Human Computer Interaction

UNIT:3

Lecture : 1

Guidelines in HCI

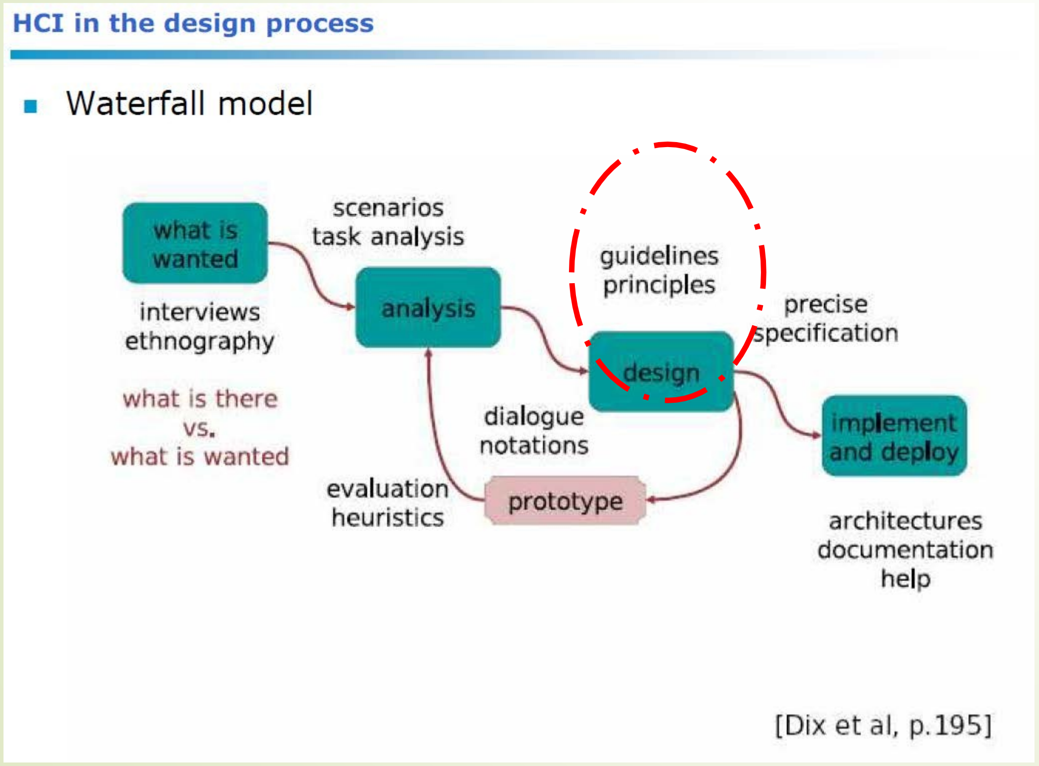
Mr. Nachiket Sainis / Ms. Reena Saini

B K Birla Institute of Engineering & Technology, Pilani

Eight Golden Rules of User Interface Design stated by Ben Shneiderman.

Background

We use Dix et al's version of the Waterfall model (reproduced below from Dix et al) to illustrate where exactly in the design cycle, the Rules - Guidelines & principles being discussed in this module, become important.


HCI in the design process

- Waterfall model

[Dix et al, p.195]

Design is both qualitative as well as quantitative.

The terms 'Guidelines' and 'principles' rather than precise' laws & rules' are used in Design.

Ben Shneiderman formulated eight such guidelines that can be used in Interface designing.

Introduction

Ben Shniderman* consolidated known tacit knowledge and practice guidelines which are used intuitively by graphic Interface designers - into a set of eight general guidelines for the use of computer science specialist who were being introduced to Visual Graphic designers' work of designing interactive GUI interfaces. Along with 'looks' the usability of a software depended on functionality.

Bend Shneiderman founded the HCI Lab at the University of Maryland , USA. He is known for Nassi-Shnerderman diagrams used in the field of Software Engineering.

