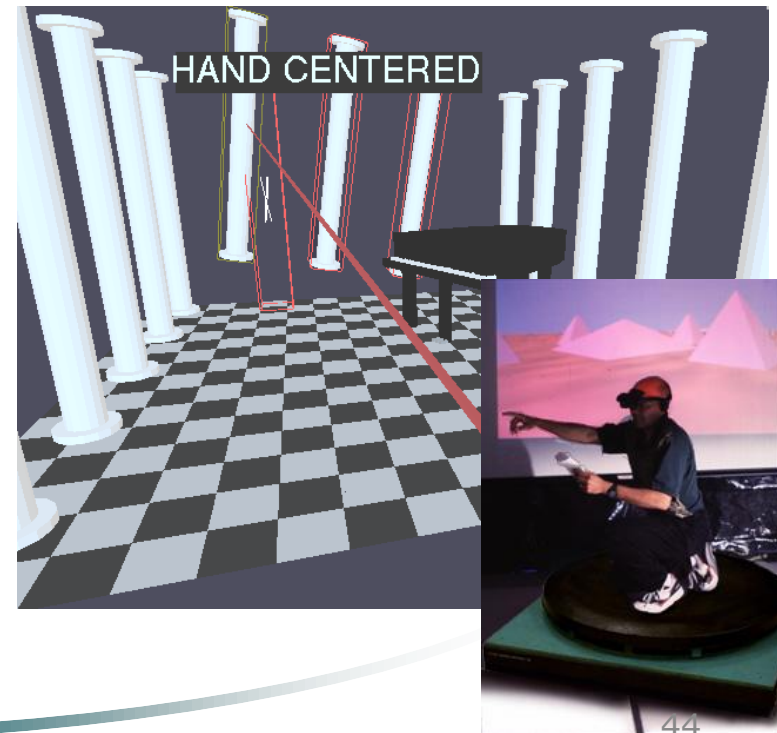
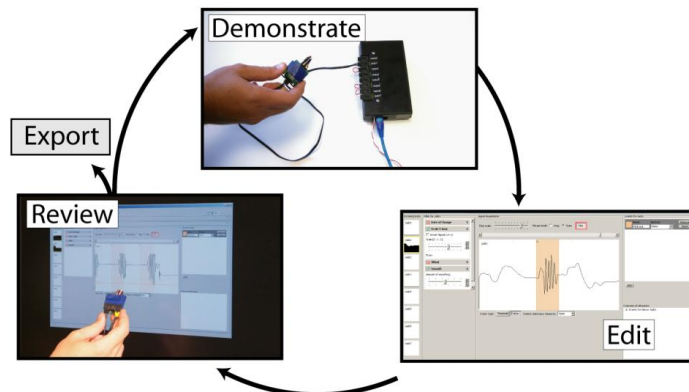


4.5. Mô thức điều khiển trực tiếp

- Nguyên lý:
 - Tính nhìn thấy được của đối tượng quan tâm:
 - Gia tăng hoạt động với phản hồi nhanh chóng của mọi hành động
 - Khuyến khích người dùng khám phá mà không phải các hậu quả nặng nề
 - Mọi hành động có cú pháp chính xác
 - Thay thế ngôn ngữ dòng lệnh bởi việc điều khiển trực tiếp các đối tượng nhìn thấy được

Ví dụ

- Giao diện: kéo thả một file vào thùng rác thay vì sử dụng lệnh rm ? (Vì sao lại tốt hơn)
- Một số lĩnh vực ứng dụng của điều khiển trực tiếp
 - Trò chơi
 - Hiển thị, Mô phỏng
 - Hiện thực ảo (Cử chỉ, găng tay)
 - v.v





4.6. Giao tiếp dựa vào tác tử

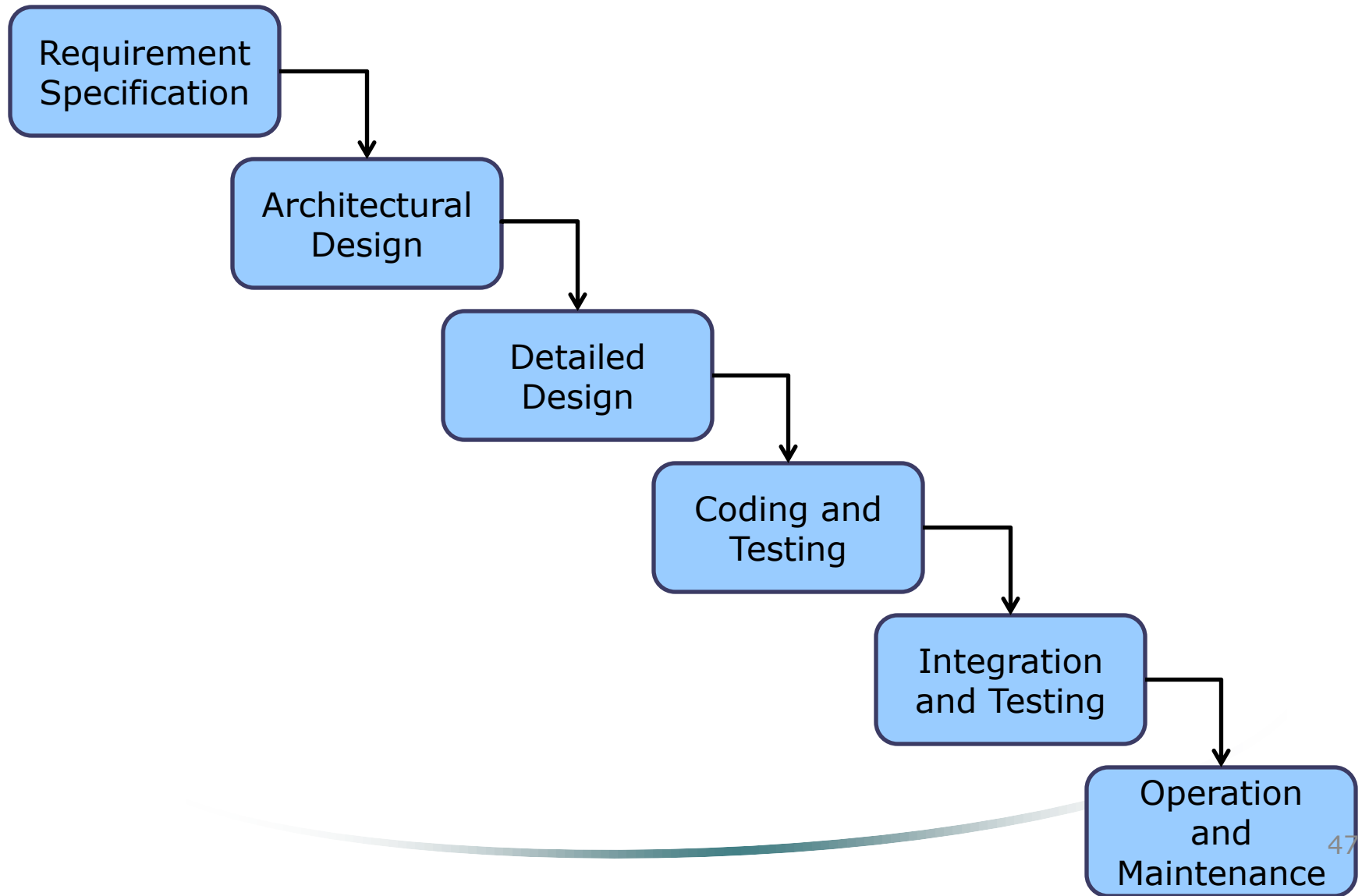
- Ví dụ
 - Tác tử thư tín làm nhiệm vụ lọc thư
 - Tác tử web dò tìm các trang web
 - Excel: tác tử tính tổng của các số
- Ưu điểm
 - Có thể thay thế các hoạt động của người dùng để thực hiện một số nhiệm vụ



