HCI in Software Development Usability Laws

Interesting and fun sources:

Sakshat vIrtual labs - Creative Design, Prototyping & Experiential Simulation In Human Computer Interaction (HCI)

- http://iitg.vlab.co.in/?sub=72&brch=170

Today: Seven Usability Laws

1. Miller’s Law of Short-Term Memory Load
2. Fitts’ Law
3. Keystroke Level Model (KLM)
4. Hick’s Law
5. Power Law of Practice
6. Pareto and Zipf Laws
7. Peak End Rule

1. Miller’s Law of Short-Term Memory Load

George Miller

Princeton Professor

“Magic Number” published in
‘Psychology Review’ in 1956

1. Miller's Law of Short-Term Memory Load

Chunking:

- A psychological phenomenon whereby individuals group responses when performing a memory task.

1. Miller’s Law of Short-Term Memory Load

Measured short-term memory in terms of “Chunks”.

Chunks are a single unit of information.

- Part of a phone number
- A name
- Etc…

Column A – Without Chunking	Column B – With Chunking
Patient ID:678290234	Patient ID 6782 9023 4
Name: Joe Jones	Name: Joe Jones
DOB:02111973	DOB 02 / 11 / 1973

1. Miller’s Law of Short-Term Memory Load

Discovered, among other things, that users are generally able to remember 7 plus or minus 2 ‘chunks’ of information.

More efficient “chunking” equals more memory!

Users typically have to practice memorizing.

1. Miller's Law of Short-Term Memory Load

Likewise, Miller concluded that humans can only categorize stimuli into about 7 categories.

- E.g., People can categorize sounds into one of 7 categories of pitch effectively, but get worse at the task very quickly after that.

What Does Miller’s Law Mean For HCI Designers?

Miller’s law does NOT mean that you should have 7’ish menu items in a drop down display.

- The law is about HOLDING items in a user’s memory.

A good example:

- Motions / # of fingers used to perform various actions on an iPad. In general, users will only be able to remember 7 +- 2 of these quickly.
- Yes, with more practice they would probably memorize more.
- E-learning applications should make liberal use of chunking to aid in end-user memorization. Chunking is also ideal in environments where an interface must compete against other stimuli for the attention or working memory of the end user (car navigation systems, cell phone, public kiosks).
- Short term memory and web usability

2. Fitts' Law

In short, states the time it takes to reach a target with a pointing device given its distance and size from the current pointer position.

Fitts' Law is an example of a predictive model because it predicts things without the need for users.

2. Fitts' Law

$$T = a + b \log_2 \left(1 + \frac{D}{W}\right)$$

T = time to move pointer to target

D = distance between the pointer and target

W = size of target

2. Fitts' Law

Things done more often should be assigned a larger button. This seems an intuitive principle, but it needs to be used very carefully, since it harms the consistency of the interface.

2. Fitts’ Law

Things done more often should be closer to the average position of the user's cursor. The Distance (D) of a widget allows more control from interface designers compared to the width (W). Again, this needs to be used with caution, since frequency-based widget arrangements may slow down the user from finding things compared to logic-based arrangements.

2. Fitts’ Law: Hmmmmmmm…

2. Fitts’ Law

The top, bottom, and sides of the screen are infinitely targetable because of the boundary created by the edges of the screen (unless a virtual screen exists). They should be fully utilized.

Fitts’ Law: Predictive Models

Fitts’ Law describes one of many predictive models.

- These models attempt to highlight how well a user will use a system without the need to find actual users!

Fitts’ Law summary

Mathematically predicts how long it will take to "acquire" a target based on its distance and size.

It takes users longer to point to links and buttons on a screen if the objects are smaller in size or farther away from the home position. This also causes more selection errors.

The size of a button should be proportional to its expected frequency of use.

- Fitts’ Law video

3. Keystroke Level Model (KLM)

A model that provides numerical predictions of user performance.