- These are intended more as guidelines rather than ‘rules’ to be strictly adhered to at every step.
- They are useful for designers as well as software engineers involved in design of interfaces.
- Using these eight guidelines it is possible to distinguish a good interface design from a bad one especially from the Human - User interaction point of view.
- These have been put forth in a concise and understandable manner by Ben Shneiderman.
- While merely or blindly applying these eight guidelines is not necessarily going to gurantee a good interface ‘design’, they are useful in heuristic evaluation to identify GUIs that fall out of normal ‘pattern’. The guidelines can be used to rate GUI’s as good or bad.

Eight Golden Rules by Ben Shneiderman

1. Strive for Consistency
2. Cater to Universal Usability
3. Offer Informative feedback
4. Design Dialogs to yield closure
5. Prevent Errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load

Let us attempt to understand each one……………..

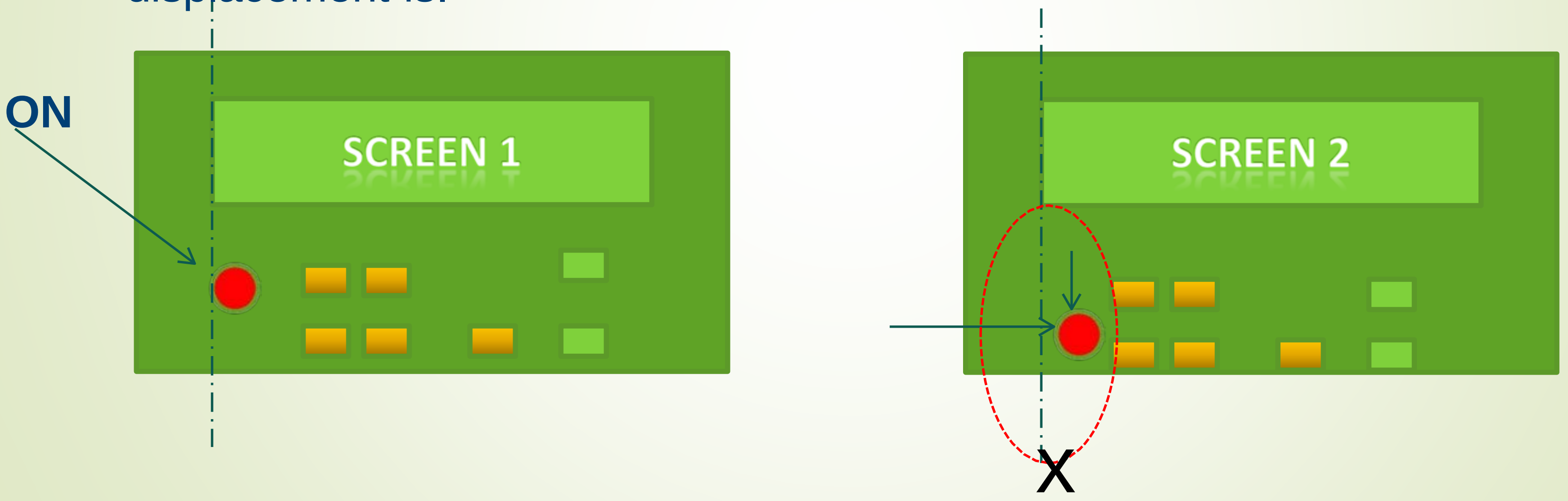
1. Strive for Consistency

- Users need to be able to do the same thing the same way that they have been doing.- every time.
- Interfaces need to exhibit ‘consistent’ quality across screens/ applications both visually as well as behaviorally.
- Consistency leads to a pattern which is easier to handle cognitively.
- Consistency such as ‘similar sequence of actions in similar situations’ makes it easy to learn.

Continued……..

Consistency can be achieved through graphical elements such as fonts, colour, shape, position being consistently same in all menus & screens, across categories for a particular software.

