CI : Intro to HCI

Goals of HCI

- Increase individual & organizational productivity
- Improve safety & reduce health hazard
- Reduce costs associated with development of user interfaces

User Interface

- A part of the product / system / software that user see, hear and touch.

Human

- The one whom computer systems are designed to assist.

Usability

- Effectiveness, efficiency and satisfaction with which specified users achieved specified goals in particular environments.
- A quality attribute that indicates how easy the user interfaces are to use.
- ISO : Effectiveness, efficiency, satisfaction

- Pros to user :
  - ↳ productivity ↑
  - ↳ learning time ↓
  - ↳ errors ↓
  - ↳ stress ↓

- Pros to developers :
  - ↳ Cost of customer supports ↓
  - ↳ Training time ↓
  - ↳ Reputation ↑

Effectiveness

- Completeness and accuracy with which users achieve specified goals.
- Users' goals are met ?
- All works are correct ?
- Desired output = actual output ?

Efficiency

- The resources used in relation to the results achieved.
- How fast the task can be done?
- Productivity ↑ while using system?

* Since human effort can't be measured, we use time as measurement.

Satisfaction

- User's physical, cognitive and emotional responses that result from the use of a system, product or service meet the user's needs and expectations.

- Comfort or acceptability of work system to users

Effect of poor UI design / usability problem

- Users make more errors
- Users' anxiety and inconvenience
- User productivity ↓
- Product is difficult to learn and use
- Low quality and value product

Can relate to the pros to users in previous page

Disciplines contributing to HCI

Computer Science

- Study of software construction
- Provide knowledge about capabilities and limitation of technology.

Cognitive Psychology

- Study of how info is processed and represented in mind. (how human interprets information)
- Provide knowledge about capabilities and limitation of users.

人体工学
Ergonomics / human factors

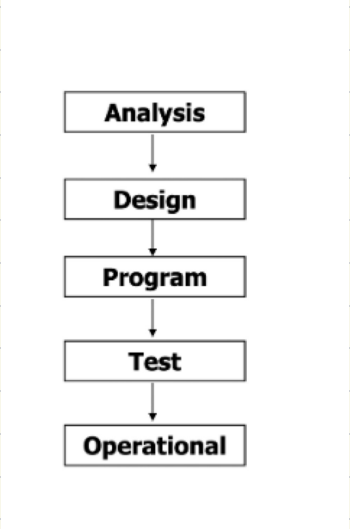
- Study of human beings in relationship to their working environment.
- How to design tools to suit the capacities and capabilities of users.
- How to design working environment to suit the users.

Social and Organizational Psychology

- Ideal designers of interactive system should have expertise in variety of disciplines.

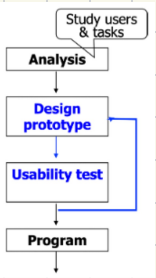
Waterfall Approach

- Development process driven by functional processing requirements.
- Deliver what designer think what the users want.



Vs

HCI Framework



- Focus on users' need
- Analyze user's tasks to provide functionality to support those tasks.
- Build prototype
- Carry out early testing & evaluation with users
- Iterate until a good design is achieved.

PACT Analysis

People

- Target user
- What information get from user
- e.g. sociology, culture, biology, language, psychology

Activities

- Interaction in context of some "community of practice"
- Behaviors, daily routine, activities done by users.
- e.g. Time, complexity, safety critical

Context

- Environment where users are in
- Background
- Derived from activity
- e.g. Physical environment, language, culture

Technology

- Selecting suitable software and hardware.
- Review new technology that may ease your system.
- e.g. Input (touch screen, mouse)
  Output (monitor, speaker)

(2: User Analysis & Task Analysis

User Analysis

- Who are the users?
- What are the tasks?
- What is the environment where the system operates in?

e.g. User : Teenager
Task : Video call with friends
Environment : At house

User Types

- Novice users : Users who attempt to use a system for the first time.
  - ↳ Unfamiliar with system
  - ↳ Well understand the goals to be accomplished

间歇性

- Knowledgeable intermittent users
  - ↳ Have stable knowledge of task
  - ↳ Difficult to remember details of system operations
  - ↳ Not using systems regularly

- Expert / Frequent users
  - ↳ Very familiar with systems and tasks
  - ↳ Uses computer terminologies, syntax, abbreviation.

Design for novice / intermittent / expert users

Novice Users

↳ All initiatives should come from the computer
- System tells them what to do
- System give simple response to instructions

↳ Each required input should be brief

↳ No special training should be necessary
- Users easily know how to use system
- System make info clear
- Low learning time

↳ Clear and unambiguous 不含米糊 system messages

↳ User decisions should be made from a small set of options
- Don't give too many options to users
- Users wouldn't feel overwhelming of huge system
- Users feel limited system size

↳ Sufficient feedback help

Intermittent Users

↳ Consistent system

↳ System provides good help facilities and good documentation

Expert Users

↳ Brief feedback

↳ Command sequences in abbreviated and meaningful form

↳ Keyboard shortcuts

Type of requirements

① Functional Requirements : What system should do (e.g. save historical data)

② Non-functional Requirements. How the system works (e.g. usability goals, response time)

Gathering Data for Requirements

① Interviews

- Determines the effect of experience of using the current version of products to users
- Determine how users behave and think about things
- Determines how the products fit into users' live.
- Determines goals and motivations for using products
- Determines mental model (how users think about their jobs & activities, expectation about the products)
- Determine problems & frustration with the products

② Focus groups

- Choose representative users
- Gather together in a room
- Ask structured set of questions
- Provide structured set of choices

③ Card Sorting

- Understands how users organize information & concepts.
- Ask users to sort a deck of cards, each contains a piece of functionality or information related to the product or website.
- Determine user's mental model

④ Questionnaires

- Pros :
  - ↳ Obtains feedback from the point of view of the user
  - ↳ Trustworthy feedback from users if questionnaire is reliable (like-minded people give same results in similar circumstances)
- Cons :
  - ↳ Requires time and effort to set reliable questionnaire

⑤ Observation

- Investigator view users as they work in field study
- Investigator takes notes on the activity
- Direct : Investigator is present during task.
- Indirect : Investigator watch video recording of the task.
- Useful for studying currently executed tasks and processes.