3. Keystroke Level Model (KLM)

Operator name	Description	Time (s)
K	Pressing a single key or button Skilled typist (55 wpm) Average typist (40 wpm) User unfamiliar with the keyboard Pressing shift or control key	0.35 (average) 0.22 0.28 1.20 0.08
P	Pointing with a mouse or other device to a target on a display	1.10
P_1	Clicking the mouse or similar device	0.20
H	Homing hands on the keyboard or other device	0.40
D	Draw a line using a mouse	Variable depending on the length of line
M	Mentally prepare to do something, e.g. make a decision	1.35
R(t)	System response time – counted only if it causes the user to wait when carrying out his/her task	*t*

3. Keystroke Level Model (KLM)

$$T_{\text{execute}} = T_{\text{K}} + T_{\text{P}} + T_{\text{H}} + T_{\text{D}} + T_{\text{M}} + T_{\text{R}}$$

3. Keystroke Level Model (KLM) Example

How long does it take to type this sentence:

- ***Eating a gallon of ice cream for dinner is normal.***

And change it so that it becomes:

- ***Eating a gallon of ice cream for dinner is not normal.***

3. Keystroke Level Model (KLM) Example

List all of the sub-tasks necessary to perform the action (according to the KLM).

Simply sum up the predicted times to perform each individual sub-task.

3. Keystroke Level Model (KLM) Example

Mentally prepare (M)	1.35
Reach for the mouse (H)	0.40
Position mouse before the word ‘normal’ (P)	1.10
Click mouse (P_1)	0.20
Move hands to home position on keys (H)	0.40
Mentally prepare (M)	1.35
Type ‘n’ (good typist) (K)	0.22
Type ‘o’ (K)	0.22
Type ‘t’ (K)	0.22
Type ‘space’ (K)	0.22
Total predicted time:	5.68 seconds

$2(M) + 2(H) + 1(P) + 1(P_1) + 4(K) = 2.70 + 0.80 + 1.10 + 0.2 + 0.88 = 5.68$ seconds.

4. Hick's Law

Reaction time

Describes the time it takes for a person to make a decision based on the number of choices available.

$$T = b \cdot \log_2(n + 1)$$

T: average reaction time to choose among 'n' equally probable choices.

b: constant determined empirically by drawing best fit line to measured data.

Log2: For small 'n', additional options add a lot of time with eventual diminishing returns.

+1: due to uncertainty when making selections.

4. Hick’s Law

Essentially, users quickly categorize and halve the number of choices iteratively until they select what they want.

- This is a bit of a mis-categorization, but an ok way to think about it.

‘b’ is a constant that is empirically derived.

Hicks law states that the more options a user has to choose between and if they are not grouped, the longer it will take to select an option.

4. Hick's Law

Law can be generalized to menu items with unequal probability of being selected:

$$T = bH$$

$$H = \sum_{i}^{n} p_i \log_2(1/p_i + 1)$$

4. Hick-Hyman Law

Be Careful!

- This law only applies to groups of items that can be reasonably categorized.
- For example, the law would probably not work with a randomized list of unrelated options.
- But would work for an alphabetized list of options.
  - If menu items were alphabetized