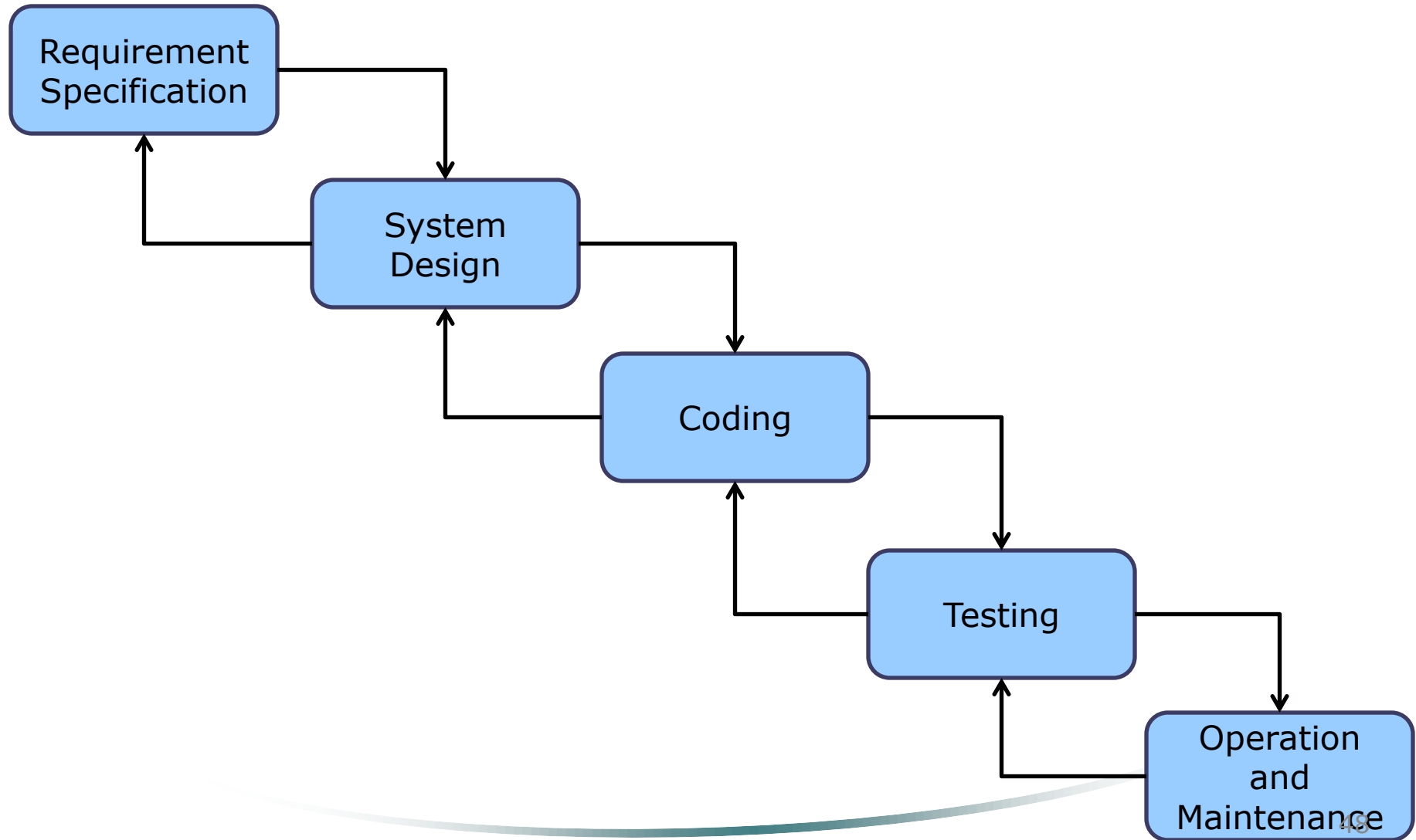
5. Quy trình thiết kế phần mềm

- Các mô hình vòng đời của phần mềm
 - Mô hình thác nước
 - Mô hình vòng đời phần mềm của Boehm
 - Mô hình vòng đời hình sao