* Direct observation might alter users' behavior as the presence of observer.

Persona

- User models that are represented as specific, individual human beings.
- Based on real-world observation.
- What you observe?
- About what frustrate or satisfy a customer
- About skills, attitude, motivation, environment and goals
- Presenting behavior patterns

Types of Persona

Marketing personas

- Focus on demographic information, buying motivations and concerns, shopping or buying preferences
- Determine what type of customers will be receptive to certain products or messages

Proto-personas

- Used when there is no money or time to create true research.
- Mostly based on assumptions / secondary research

Design personas

- Focus on user goals, current behavior and pain points
- Based on field research and real people.

Identify behavior variables

- Activities : What user does
- Attitudes : How user thinks about product domain and technology
- Motivations / Goals : Why user is engaged in product domain
- Skills : User's capabilities related to product domain and technology
- Pain points / frustration

Scenario

- Use scenario to represent the task.
- Describe how users accomplish tasks
- Describe environmental setting
- Includes agents or actors (persona)

Environment Analysis

- Scenarios describe the broad context in which usage patterns and include environmental considerations.

- Focus on high-level actions from user's perspective for identifying user requirements and design appropriate interactions and interfaces

Address questions :

- In what setting (environment) will the product be used?
- Will it be used for extended amounts of time?
- Is the persona frequently interrupted?
- Are there multiple users on a single workstation or device?
- With what other products will it be used?

Task analysis

- Investigate an existing situation

- Focus on :
  - ↳ What are people trying to achieve? (goals)
  - ↳ How are they going about it? (how they achieve)

- Modelling technique : Hierarchical Task Analysis (HTA)

- Focus on physical and observable actions.

Hierarchical Task Analysis

- Break task down into subtasks, then sub-sub-tasks and so on.
- Focus on physical and observable actions
- Look at actions not related to software or an interaction device

HTA in list form:

0. in order to clean the house
    1. get the vacuum cleaner out
    2. get the appropriate attachment
    3. clean the rooms
        - 3.1. clean the hall
        - 3.2. clean the living rooms
        - 3.3. clean the bedrooms
    4. empty the dust bag
    5. put vacuum cleaner and attachments away

Plan 0: do 1 - 2 - 3 - 5 in that order. when the dust bag gets full do 4
Plan 3: do any of 3.1, 3.2 or 3.3 in any order depending on which rooms need cleaning

Steps:

① Identify the task to be analysed.

② Break this down into subtasks. These subtasks should be specified in terms of objectives and, between them, should cover the whole area of interest.

③ Draw subtasks as a layered diagram ensuring that it is complete.

④ Decide upon the level of detail into which to decompose.

⑤ Continue the decomposition process, ensuring that the decompositions and numbering are consistent.

Uses of Task Analysis in HCI

- To understand how users perform their current job with existing system
- Arrange new system in a way that is compatible with user's accustomed 习惯 behaviors.
- Transfer skill learnt from the old system to new system.
- User doesn't need to alter his/her thought significantly to use the new system.
- Produce documentations and training materials

C3 : Users : Physical Capabilities

Information is received and responses given via :

- Visual channel
- Auditory channel
- Haptic channel
- Movement

Vision

- mechanism for receiving light and transforming it into electrical energy.
- light reflects from objects
- Peripheral vision
  ↳ Far better at perceiving movement (detect danger)
- Context is used to resolve ambiguity 歧义
- Determine the size and depth of objects (sometimes it might cause optical illusions due to over compensation)
  视觉错差
- Reading
  ↳ Words can be recognized quickly as single characters
  ↳ Familiar words can be recognized using word shape.

↳ Capitalized words will slow down reading speed and accuracy.

Hearing

- Locate things
- Provide feedback (focus on one thing and wait for its response)
- Attract attention (focus on second thing. Once the first thing respond, your attention will go to the first thing)

Use sound for FEEDBACK and attracting ATTENTION when:

- Info short and simple
- Immediate response is needed (alert / warning)
- Visual system overburdened

- User is moving
- Poor illumination 光线不足

* Sound can be annoyance / distraction

Haptic (Touch)

- Provides feedback about environment.
- Essential for people who are visually impaired. 視障人士
- Affect comfort and performance.

Movement

- Human are sensitive to movement.
- Movement may be different depending on age / fitness.
- Response time :
    - ↳ 0.1 sec : feel immediate reaction
    - ↳ 1 sec : feel system delay
    - ↳ 10 sec : feel slow computer response

Movement : Fitts Law

- Distance between user and target object ↓, movement time ↓
- Width of target object ↓, movement time ↓
- e.g. wheel menu, list menu, desktop, OS desktop taskbar

* edge / corners have "infinite" width.

C4 : Users : Cognitive Capabilities

Cognition

- Describe the interpretation of information in the mind.

- Process :

    1. Receive info (eyes, ears, touch)
    2. Interpret
    3. Plan action
    4. Execute action

Core Cognition Aspects

Attention

- Allow us to focus on information which relevant to our actions. (involves audio / visual senses)

- Info at interface should capture users' attention.

- e.g. colour, sound, flashing lights

Design Guidelines for Attention

- Make info noticeable
- Use techiques to make things stand out (eg. colour, ordering, spacing, underlining)
- Avoid cluttering (messy) interface
- Menu choices / info on screen arranged in logical way

Perception

- Ability to see, hear or become aware of something through senses.
- To design representations that are perceivable
- e.g. text is clear to read, icons can be easily read and distinguished

Design Guidelines for Perception

- Information should be represented in perceptible and recognizable way.
- Users are able to understand the icons / graphical representation meaning.
- Users can distinguish them.
- e.g. bordering, spacing, text font style, contrast (bg & fg)

Gestalt Law of Perception

- Mind understands external stimulus as whole rather than the sum of their parts.
  看呈现出来的整体，而不是单个物体

- Group of grouping laws created via aspects:
  - ↳ Proximity 距离/间隔
  - ↳ Continuity
  - ↳ Similarity
  - ↳ Figure-ground

Proximity

- Objects which are closed together in space or time are perceived together.
- If items in UI are not intended to be "read" as a group, we should make enough space between them.

