Question (1-5)

* Is there a specific explanation? (1)

Good examples:
- it improves learnability
- it helps error prevention

Bad examples:
- just because it makes things more usable
- just because it is useful

* Is there a correct example? (1)
* Is the example specific? (1)

Good examples:
- programmers may know what mb/s means, but novice users only care about how many seconds left.

Bad examples:
- there might be some feature in some tool needed by some users

1. memory
   - learnability
   - old is slower, young is faster
2. left-to-right
   - speak different languages
3. advanced users, expert mode, command line, care about efficiency
   - beginner users, GUI, learnability is important
4. efficiency
   - satisfaction
5. more frequent, more familiar with the system, cares about efficiency
   - less frequent, memory, learnability

Questions (6-10)

* is there a usability attribute mentioned? (1)
* is there a design principle mentioned? (1)
* are they related? (1)

6. efficiency or few errors, or learnability
   - reversible actions
7. learnability or memory
   - not speaking the user's language
   - no match between system and real world
   - no error prevention
8. learnability or memory
   - consistency
   - following the standards
   - recognition rather than recall
9. efficiency or memory
   - recognition rather than recall
   - visibility of system status
10. error prevention
help users recognize and fix errors

Questions (11-15)

* is there a usability issue mentioned? (1)
* is there a correct solution? (1)
* is the solution specific enough? (1)

11. Memory
   change color
12. Memory
   file names
   keep the steps visible
13. Learnability or efficiency
   tool tips
14. few errors or subjective pleasing
   have a toolbar that allow users to adjust spacing
   make spacing larger
15. satisfaction
   progress bar