**<Ch1. Introduction to HCI>**

**HCI**: **H**uman **C**omputer **I**nterface/**I**nteraction

Interface/Interaction = User Interface (UI) + User Interaction (UI) + User Experience (UX)

- **UI/UX**는 **usability** 와 **Human Factors/Ergonomics**(인체공학)을 포함한다.

- **UI/UX**는 사람들이 컴퓨터와 **상호작용**할 수 있게 해주는 애플리케이션의 일부이다.

HCI = design, prototyping, evaluation & implementation of UIs을 포함한다.

**\* HCI의 목표**

: **컴퓨터 인터페이스의 사용성(usability)**을 향상시키는 것을 목적으로 한다.

-> 인간의 적절한 연구는 디자인의 과학에 있다. 결국 인간을 잘 이해하고, 인간이 사용하기 쉽게 만드는 디자인이 좋은 디자인이라 할 수 있다.

**\* Usability**: 그저 새롭게 만들고, dry한 디자인을 만드는 것 보다 **사용자의 목적에 맞게** 만드는 것이 중요하다. (it all depends on your goals)

=> Robust, Discoverable, Pleasing, Fun, Learnable, Memorable, Flexible, Efficient

**\* 좋은 UX/UI를 만들기 위해서는?**

1) 사용자를 잘 고려해서 만들어야 함.

2) 사용자의 인지 능력을 이해해야 한다, (perception, Physical manipulation, Memory)

3) 조직적인 업무 능력과 기술이 있어야 한다. (사용자의 참여를 유지하는!)

4) 개발자는 target customer와 함께 소통하며 작업해야 한다.

(user 관점에서 바라보고, 기술 이나 기능을 중점적으로 보지 않는다.)

5) 빠르게 prototype을 만들어야 한다. (설계 모형 구축, 종이 스케치 등등)

**\* HCI의 중요성**

**<이론적(Theoretical) 측면>**

1. “real” 프로그램의 대부분의 문제는 나쁜 user interface가 원인이다.
2. 사용자의 만족도를 충족하는 것이 이익을 향상시켜준다.
3. 하지만, user interfaces가 제대로 작동하지 않는다.

- 사람들은 예측하기 어렵다.

- 때때로, 디자이너들의 의도는 잘못 전달된다.

**<사업(business) 측면>**

Apple이 어떻게 Microsoft를 따라잡았는가?

- usability에 대한 종교적 불신

- “Maria” 고객을 기반으로.

- 핵심 역량에 집중. (software, hardware design, 제품 및 앱 스토어에 대한 관리)

- 기술 패러다임을 시리즈화. (iPod, iPhone, iPad)

\* HCI 관련 분야

초기: 컴퓨터 과학과 심리학을 공동으로 연구함.

최근: HCI 연구 분야에 포함된 컴퓨터 과학, 행동 과학 등등.

- Understanding human

- Understanding computer

- Understanding usability

- Design (product design, web design, UX design)

**<ch2. History of HCI and fundamentals>**

\* Vannevar Bush: **Memex** 아이디어 고안

- 인간의 **지식을 유지**하기 위한 연구의 필요성 제시

: 사용자가 여러 세대의 인간에 걸쳐 저장된 방대한 양의 데이터를 빠르고 쉽게 검색할 수 있다면, 인간의 지능은 현저하게 증가할 것.

- **Memex 개념**: 저장된 데이터를 찾아 화면에 투사하여 정보를 표시한다.

마이크로 필름 타입의 데이터를 저장하는 개념, 정보를 보여주고 스크린에 투영하는 것까지 개념으로 상상함.

**\* SketchPad (Ivan Sutherland, 1963)**

: 어셈블리 언어로 직접 그래픽을 코딩하는 것이 아닌, sketchPad를 사용하여 직접적으로 구현할 수 있게끔 함. 첫번째 **direct manipulation interface**(ex.스타일러스 펜)이다. 컴퓨터 그래픽스 역사의 시작.

