Why usability?

- Cost savings

  - Telephone company: redesign phone interface
    speed up dialing, 0.15 sec/digit → $1 million savings from reduced demand on switches.

  - Insurance company: redesign application forms
    reduce customer errors, original forms had avg. 7.6 errors per form & 1 hr to fix errors.
    Saving from new form A$ 536,023 (decrease support costs)

  - Boeing 757 flight deck redesign
    operation by 2 pilots instead of 3

- Increase membership/sales

  - Efficiency to accomplish task
  - Easy to use / interface

Amerritrade focus on website performance, user base 400,000 to 1.4 million
IBM sales increase by 400% with easier to navigate website
What is usability?

Initially idea was user friendly, not quite accurate

- machines don't need to be friendly, they just need to stand in the way of getting task done
- not 1-dimensional metric (more friendly vs less friendly)

New terms introduced

CHI → computer-human interface
HCI → human-computer interface, preferred since human is first
UCD → user centered design
MMI → man-machine interface
OMI → operator-machine interface
UID → user interface design
HF → human factors
Ergonomics

Usability used as a broader term to describe the quality of a user's experience in interacting with a system/product and ability to achieve specified goals.
Figure 1 A model of the attributes of system acceptability.

How to measure usability

- Learnability: ease to learn so user can rapidly start getting work done
- Efficiency: once user learns how to use system, can have high level of productivity
- Memorability: casual user should be able to return to system after some period of inactivity without having to completely re-learn everything
- Errors: low error rate while using system, easy to recover from errors, no catastrophic errors
- Satisfaction: pleasant to use

Also part of a larger picture, many times these concerns can be competing (ie trade offs)
By having precise and measurable components arrive at engineering discipline where usability systematically approached, improved, evaluated.

**Learnability**

fundamental attribute of usability.
most people's first experience with new system is learning to use it

![Learning Curve Graph](image)

**Learning curve.**

highly learnable systems have steep incline at first part of curve

walk-up-and-use systems (museum info) have zero learning time and intended to use once

transfer skills when upgrading will not have standard learning curve

**How to measure/evaluate?**

pick users who haven't used system, measure time needed to reach specified level of proficiency. (task, not complete mastery of system) testers should be representative of user group
Efficiency: of use

steady state performance when learning curve flattens out
possible to continue learning forever, most plateau once
they learn "enough."

how to test?

bring in experienced users, ask them to perform task
(⇒ how do you know if they are experienced?)
- self proclaimed, use of system for certain amount
of time.

train users for X amount of time, then measure
bring in users and keep measuring until recorded
time levels off.

Memorability

Casual user 3rd category of user, behind novice & expert.
used system before, just need to remember based on
previous experience (diff then 1st time)

⇒ utility programs, that are supplementary
activities, not part of regular work

⇒ Inherently, used at long intervals like quarterly reports

ex "kiss and Ride" outside of Washington DC metros.

Sign has poor learnability without assistance.
but once you know it, drop-off zone easy
to find at other stations.
how to test?
not tested as much as other metrics
• casual user away from system for predetermined time asked to perform task
• after user finished with test session, try memory test by having them explain effects of various commands, name command (draw icon)

flawed, some testing showed users cannot name menu items when away from system but able to use same system when sitting at computer.

Few & Non-catastrophic Errors.
error - any action that does not accomplish desired goal
how to test?
• measured by counting actions, can be incorporated as part of testing other usability attributes.

impact of error varies
• simple errors that can be corrected easily slow down user and can be part of efficiency measure - can skip counting these
• catastrophic - not discovered and lead to faulty work product, destroy users work should be counted differently
Subject Satisfaction

how pleasant is it to use the system?

especially important in non-work environments

home computing, games, interactive fiction, creative painting

user should have entertaining/moving/enriching experiences since there is no other goal.

how to measure?

psyche physiological measures - EEGs, pupil dilation, heart rate, (rate)

skinfold, blood pressure, adrenaline levels

asking for opinion -

(common) single response may be subjective, average from multiple responses is objective measure of system's pleasantness.

after session give questionnaire or to user if installed system not before use, answers for before & after not highly correlated.

rating closely related to peak difficulty encountered, tends to be most memorable.

questionnaires typically very short, asked to rate system based on 1-5 or 1-7 rating scales, that are normally Likert scales or semantic differential scales.
Likert scale — postulate some statement, user asked to agree or disagree.

e.g. "I found this system very pleasant to use."

