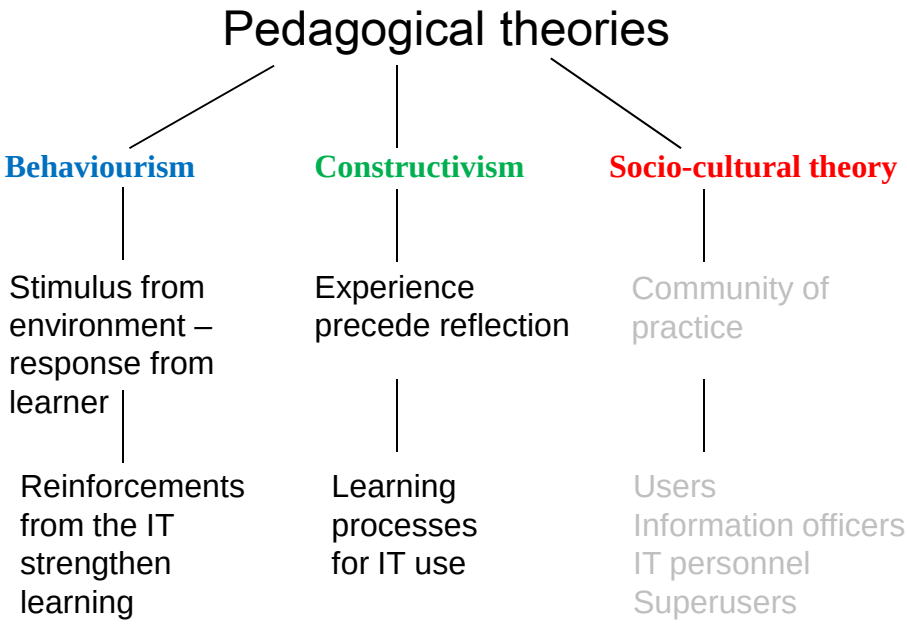
# User interface for learning

- Aim:
  - Design for and evaluate learnability
  - Writing inline help
  - Basis for Assignment 5
- Core literature
  - Chapter 9
- Additional literature
  - Grossman et.al. (2009) A Survey of Software Learnability: Metrics, Methodologies and Guidelines
  - Furnas et.al. (1987) The vocabulary problem in human-system communication

1




2

# Behaviourism – 1920

- Observable behaviour
  - No concern for thinking
- The learning cycle
  - Stimulus from the environment
    - Login:
    - Password:
  - Response from the learner
    - Writing user name and password
    - If this response differed from previous ones and the new response is repeated, learning has occurred
  - Reinforcement from the environment
    - Other windows appear



## Reinforcement and learning

**Reinforcement** strengthens learning

A wanted event or the removal of an unwanted one

- Wanted
  Output which we wanted
- Removal of unwanted
  No more error messages
- Informative reinforcement
  `You are now logged in`
- Immediate reinforcement
  < 1 second
- Repeated reinforcement at variable ratio or interval

**Punishment** weakens learning

Unwanted event or removal of a wanted one

```
> prnt rapport.text
        ILLEGAL USER ACTION!
          PRINTER DAMAGED!
```

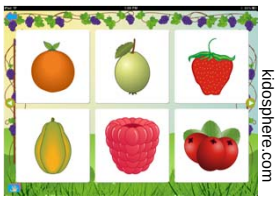
- Extinction
  - No new stimulus from the environment

```
> cd MyFiles
>
```