- **Direct Manipulation** (defined by Ben Shneiderman, 1983)

1) Visibility of objects (object가 무엇을 하고 있는지를 보여줌.)

2) Incremental action and rapid feedback (잘못 그려진 경우에 재빠른 feedback을 받을 수 있음)

3) Reversibility(반전성) (지금 하고 있는 기능을 undo하는 것까지 제공)

4) Exploration

**\* Head tracked VR/AR (Ivan Sutherland, 1965-70s)**

: 컴퓨터 그래픽스의 아버지다. Stereo, see-through head-worn display. 양눈을 이용해서 밖을 볼 수 있는 디스플레이. 가상으로 만든 이미지가 실제 세계에 적용되도록 함. HWD를 사용하기 위해서 먼저 head tracking이 이루어져야 함. AR기능이 이때 구현이 됨.

\* Tracking (Robert Burton, 1973)

: Scene scannin/tracking 발명. 다수의 LED들을 실시간 3D 추적, 장면 레이저 스캐닝=> 손 끝에 불빛을 달고, 공간에서 scanning을 해서 손의 위치를 3차원 공간상에서 찾음

**\* Augmenting Human Intellect (Douglas Engelbart, 1960)**

: Stanford Research Institute (SRI)=> 키보드와 마우스에 대한 개념. 점점 컴퓨터로 해야 하는 일들이 복잡해지면서, 컴퓨터에 입력하는 것을 쉽게 하도록 해야겠다 하고 생각하게 됨.

- 최초의 키보드와 마우스가 이 때 처음 제안됨.

- 키보드랑 마우스 등의 형태에 동등한 개념을 적용해서 입력을 보다 더 쉽게 할 수 있게 함.

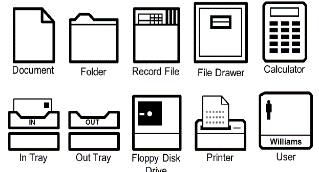
- 마우스 휠의 움직임을 통해서, 커서의 위치를 조종하는 마우스를 이때 처음 개발.

Ex) **first** mouse, **first** 2D editing & windows, **first** hypertext, **first** word processing, **first** document version control, **first** distributed client-server 등등

인간의 지능이 향상되기 시작하면서, 여러 기능들이 시작됨.

\* Dynabook (Alan kay, 1972)

: 현재 알고 있는 아이패드나 탭에 대한 전신. 모든 연령대에서 사용할 수 있는 개인적인 컴퓨터를 처음에 제안함.



**\* Xerox Star (1981)**

- 첫번째의 상업적인 GUI.

- 윈도우 기반으로, GUI 모양을 제시해서 사용자가 특정 기능에 편리하게 접근할 수 있도록 함.

- **Desktop metaphor**를 차용함 => 우리가 일반적으로 알고 있는 desktop 환경에서 사용하던 것들을 차용해서 사용했다는 의미.

**\* HCI의 정의**

- Human – Computer interaction

: 사람과 컴퓨터의 **interaction을 중심**으로 컴퓨터 기술을 설계하는 연구

\* Ergonomics(인체공학): 기본적으로 인간의 한계를 고려해서 어떤 식으로 기계를 설계하는 것이 좋다, 하는 것을 연구하는 학문. HCI측면에서 설계를 할 때, 인체의 편리함을 고려해서 구성이 된다.

**\* UI/UX**

: 사용자가 product를 사용하는 수단. **사용자의 경험**이 그 product에 대한 의견을 이끈다.

**\* User interface**: 머신의 part. 보통 physical한 부분을 말함. Ex) switch, keyboard, touchscreen 등

**\* User interaction**: User interaction research ways in which humans interact with computers and design technologies that let humans interact with computers in novel ways.

=> 사용자와 컴퓨터가 상호작용하는 방법 및 인간이 새로운 방식으로 컴퓨터와 상호작용할 수 있도록 하는 설계 기술.