1 - strongly disagree
2 - partly disagree
3 - neither agree or disagree
4 - partly agree
5 - strongly agree

Semantic differential scale — two opposite terms along some dimension, user place system on most appropriate rating along that dimension.

possible to use more sophisticated methods based on rating scale theory from sociologists & psychometrics

Please indicate the degree to which you agree or disagree with the following statements about the system:

"It was very easy to learn how to use this system."
"Using this system was a very frustrating experience."
"I feel that this system allows me to achieve very high productivity."
"I worry that many of the things I did with this system may have been wrong."
"This system can do all the things I think I would need."
"This system is very pleasant to work with."

Table 3 Questions users might be asked to measure subjective satisfaction using a Likert scale. Users would typically indicate their degree of agreement on a 1–5 scale for each statement. One would normally refer to the system by its name rather than as “this system.”

Please mark the positions that best reflect your impressions of this system:

| Pleasing | Irritating |
| Complete | Incomplete |
| Cooperative | Uncooperative |
| Simple | Complicated |
| Fast to use | Slow to use |
| Safe | Unsafe |

Table 4 Some semantic differential scales to measure subjective satisfaction with computers. See [Coleman et al. 1985] for a list of 17 such scales.
make sure to pilot test
- interpreted properly what you are asking.

need anchor or baseline to calibrate scale
- other systems/diff versions of system available, consider rating in relation to these results

- Single user, take care in interpreting results. tend to be too polite unless really unpleasant experience. If rested in system try to be more positive.

- counteract (partly) by having some questions to which agreement would be negative rating of system.
Usability activities take place throughout lifecycle of a product. Design decisions have ripple effect on subsequent products and products that need to be backwards compatible, company reputation, support, etc...

1. Know the user
   a. Individual user characteristics
   b. The user's current and desired tasks
   c. Functional analysis
   d. The evolution of the user and the job
2. Competitive analysis
3. Setting usability goals
   a. Financial impact analysis
4. Parallel design
5. Participatory design
6. Coordinated design of the total interface
7. Apply guidelines and heuristic analysis
8. Prototyping
9. Empirical testing
10. Iterative design
    a. Capture design rationale
11. Collect feedback from field use

*Table 7 The stages of the usability engineering lifecycle model.*

Don't rush into design!

Avoid redesign due to usability issues, development of parts that will not be used.

Pre-design activities can be part of market research or product planning.
1) know the user

Study, users and intended use

→ installers, maintainers, sys admin, support staff, included

Individual user characteristics & variability in tasks two largest impact factors in usability.

Access to users can be difficult

- keep developers from customer so not to bypass tech support and distract developer from job

- reluctance of sales rep to allow others to talk to "their" customers from fear of offending customer, create dissatisfaction

- highly paid execs short on time, unionized and limit time/client like to be studied

Avoid hearsay, important to get direct access

(1a) user characteristics - product going to specific dept/company/age group

Know user's work expr, education level, age, previous computer experience

Anticipate learning difficulties, limit complexity, language skills

Young audience - visual vs textual interface

Older audience - simplify interface, increase text size

Is there training?

What is the work environment like?

Audible "bleeps" distracting in open envir

Frequently disturbed - good way to switch tasks
characterize from market analysis or questionnaire
beneficial to observe in working choir - find things they may not realize or think important.