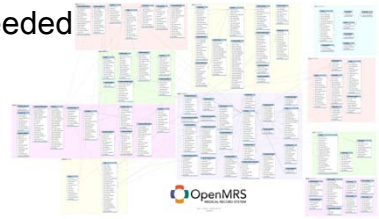


# Design for learnability

When simple things need instruction, it is a certain sign of poor design



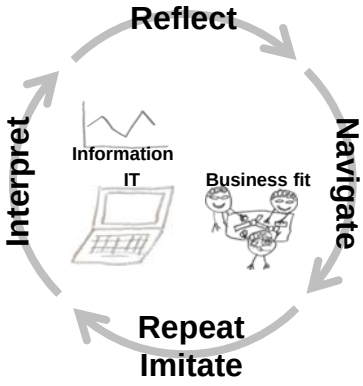

- Some computer applications are complex
  - Additional help needed




5

# Inline help in the program

- Responding to the the users' current problem
  - Guidance
  - Not a tutorial primarily designed for teaching
- Minimal distraction from task
  - Recognizable language
  - Recognizable graphics
  - Choice of contents
- For which learning process?



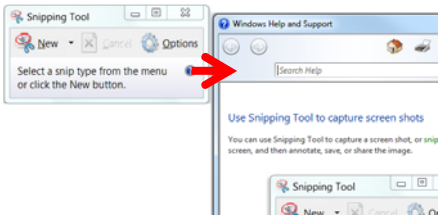

6

# Help – How

**Inline – Context-sensitive**

- Tooltip
- Wizard
- Help button
- System-initiated

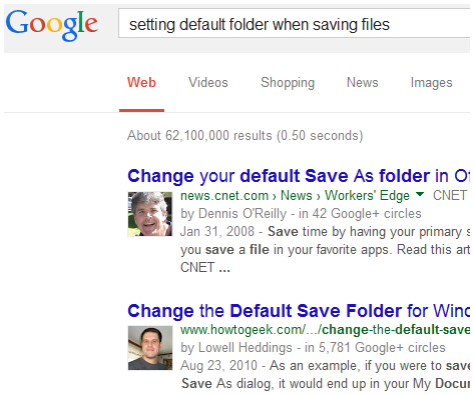
→ Help where you are



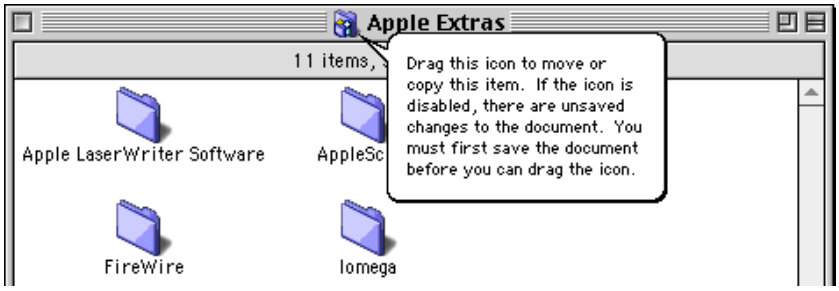

**Context-free**

- Help system
- Web

→ Search if you don't know where to be



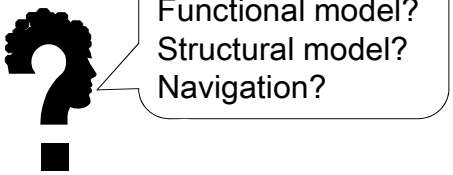

# Balloon help

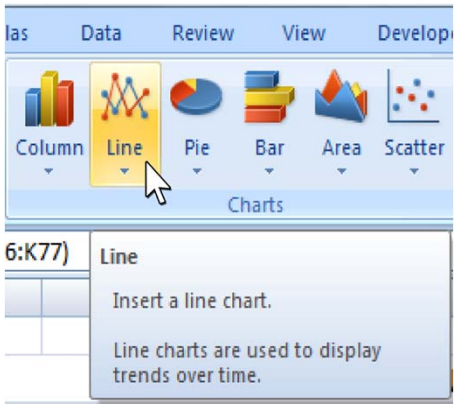



- Appeared immediately on mouseover
- Cluttered the screen

# Tooltip – Screentip

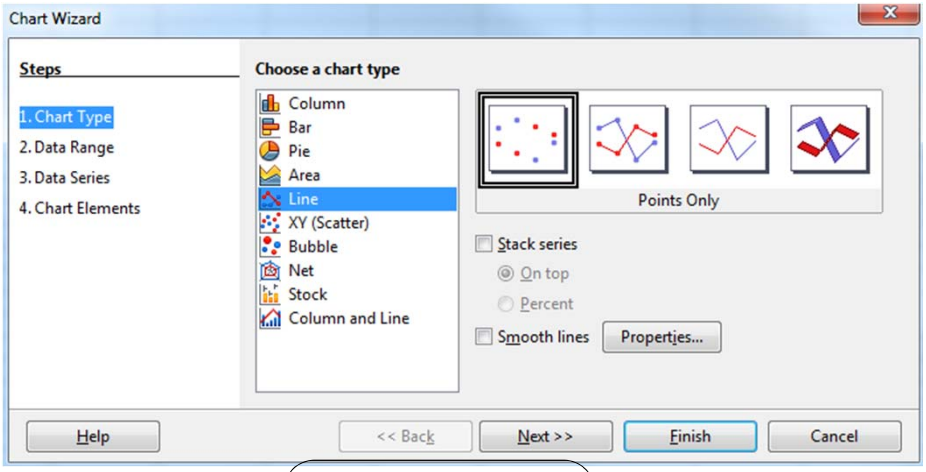
- Help where the user is at the moment
- No need for search
  1 s delay
  → Avoiding distraction