**\* User experience:** A person’s perceptions and responses that result from the use or anticipated use of a product, system or service.

=> **사용자의 경험**을 다루는 용어를 포함할 때, UX라고 한다. 사용자가 어떻게 인지하고, 어떻게 경험을 하는지를 다루는 용어다. UI보다 큰 개념.

**\* UI/UX의 중요성**

1) 사용자가 기능을 제공받는데, 그 기능을 충분히 잘 사용할 수 있도록 해주기 위해 UI/UX가 중요하다.

2) UI/UX는 사용자의 브랜드 충성도 또는 이탈의 중요한 요소이다.

: bad UI/UX는 사용자가 떠나게 만들 수 있기 때문.

\* UI/UX는 사용자들에게 **익숙한 메타포(metaphor)**를 이용해서 사용하게 쉽게 만들어야 함.

\* UI/UX의 도전 과제들

- 비전문가들도 사용하기 쉬운 UI/UX를 제공하는 것이 요구됨.

- 일반적으로, 누구나 활용하기 쉬운 UI/UX를 만들어야 함. (it all depends on your goals)

**\* UI/UX의 어려움**

1) 실제 구현해야 하는 feature/function의 수가 많음.

2) many technology-based features/functions

3) input & output restrictions

4) Wide variety of user reactions => preferences, needs, accessibility 등이 모두 다름.

**<ch3. Understanding Human1>**

\* perception: 인지한다는 것은 환경을 표현하고 이해하기 위한 감각 정보의 구성, 인식 및 해석하는 것이다.

**[Factors affecting perception]**

**-** **sensory limitations** 감각 한계

인간이 가진 기본적인 한계들을 다 고려해서 설계해야 한다. (seeing, hearing, smelling, touch)

**- Brain Limitations** 뇌의 한계

(remembering 기억, attention span 집중력의 한계, information overload 정보 과부하, logical thinking 논리적 사고)

**=> 이런 한계들이 perception에 영향을 미침.**

**- (Simply) Being Human 인간으로서**

(emotions, prior experiences 기존의 경험, mood, stress, decision without enough data)

=> 우리가 센싱하는 정보를 제대로 이해하지 못할 수 있음.

**- External Factors 외부 요소들**

(alcohol, drug, influencers ....)

**\* Visual Perception**

: 눈으로 인지하는 것. 단계는 두가지

1. 물리적으로 자극을 받아들이는 것.
2. 그 자극을 가공하고 해석하는 것.

\* Eye – physical reception

1) 빛을 받아들이고, electrical energy로 변환하는 mechanism.

2) 빛을 객체로부터 받아들인다.

3) 이미지들은 망막에 거꾸로 맺힌다.

4) 망막은 낮은 빛에서 **rod(명암구분세포)**와 색을 위한 **cone(색상구분세포)**을 포함한다.

5) 골수세포(Ganglion cells)는 패턴과 움직임을 감지한다.

**\* Visual Perception**

=> 수정체를 통해 망막에 상이 맺히고, 그곳에 시신경이 있어서 **망막이 processing**을 함. 그래서 우리가 본 것에 대한 판단이 가능해짐.

- Scotopic(암순응): Rods 사용.

- Photopic(광순응): Cones 사용. 세가지의 cone이 존재, RGB가 아니라 거기에 가까운 색을 나타냄.

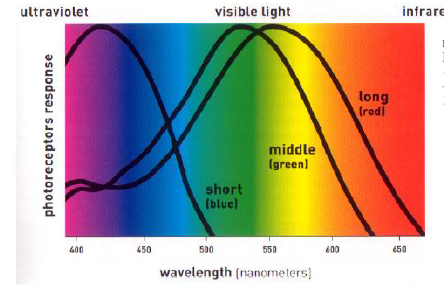
- Mesopic: Combined. 빛의 양이 상당히 적을 때, rod와 cone이 함께 반응을 해서 색 인지.

ex) 새벽, 해질녘.

