

01

▼ HCI?

- Interaction + Interface
 - Interface
 - 유저와 컴퓨터 시스템이 상호작용하는 특정한 경로
 - Interaction
 - 유저가 특정한 작업을 인터페이스를 통해 성취하게 하는 작업 모델
- HCI?
 - 상호작용을 디자인하고, 인터페이스를 구현하는 작업
 - 컨셉은 간단하지만, 실 구현은 어려움
 - 뭐가 중요?
 - 시간, 비용, 건강, 심지어 생명까지 고려해야함(클릭 한번을 아끼는게 중요)
- Usability
 - HCI의 최중요 개념 중 하나.
 - 신뢰성, 오류 처리, 일관성, 성능, 학습 등등...
 - 기술, 개인 요구나 훈련 시간이 적어야함.
- Usability가 어디서 중요한가?
 - 생명-직결되는 시스템
 - 실패의 비용이 너무 큰 경우, 오류를 줄이고 신뢰성을 높이는게 중요함.
 - 훈련 기간이 길더라도, 오류 없는 성능을 위해 비용이 수반될 수 있음.
 - 산업, 상업적 사용
 - 쉬운 학습 + 성능의 속도 + 주관적 만족도가 중요할 수 있음
 - 오피스, 가정, 엔터테인먼트등.
 - 쉽게 학습하고, 오류가 적어야 하며, 만족도가 다 높아야됨.
 - 특정 어플리케이션을 자주 사용하지 않으므로, 인터페이스가 직관적이고 온라인 도움을 받기 쉬워야함.
 - 최고의 구매 경험을 위해서 디자인과 기능 사이에서 한 쪽에 치우치지 않아야함.
- HCI가 어떤 파급효과를 불러올 수 있는가?
 - 성공 스토리
 - MS, Linux 등등의 HCI에 투자한 기업의 성공
 - 경쟁
 - 사고를 방지 및 유발
 - 저작권 침해 소송
 - 기업 인수.
- Universal Usability
 - 모두를 위한 UI ⇒ 모든 사용자가 시스템을 사용할 수 있게 보장하는 것.
 - 평균 사용자라는건 존재하지 않는다.
 - 신체적 능력이 각자 다르다.
 - 타협이나, 여러 버전의 시스템을 만들어야 할 수 있다.

- 손이 닿는 범위, 힘, 속도와 같은 동적 측정도 고려해야할 수 있다.
- 기타 기준
 - 성격의 차이
 - 문화, 국제적 다양성
 - 언어, 관습 등등....
 - 장애를 가진 사용자들
 - 미국의 장애법을 준수해야할 수 있음.
 - 고령의 사용자들
 - 소리, 색상, 밝기, 폰트 사이즈 등등을 조절할 수 있어야할 수도.
- 어케 달성할 수 있음?
 - 가치가 있긴 함?
 - 단순히 서비스를 제공하는 것 만의 문제가 아님.
 - 여러 시스템을 얼마나 많이 만들어야함??
- UX
 - 사용자가 어떤 시스템, 제품, 서비스를 직 간접적으로 이용하면서 느끼는 총체적인 경험.
 - 유저가 겪는 모든 것
 - 단순 욕구의 만족을 넘어 브랜드 충성도, 시장의 지속적 성공이나 Subculture를 창출할 수도 있다.
 - 긍정적인 사용자 경험을 개발 창출하기 ⇒ 사용자 경험 디자인.
 - 그러나 사용자 경험은 다학제적이다 ⇒ 여러 학문 분야가 뒤섞여있다
 - 중요 사항
 - 일관성 ⇒ APPLE같은 Product Familiy
 - 차별성
 - 유저 스테레오 타입
- UX vs Usability?
 - Usability는 시스템의 사용이 얼마나 쉽고 효율적인지를 평가함
 - UX는 그 보다 더 큰 범주의 경험 모두를 평가함.
- 사용성을 위한 설계
 - 디자인 선택에 있어 절충은 자주 발생함 → 인터페이스의 변경이 일관성을 해쳐도 새로운 인터페이스나 기능을 도입하는 기회가 될 수 있음.
 - 설계 대안 ⇒ 모형이나 고충실도 프로토타입을 통해 디자이너와 유저가 설계 대안을 평가할 수 있음
 - 개발 초기 단계에서의 피드백으로 초기 개선이 가능하지만, 더 진짜같은 인터페이스를 위해 추가 비용이 발생할 수도?
- 요약