Continuity

- Perceive smooth, continuous patterns instead of disjoint and interrupted one.
- Our mind fills in missing segments / incomplete form automatically.

Similarity

- Elements that appear similar are perceived as one unit.
- Useful for grouping / categorizing elements

- e.g same color / shape

Figure-Ground

- The whole is greater than sum of its parts.
- Our mind separates visual field into foreground and background.

* Adjusting color / contrast of bg / fg, people may perceive two definitely different things.

* People can also differentiate the main dialog via making shadows, contrasts.

Memory

- Sensory memory (SM)
- Short-term memory / working memory (STM/WM)
- Long-term memory (LTM)
- Overtaxing on memory → inefficient, anxiety, stress
- Sensory Memory: Area of conscious memory that deals with information from senses (eyes, nose, haptic, ears)

Short-Term Memory (STM)

- Area of memory which hold limited information for very short period of time
- Limited capacity (7 ± 2 pcs of info)
- Info can be accessed rapidly
- Info decay (lost) rapidly

Long-Term Memory (LTM)

- Area of memory which hold and retrieve infinite information over very long periods of time
- Infinite capacity
- Need time to retrieve info
- Info less accessible after long period of time

Recognition VS Recall

- command-based interface needs users to recall name / command
- GUI provides visual-based options
- People are very good at remembering visual things
- Recognize things > Recall things
- Remember images > Remember words
- GUI > CLI

Chuncking 组块

- Grouping together pieces of information into sections
- More can be remembered
- Miller's magic number : 7 ± 2

Mental model

- how thing works ?

- Allow people to predict how things work

- Knowledge often described as mental model :
  - ↳ how to use the system (what to do next)
  - ↳ what to do with unfamiliar systems or unexpected situations (how system works)

* People uses past experiences to deal with unfamiliar systems

Design for Mental Model

Metaphors (Transfer Effect)

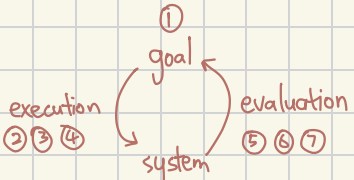
- Describe unfamiliar concepts or systems in terms of familiar elements
- e.g. use familiar element terms to describe something abstract, "files" & "folders" in OS.

- Limitation :
  - ↳ Does not have well scale. (metaphor only works well for simple process in simple programs)
  - ↳ Might cause misleading due to different culture / background of user and designer (white = funeral (china) but = wedding (America))

C5 : UI Design Guidelines & Principles

Interaction Model : Donald Norman's model

- Major phases : execution and evaluation
- Seven stages :
  1. user establishes goal
  2. formulate intention 意, 圖
  3. specifies actions at interface
  4. executes action
  5. perceives system state (understand signal)
  6. interprets system state (user know what's happen via signal)
  7. evaluates system state with respect to goal

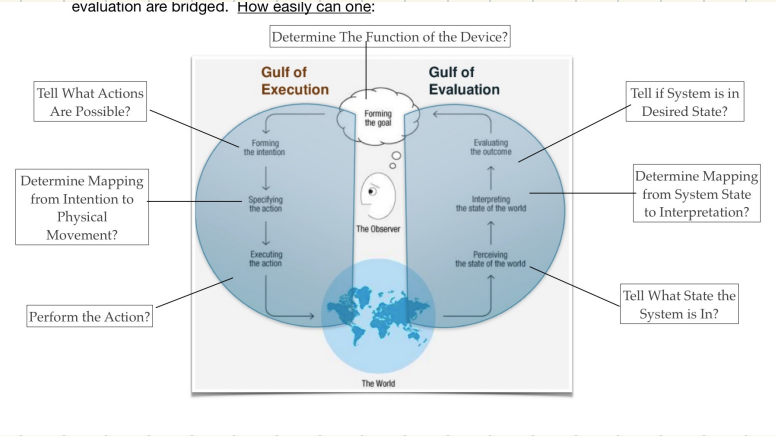


* Keyword
- ↳ Goal (what to achieve ?)
- ↳ Intentions (using what method ?)
- ↳ Actions

Reason of systems are harder to use :

- Gulf of Execution : Action formulated by users ≠ Action allowed by system.
- Gulf of Evaluation : User's expectation (system state) ≠ actual presentation of this sate (from system)

It is distance between the physical presentation of the system state and the user's expectation.

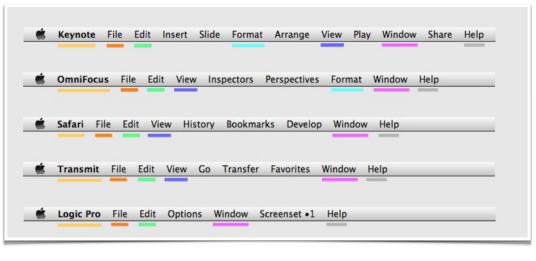


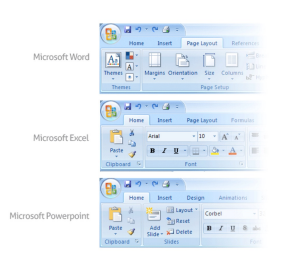
UI design guidelines & principle can:
- Guide UI design
- Evaluate UI design

Shneiderman's Eight Golden Rules

① Strive (争取) for consistency

Consistent sequences of action should be required in similar situations; identical terminology should be used in prompts, menus, and help screens.





* Don't simply change the layout if there is not necessary.
  Inherit back the previous version of layout design is fine.

✓ Easy for user to catch up the steps even though version always update

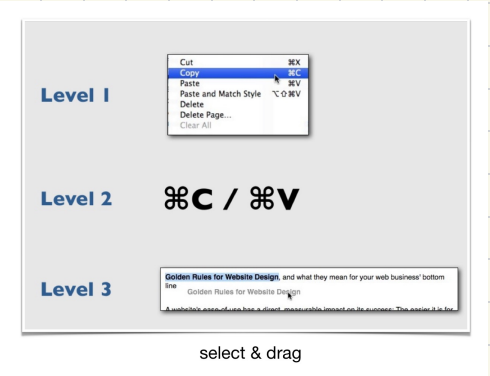
✓ Less learning cost & time for user

② Enable frequent users to use shortcuts

As the frequency of use increases, users would desire to reduce number of interaction and increase the pace of interaction. (加快交互速度)

e.g. Abbreviations (缩写), function keys, hidden commands helpful for expert user

* This might not be helpful for novice users.



select & drag

③ Offer informative feedback

For every operator action, there should be some system feedback.