**\* Color sensitivity 색감 (cone을 사용하여 색감을 구분함)**

: 눈이 가진 한계로 인해, wavelength의 degree가 다양함.

**L**ong(red), **m**iddle(green), **s**hort(short) => RGB랑 비슷함.



만약, red light이 비췄다면, **L-cone**이 M-cone보다 더 많은 자극을 받아서 활성화되지만, S-cone은 전혀 관계가 없어진다. L-cone이 더 많은 자극을 받기에 빨간색이라 인지가 가능해진다.

- 고르게 분포되지 않은 color wave에서, 매우 짧은 wavelength는 예민해서 잘 인지할 수 없다. 나이가 들수록, short wave는 거부된다. 잘 인지하기가 어렵다. 이런 이유로, 나이가 들면 색을 구분하기가 어려워진다.

- 가장 sensitive한 색들은 대부분 스펙트럼의 가운데에 있다. Green & yellow는 스펙트럼의 가운데에 있어서, 빛을 소량만 쐐도 인지를 잘할 수 있지만, red나 blue는 그게 어렵다. 이런 한계들을 극복하기 위해서 brightness나 color difference(보색대비)를 사용한다.

\* Field of View & resolution

1) **Temporal resolution**: 눈은 기본적으로 temporal resolution을 가지고 있다. 눈 깜빡임 같은.

2) **Spatial resolution**: 공간에서 우리가 어느 정도까지 볼 수 있느냐.

=> HCI측면에서, 눈의 **temporal resolution**과 **spatial resolution**을 잘 고려해서 설정해야 한다.

**\* Interpreting the signal (신호 해석)**

- size and depth: 크기나 깊이 정보를 가지고 어떤 것 위에 어떤 게 overlapping 되는지에 대한 cue를 획득할 수 있음.

- Brightness: 빛의 수준에 대해 주관적으로 반응함. 사물의 밝기에 영향을 받음.

- Color

=> 센서로부터 신호가 들어온 다음에, 우리 **스스로 interpreting**을 함.

**\* Depth Cue (깊이 단서)**

: 물체의 깊이를 이해하는 데에 도움을 주는 단서.

**- Monocular, Static cues**

1) relative size: 사이즈 다르게 하기

2) Occlusion: 겹쳐진 정도

3) Relative height to the horizon

4) Linear perspective: line을 더 그려주면서 3차원 공간에서 어느 정도의 깊이를 가지고 있는지에 대한 단서를 줄 수 있음.

5) Shadow

=> 이러한 단서들로 depth 정보나 높이 정보, 크기 정보 등을 알 수 있음.

**\* Sound**

소리는 압력의 파도에서 공기, 물 그리고 다른 문제들을 통해 움직이는 에너지의 한 형태.

**\* Audio Perception:** 거리, 방향, 물체에 대한 정보를 제공한다.

**- Monaural, State Cues**

1) Volume: 강렬함 -> 근거리

2) Spectrum: 고주파수 감소

3) Reverberation(반향)

4) Doppeler effect: 소스로부터 상대적인 모션에 따라 들리는 소리가 변하기도 함.

**\* Touch**

: 환경에 대해 중요한 feedback을 제공함.

- Haptics Display: 우리가 느끼는 **압력**을 액츄에이터를 사용해서 인지할 수 있도록 함.

**[Human memory]**

**\* Memory**

**Sensory memories**에 센싱된 메모리를 잠깐 넣어둠 -> **Attention** 집중을 하게 되면, 센싱 메모리 중 일부가 **short-term memory**또는 **working memory**에 들어감. 이 후, **Rehearsal**(반복연습)을 하게 되면 **long-term memory**에 들어감.

**- Sensory memory**

: 여러 자극들을 잠깐 저장해두는 기능을 가지고 있음. 자극에 대한 **Buffer 역할**.

ex) “sparkler” trail(향흔적): 향에 대한 것을 기억해두기 때문에 향을 따라갈 수 있음

stereo sound: 깊이 정보를 알 수 있는 이유도 데이터가 들어왔을 때, 잠깐 저장해뒀다가 사용하기 때문이다.