1. HCI 개념 및 정의

- **HCI(Human-Computer Interaction)**: 인간과 컴퓨터 사이의 상호작용을 설계하고 구현하는 학문입니다. 사용자가 컴퓨터 시스템을 효과적으로 사용할 수 있도록 돕는 것을 목표로 하며, **사용자 경험(UX)**과 **사용성(Usability)**을 중심으로 발전해왔습니다.
- **인터페이스(Interface)**: 사용자와 컴퓨터 시스템 간의 상호작용을 가능하게 해주는 물리적 또는 소프트웨어적 경로입니다. 예: 화면, 버튼, 마우스 등.
- **인터랙션(Interaction)**: 사용자가 인터페이스를 통해 시스템과 작업을 수행하는 과정을 의미합니다. 상호작용 모델은 사용자가 목표를 달성하기 위한 작업의 순서, 맥락, 단계 등을 포함합니다.

2. 사용성(Usability)

- **사용성의 정의:** 사용자가 시스템을 쉽게 배우고 효과적으로 사용할 수 있도록 돕는 것이 목표입니다. 사용성의 주요 요소는 **효율성, 효과성, 학습 용이성, 오류 관리, 사용자 만족도**입니다.
- **사용성의 구체적인 요구 사항:** 미국 군사 표준에서는 사용성을 통해 작업자의 성능 달성, 훈련 시간 최소화, 사용자 장비의 신뢰성 확보 등을 제시합니다.
- **도메인별 사용성 동기:**
 - **생명체 안전이 중요한 시스템:** 항공 교통 제어, 원자력 시설 등은 오류 방지와 신뢰성 확보가 매우 중요합니다.
 - **상업적 사용:** 은행, 재고 관리 등에서는 학습 용이성과 속도, 오류율이 중요한 요소입니다.
 - **오피스 및 가정용 애플리케이션:** 워드 프로세서, 이메일 등에서는 학습 용이성, 낮은 오류율, 사용자 만족도가 중요합니다.

3. 사용자 경험(User Experience, UX)

- **UX의 정의:** 사용자가 시스템, 제품, 서비스를 직접접적으로 이용하면서 느끼고 생각하게 되는 총체적 경험을 의미합니다. UX는 단순한 사용성 이상의 개념으로, **브랜드 충성도, 감정적 반응, 심미성** 등을 포함합니다.
- **UX와 사용성의 차이:** 사용성은 시스템의 효율적인 사용을 평가하는 반면, UX는 브랜드 이미지, 감정적 반응 등 포괄적인 사용자 경험을 다룹니다.
- **다학제적 접근:** UX 설계는 **컴퓨터 과학, 인지 심리학, 디자인, 사회학** 등의 다양한 학문이 융합되어 사용자의 요구와 행동을 이해하고, 긍정적인 사용자 경험을 제공하는 것을 목표로 합니다.

4. 보편적 사용성(Universal Usability)

- **모든 사용자에게 사용성을 보장:** 신체적 능력, 성격 차이, 문화적 다양성, 장애가 있는 사용자 등 다양한 사용자 그룹을 고려해야 합니다.
- **고령자와 장애가 있는 사용자:** 시스템이 가변적 설정(소리, 색상, 밝기, 글꼴 크기 등)을 제공해 다양한 사용자 그룹의 요구를 충족해야 합니다.
- **사용성 보장 방법:** 맞춤형 시스템을 제공하거나 자동화된 설정을 통해 사용자 개인의 요구를 충족할 수 있는 방법을 고민해야 합니다.