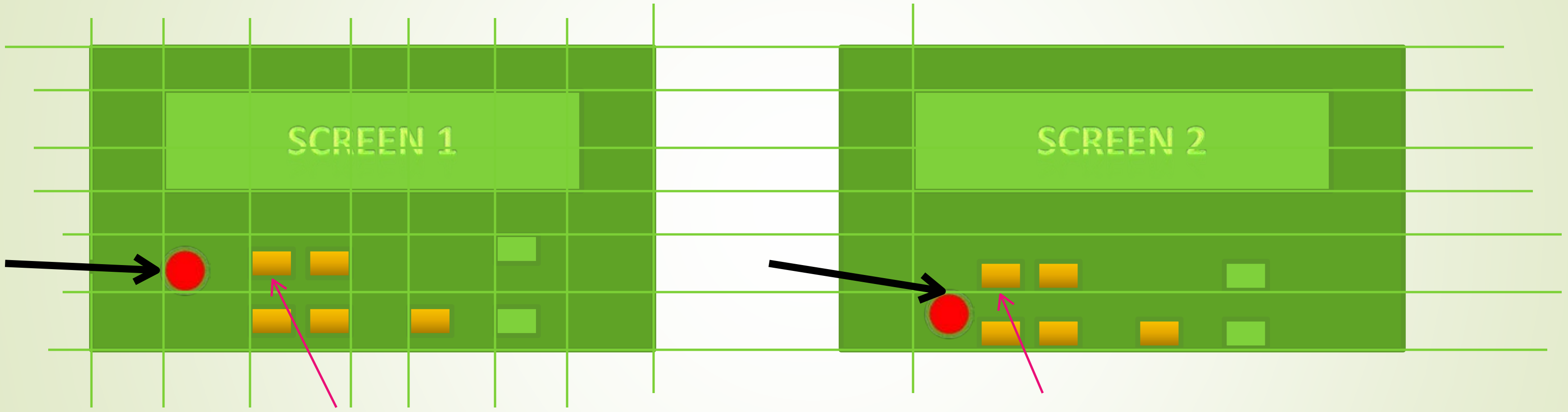
For example: If the **ON** button is on the right in the first screen and moves towards middle in the second screen then positional inconsistency is said to have occurred - however small the displacement is.



GUI designers use a simple technique to maintain consistency of control elements in successive screen.

Consistency….. continued

GUI designers use a background grid to place interactive elements in a consistent and orderly way so as to make them appear both physically as well a visually at the same place across the entire software package.



Inconsistent positioning of GUI elements is evident when observed against a grid. Grids are used as background reference to place the elements consistently

In case certain exceptions in maintaining consistency are required to be made in a subsequent screen, they should be such that they are comprehensible, distinct and limited in number.

2. Cater to wide range & type of Users

Universal design strives to cater to as wide a range of human users of different characteristics (age, culture, educational level, disability) with a single design.

While this may not be feasible or possible in all contexts, Shniderman's rule none the less needs to be followed so as not to leave out taking into consideration a section of users, other wise competent, who cannot use the interface due to no fault of theirs.

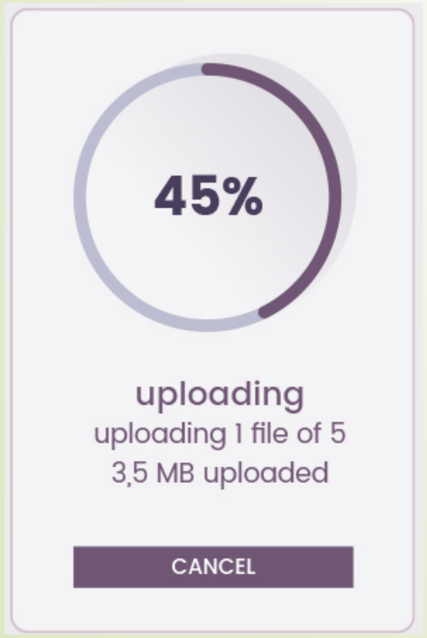
Users are classified as **Novice**, **Intermediate** and **Experts**. Experts tend to use lesser actions at a faster pace. Abbreviations short cuts keys etc are some of the techniques used.