**- Short-term memory (STM)**

: **임시적인 기억**을 위한 scratch-pad

**- Long-term memory (LTM)**

: 우리의 **모든 지식에 대한 저장소**. 감소가 천천히 일어난다.

1. Episodic: 일련의 사건에 대한 메모리
2. Semantic: 사실, 개념, 기술의 **구조적**으로 되어있는 메모리

=> 정보에 접근을 쉽게 해준다. 정보들 사이에 관계를 표현해서 추론이 쉽다.

**\* LTM**

**- Storage of information**

1) Rehearsal: STM에서 LTM으로 정보를 이동시킴

2) Total time hypothesis: rehearsal time에 비례해서 얻게 되는 양

3) Structure, meaning and familiarity: 기억하기 쉬운 정보들

**- forgetting**

1) Decay: LTM에 저장되어 있는 정보는 느리게 잊혀진다.

2) Interference: 새로운 정보를 받아들이려 할 때, 옛날 정보로 인해 억제될 수도 있다.

3) affected by emotion: 메모리들이 선택적으로 지워지게 할 수 있는데, 이때 영향 중 하나인 emotion이 많은 영향을 미친다.

**- Retrieval** 저장된 정보를 뽑아오는 방법

1) Recall: 어떤 cue를 주고 그 cue에 의해 정보가 다시 생성된다. 회상

2) **Recognition**: 인지. 이전에 봤던 지식을 사용해서 현재 정보를 인지.

=> UI 디자인을 할 때, 기능을 사용하는 것에 있어서 어떻게 해야 하는지 쉽게 알게 해주는 cue를 줘서 recall보다는 recognition해서 정보를 쉽게 받아들일 수 있도록 해야함.

**[Cognitive processes]**

**\* Attention 주의 집중**

: 우리 주변에는 자극이 많음. 이것에 포커싱하게 하는 게 attention

사용자의 attention을 이끌 수 있도록 디자인하는 것이 중요함.

Ex) Gestalt psychology: 실제로 우리가 visual한 반복적인 이미지를 통해 특정 정보를 grouping 하거나 하나의 통일된 것으로 인지하는 이론.

(similarity, anomally, continuation, closure, multistability)

**\* 디자인할 때 Attention 측면에서 고려해야 할 점.**

1) 집중이 필요한 부분을 구분하게 만들어야 함.

2) color, ordering, spacing등을 사용해서 강조할 부분은 pop out이 되도록 해야함.

3) 너무 많은 정보로 interface를 복잡하게 하는 것을 피해야 함.

**<ch4. Understanding Human2>**

**[Human motor behavior]**

**\* Movement**

- 자극에서 respond 되는 시간: reaction time + movement time(정보를 받는 시간+행동하는 시간)

- movement 시간은 나이나 건강에 의존한다.

- 자극의 유형에 따라 reaction time이 다르다.

**\* Motor Behavioral models**

: 특정한 종류의 모델을 만들어서 인간의 behavior을 분석함.

1. **Predictive models**: 분석적으로, 숫자로 얘기할 수 있는 model

특정 가설을 세워서 만들고, 미리 예측을 할 수 있도록 해주는 기능을 함

* 1. **Paul Fitts’s Law (1954)**

: human movement를 모델링한 것.

**[ T = a + b log2(2 \* D/W) ]**

어떤 타겟에 도착하는 시간 T는 Distance에 비례하고 width에 반비례함.

어떤 타겟에 도착하는 시간은 거리가 짧고, 타겟의 크기가 클 수록 빠름.

* 1. **Keystroke Level Model (1980)**

: 특정한 task를 하는데 걸리는 시간을 모델링함.