5. HCI 설계와 평가

- **디자인 선택의 절충(trade-off):** 인터페이스 변경 시 이전 버전과의 일관성 문제가 발생할 수 있지만, 새로운 기능 도입 및 개선이 필요할 수 있습니다.
- **설계 대안 평가:** ****모형(mockups)****이나 ****고충실도 프로토타입(high-fidelity prototypes)****을 통해 사용자 피드백을 받고 설계 대안을 평가하는 것이 중요합니다. 이는 개발 초기 단계에서 비용을 절감하는 동시에 더 나은 사용자 경험을 제공하기 위함입니다.

6. HCI의 파급 효과(Ramification)

- **성공 사례:** Microsoft, Linux, Amazon, Google과 같은 기업들이 HCI를 통해 사용자 경험을 개선하고 성공을 거둔 예입니다.
- **경쟁과 사고 방지:** Netscape와 Internet Explorer의 경쟁, Apple vs. Microsoft의 저작권 분쟁 등에서 HCI 설계의 중요성을 확인할 수 있습니다. 잘못된 인터페이스 설계는 사고를 초래할 수 있으며, 이는 사용자의 안전과 직결됩니다.

▼ HCI Principles

- HCI의 목적 = 사용성 + ? = UX
 - 최종적으로 UX를 개선하는데 목적을 둔다.
- 원칙, 가이드라인, 이론(규칙 등등,...)
 - 원칙은 가장 높은, 기초적이고 폭 넓게 반영가능
 - 경험 기반
 - 어길 수 있음

- 가이드라인 ⇒ 특정 조직, 작업, 상황 등에 적용
 - Practices 기반 = 유연함.
- 이론
 - 과학 기반 = 그렇게 실용적이지 않을 수도.
- Know Thy user
 - 사용자의 성별, 연령, 여러 특징들을 고려해라.
 - 추측하지 마라 = 안하느니만 못하다.
 - 사용자의 기술 수준에 따라 다른 설계 목표를 세워야 함.
 - 초보자, 간헐적 사용자, 전문가.
 - vs Universal
- Identify the tasks
 - 사용자가 수행하는 작업을 분석(분해)하고, 모델링 해라
 - 작업의 순서와 빈도를 이해해라.
 - 같은 작업이여도, 다른 모델을 가질 수 있다.
 - Task/Goal 구조 ~ UI 구조와 비슷해야함
 - 멘탈 모델과 실제 시스템 디자인이 비슷해야 습득이 쉬움,.
- Minimize memory load
 - 장기 기억보다 단기 기억 → 인식을 위주로 설계
 - 멀티태스킹시, 지속적인 맥락 제공이 필요
- Consistency
 - 인터페이스의 일관성을 유지해서 사용자의 혼란을 방지.
- Provide Closure
 - 작업 완료를 알려줘라.
- Prevent Error
 - 세 종류의 에러가 있다.
 - slips = failure of execution
 - 피할 수 없는 휴먼 에러.
 - 오타와 같은 에러들
 - 이를 해결하기 위한 특정한 기능을 제공할 수 있다.
 - 이메일 자동 입력, 자동 완성 기능
 - 혹은 작업을 취소할 수 있는 기능.
 - 에러가 발생했음을 알려 주는 기능.
 - Strong-but-wrong effect
 - 유저가 기억을 잘못하고 있는 경우
 - 특히나 전문적인 분야에서, 잘못된 지식을 가지고 있는 경우 = 근자감
 - 한번 수정할 경우, 이후에는 똑같은 실수를 반복하지 않음.
 - Inattention or inappropriate attention
 - 특정 작업을 빨리 수행할 수록, 많은 실수를 하게 됨
 - Speed / Accuracy tradeoff