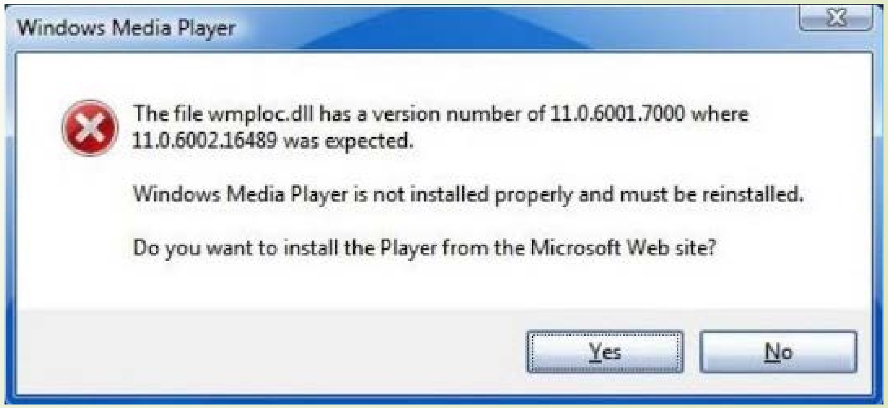
Interfaces need to cater to all levels & classification of users: novice to experts.

3. Offer Informative feedback

1. Strive for Consistency
2. Cater to Universal Usability
3. **Offer Informative feedback**
4. Design Dialogs to yield closure
5. Prevent Errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load

- Interfaces need to not just to be communicative but also need to inform the 'user' in terms of learning & feed back which tells them that they are proceeding in the right direction.
- For every action of the user there needs to be a feedback – only then 'interaction' (in HCI) is said to take place. Specific error messages composed in a appositive tone give affirmative feedback without having to feel punitive.
- Unless the user gets a feed back he/she cannot proceed or becomes unsure of the correctness of the action.




Windows Media Player

The file wmploc.dll has a version number of 11.0.6001.7000 where 11.0.6002.16489 was expected.

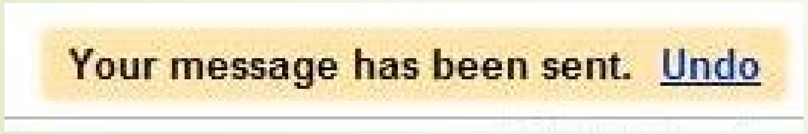
Windows Media Player is not installed properly and must be reinstalled.

Do you want to install the Player from the Microsoft Web site?

4. Design Dialogs to yield closure

- In an interaction - dialogue needs to have a closure which is recognized by the user as end of an action.
- Sequence of actions need to proceed in a dialogue by engaging the user in a step by step manner.
- Like in a mathematical expression, every enclosing bracket needs a corresponding closing bracket. So also subsequence of actions needs to be grouped with intermittent closing of each sub group followed finally by a closer action of the group.

Ex: A message at the end of a sequence of events gives a feed back & closure of sending a SMS**.**


Your message has been sent. Undo

Continued…….

Example 2: Un closed dialogue

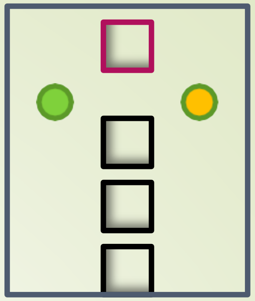
Press ON button

Look at the green lamp.

If green glows press next push button - yellow lamp will glow

Push 3rd button and continue till green lamp stops glowing.

End of task…………



An Example of a closed dialogue:

Press ON button

Look at the green lamp.

If green glows press 2nd push button and yellow lamp will glow.

Press 3rd button and continue with other 3 buttons till green lamp stops glowing.