**[ Texecute = tK + tP + tH + tD + tM - tR ]**

키보드에서 타이핑할 때 사용할 수 있는 모델. 실행되는 타임을 위와 같이 정의.

인간이 생각하는 시간 tR은 뺀다.

* 1. **GOMS Model (1983)**

(Goals, Operators, Methods, Selection Rules)

: GOMS 모델은 사용자가 이미 숙달한 작업을 수행할 것으로 예상되는 상황에 적용. Quantitatively와 Qualitatively를 사용할 수 있음.

Quantitatively: time을 측정하고 있기 때문에, 좋은 예측을 할 수 있음.

Qualitatively: 훈련 프로그램을 디자인하는 것을 도울 수 있고, 연구의 결과를 분석하는 일에 도움을 줄 수 있음.

* 1. **Hick-Hyman’s Law (1951)**

: 시간은 가능한 선택지의 수에 비례한다. 우리가 선택해야할 초이스가 증가할 수록 결정하는 시간도 증가한다.

우리가 화면상에서 보여줘야 하는 **메뉴의 개수를 예측**하는 데에 사용할 수 있음.

**[RT = a + b log2(n)]** (RT=response time; n = number of choices)

1. **Descriptive models**: 설명하는 model. 일반적으로 문제에 대해 생각할 수 있는 framework를 제공해주는 model.

**2-1) Key-Action Model (KAM)**

: 컴퓨터와 어떻게 인터렉션 하는지를 보는 model.

Symbol keys, modifier keys, executive keys 같은 key들을 잘 배치해서 사용. 세가지 framework.

**2-2) Bill Buxton’s 3-state Model (1985)**

: 인터랙션을 하는 동안 발생하는 state와 그 state가 발생하는 동안을 나타내는 model. Out of Range, Tracking, Dragging 총 3가지 state로 나눠 짐.

**2-3) Yves Guiard’s Model (1987)**

: 두 손이 서로 다른 일을 하고 있다, 라는 것에서 착안. 일반적으로 사람들이 많이 사용하는 손과 보조로 사용하는 손의 역할이 다르므로, 역할에 맞게 interface를 구성해야 한다. **두 손의 상호작용**에 초점

**<ch5. Usability (Good and Bad design)>**

**\* 좋은 디자인이란?**

: 사용자의 요구에 부합하도록 제품과의 **상호작용**을 최적화하는 것.

**\* UI/UX 디자인의 목표**

사용자들이 모든 기능들을 잘 사용할 수 있도록 해야 함.

1. Easy to perceive 알기 쉽게
2. Easy to use 사용하기 쉽게
3. Easy to learn 익히기 쉽게
4. Easy to retain 유지보수가 쉽게
5. Difficult to misuse 오용되지 않게

**[Design rules]**

**\* Principle**

: 추상적인 디자인. **Low authority, high generally**

원칙인데 지켜도 그만, 안 지켜도 그만. 일반화 해서 이야기할 수 있는 것.

**\* Guidelines**

: **Lower authority, authority**

지켰으면 좋겠다 하는 것. recommend에서 지켜주는 것이 좋음. principle보다는 더 지켜야 함.

**\* Standards**

: 구체적인 디자인. **High authority, Limited application**

Authority가 굉장히 높음. 반드시 따라야 할 표준.

=> principle은 지향하는 바. Guideline은 application dependent에서 지켜줘야 하는 것. Standard는 100% 지켜야 하는 것.

**\* Principle of usability**

**1) Learnability:** 사용성을 최대화하기 위한 규칙. 사용자가 기능을 딱 보고 어떻게 사용해야 한다. 하고 바로 알 수 있게 하는 것.

**1-1) Predictability**

: 과거에 interaction한 history를 통해 쉽게 미래를 예측할 수 있도록 하는 정보 제공.

* + 1. **Visibility**

: 사용자에게 제공할 정보를 visible 하게 제공하는 것. 사용자가 어떻게 그 기능을 사용해야 하는지에 대한 정보를 시각적으로 제공.