- lapses = failure of memory
 - 유저에게 너무 많은 것을 기억하도록 요구하는 경우.
 - skilled behavior에서 주로 발견됨
 - Mistakes
 - 목표를 이루기 위한 잘못된 절차를 밟는 경우
 - Misdirection
 - 유저가 생각하는 mental model과 UI가 다른 경우.
 - 유저가 생각하는 절차와 실제 절차가 다를 수 있다.
 - lapses와 Mistakes는 상호작용 모델이 좋지 않아 발생한다.
 - slips는 항상 발생할 수 있는 에러이니 괜찮다.
- Choose the right interaction style
 - 적절한 상호작용 방식을 선택해라.
- Provide Useful Feedback.
- Information Scent?
- Natural and Intuitive
- 8개의 황금 법칙.
 - 일관성 유지
 - 보편적 사용성 충족
 - 정보성 있는 피드백 제공
 - 작업 완료를 명확히 하라.
 - 에러를 방지해라.
 - 쉽게 행동을 되돌릴 수 있게
 - 내적 통제감을 지원해라
 - 사용자가 통제한다는 느낌을 줘야함
 - STM을 줄여라.
 - + 작업을 적절히 분할해라
 - 자동화 vs 인간 통제
 - 반복적이고 실수가 잦음 ⇒ 자동화
 - 중요한 결정 ⇒ 인간 통제.
- 참여적 디자인 + KISS + 의인화 X
 - 의인화할 경우, 시스템이 인간과 동일하게 생각할 것이라는 잘못된 착각
 - 시스템의 의도를 곡해하는 경우가 생길 수 있음.
- 접근성 가이드라인
 - 장애인을 위한 지침.
 - 지각 가능 + 운용 가능 + 이해 가능
- 중요 이론
 - 인지 과학(정신) + 휴먼 팩터(신체)
 - 어떻게 정보를 처리할 것인가? ⇒ Fitt's Law
 - 인간이 목표 지점에 도달하는 데 걸리는 시간

- 인간의 정보 처리 모델
 - Rule based reasoning ⇒ sequence of rules
 - 서브 태스크들을 완료해서 최종 목적을 달성하는 것.
 - 액션 = 외부 상황 + 내가 가진 지식을 통해 플랜에 맞는 액션을 취함.
 - Meta Knowledge
- Subgoalting
- Norman's Theory
 - 목표 설정
 - 의도 설정
 - 행동 구체화
 - 행동 실행
 - 외부 상황 인식
 - 외부 상황 해석
 - 결과 평가
- Gulf of Execution
 - ⇒ 사용자의 의도와 가능한 행동 사이의 간극
- Gulf of Evaluation
 - ⇒ 시스템의 상태나 피드백이 명확하지 않아서 현재 상황을 잘못 평가하는 것.
- STM / WM
 - 단기 기억은 7 +- 2개의 청크가 한계임
 - 청크 = 미지의 정보 단위
 - 모든 기억은 새로고침이 필요함
 - Chunking
 - 여러 액션을 묶어서 하나의 청크로 만들.
 -
- ▼ GOMS
 - Goals, Operators, Method, Selection Rules
 - Card, Moran, Newel이 개발했으며, 인공지능 모델에 영감받음
 - 인지적인 워크스루를 통해서 사용자가 목표를 달성하는 과정을 시뮬레이션 함.
 - ACT-R?
 - KLM
 - Key-Stroke Level Model
 - 키 입력, 마우스 움직임, 버튼 누르기 ⇒ Operators
 - 이 연산자들을 통해서 작업 수행 시간을 예측하는 모델
 - 숙련된 사용자를 가정함.
 - GOMS model
 - 특정 목적을 달성하기 위한 Method들의 설명
 - 메소드는 계층적 구조를 가짐.
 - Selection rule을 통해 여러 메소드 중 하나를 고름