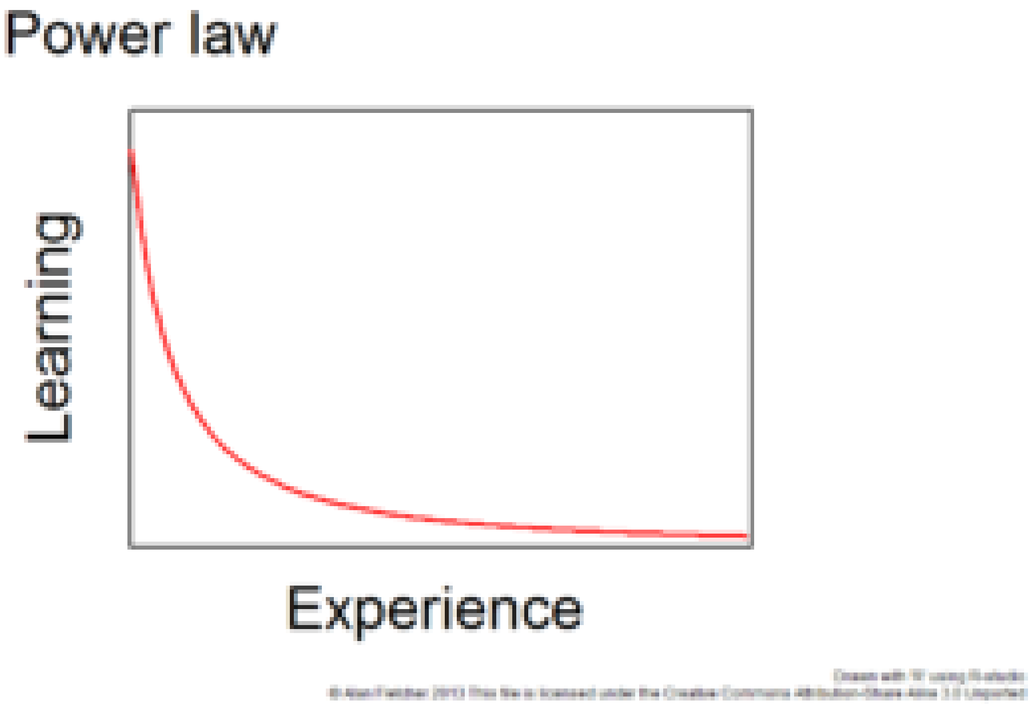
http://iitg.vlab.co.in/?sub=72&brch=170&sim=769&cnt=1

5. Power Law of Practice

The **power law of practice** states that the logarithm of the reaction time for a particular task decreases linearly with the logarithm of the number of practice trials taken. It is an example of the learning curve effect on performance.

This law is not necessarily used for guiding interface design, but it is rather a knowledge we must learn for interaction designers : a quantified measure of how our users can learn to use the user interface.

5. Power Law of Practice



6. Pareto and Zipf Laws

Vilfredo Pareto, Italy

- observed that wealth was unequally distributed in Italy. He noted that 80% of the land and wealth was owned by 20% of the people (early 1900's).
- Sound familiar?
- Top 20% of earners make 51% of all income in the U.S.