(b) Task analysis—what are the goals? how to approach task?
how to deal with exceptional circumstances? "workarounds"?

outcome is a list of all things users want to accomplish, information needed to achieve goals, steps, interdependencies, between steps, outcomes/reports to be produced, criteria to determine quality of output, communication needs to exchange info.

concrete examples to illustrate needs, supplement with interviews/observations

- Why do you do this? (large goal)
- How do you do this? (decompose subtasks)
- Why not do this in such a manner? (alternatives)
- Do errors occur while doing this?
- How to correct errors?
(1e) Functional Analysis

What needs to be done? Not just how user does these tasks.

c.x. search manual.

Users flip through pages, should program just have fast way to move through pages or should we take advantage of technology strengths and provide search to jump directly to location?

(1d) Evolution of user

Coevolution of tasks and artifacts - using the system changes the user; users change how to use system.

Typical change - novice to expert user

Want interaction shortcuts (accelerators)
2) Competitive Analysis

Existing products can be best prototyped for our products analysis. Provide usability guidelines, avoid for testing. *not steal* — we want to do better based on analysis.

3) Goal Setting

Compromise of conflicting components, which ones are most important.

For each component, diff levels of performance can be identified.

Min level for product release outlined / planned achievement / competitors.

![Diagram](image)

Figure 7 An example of a usability goal line in a notation similar to that used by Rideout [1991].
3(a) Financial Impact

# users
loaded salary (salary + pension + benefits + overhead + ...)
time using system

e.g.: develop software system for a group of 3,000 users
processing orders

loaded cost technician $25/hr

5/6 of working day

1/2 * 8 hr/day * 230 working days/yr = 629.3 hrs/yr

$25/hr * 3,000 users * 629.3 hrs/yr = $47 million

annual financial impact

assume 2 years before introduction, used for 4 years before upgraded/redesigned, 10% deflation

<table>
<thead>
<tr>
<th>yr1</th>
<th>yr2</th>
<th>yr3</th>
<th>yr4</th>
<th>yr5</th>
<th>yr6</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>42.3</td>
<td>39.07</td>
<td>34.26</td>
<td>30.837</td>
<td>27.753</td>
</tr>
</tbody>
</table>

\[ \sum_{yr 3} - \text{yr} 6 = 129 \text{ million} \]

(drop decimal)

total impact
cut down learn time by 1 day.

\[ 4.25 \text{ hr} \times 3,000 \text{ users} \times 8 \text{ hr/day} = \$600,000 \]

\[ (600,000) \times .9 \times .9 = \$486,000 \text{ (250K in book)} \]

Alternatively, we increase productivity by 10%.

\[ 3.8 + 3.4 + 3.0 + 5.7 = \$12.9 \text{ million} \]

(remember deflator & 2 yr delay.

calc shows its worth more to improve productivity
vs. training time.

Financial impact for open market
- impact on development organization (incl. service to customer)
- impact on user organization
Parallel Design

explore design alternatives before settling on single one
which is developed further

Figure 8 Conceptual illustration of the relation between parallel and iterative design. Normally, the first prototype would be based on ideas from several of the parallel design sketches.

designer spend few hours or days for initial design
- work individually to generate more ideas
usually find at least 2 fundamentally different ways to approach problem
- pursue a little further in usability evaluation
similar designs combined to take best of each

diversified parallel design - different design focus on different aspects of design problem
  (novice interface vs expert vs nonverbal)
5) Participatory Design

even though take time to "know the user" cannot
know user well enough to answer all issues that come up.
user also come up with question developers/designer
may not have thought of

SME (subject matter experts) users involved in design
process
not for designing interface
good for reaching to concrete designs/paper mock-ups/
screen shots

refresh pool of users
user changes w/ use not representative of avg population
too much time becomes willing to accept developer
ideas though akward

6) Coordinating Total Interface

consistency! important usability characteristic
across UI, documentation, online system,
video tape tutorials, across new releases
or product families

interface standards helpful, dictionary to outline
proper terminology, coordinator or representative, existing exs.
not an excuse to propagate bad design.
7) Guidelines & Heuristic Evaluation

- List well known principles for UI design
  - General: all interfaces
  - Category-specific: type of system (data proc, keypad, ...)
  - Product-specific: specific product

Ex. "Provide feedback" guideline is general

In graphical user interface, ensure main objects of interest visible on screen/most important attributes shown (category specific).