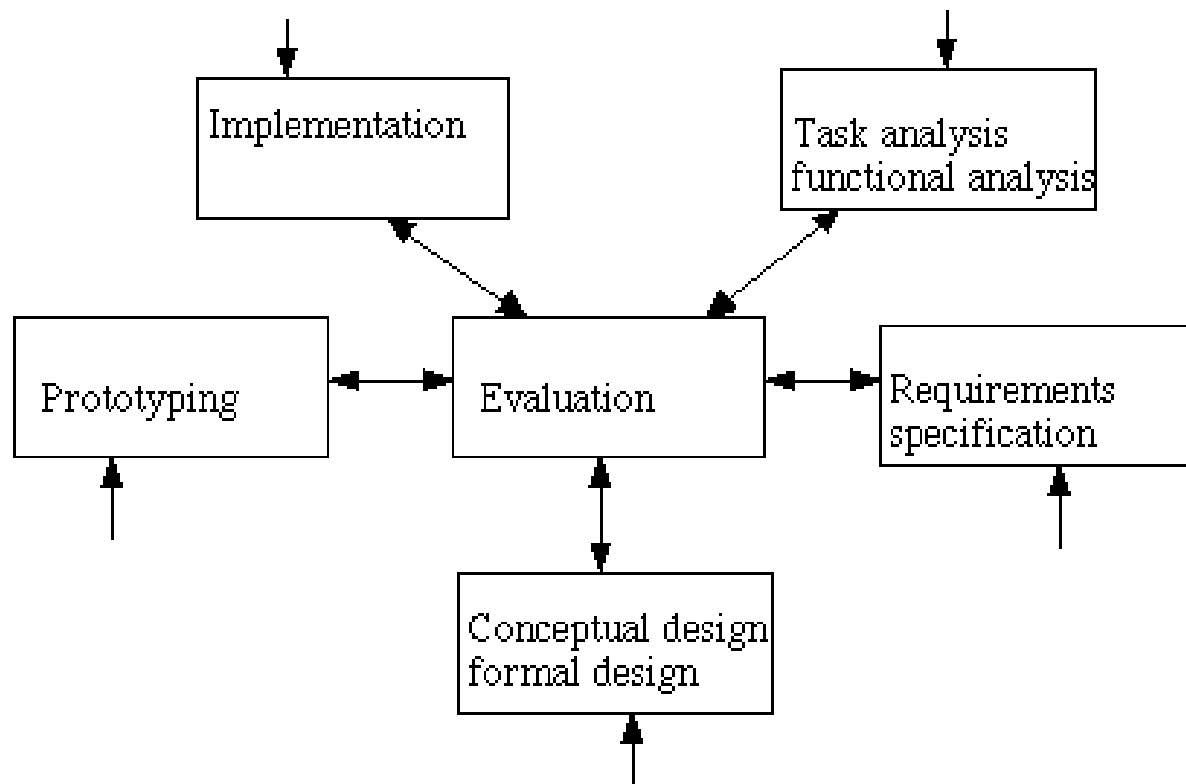
5.1. Mô hình thác nước



5.2. Mô hình Boehm

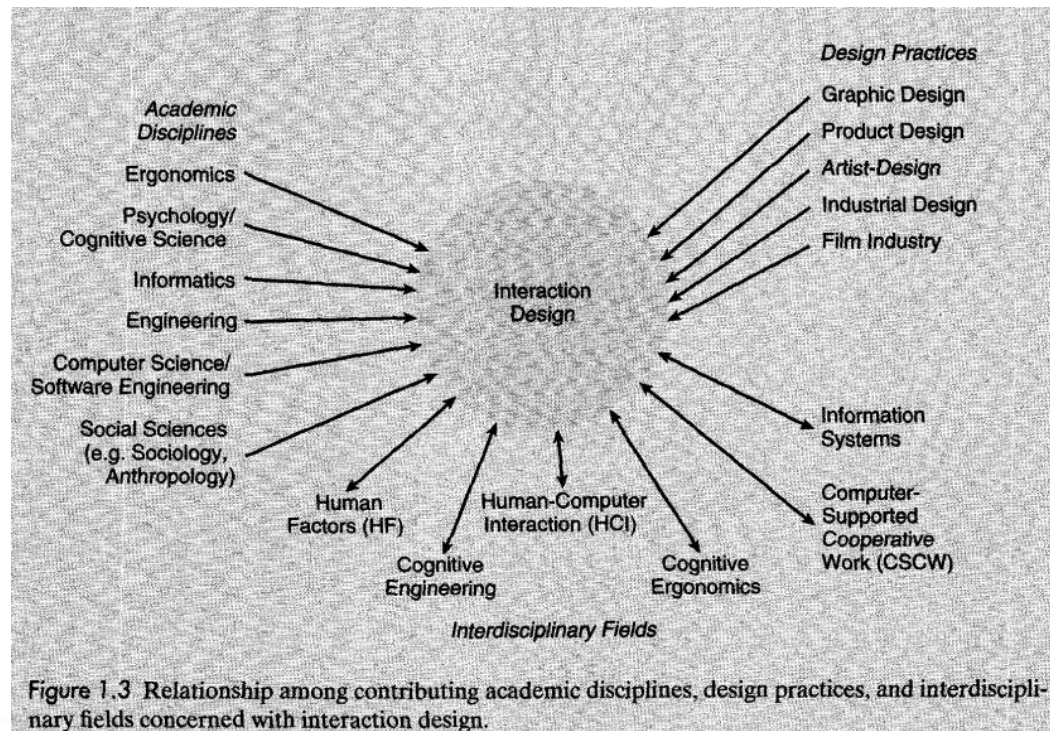


5.3. Mô hình hình sao

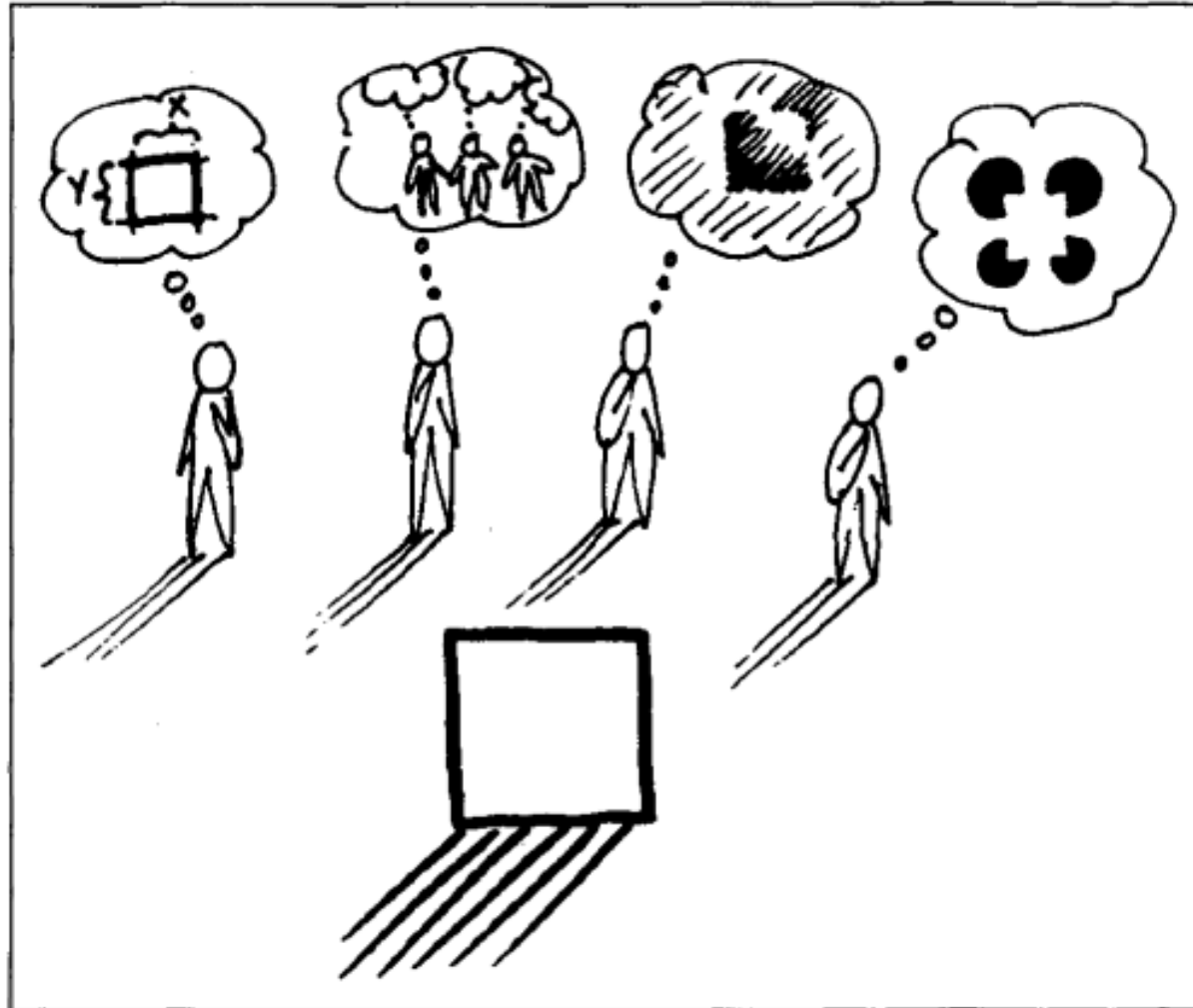