* + 1. **Feedback**

: 사용자에게 어떤 일이 일어났는 지에 대한 내용을 다시 사용자에게 알려주는 것. 버튼을 눌렀을 때, 색이 변하는 것 또한 feedback의 일종

* + 1. **Constraints**

: 수행할 수 있는 작업을 제한함. 옳지 않은 옵션을 선택했을 때, 못하게 막거나, 사용자가 사용할 수 있는 기능만 enable하게 보여주는 것 등이 있음.

* + 1. **Affordance**

: 이 기능을 어떻게 사용할지 보자 마자 유추 가능하게 하는 것. 행동유도성을 말함. 기능의 사용법에 대해 clue를 주는 것을 affordance를 제공한다고 함.

Virtual affordance=> 실생활에서 이미 인지된 것을 통해 가상의 affordance도 적용할 수 있음.

**1-2) Synthesizability**

: 이전 작업이 현재 상태에 미치는 영향을 평가하기 위해 제공.



=> 이런 거!

기존에 했던 일을 통해서 내가 왜 이 페이지에 있는지, 이런 것들을 알려주는 것

**1-3) Familiarity**

: 유사함. 사전 지식이 어떻게 새로운 시스템에 적용되는 가! Physical world에서 보고 있는 것들을 metaphor를 활용해서 적용시킨 경우 familiarity를 고려했다고 할 수 있음.

**1-4) Generalizability**

: ctrl+c, ctrl+v 같은 것. 많은 다양한 애플리케이션을 걸쳐서 공통적으로 사용될 수 있는 것

**1-5) Consistency**

: input/output 기능들이 같은 상황에서 일관성 있게 행동할 수 있도록 제공해야 함.

**2) Flexibility:** 사용자와 시스템이 정보를 교환하는 다양한 방법들을 제공해야 함.

**2-1) Multithreading**

: 한번에 한 개 이상의 일을 할 수 있게 함. Ex) street map and voice for navigation

**2-2) Task migratability**

: task에 대한 책임을 누구에게 넘기는 가. Task가 migration 됐다는 것은 authority가 넘어갔다는 것을 의미함.

**2-3) Substitutivity**

: input과 output형식을 맞춰서 서로 상호 보완적인 역할을 하도록 하는 것.

**2-4) Customizability**

: 사용자가 customize 할 수 있게 하는 경우도 만들어야 함.

**3) Robustness**

: 목표 지향적 행동을 성공적으로 달성하고 평가하는데 있어 사용자에게 제공하는 것.

**3-1) Observability**

: 시스템의 표현으로부터 시스템의 내부 상태(진행도 같은)를 평가하도록 보여주는 것. 안에서 시스템이 어떻게 동작하고 있는지에 대한 정보를 바 그래프 형태로 보여줄 수도 있음.

**3-2) Recoverability**

: 원하지 않는 일을 했거나, 에러가 발생했을 때 잘 recover할 수 있는 기능도 제공해야 함

**3-3) Responsiveness 반응성**

: 사용자가 하고 있는 기능이 어느 정도 시간안에 끝낼 수 있는지를 반응해서 알려주는 것

**3-4) Task conformance (준수)**

: task를 얼마나 잘 준수하느냐, 어느 정도로 사용자의 task를 support하고 있는지.

**\* Guidelines**

**- UX guideline from Apple**

**<architecture>**

1) Clarity: 정보를 명확하게 제공해라

2) Deference: content가 전체 스크린상에서 명확하게 보이는 것이 중요하다.

3) Depth: 깊이 정보를 잘 활용해라.

4) Accessibility: 장애를 가진 사람들을 위한 높은 접근성을 제공해라

5) Modality: alert 메세지를 띄우는 등의 modality를 최소화해라.

6) Loading: 로딩화면을 clear하게 유지해라

등등

**<interactions>**

: 3D touch, Feedback, Gesture을 사용자와 상호작용하기 위해 적절하게 제공해라.