Product specific: have each file and subdirectory represented by icon.
Use different icons for different file types.

Standards vs guidelines.
- Standards specify how interface appears to user.
- Guidelines provide advice about usability characteristics of interface.

Ex. Guideline: provide way to back out of undesired system state.
Standard: "undo" command need to be available at top right of screen. "Escape" key provides undo mechanism.

Heuristic evaluation:
Looking at interface & come up with good/bad based on guidelines provided.
Develop list of items to address.
8) Prototyping

- not full scale implementation
- used to get better understanding of system so final system can be developed more fast at cheaper
- user interface better than with abstract specs.

Figure 9 The two dimensions of prototyping: Horizontal prototyping keeps the features but eliminates depth of functionality, and vertical prototyping gives full functionality for a few features.

How to save cost/time?
- cut down on number of features - vertical prototyping in depth implementation for a few features
  test in realistic circumstance with real tasks.

- reduce level of functionality - horizontal prototyping includes entire interface but no func, just a simulation of tasks (no real work done)
- less efficiency (larger code size, slower response time)
- lesser quality code (bugs/crashes)
- simplified algorithms (ignore special cases)
- "wizard of OZ" method
- low-fidelity media (scan images vs. live video)
- fake data
- paper mockups
- imaginary prototypes (describe possible interface orally w/ what if questions)

"forward scenario simulation"

**Scenarios**

Ultimate minimal prototype in a single interaction session w/o any flexibility for user

Includes:
- individual user
- set of computer facilities
- achieve a specific outcome
- under specified circumstances
- over a certain time interval

Can be used w/ mockups to get more detail
Interface Evaluation

The best way to evaluate is to just do it.

Classify evaluation methods:
1. Real users or not
2. Interface implementation or not

Outcome is a list of usability problems in the interface (maybe hints to support success).

Can't solve all, so need to prioritize by severity rating:
- 0: Not a usability problem
- 1: Cosmetic
- 2: Minor usability problem
- 3: Major usability problem
- 4: Usability catastrophe

Based on how many users expected to have the problem and extent it hurts these users.

<table>
<thead>
<tr>
<th>Impact of problem on the users who experience it</th>
<th>Proportion of users experiencing the problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>Low severity</td>
</tr>
<tr>
<td>Large</td>
<td>Medium severity</td>
</tr>
</tbody>
</table>

Table 8: Table to estimate the severity of usability problems based on the frequency with which the problem is encountered by users and the impact of the problems on those users who do encounter it.
results from testing help produce new version of interface
changes made to solve problems may fail and/or introduce new problems
changes may make interface worse for users w/o problems
by trade off analysis of how many had problem vs how many suffer from change

capture design rationale
rationale for design decisions, available for later reference
important change not undone, help tech writers or translators, maintain consistency.

Figure 10 A partial example of a design rationale for a small part of an interface design for a hypothetical color paint program. The full design rationale might include more sample screens and links to additional design questions like the “How to appropriate…” question hinted at here. The lines might denote hypertext links in an online representation, or they could be supported by simple proximity in a paper document.
11) Follow up studies of installed systems
	note: objective of usability after product release to gather
data for next version, data that not avail
in a laboratory.

may find changes over time

A US company found change in bill interface
increase 67 to 84% after time

other mechanisms to follow up:

- from user complaints, modification
requests, or call to help line

MISC

* Supplement activities with following metamodels
  methods that apply to methods
  - write down explicit plan
  - get independent reviewer to critique plan
  - pilot w/ 10-15% resources, revise plan for
    remaining

* prioritize usability activities

* be prepared
  - tools avail?
  - eval tech?