6. Thiết kế tương tác

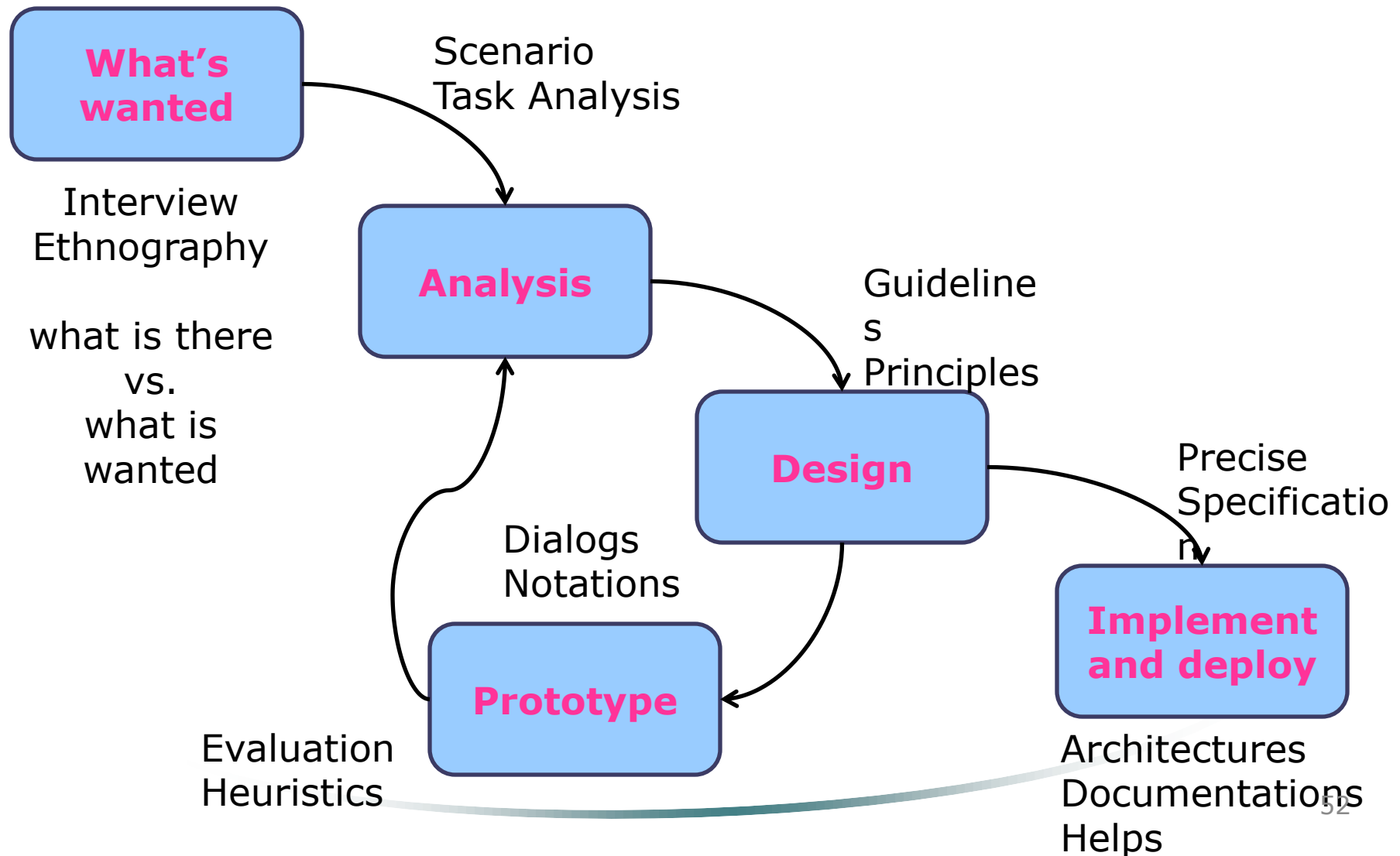
- Định nghĩa: Designing interactive products to support people in their everyday and working lives.