For frequent and minor actions, the response can be modest 适度

e.g. clicks / taps, loading, scrolling, icon changes, animation

For infrequent and major actions, the response should be more substantial 实质性

e.g. page changes, color changes, alert windows

✓ System show meaningful, clear reaction

✓ Immediate feedback from system (look fast)

✗ Messy / long feedback message (Misleading users)

✗ Insignificant feedback (can't be noticed by users)

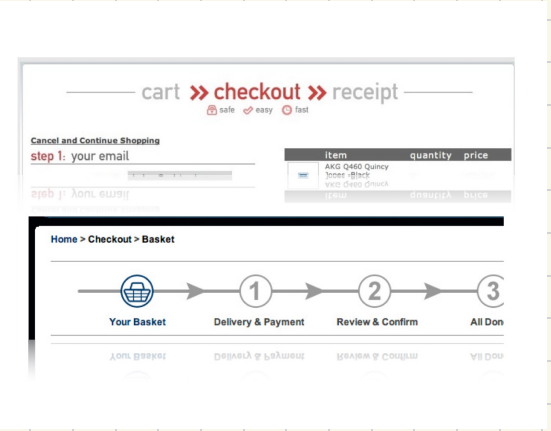
④ Design dialog to yield closure

Sequence of actions should be organized into groups with a beginning, middle and end.

✓ Users can view the progress / completion status feedback for an action (provide satisfaction and sense of relief)

✓ Users can clearly know what is going on in the system.

e.g. Progress bar showing the image uploading progress, progress bar which helps users to track their parcels delivery status



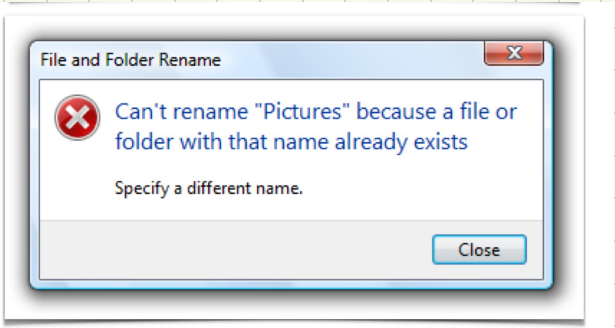
* Encouraged to design interactive dialog to attract users' attention for longer time

⑤ Offer simple error handling

If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling error.

* Don't give the message which is difficult to understand such as professional terms or programming language.

* Just let users know what is happening and how to solve it using simple sentence.

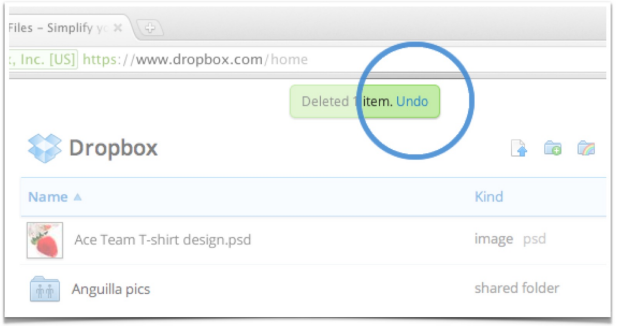


⑥ Permit easy reversal of actions

If users made some mistakes, they can easily undo their operations.

✓ Relieves anxiety

✓ Allow more exploration of unfamiliar options



⑦ Support internal locus of control

Experienced operators desire the sense that they are in charge of the system and the system responds to their actions.

Design the system to make users the initiators of action rather than the responders.

> 用户：我要我觉得，不是你觉得。
>
> 用户：系统按照我的需求走，而不是我按照系统的方式走。

e.g. Users can decide when to update the system, instead of system decides when to start updates.

⑧ Reduce short-term memory load

Human can only process info in short-term memory when the displays are simple.

e.g. Chunking 分块, recognition rather than recall

✗ Multiple page displays

✗ Long messages / text

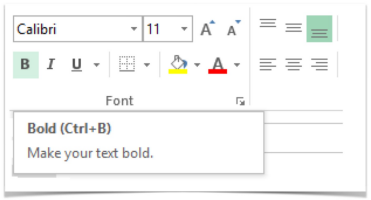
Norman's Design Principles

① Visibility

Refers to visible of function and clear controls or information.

The more visible the functions, the more likely the users will know what to do next.

X Users will be difficult to find and know how to use if they can't see the functions.



② Feedback

Send back information about what action has been done.

e.g. Sound, highlighting, animation, combinations of these

③ Constraints

Restrict allowed user interactions / action at specific moment

✓ To prevent invalid data entering

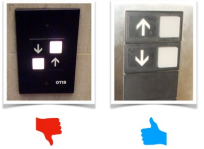
✓ To prevent invalid actions performed

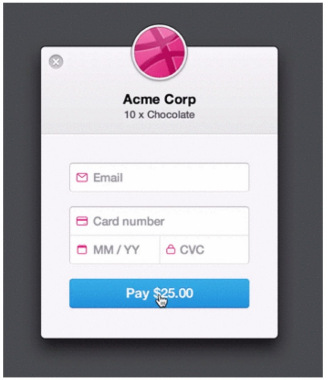
e.g. Software
- set length limit for text field of tel no.
- disable invalid date when users are choosing date for booking movie ticket.

Hardware
- Design of USB drive / HDMI cable head to prevent users from plugging the cable with a wrong side.

④ Mapping

Relationship between controls and their effects in the world.