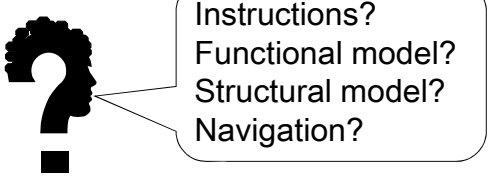




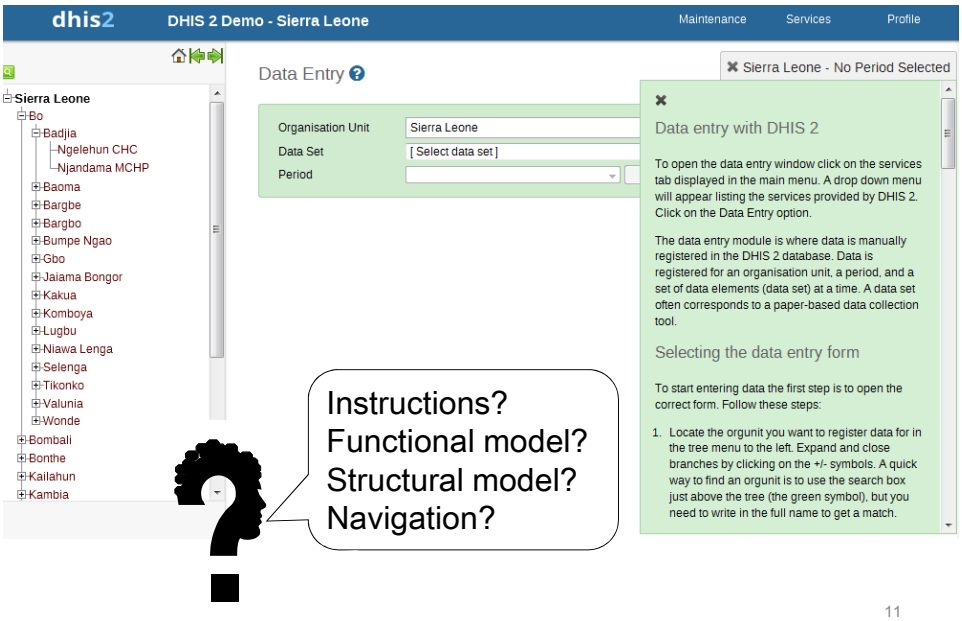

# Wizards carrying out the operatons



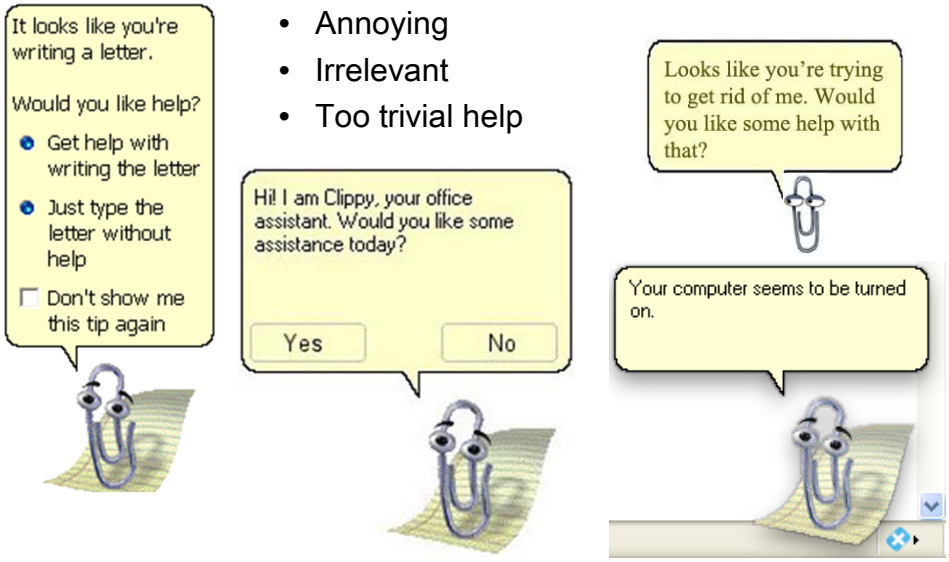



# Help button → Document




11

# System initiated – Clippy

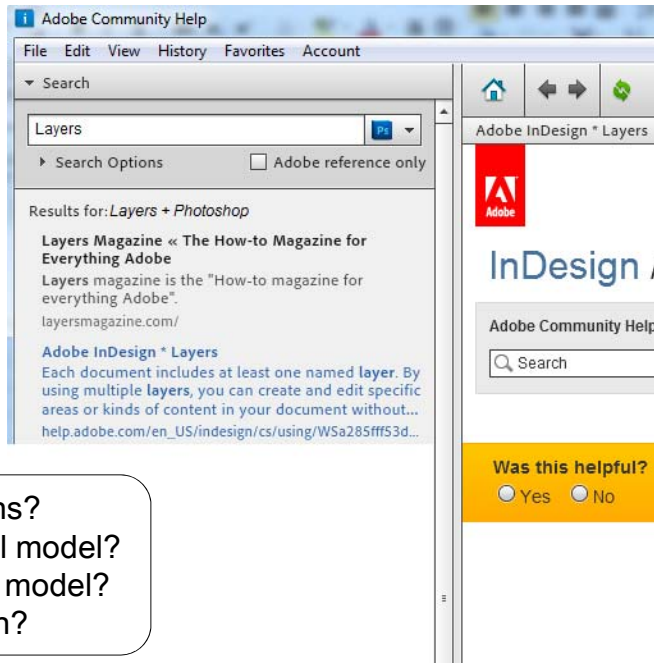


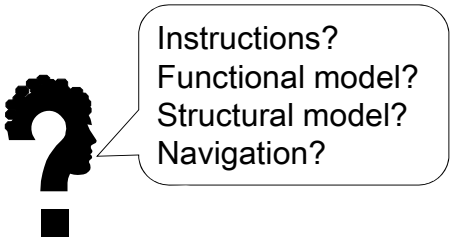

- Annoying
- Irrelevant
- Too trivial help

12

# Help system

1. Click Help in the application
2. Wait for the help system to start
3. Select software
4. Search
5. Select hit

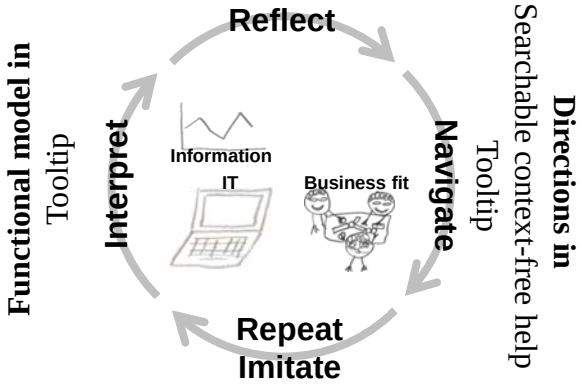



13

# Targeting learning in help functionality

**Structural model in**
Context-sensitive/free document help




**Instructions in**
Wizard
Context-sensitive/free video

14

# Qualities of IT applications

- Learnability
  - From novice to expert user
    - Time from first encounter to use
    - Time before understanding what the application can be used for
    - Time before understanding how the representation of the domain enables and restricts operations
  - Intermittent users
    - Time to recall

Trouble shooting

- Usability
  - Efficiency
    - Time and effort used to achieve the result
  - Satisfaction