- GPS
 - 일반적 문제 해결
- KLM 모델링 방법
 - 상태 가정
 - 사용자의 초기 상태
 - 시나리오 작성
 - 어떤 연산자들을 통해 목적을 달성할 것인가?
 - 시간 계산
- KLM 연산자의 시간
 - K = Key
 - 숙련자 ~ 0.12
 - 평균 ~ 0.20
 - 평균 비서 ~ 0.28 sec
 - 독수리 ~ 1.2
 - T
 - 문자 입력 = 280 msec
 - P
 - 마우스 포인팅 = 1100 msec
 - B
 - 버튼을 누르거나 떼는데 = 100 msec
 - BB
 - 마우스 클릭 = 200 msec
 - H
 - 키마로 손을 이동하는데 = 400ms
 - M
 - 생각 = 1200ms
 - W
 - 시스템의 응답 t msec
- 최종 결과
 - 키 입력은 빠름 = 명령을 기억하고 있는 경우
 - 생각과 마우스는 느림
 - 손 위치 변경도 느림.
- KLM
 - 다른 시스템이나 전략의 효율성 비교할 때 적합함
 - NGOMSL
 - 자연 언어 기반
 - CPM-GOMS
 - 지각, 인지, 운동 과정 사이의 순차적 의존성 맵핑.
 -

▼ Desicion Trees

- 분류나 회귀에 사용되는 기계학습 알고리즘
 - 주어진 데이터를 통해서, 새로운 데이터를 분류하는 가장 최적화된 모델을 찾는 것
 - 동일 데이터가 여러 개의 트리를 만들 수 있음 = 분류 기준을 어떤 속성으로 하나에 따라
- 여러 가지 알고리즘으로 트리를 만듦
 - 유도 + 연역을 통해 만듦
 - 유도 = 분류 기준 만들기
 - 연역 = 만들어진 트리로 새로운 데이터 분류하기.
 - 알고리즘들
 - Hunt's Algo
 - CART
 - ID3, C4.5
 - SLIQ, SPRINT
- 트리 분할 및 정지 기준
 - 그리디
 - 분할 최적화를 위해 속성을 기준으로 분할함
 - 이슈
 - 어떤 질문을 할 것인가?
 - 언제 트리 분할을 멈출 것인가??
 - 기준
 - 모든 레코드가 동일 클래스
 - 모든 레코드의 속성 값이 비슷
 - 이 경우, 조기 종료를 통해 트리 성장을 멈춤.
- 장단점
 - 구축 비용이 싸고, 분류 속도가 빠름
 - 트리 크기가 작으면, 해석이 용이
 - 간단한 데이터 셋일 경우, 다른 분류 기술과 비슷한 정확성
 - 문제
 - 언더피팅 + 오버피팅
 - 값 손실
 - 분류에 비용 발생
- 언더피팅
 - 모델이 너무 단순해서, 훈련 데이터와 테스트 데이터 모두 오류율이 높은 경우
- 오버피팅
 - 모델이 너무 복잡해져서, 훈련 데이터에서는 분류가 정확하지만, 새로운 데이터는 오류가 나는 경우
 - 왜?
 - 노이즈 데이터에 의해 왜곡된 결정 경계가 생긴 경우
 - 훈련 데이터가 부족해서, 특정 영역에서 정확한 예측을 못하는 경우
 - 해결?