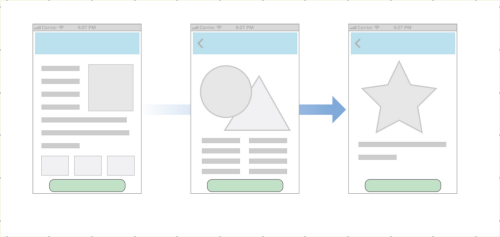


⑤ Consistency

Similar tasks with similar operations should apply similar interface design.

✓ Enhance readability

✓ Improve learnability



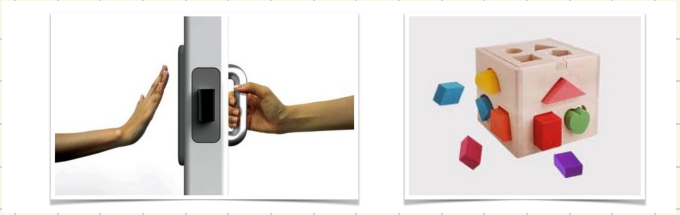
e.g. browsing different product pages in online shopping platform.

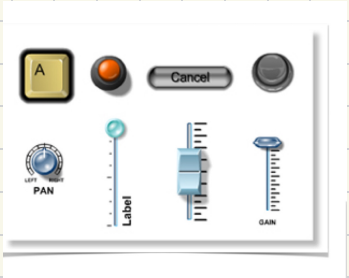
⑥ Affordance

An attribute of an object that allows people to know how to use it.

* Human will deal with unfamiliar objects based on their previous experience and knowledge.

e.g. Human see a table, they know it is made from wood because they feel the texture and color. They can pour water / put heavy things on the wooden table, they know the table is strong enough to carry the load.





Techniques to draw user's attention

① Blinking / Flashing

✓ Urgent messages only

✓ Strongly and effectively draw attention

✗ Not suitable for long message

✗ Annoying if frequent use

e.g. cars' signal lamp, phone light blinking when there is incoming calls

② Bold

Make something stand out

e.g. Report title / caption

③ Underlining

Simple way to make a text stand out on a display

* Use it carefully (might reduce readability)

④ Sound

✓ Suitable for error messages

✗ Can be annoying
(System should allow users to turn off the option)

e.g. Errors occured in Windows 11.

⑤ Colour

✓ Highlight differences between information

✓ Group information

✓ Code simple messages (error / successful message)

✓ Indicate status

e.g. Red = stop, hot, danger, fire

Errors

2 types :

↳ Slip : User understands system and goal, has correct formulation of action but incorrect action.

e.g. choosing Quit when the user meant to choose the command next to it.

closing a window when the user meant to resize it.

↳ Mistake : An incorrect action is taken based on incorrect decision (User may not even have right intention)

Users don't understand the actual purpose of the system.

e.g. using MS Word to design name card.

Solution :

For Slip : Better interface design

① Provide sufficient separation between elements to be selected.
② Minimize typing by using menus.
③ Provide clear feedback on system's current state.

* Slip happens at "executes action" stage.

For mistake : Better understanding of system

* Mistake happens at "formulates intention" stage.

Error Message Design

① **Explicit**

Indication that something has gone wrong.

② **Human-readable**

Express clearly in plain language using words, phrases and concepts familiar to user.

X No specific / professional terms

* Users can understand the content, it is not the problem with the language used.

③ **Polite**

Don't use phrases that blame or imply that users are either stupid or doing something wrong.

④ **Precise**

Describe exact problem instead of general / unclear message.

⑤ **Constructive**

Provide advice / solution how to fix the problem.

C6 : The Computer & Interaction Style

Input Devices

- Keyboard
- Stylus
- Microphone
- Mouse
- Trackball
- Touch screen
- Camera

Output Devices

- Computer Screens (Cathode Ray Tube (CRT), Liquid Crystal Display (LCD), Touch Screen, OLED)
- Speakers (Speech output, Audio)
- Braille Display
- Haptic Feedback

Keyboard

common input device for entering text

QWERTY

- keypress closes connection, causing a character code to be sent
- allow rapid entry of text by experienced users
- QWERTY layout is not optimal for typing
- Layout due to typewriters
- biased towards left hand

DVORAK

- common letters under dominant fingers
- common combinations of letters alternate between hands
- biased towards right hand
- 10 - 15% improvement in speed and reduction in fatigue

Alphabetic

- keys arranged in alphabetic order
- not faster for trained typists.
- not faster for beginners
- mainly used in handheld devices.

Interaction Types

Instructing

- issuing commands and selecting options
- users instruct a system and tell it what to do

Benefits:

- supports quick and efficient interaction
- good for repetitive kinds of actions performed on multiple objects

When to use:

- Repetitive tasks
- e.g. spell-checking, file management

Conversing

- interacting with a system like having a conversation
- underlying model of having a conversation with another human

Benefits:

- allow users, especially novices and technophobes 技术恐惧者, to interact with the system in a familiar way
- makes them feel comfortable, at ease and less scared

Problems:

- System might not able to understand or parse what users say due to pronounciation / one sentence with mixing languages

When to use:

- children
- computer-phobic
- disable users
- specialised application (e.g. phone services)

Manipulating

- Interacting with objects in a virtual or physical space
- Involves dragging, selecting, opening, closing and zooming actions on virtual objects
- Direct Manipulation

Why Direct Manipulation?

- Users experience less anxiety
- Users gain confidence and mastery and feel in control

Disadvantages:

- Not all actions can be done directly

When to use:

- 'doing' types of tasks
- e.g. designing, drawing, flying, driving, sizing windows

Exploring

- Moving through a virtual environment
- Similar to how people browse information with existing media
  e.g. multimedia, web

Interface Type

Command Line Interface

↳ original / traditional style human computer interface
↳ types in command using artificial language with its own 语义 semantics, vocabulary & syntax

Advantages:

↳ powerful, fast & efficient
↳ flexible & user controlled
↳ fast for experts

Disadvantages:

↳ hard to learn & remember
↳ invisible enhancement
↳ requires typing skill
↳ hard for novice users

Direct Manipulation

↳ directly manipulate objects at the interface using pointer device
↳ e.g. mouse, stylus, fingers
↳ replace the need of entering command via keyboard

Advantages:

↳ can immediately see if their action is furthering users' goal. If not, can simply change the direction of their activity.
↳ visible system state
↳ easy to reverse actions

Disadvantages:

↳ not all actions can be done directly
↳ requires graphic display and pointing devices
↳ slow in handling repetitive tasks compared to CLI.