6. Pareto and Zipf Laws

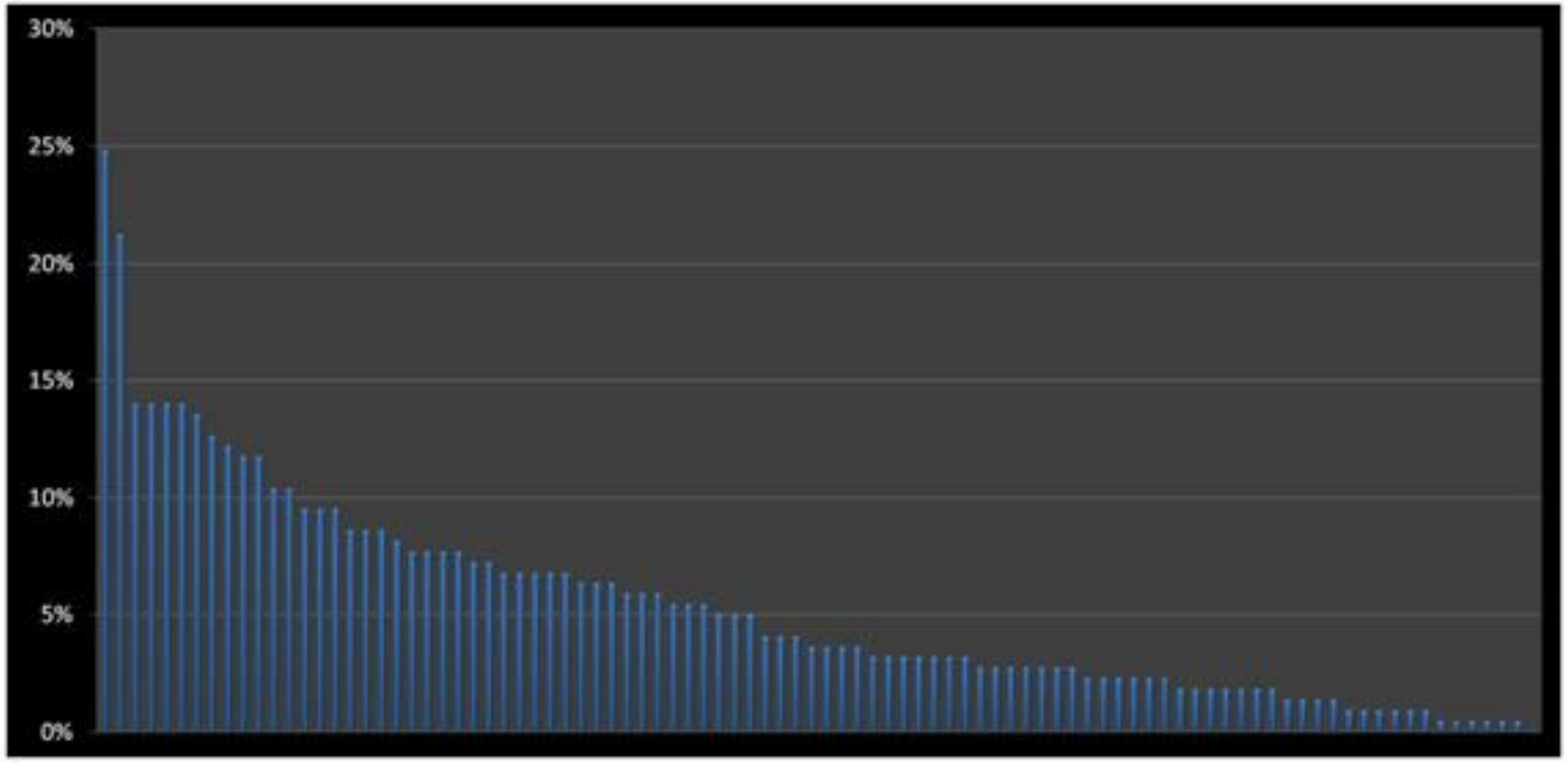
Examples of the Pareto Principle.

- Microsoft noted that by fixing the top 20% of the most reported bugs, 80% of the errors and crashes would be eliminated
- The top 10% of cell phone users consume 90% of wireless bandwidth.
- The top 20% of U.S, taxpayers pay 68% of all taxes

6. Pareto and Zipf Laws

Applications to HCI:

- Of 94 tasks (on a large informational website) presented to users, just 15 account for 41% of all votes (users asked which tasks were most important).