    - Comfort and acceptability amongst users
- Usefulness
  - Effectiveness
  - The quality of the result achieved

15

# Learnability evaluation

| | Software | Help functionality |
|---|---|---|
| **Heuristic evaluation** | Specialists checking software | Specialists checking help functions |
| **Question-suggestion** | Software tasks | Help tasks |
| **Measuring learning** | Software tasks | Help tasks |

- Help functionality
  - Information: understandability of the help contents
  - IT: way of accessing the help
  - Business fit: does the help fix the problem

16

# Heuristic evaluation – software and help

- 2 - 3 usability specialists
- Inspect every detail of the application
- Compare to known guidelines (heuristics)
  - Each guideline broken
  - a possible problem of learning is noted
- Cheap, first evaluation

| | Software | Help |
|---|---|---|
| **Guidelines (heuristics):** | Rogers, Sharp, Preece (2007) Interaction Design: Beyond Human - Computer Interaction | Chapter 9. Furnas et.al (1987) |

17

# Question-suggestion – software (incl. Help)

- Small number of test persons, stop when no news
  - Right selection of users?
- Design tasks to perform

**Question-suggestion Protocol Instructions to Participant:**

1. Ask relatively specific, procedural questions.
2. Try to answer your own questions first,
   Software only: but do not engage in extensive problem solving.
   Help: Look for help if needed
3. Focus on getting the task done, as you would in the real world.

- Video-recording, time taking, notes
- Possible interview before and after the session
- Analysis of the users' understanding, misunderstandings and mistakes
- Consumes more time than heuristic evaluation
- For systems to be extensively used

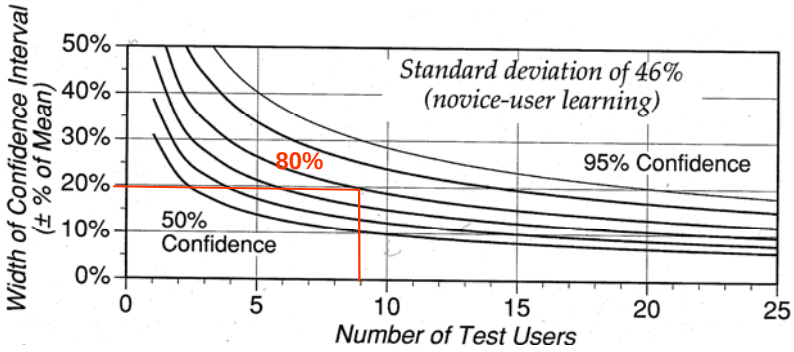
18

# Measuring skills learning – software incl. help

- Design tasks to perform
- Representative selection of users
- Way of measuring
  - Time taking
  - Counting keystrokes
  - Counting errors
  - Scaled response to questionnaires


19

Number of test users for measuring time for learning for novices
The number increases with standard deviation




80% confidence level

20% confidence interval → 9 users

Example

Mean 5 minutes

80% surety that the real mean lies within the 4 – 6 minutes interval

Jakob Nielsen (1993) Usability Engineering. AP Professional, Boston

20