Menu

↳ list of options which user selects the desired choice

Advantages:

↳ visible option

↳ less recall → easy to use

↳ rely on recognition using meaningful names

↳ visible enhancements → shown on the screen if added new functions into system

Disadvantage:

↳ inflexible → menu forces user through set sequences of steps

Menu Design Guidelines:

↳ graying out inactive menu items

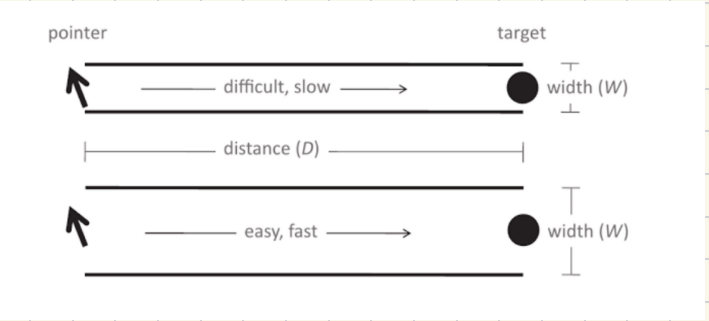
↳ use familiar terminology but all items must be distinct from one another

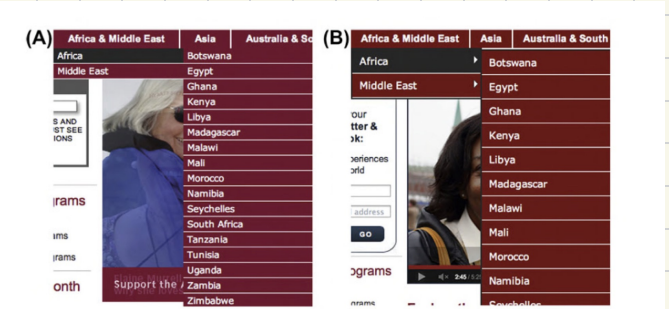
↳ ordering menu items:

- ↳ alphabetical
- ↳ categorical
- ↳ conventional
- ↳ frequency of use

Accot-Zhai Steering Law

- A narrow and long path increases selection time.
- Conversely, a wider and short path reduces selection time.





A) Narrow menu items B) wider menus items

Fill-in Form

↳ screen like paper form

↳ contain user input fields

↳ contain label for each field

↳ useful in data entry application

Advantages:

↳ efficient use of screen real estate (can ask many questions)

↳ provide context (fields)

↳ visible enhancements

Disadvantages:

↳ requires typing skill

↳ more user errors

↳ require knowledge of special keys (e.g. Tab, Enter, Backspace)

Fill-in Form Design Guidelines

- Organisation & layout
  - ↳ organize form to support task
  - ↳ online version & paper have similar layout
  - ↳ utilise spaces to create symmetry and balance.
  - ↳ separate logical groups
  - ↳ reduce gap between label and field

- Caption & field design
  - ↳ adjust label position for better user readability
  - ↳ provide distinctive section headings
  - ↳ indicate optional & mandatory fields (*).
  - ↳ avoid overuse of uppercase letter → readability
  - ↳ alignment of numeric fields (money, decimal points, telephone number, IC No)

- Error handling
  - ↳ show meaningful error messages

- Completion signal to support user control
  - ↳ provide clear messages to users about what they must do after filling form.

C7 : Prototyping

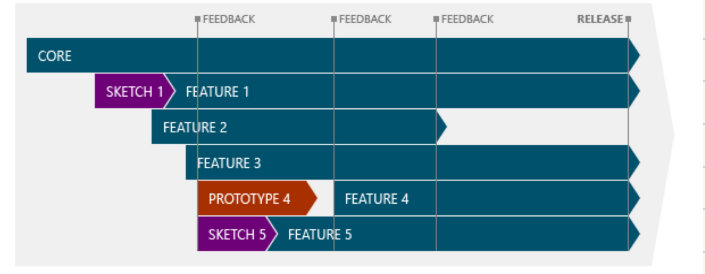
- Prototype is a primitive 原始 version of a system

Types of prototypes:

- throw-away
- incremental
- evolutionary

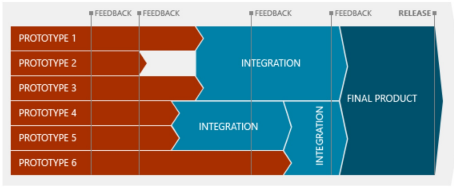
Throw-away prototype

- prototype is built & tested.
- design knowledge gained from this exercise to build final product
- actual prototype is discarded



Incremental prototype

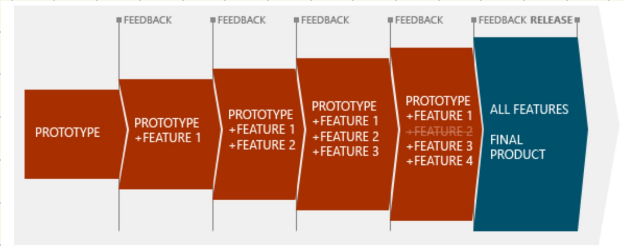
- Final product is built as separate components, one at a time
- Only one overall design for the final system, but it is partitioned into independent and smaller components.
- Final product is released as a series of products, each subsequent release including one more component.



Incremental prototyping consists of building several prototypes, each one representing a part of the future system, and then combine them.

Evolutionary prototype

- Prototype is not discarded and serves as the basis for the next iteration of design
- Actual system evolves from a very limited initial version to its final release.



Why prototype?

- Evaluation and feedback
- Stakeholders can see, hold, interact with prototype
- Team members can communicate effectively
- Can test out own ideas
- Examines the structure of design
- Prototypes answer questions and support designers in choosing between alternatives

Prototype

- Design technique where users can be involved in testing design ideas by using experimental & incomplete designs known as prototypes.