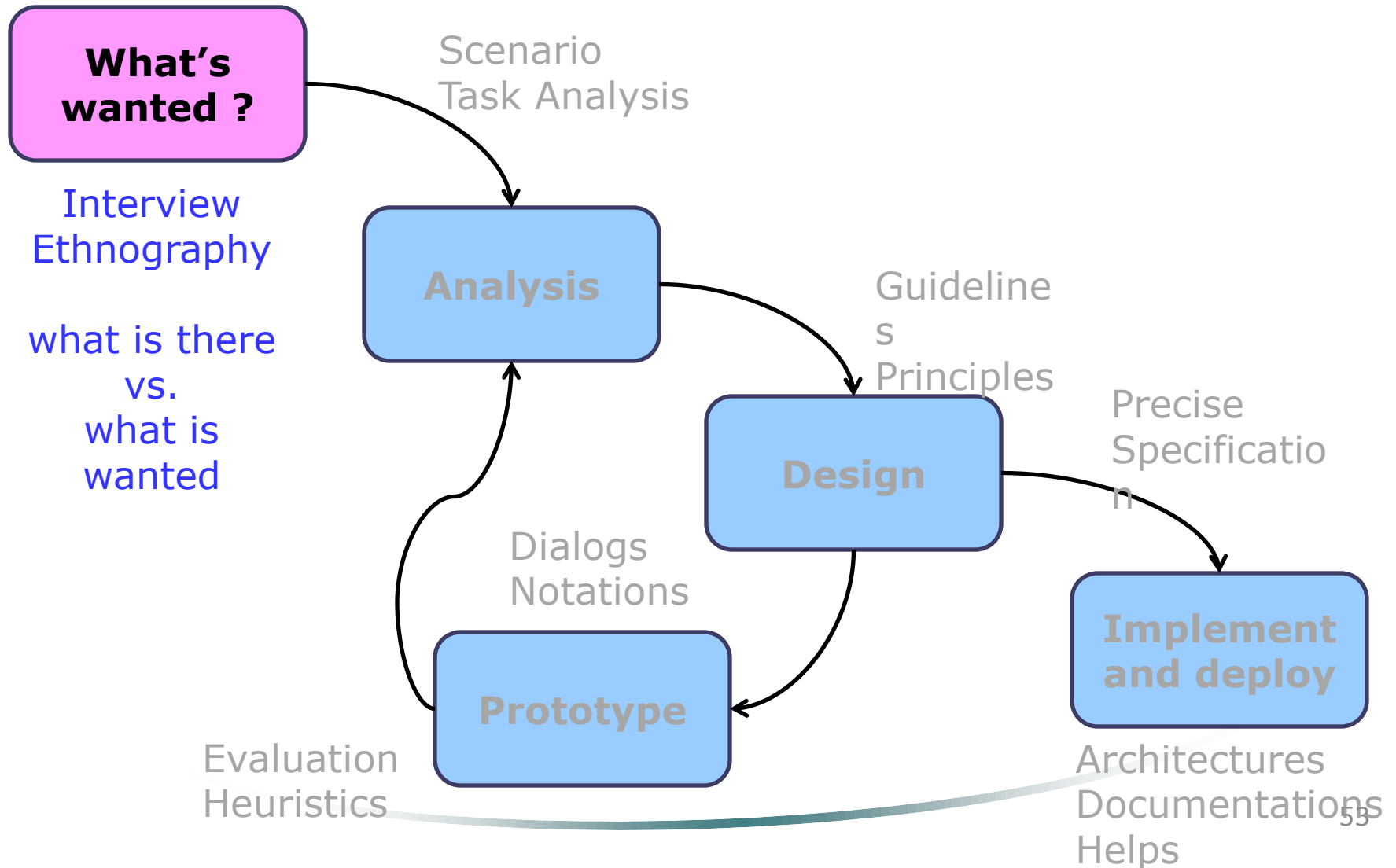
Thiết kế tương tác để cùng làm việc



7. Quy trình thiết kế hệ tương tác



7.1. Người dùng muốn gì

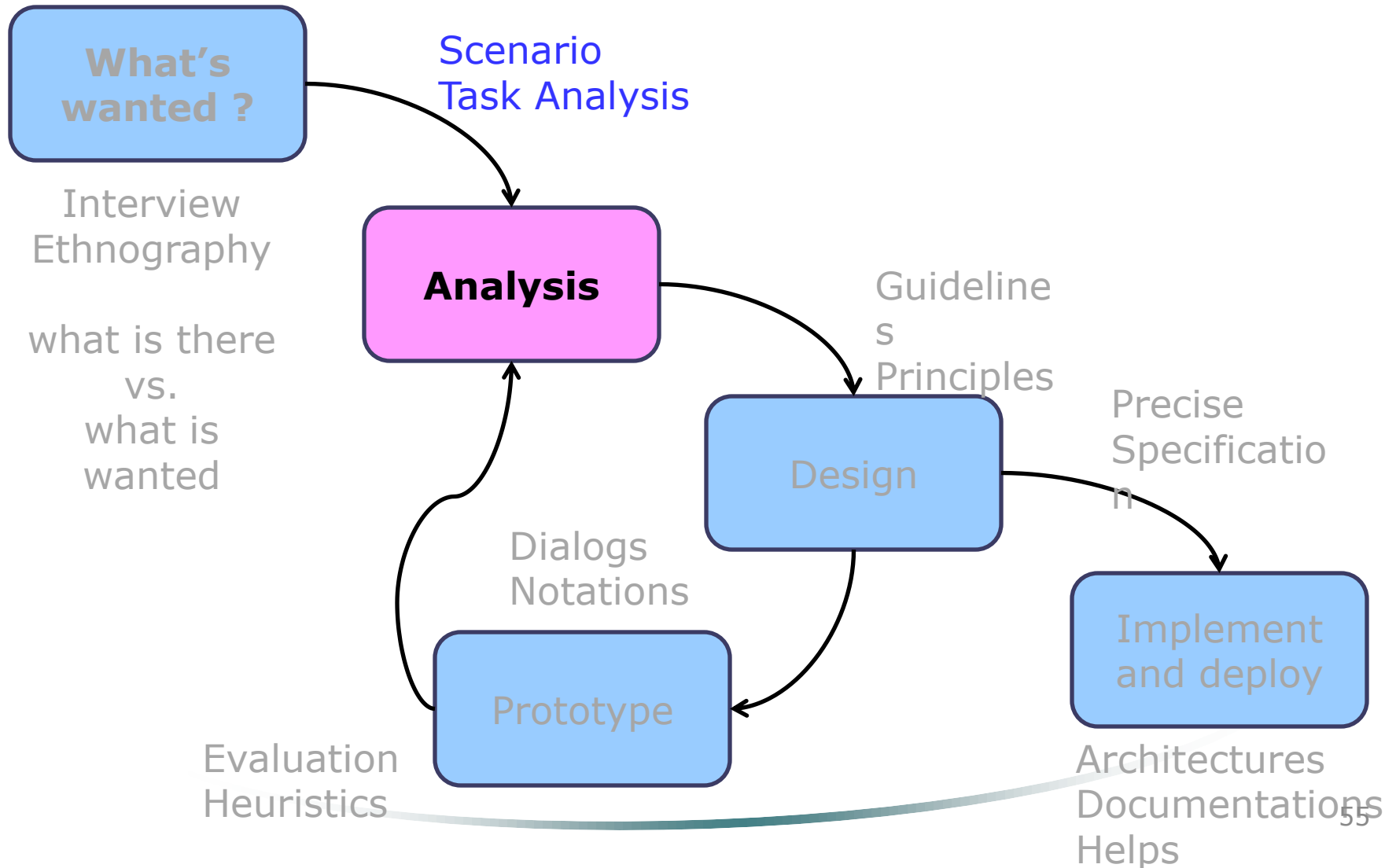




7.1. Người dùng muốn gì

- Requirements – What's wanted ?
- Các phương pháp thực hiện
 - Phỏng vấn
 - Videotaping
 - Tìm kiếm và tra cứu tài liệu về vấn đề liên quan
 - Quan sát trực tiếp