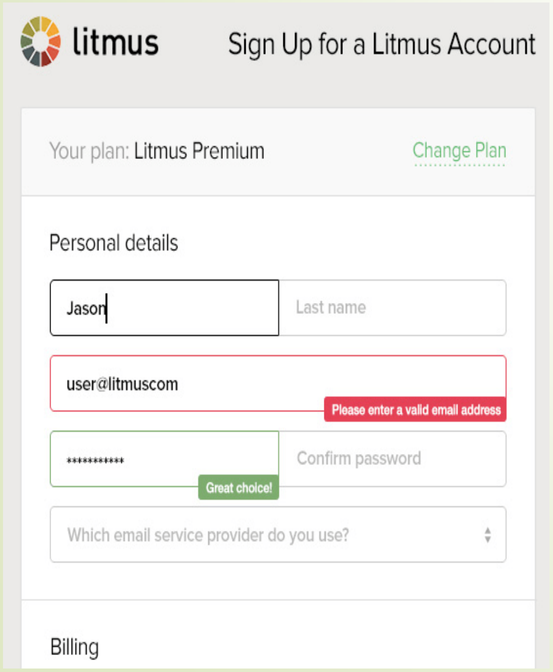
When Yellow lamp stops glowing it indicates sequence over.

End of task.

Notice the yellow lamp feed back dialogue above being not closed ?
What happens to yellow lamp ? Did it stop glowing? or why it continues glowing when the task is over ?

….. are some of the questions that may arise due to non closure of dialogues which can lead to confusion for a user

5. Prevent Errors



Sign Up for a Litmus Account

Your plan: Litmus Premium

Personal details

Jason

Last name

user@litmuscom

Please enter a valid email address

****

Great choice!

Confirm password

Which email service provider do you use?

Billing

Interfaces need to minimize errors. Human Computer dialogue can be designed to minimize and prevent errors made by users.

There cloud be many reasons for users errors but the user himself or herself is not one of them ! Users can make errors while interacting with computers as well as while inputting / interpreting information.

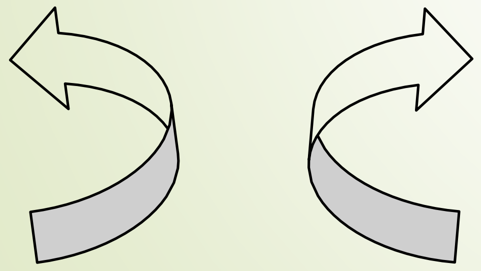
Even if the user makes an error the system needs top be designed to detect it, take corrective or precautionary steps to arrest it. It also needs to offer a way out for recovery from the error.

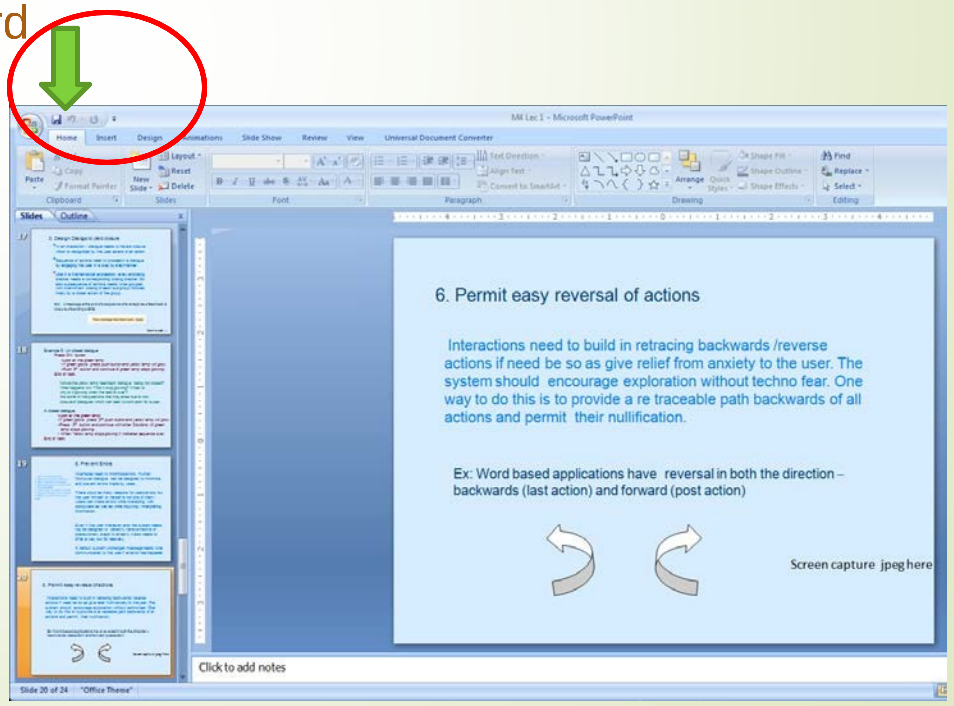
A default system unchanged message needs to be communicated to the user if an error has happened .