Examples of prototypes:

- Hand-drawn prototype
- Tool-drawn prototype
- Screen prototype
- Functional prototype

Hand-Drawn Prototype

- designer uses paper & pencil to draw screens
- user "enters data" by writing in the fields with pencil
- "click" on buttons and menus using pencil

Tool-Drawn Prototype

- designer draws screens on computer using the same tool which will be applied in final product
- software tools example: MS Access, Visual Basic
- print the screens & uses them in the same way as hand-drawn prototype
- Pros: look more real than hand-drawn prototype

Screen Prototype

- use computer to show screens but have little functionality
- user may enter data into some fields but nothing happen when pushes button or select item menu
- simulate response by:
  - ↳ using "secret keys" to open & close specific windows
  - ↳ putting yellow stickers on the screen with necessary data

Functional Prototype

- similar to screen prototype but many buttons and menu items have functions
- may update data & bring data forth from real database

Low Fidelity Prototypes

- paper-based
- range from a series of hand-drawn mock-ups 手绘模型 to printouts
- does not look very much like final product
- uses materials that are different from the intended final version
- used during early stages of development
- quicker to create
- enable early visualization of alternative design solutions.

Example of low-fi prototypes:

- Sketching
- Storyboards
- Wireframes
- Index cards
- "Wizard of Oz"

} detail explanation in lecture note

High Fidelity prototype

- As close as possible to a true representation of user interface
- looks more like final system in appearance
- more effective in collecting true human performance data
- common development environments:
  - ↳ Adobe XD
  - ↳ Visual Basic
  - ↳ Photoshope
  - ↳ Xcode / Swift

Advantages & Disadvantages of low and high fidelity prototype

Type	Advantages	Disadvantages
Low-Fidelity	-less time & lower cost -evaluate multiple design concepts -useful communication device -address screen layout issues	-limited usefulness for usability tests -navigational and flow limitations -facilitator-driven -poor detailed specification
High-Fidelity	-partial/complete functionality -Interactive -user-driven -clearly defines navigational scheme -use for exploration and test -marketing and sales tool	-time-consuming to create -inefficient for proof-of-concept designs -blind users to major representational flaws 使用户看不到主要的表述缺陷

Compromises

2 common types of compromise:

- 'horizontal'
  - provide a wide range of function, but with little detail (broad scope, limited functionality)
    覆盖多种功能，但是每一个都未深究

- 'vertical'
  - provide a lot of detail for only a few functions (narrow scope, deep functionality)
    覆盖少许功能，但是每一个都有深究

C8 : Evaluation

- Assessing the usability of a system.

Importance of Evaluation

- ensure that people can use and like the system / product
- allow designers to concentrate on real problems rather than rather than imaginary one
- suggest improvements
- ensure that problems are fixed before the system / product is shipped out

Goals of Evaluation

- Assess extent of system functionality
  - ↳ match with user's requirements
  - ↳ matching the use of system to user's expectations of task

- Assess effect of interface on user (usability)
  - ↳ consider aspects such as how easy the system is to learn, its usability and user's satisfaction with it
  - ↳ enjoyment and emotional response

- Identify specific problems
  - ↳ determine which aspects of design would cause unexpected results or confusion amongst users when it is used in their intended context

Types of Evaluation

- Formative evaluation
  - ↳ Evaluation done during design phase. It can be done by design team or by involving real users.
    (ongoing, during development, focus on improvement)

- Summative evaluation
  - ↳ Evaluation performed with finished product. It is mostly done by external users.
    (final, after development, focus on assessment of success)

Types of Methods

- Expert analysis
  - ↳ to identify any areas that are likely to cause difficulties
  - ↳ used at any stage in development process
  - ↳ cheap, since not require user involvement
  - ↳ does not assess actual use of system, only checks whether a system fulfils the accepted usability principles.

- User participation
  - ↳ used in later stages of development when there is at least one working prototype of system in place

Styles of Evaluation

- Laboratory studies
  - ↳ users are taken out of their normal work environment to take part in controlled tests, often in a specialist usability laboratory

Pros:
- ↳ specialist equipment available
- ↳ uninterrupted environment

Cons:
- ↳ lack of context
- ↳ difficult to observe several users cooperating

Use it when:
- ↳ system location is dangerous
- ↳ very constrained 有限 single-user tasks may be adequately 充分 performed in laboratory
- ↳ compare alternative designs within a controlled context
- ↳ deliberately 有意. want to manipulate the context to uncover 揭露 problems or observe less used procedures

Field Studies

Pros:

↳ natural environment

↳ context retained

Cons:

↳ distractions

↳ noise

Use it when:

↳ where context is crucial

Observation + Think Aloud

- Observation : user observed performing task
- Think Aloud : user asked to describe what he is doing and why, what he thinks is happening

- **Advantages:**
  - ↳ simplicity → requires little expertise
  - ↳ can provide useful insight
  - ↳ can show how system is actually use

- **Disadvantages:**
  - ↳ subjective, selective (user may think before describing)
  - ↳ act of describing may alter task performance

- **Use it when gathering info such as:**
  - ↳ what facilities are frequently used
  - ↳ how much time is needed to complete a task
  - ↳ which interfaces are confusing & frustrating
  - ↳ how users react to error messages

- **Avoid Observer Effect:**
  - ↳ minimise distractions
  - ↳ make it clear that you are not evaluating personal performance instead evaluate the UI

Query Techniques

- Interviews
  - ↳ analyst questions user on one-to-one basis (f2f or via telephone)

Pros:

- ↳ varied 变化 to suit context (change questions based on user's reaction)
- ↳ fully explore issues
- ↳ elicits user views 征求用户意见 and identify unanticipated problems

Cons:

- ↳ interviewee can be very subjective and often reluctant to criticise openly
- ↳ time consuming

- Questionnaires
  - ↳ set of fixed questions given to users

Advantages:

- ↳ quick and reaches large user group
- ↳ can be analyzed more systematic

Disadvantages.