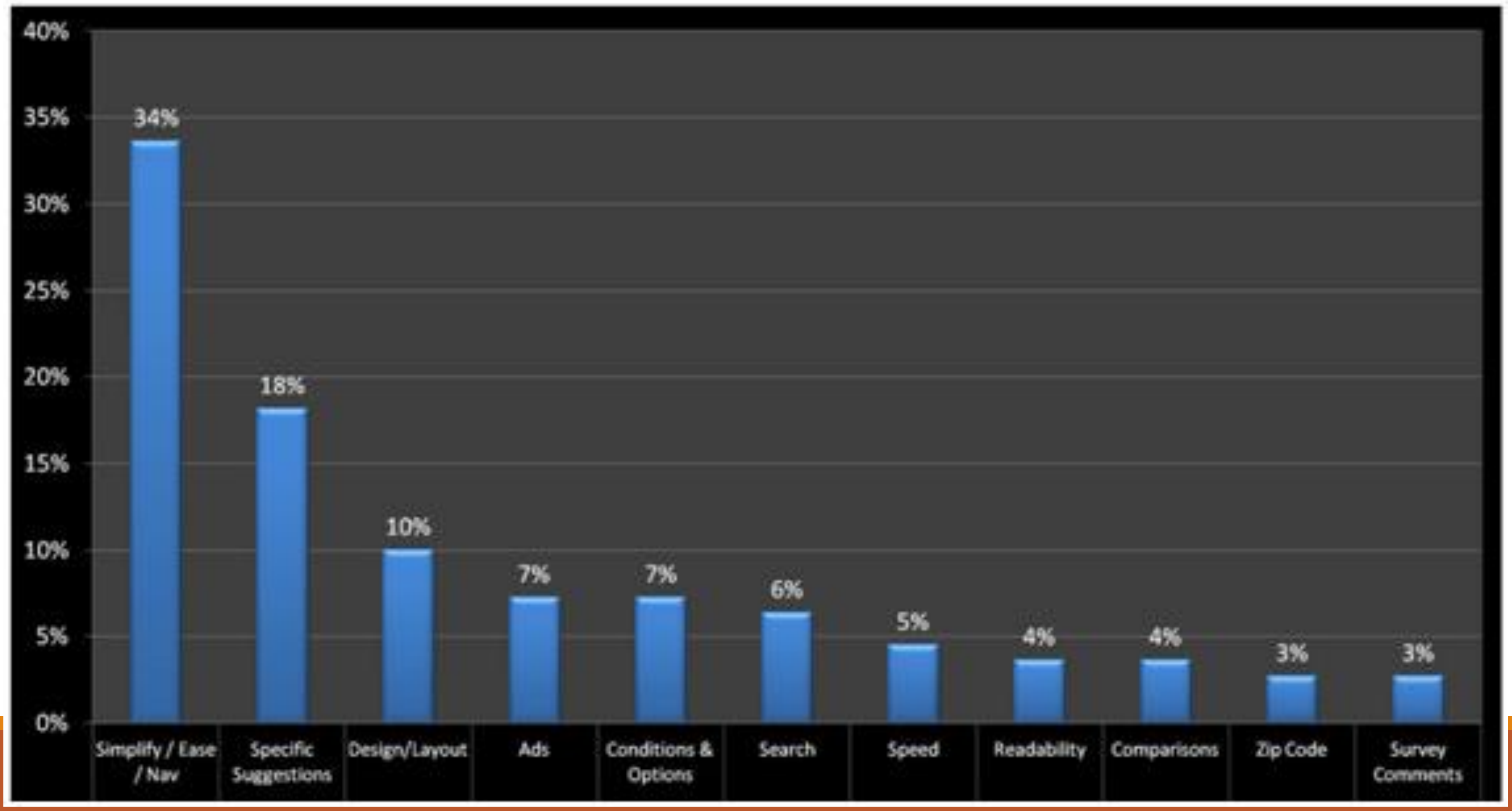


*http://www.measuringusability.com/blog/pareto-ux.php

6. Pareto and Zipf Laws

Applications to HCI:

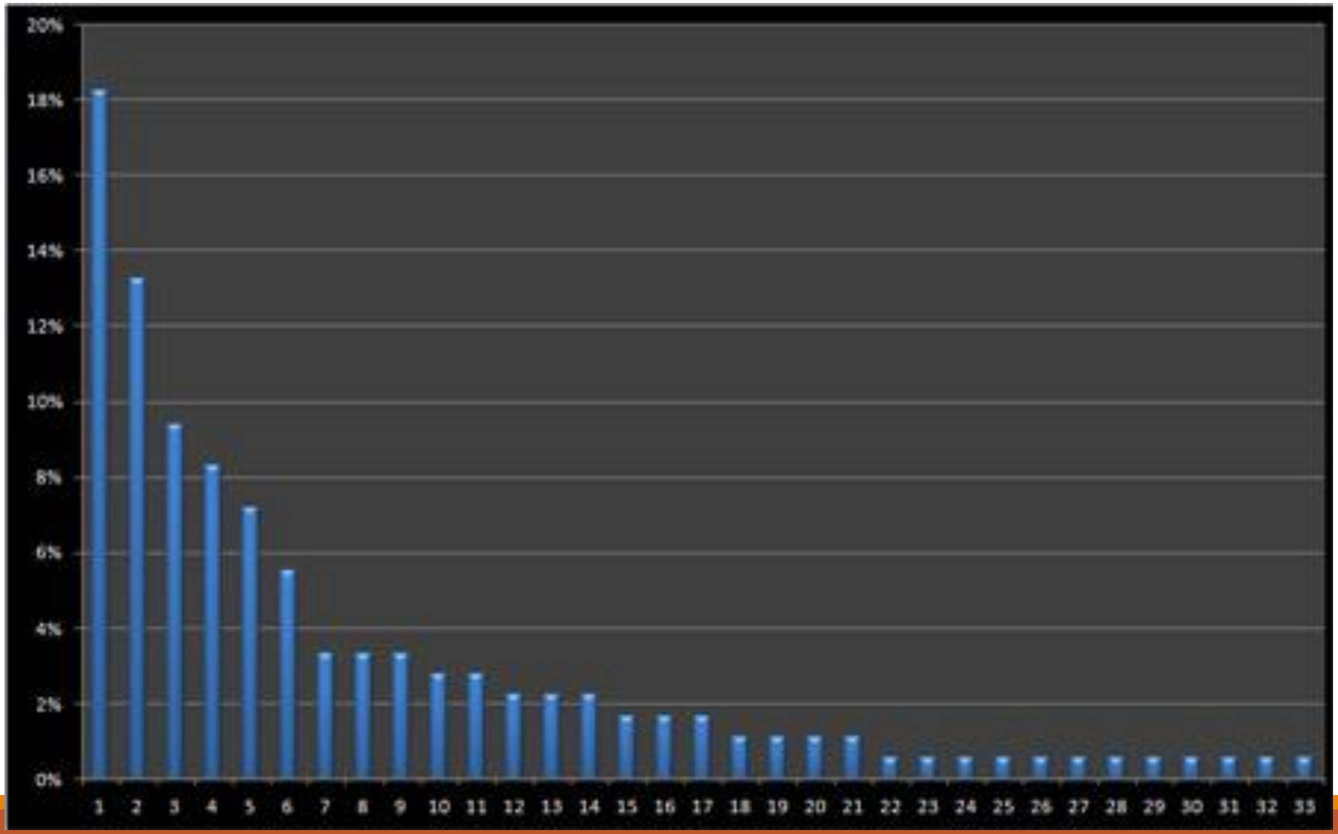
- When asked what users would fix on a website, two issues account for 50% of all comments and five account for 75%.



6. Pareto and Zipf Laws

Applications to HCI (Large Informational Website):

- Of the 33 usability problems identified across 50 users, fixing nine of them would address 72% of all poor interactions.



6. Pareto and Zipf Laws

The bottom line:

- There is almost always a time-budget tradeoff when designing.
- Often times, Pareto’s law can help designers focus on fixing the majority of issues with an interface quickly.

6. Pareto and Zipf Laws

George Kingsley Zipf

- American linguist
- Noticed that the most frequent word occurs about twice as much as the second most frequent.
- In general, nth most frequent word occurs 2x more often than the (n+1)th most frequent.
- Zipf law

Note: In the English language words like "and," "the," "to," and "of" occur often while words like "undeniable" are rare. This law applies to words in human or computer languages, operating system calls, colors in images, etc., and is the basis of many (if not, all!) compression approaches.

6. Pareto and Zipf Laws

As it turns out, the most frequently used command used in a piece of software is used about twice as often as the second most frequently used.

6. Pareto and Zipf Laws

Why does it matter?