7.2. Phân tích

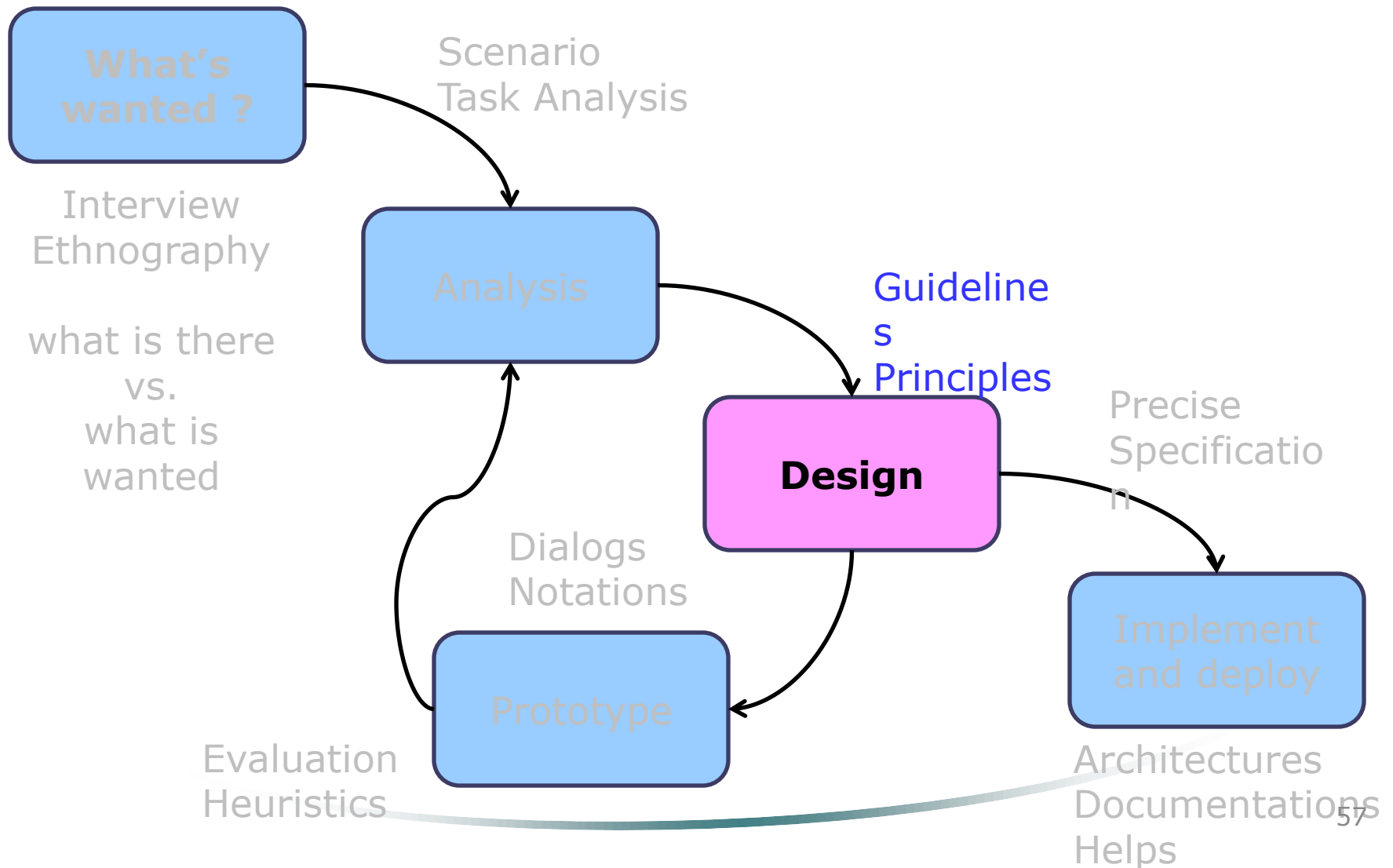




7.2. Phân tích

- Phân tích: Các kết quả thu nhận được từ pha xác định nhu cầu sẽ được sắp xếp theo cách thức nào đó để đưa ra các vấn đề chính và trao đổi với các khâu sau của quá trình thiết kế
- Các phương pháp:
 - Xây dựng kịch bản