- ↳ less flexible
- ↳ less probing 探查
- ↳ require skill to produce a good questionnaire
- ↳ Poor returned rates

C9: Evaluation (Expert Analysis)

Expert Analysis

- fast & inexpensive
- not require use of well-equipped lab
- not require user involvement
- used at any stage in development process
- Not assess actual use of system, only check whether a system fulfils the accepted usability principles

- Techniques:
  - ↳ Cognitive Walkthrough
  - ↳ Heuristic Evaluation

Cognitive Walkthrough

- evaluate walk-up-and-use systems
- usability evaluation method in which one or more evaluators work through a series of tasks & ask a set of questions from the perspective of user
- Main focus: establish how easy a system is to learn (learnability)
- find out most severe usability problems and desirable new features / functionalities

Requirements before walkthrough:

- a prototype
- indicate who are the users and what kind of experience & knowledge they have
- description of task the user is to perform on the system / prototype

Considerations:

- is the effect of the action the same as the user's goal at that point? (goal match)
- will the user try and achieve the right outcome, (goal match)
- will users see that the action is available (visibility)
- once users have found the correct action, will they know it is the one they need? (identify the correct action with the outcome they expect to achieve)
- after the action is taken, will users understand the feedback they get? (feedback)

Heuristic Evaluation

- involves having a small set of evaluators examine the interface & judge its compliance 是否符合 with recognised usability principles
- useful for evaluating early design
- can be used on prototypes, storyboards & fully functioning systems
- evaluators indepently critique a system to come up with potential usability problems.

- Nielsen's Ten Heuristics :
  - ↳ Visibility of system status
  - ↳ Match between system and the real world
  - ↳ User control and freedom
  - ↳ Consistency and standards
  - ↳ Recognition rather than recall
  - ↳ Flexibility and efficiency of use
  - ↳ Aesthetic and minimalist design 美观简约的设计
  - ↳ Help users recognize, diagnose, and recover from errors
  - ↳ Help and documentation

Heuristic Evaluation Stages

3 stages :

- Briefing : introduce the system / prototype to the experts & tell them what to do
- Evaluation : each expert evaluate the system / prototype independently & record potential usability problems
- Debriefing :
  - ↳ experts come together to discuss their findings
  - ↳ prioritized the problems found
  - ↳ suggest solutions

Nielsen's Ten Heuristics:

- Visibility of system status
  - ↳ system should always keep users informed about what is going on, through appropriate feedback within resonable time

- Match between system and real world
  - ↳ system should speak users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms
  - ↳ follow real-world conventions, making information appear in a natural and logical order

- User control and freedom
  - ↳ system should support undo and redo since users often choose functions by mistake

- Consistency and standards
  - ↳ users should not have to wonder whether different words, situations, or actions mean the same thing
  - ↳ follow platform conventions

- Error prevention
  - ↳ eliminate error-prone conditions 消除易出错的条件
  - ↳ display confirmation option before users commit to the action
  - ↳ even better than good error messages
  - ↳ display mandatory fields, form validation only

- Recognition rather than recall
  - ↳ minimize user's memory load by making objects / actions / options visible
  - ↳ user no need to remember information from one part of dialogue to another

- Flexibility and efficiency of use
  - ↳ system can cater 满足 to both inexperienced & experienced users. Allow users to tailor 定制 frequent actions

- Aesthetic and minimalist design
  - ↳ dialogues should not contain information which is irrelevant or rarely needed

- Help users recognize and recover from errors
  - ↳ error messages should be expressed in plain language
  - ↳ precisely indicate problem & suggest constructive solution

- Help and documentation
  - ↳ information should be easy to search
  - ↳ focus on user's task & list concrete steps to be carried out

C10 : Usability Measurement

Usability

- the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use

- Effectiveness : accuracy and completeness with which users achieve specified goals

- Efficiency : resources expended in relation to the accuracy and completeness with which users achieve goals.

- Satisfaction : comfort and acceptability of use

Why need usability metrics ?

- track progress between releases.

- assess your competitive position

- are you better or worse than other companies ?

- where are you better or worse ?

- make a Stop / Go decision before launching system (is the design good enough to release ?)

How to measure?

- Task time
  - ↳ record how long it takes a user to complete a task in seconds and or minutes

- Errors
  - ↳ Record any unintended action, slip, mistake a user makes while attempting a task.

- Completion Rates / Task success
  - ↳ completion rates / task success are a simple measure of usability.
  - ↳ recorded as binary metric (1 = Task Success, 0 = Task Failure)

- Page Views / Clicks
  - ↳ fundamental tracking metrics for websites and web-applications

- Conversion Rate
  - ↳ Measuring whether users can sign-up or purchase a product

- Test Level Satisfaction
  - ↳ measuring users' satisfaction

Qualitative user testing

- ↳ 2 - 5 users
- ↳ obtain users insight on the system
- ↳ to explore users' needs, preferences and reactions
- ↳ to gather feedback about user experience, usability issues and overall impressions
- ↳ to allow designers to quickly iterate based on feedback gathered from interviews, observations & usability test
- ↳ used at early design stages

Quantitative user testing

- ↳ large sample size
- ↳ validate design decisions by measuring performance, satisfaction & task completion
- ↳ identify trends, effectiveness and usability issues
- ↳ used at later design stages

C10 : Ergonomics , Health & Safety

Ergonomics

- the study of human beings in relationship to their working environment and the engineering of that environment for comfort, efficiency and safety

Goals of Ergonomics

- to make the work more comfortable
- to improve workers production
- to improve both health and safety

Impact of well designed working environment

- happier & healthier
- produce better work
- quit job less frequently

Impact of poorly designed working environment

- Repetitive strain injury (RSI)
- Eyes disorders
- Back & shoulder pain

The Office Environment

- Lighting
  - ↳ adequate lighting should be provided
  - ↳ more detail the user can see
  - ↳ less discomfort or eyestrain

- Noise level
  - ↳ maintain noise at comfortable level
  - ↳ stress / high blood pressure / poor concentration if excessive noise

- Temperature
  - ↳ keep office temperature in comfortable range
  - ↳ extreme high or low temperature affects performance

- Working places
  - ↳ top of screen should be positioned at eye level with specified distance from face
  - ↳ keyboard should be kept at elbow height

- Tables & Chairs
  - ↳ seat should be padded for comfort
  - ↳ enough space under the desk
  - ↳ smooth and rounded edges of tables