6. Permit easy reversal of actions

- Interactions need to build in retracing backwards /reverse actions if need be so as give relief from anxiety to the user.
- The system should encourage exploration without techno fear. One way to do this is to provide a re traceable path backwards of all actions and permit their nullification.

Ex: This PPT application has reversal in both the direction – backwards (last action) and forward (post action)





6. Permit easy reversal of actions

Interactions need to build in retracing backwards /reverse actions if need be so as give relief from anxiety to the user. The system should encourage exploration without techno fear. One way to do this is to provide a re traceable path backwards of all actions and permit their nullification.

Ex: Word based applications have reversal in both the direction – backwards (last action) and forward (post action)

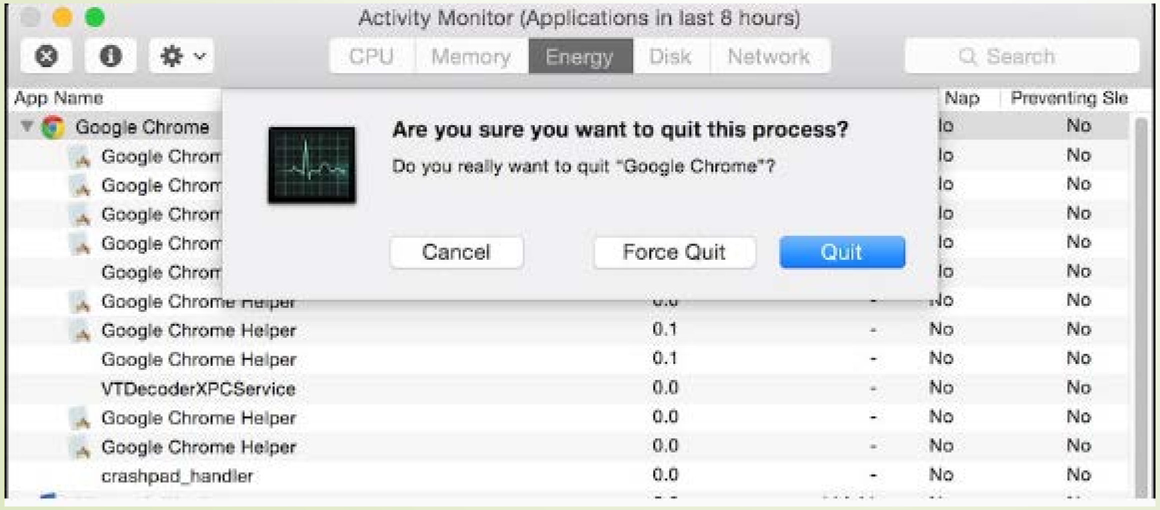
Screen capture jpeg here

Click to add notes

Slide 20 of 24 "Office Theme"

7. Support internal locus of control

- Allow user to always feel ‘in control’ of the system and of the situation.
- Make the user aware that he/she is in control. User should believe that they are controlling the system and not the other way around. This is achieved by more opportunities for ‘interactions’.
- The bearing of where the user presently is helps the user to orient or reorient the interaction. The user should never be allowed to feel lost.