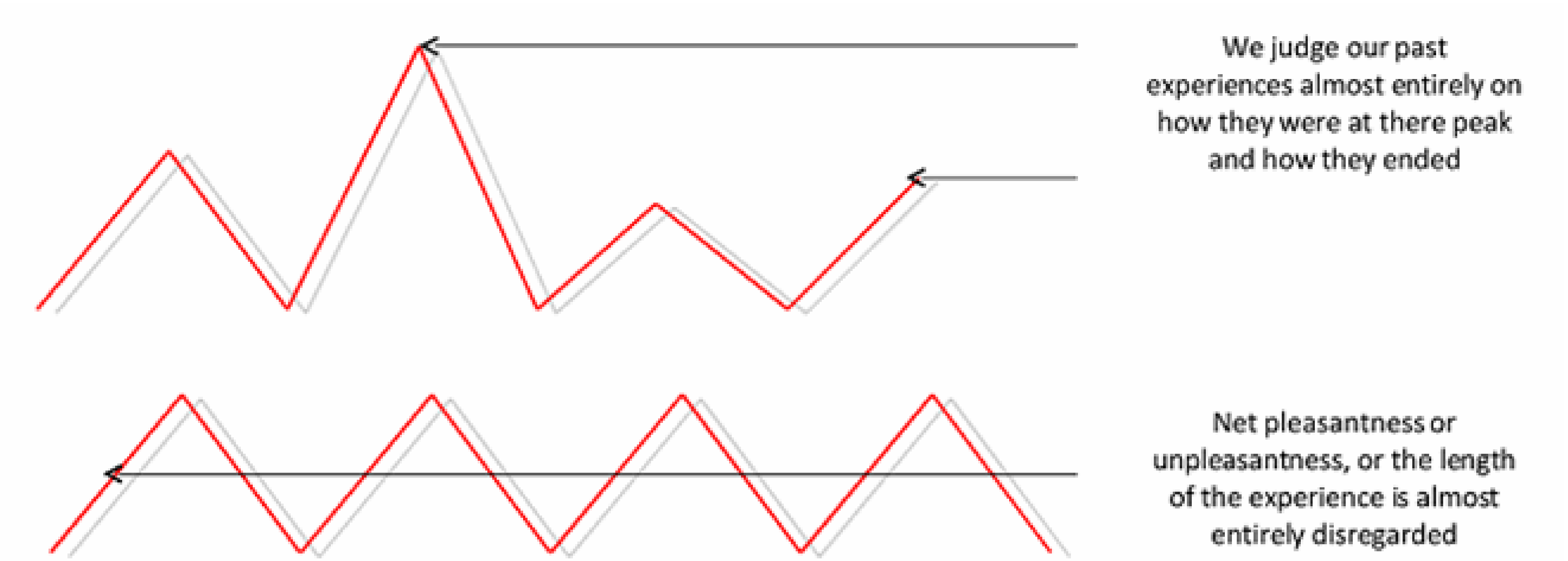
- Means that generally, you can design for the masses.

7. Peak End Rule

The Peak-End Rule states that humans judge their past experiences on the peak of that experience (whether pleasant or unpleasant) and how it ended. The net pleasantness or unpleasantness or the length of experience is disregarded.

*http://www.peakusability.com.au/articles/usability-heuristics-rules-laws-and-things-to-remember

7. Peak End Rule



7. Peak End Rule

The best or worst moment in their interaction, whether that's a moment of delight created by skillful design or the moment of frustration when they have just spent 5 minutes looking for a form and now can't find a phone number to get it mailed to them.

How it ended – was it a success? Did they find what they needed? Complete that form? Watch the video they intended to? Or did it end in miserable failure. What taste does that leave in the user's mouth?

*http://www.peakusability.com.au/articles/usability-heuristics-rules-laws-and-things-to-remember

Krug’s Usability Laws: For Fun!

Krug’s 3 laws of Usability

- “Don’t make me think.”
- “It doesn’t matter how many times I have to click, as long as each click is a mindless, unambiguous choice.”
- “Get rid of half the words on each page, then get rid of half of what is left.”

From: Krug, S. (2006), Don’t Make Me Think: A Common Sense Approach to Web Usability, Berkeley, CA: New Riders

Video!

https://www.youtube.com/watch?feature=player_embedded&v=6lyudeTqggE