 - Phân tích tác nhiệm

7.3. Thiết kế

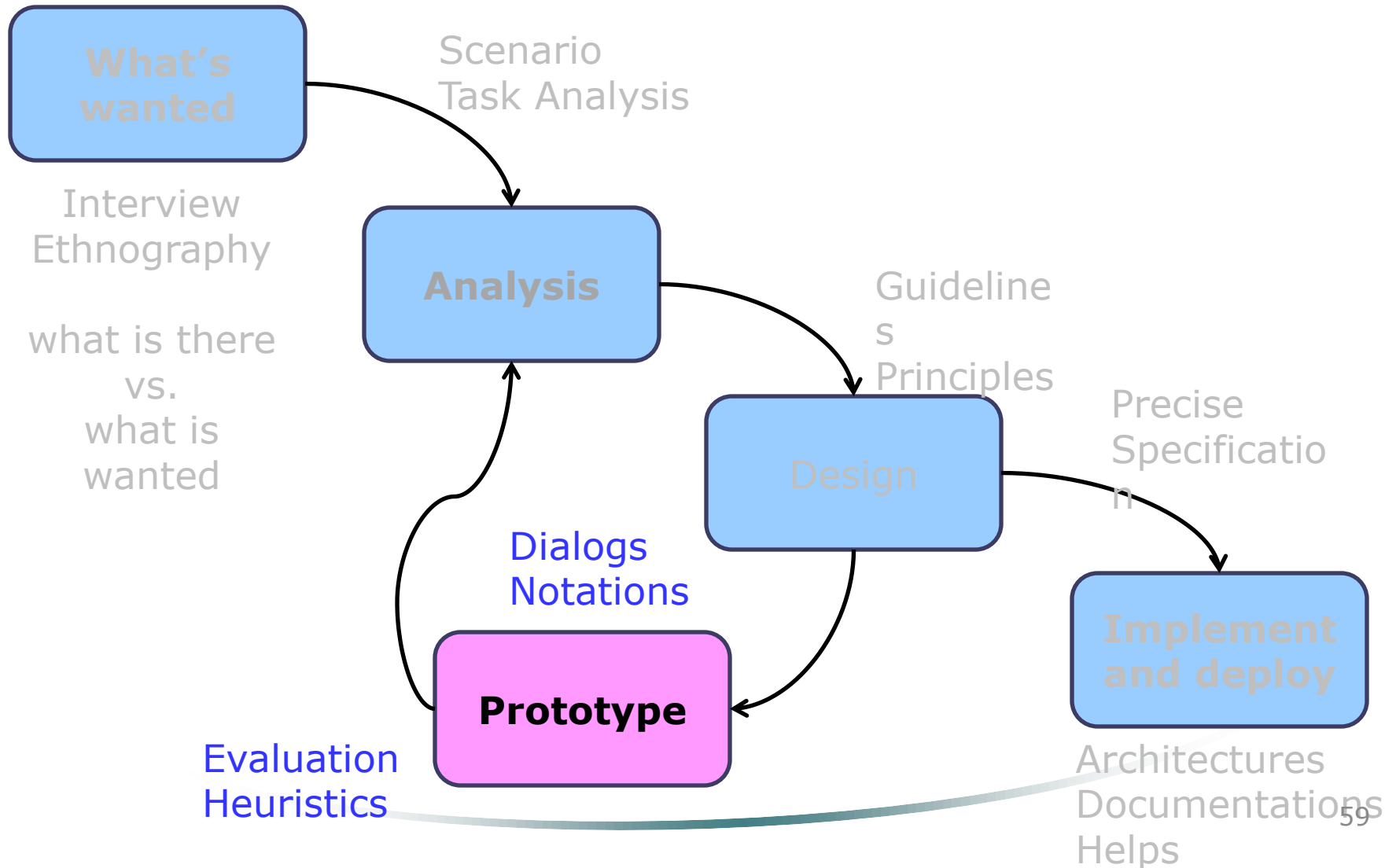




7.3. Thiết kế

- Thiết kế:
 - Mặc dù tất cả quy trình là thiết kế
 - Tuy nhiên: đây là khâu trọng yếu của quá trình
- Các phương pháp thiết kế dựa trên:
 - Luật tương tác
 - Nguyên lý thiết kế
 - Guidelines