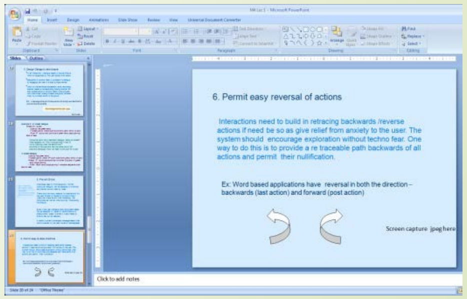


Are you sure you want to quit this process?

Do you really want to quit "Google Chrome"?

Cancel | Force Quit | Quit

App Name			Nap	Preventing Sle
Google Chrome			lo	No
Google Chrom			lo	No
Google Chrom			lo	No
Google Chrom			lo	No
Google Chrom			lo	No
Google Chrom			lo	No
Google Chrome Helper	0.0	-	No	No
Google Chrome Helper	0.1	-	No	No
Google Chrome Helper	0.1	-	No	No
VTDecoderXPCService	0.0	-	No	No
Google Chrome Helper	0.0	-	No	No
Google Chrome Helper	0.0	-	No	No
crashpad_handler	0.0	-	No	No



6. Permit easy reversal of actions

Interactions need to build in retracing backwards /reverse actions if need be so as give relief from anxiety to the user. The system should encourage exploration without techno fear. One way to do this is to provide a re traceable path backwards of all actions and permit their nullification.

Ex: Word based applications have reversal in both the direction – backwards (last action) and forward (post action)

Screen capture jpeg here

8. Reduce short term memory load

94 56 781029

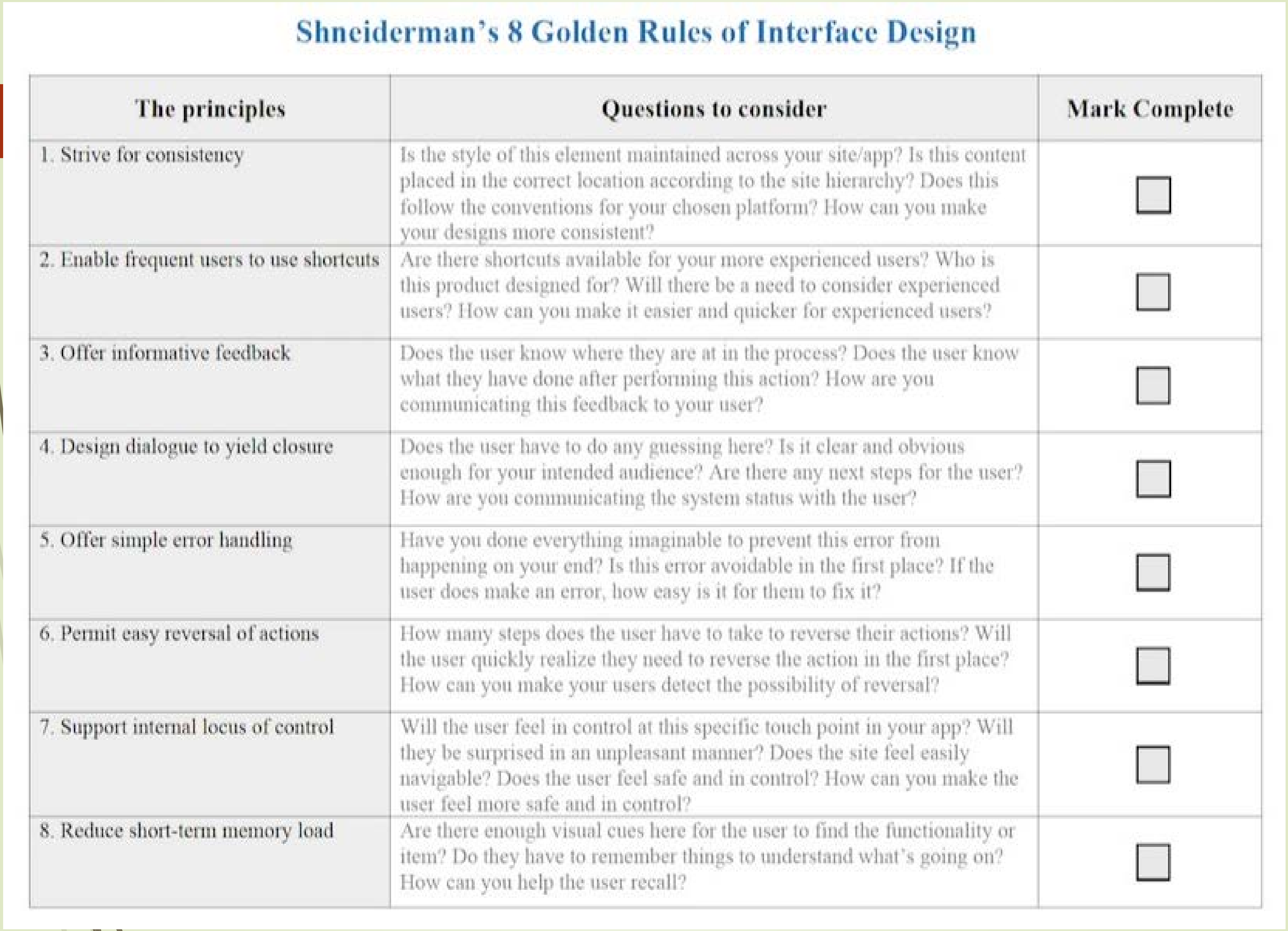
Easier to remember if chunked into smaller set