- Pre-Pruning
 - 트리가 완전히 성장하기 전에 알고리즘을 멈춰!
 - 모든 인스턴스가 동일 클래스, 혹은 속성 값이 같으면 정지
 - 혹은 최대 질문 개수를 제한함.
 - Post-Pruning
 - 트리를 완전히 성장시키고 불필요한 노드를 제거하는 방식
 - Sub-tree를 리프노드로 바꿈
 -
- ▼ AI UX design
 - 머신 러닝을 통해서 상호작용 가능성을 증가시키자
 - 인공지능 시스템의 해석 / 이해 가능성
 - AI는 사용자가 결과를 이해하고, 신뢰할 수 있게 설명을 제공해야함.
 - 설명이 설득력이 있어야 시스템을 신뢰할 것
 - AI의 작동 모델이 블랙박스인 경우
 - 인과성이 제공되지 않았을 때 신뢰를 잃을 수 있음
 - AD HOC 설명이라도 제공해야함.
 - 정확한 설명이 아니더라도, 알고리즘 설명이라도 해야함.
 - 디테일을 모르더라도, 사용자가 쉽게 예측가능하게 만들어야함.
 - 단순 입력으로 동작해야하며, 결과가 일관적으로 나와야 함.
 - 자동화 vs 인간 개입
 - 자동화와 인간 개입 사이의 적절한 균형을 만들어야함.
 - 문제는 우리가 확실적인 출력을 받는다는 것
 - 믿었을 때의 손익 vs 안믿었을 때의 손익을 계산해야함.
 - AI가 정확할 수록, 무조건 따르는게 이득에 가까워짐.
 - 단순히 한쪽을 희생시키는게 아니고, 2d 적으로 모두 향상시킬 방법을 찾는게 좋다.
 - HCAI 프레임워크
 - 과도한 인간 제어나 과도한 자동화를 피함.
 - Human Performance를 높이기 위해서!
 - AI UI/UX 디자인 원칙
 - Affordance
 - 사용자가 객체나 시스템을 보고 어떻게 사용할 수 있는지 이해시키는 단서
 - 명확한 설명
 - Error Tolerance
 - 잘못된 요구를 시스템이 수용, 복구할 수 있는 것
 - 실수해도 복구할 수 있어야함.
 - 인지 가이드
 - 사용자의 상호작용을 올바른 방향으로 유도하는 것.
 - 사용자가 알아보기 쉬워야함.
- ▼ 사용자 분석

- LUCID
 - 사용자 중심의 인터랙티브 디자인.
- 사용자
 - 메인 유저 = 실제 상호작용하는 사람
 - 이차 유저 = 메인 유저의 상호작용에 영향받는 사람.
- 뭘 알아내야 함?
 - 어떤 HCI 원칙을 사용자가 중시하는지
 - 사용자의 숙련 정도
 - 초보자, 중급자, 숙련자, 전문가
 - 사용자의 행태적 특성
 - 상호작용
 - 빈도, 깊이 등
 - 정보
 - 기능
 - 시스템을 위한 최소한의 기능
 - 사용자의 개인적 특성
 - 성격, 신체, 문화, 동기
- 사용자들이 관심있어 하는 것.
 - 유용성
 - 기능적 가치
 - 사용성
 - 정확성, 효율성 가치
 - 감성
 - 인상, 개성
- 유저 분석법
 - 시퀀스 모형 분석법
 - 사용자가 일을 하는데 필요한 세부적인 과업들을 순차적으로
 - 계층적 과업 분석 방법
 - 하나의 일을 계층적으로 분할함
 - 너무 인간답지 않음 ⇒ 획일적, 공식적임.
 - Function-central system
 - 지식 기반 분석법
 - 과업을 위한 도구와 행위를 파악, 분류 체계를 작성함.
 - Content-central
 - 시나리오 기반
 - 사용자의 경험 → 모델로 추출
- 사용 맥락
 - 사용자의 행동에 영향을 미치는 모든 요인
 - 물리 + 사회 + 문화적 맥락.

- 5W + 1H
- 세부 절차
 - 트리거를 찾아라
 - 한 행동을 이루는 청크를 찾아라
 - 추상적 단계를 만들어라
 - 의도를 파악해라
- 페르소나
 - 최종적으로 만들어낸 사용자
 - 인물적 배경 + 설계 대상과의 관계 + 목적, 요구, 태도 + 특정 지식, 능숙도를 가짐.
- 시나리오
 - 여러 Ways ⇒ Triggers + Functions
- 인터페이스 선택
 - 특정 인터페이스만 선택해야 하는 경우가 있을 수 있음
 - 선택지가 한정되어있을 경우, 통일하는게 나올 수 있음
 - 너무 인터페이스가 자주 바뀌면, 사용자가 혼란스러워함.
- 참여적 디자인?
 - 시스템에 대한 사용자의 수용성 증대 가능성 + 작업에 대한 정확한 정보
 - 비용증가 + 구현 기간이 길어짐.
- 고려할 점
 - 사회적, 법적 영향 + 데드라인