7.4. Tạo mẫu thử

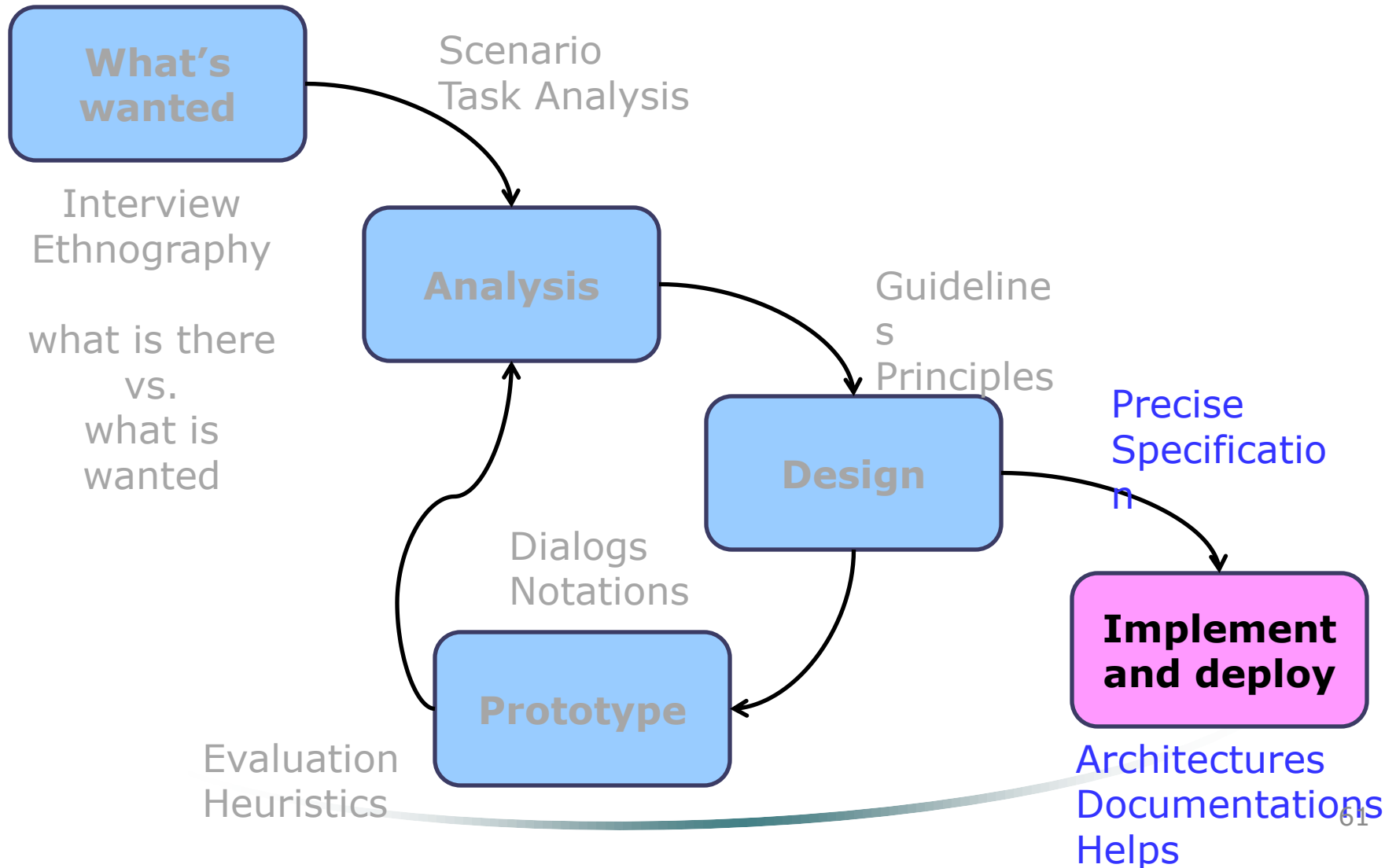




7.4. Tạo mẫu thử

- Vòng lặp và thiết kế mẫu thử:
 - Con người là phức tạp
 - Chúng ta không chờ đợi có thể có một thiết kế hoàn hảo ngay lần đầu tiên
 - Vì thế cần phải đánh giá để xem sản phẩm mẫu đạt được như thế nào và chỗ nào có thể cải thiện được
- Các phương pháp dựa trên:
 - Kỹ thuật đánh giá
 - Thu nhận thông tin phản hồi từ người dùng thử

7.5. Cài đặt và triển khai





7.5. Cài đặt và triển khai

- Cài đặt và khai thác:
 - Sau khi đã hài lòng với việc thiết kế chúng ta đi vào cài đặt và triển khai sản phẩm
- Các công việc cần thực hiện
 - Writing Codes
 - Making hardware
 - Writing documents, manuals



Tổng kết

- SV nắm bắt được
 - Thiết kế là gì
 - Thiết kế tốt/tồi
 - Các nguyên lý của tính dùng được
 - Các mô thức tương tác
 - Quy trình phát triển phần mềm
 - Thiết kế tương tác