**- UX guideline from google**

1) Material is the metaphor: 빛이나 그림자 등의 질감을 실생활에서 영감을 받아 디자인함.

2) Bold, graphic, intentional: 여러 디자인 method를 이용해서 material design을 함

3) Motion provides meaning: motion은 집중을 높이고 적절한 feedback을 주는 등의 의미 있는 것으로 제공되어야 함.

4) Flexible foundation

5) Cross-platform: platform을 거쳐가면서도 같은 UI를 유지해야 한다.

등등

**- UX guideline for VR**

=> interface를 상호작용하고, 반응성 있게 만들어야 함. 편하고 쉽게 디자인해야 하며, 실생활에 있는 것처럼 보여야 하고, 안전성이 있어야 함.

**<ch7. Interactions design principles>**

**[Investigation]**

**\* Identify needs**

: 유저의 needs는 어떻게 판단하는 가?

**1) Key questions**: 유저가 누구인지, 유저의 need가 무엇인지, 유저가 어떻게 활동하는지에 대한 질문들을 하면서 need를 판단함.

**2) Learn from people**: 인터뷰나 설문조사를 통해 알아 감.

**3) Learn from experts**: 주제에 대한 전문가를 만나 봄.

**4) Immersive yourself in context**: 나 자신이 유저라고 생각하고 실제 그 상황에 들어가서 유저의 needs가 무엇일지 예측하는 것.

**5) Use your imagination**: persona라는 가상의 인물을 하나 만들고, 실제로 어떤 유저가 뭘 원하는 지에 대해 상상을 통해서 만드는 방법.

**\* Method to understand users**

**1) Talk to them**

: 직접 사용자를 찾아가서 무엇을 원하는지 물어보는 방법

=> Data recording, **Interview** (unstructured, structured, semi-structured, Focus groups)

- Closed questions: ‘yes’, ‘no’같은 답변을 유도하는 질문. 분석이 쉽지만, 샘플링이 어려움. 모든 집단에 동등하게 설문이 돌아가는 지 확인할 수 없음.

**2) Watch them**

**- observation (관찰)**: 직접적으로 관찰 => 속해서 관찰하거나, 밖에서 관찰할 수 있음. 통제된 환경에서의 관찰도 가능. 간접적으로 관찰 => 사용자의 활동을 추적함. (다이어리나 비디오)

**- Contextual Inquiry (상황 조사)**

: 고객의 요구 사항 및 작업 관행을 이해하는 방법, master/apprentice 모델을 통해 고객은 자신이 무엇을 하는지 배울 수 있음.

1. **Use your imagination**:

- persona라는 **가상의 인물**을 하나 만들고, 실제로 어떤 유저가 뭘 원하는 지에 대해 상상을 통해서 만드는 방법.

- Scenarios: 시스템을 통해 가는 하나의 직선 경로..?

장점) 삶과 시간은 선형적이다. 시나리오를 사용하면 이해하기가 쉽다. 그리고 구체적이다.

단점) 선택지가 없고, 잔가지가 없고, 특별한 조건들이 없다. 그래서 의도되지 않은 실수가 있을 수 있다.

**\* Analysis**

**1) Quantitative analysis (양적 분석)**: Numerical method임

1-1) Average: mean(값을 다 더하고 데이터 수로 나눔), median(순위 지정 시 데이터의 중간 값), mode(데이터에 가장 자주 나타나는 숫자)

1-2) Percentages

1-3) Graphical representations give overview of data

1-4) Tools to support data analysis 데이터 분석 도구 (ex. spreadsheet)

**2) Qualitative analysis (정량 분석)**: 요소의 특성을 표현하고 테마, 패턴, 이야기로 표현함.

2-1) Recurring patterns of themes: 반복 패턴 또는 테마

2-2) Categorizing data

2-3) Looking for critical incidents

**\* Task Analysis**

- 이를 사용하여 상황 별 조회 데이터를 구성한다.