94 56 78 10 29

Care not load the cognitive short term memory of the user by expecting user to remember several sequences , actions and their consequences at a time. Means loading their short term memory while interacting.

Millers* 7 chunks of information is often prescribed as a solution to limit short term memory.
In psychological experiments it has been found that the short term memory can hold 7 +- 2 bits called chunks of information.
Long sequential actions requiring more than 7 chunks need to be broken down into smaller chunks.

*G.A. Miller; The Magical number seven, plus or minus two: some limits on our capacity to process information.



Shneiderman's 8 Golden Rules of Interface Design

The principles	Questions to consider	Mark Complete
1. Strive for consistency	Is the style of this element maintained across your site/app? Is this content placed in the correct location according to the site hierarchy? Does this follow the conventions for your chosen platform? How can you make your designs more consistent?	☐
2. Enable frequent users to use shortcuts	Are there shortcuts available for your more experienced users? Who is this product designed for? Will there be a need to consider experienced users? How can you make it easier and quicker for experienced users?	☐
3. Offer informative feedback	Does the user know where they are at in the process? Does the user know what they have done after performing this action? How are you communicating this feedback to your user?	☐
4. Design dialogue to yield closure	Does the user have to do any guessing here? Is it clear and obvious enough for your intended audience? Are there any next steps for the user? How are you communicating the system status with the user?	☐
5. Offer simple error handling	Have you done everything imaginable to prevent this error from happening on your end? Is this error avoidable in the first place? If the user does make an error, how easy is it for them to fix it?	☐
6. Permit easy reversal of actions	How many steps does the user have to take to reverse their actions? Will the user quickly realize they need to reverse the action in the first place? How can you make your users detect the possibility of reversal?	☐
7. Support internal locus of control	Will the user feel in control at this specific touch point in your app? Will they be surprised in an unpleasant manner? Does the site feel easily navigable? Does the user feel safe and in control? How can you make the user feel more safe and in control?	☐
8. Reduce short-term memory load	Are there enough visual cues here for the user to find the functionality or item? Do they have to remember things to understand what's going on? How can you help the user recall?	☐

References:

1.Shneiderman. B. ; Designing the user interface: Strategies for effective Human Computer Interaction; Addsion-Wesley Publishers Treading MA. 2004.)

2.Designing the user interface: Strategies for effective Human Computer Interaction ; Ben Shneiderman and Catherikne Plaisant , Addison-Wesley Publishers Treading MA. 2010.)