

Human-Computer Interaction

CPSC 481 - Winter 2019

Heuristic Evaluation

Heuristic Evaluation

- Heuristic evaluation (Nielsen and Molich, 1990; Nielsen 1994) is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process.
- Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the “heuristics”).

Heuristic Evaluation

- Heuristic evaluation is performed by having each individual evaluator inspect the interface alone.
- Only after all evaluations have been completed are the evaluators allowed to communicate and have their findings aggregated.
- This procedure is important in order to ensure independent and unbiased evaluations from each evaluator.
- The results of the evaluation can be recorded either as written reports from each evaluator or by having the evaluators verbalize their comments to an observer as they go through the interface.

Heuristic Evaluation

- Typically, a heuristic evaluation session for an individual evaluator lasts one or two hours.
- During the evaluation session, the evaluator goes through the interface several times and inspects the various dialogue elements and compares them with a list of recognized usability principles (the heuristics).
- In principle, the evaluators decide on their own how they want to proceed with evaluating the interface. A general recommendation would be that they go through the interface at least twice, however. The first pass would be intended to get a feel for the flow of the interaction and the general scope of the system. The second pass then allows the evaluator to focus on specific interface elements while knowing how they fit into the larger whole.

Heuristic Evaluation

- The output from using the heuristic evaluation method is a list of usability problems in the interface with references to those usability principles that were violated by the design in each case in the opinion of the evaluator.
- It is not sufficient for evaluators to simply say that they do not like something; they should explain why they do not like it with reference to the heuristics or to other usability results.
- The evaluators should try to be as specific as possible and should list each usability problem separately.

Heuristic Evaluation

- A heuristic is a rule of thumb – a principle that is a shortcut for solving a problem or making decisions
 - “never chase after a bus, another one is coming...”
 - Not always right/true, but cognitive shortcuts
- Design heuristics:
 - Broad usability statements that can guide a developer’s design efforts

HEURISTICS

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
 - 5. Error prevention
 - 6. Recognition rather than recall
 - 7. Flexibility and efficiency of use
 - 8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation



HEURISTICS



HEURISTICS

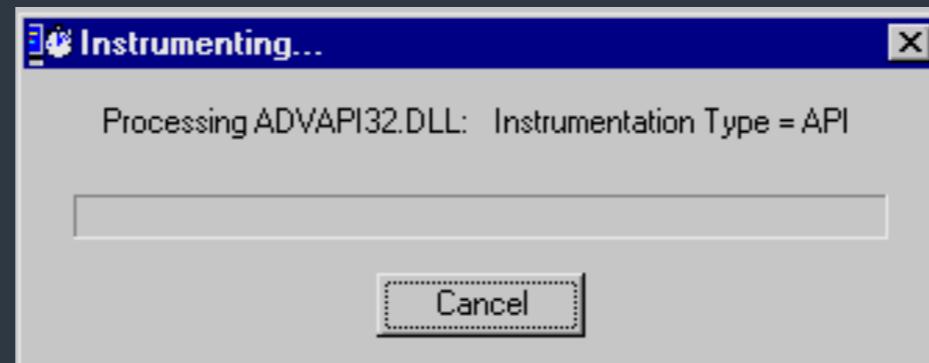
1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.



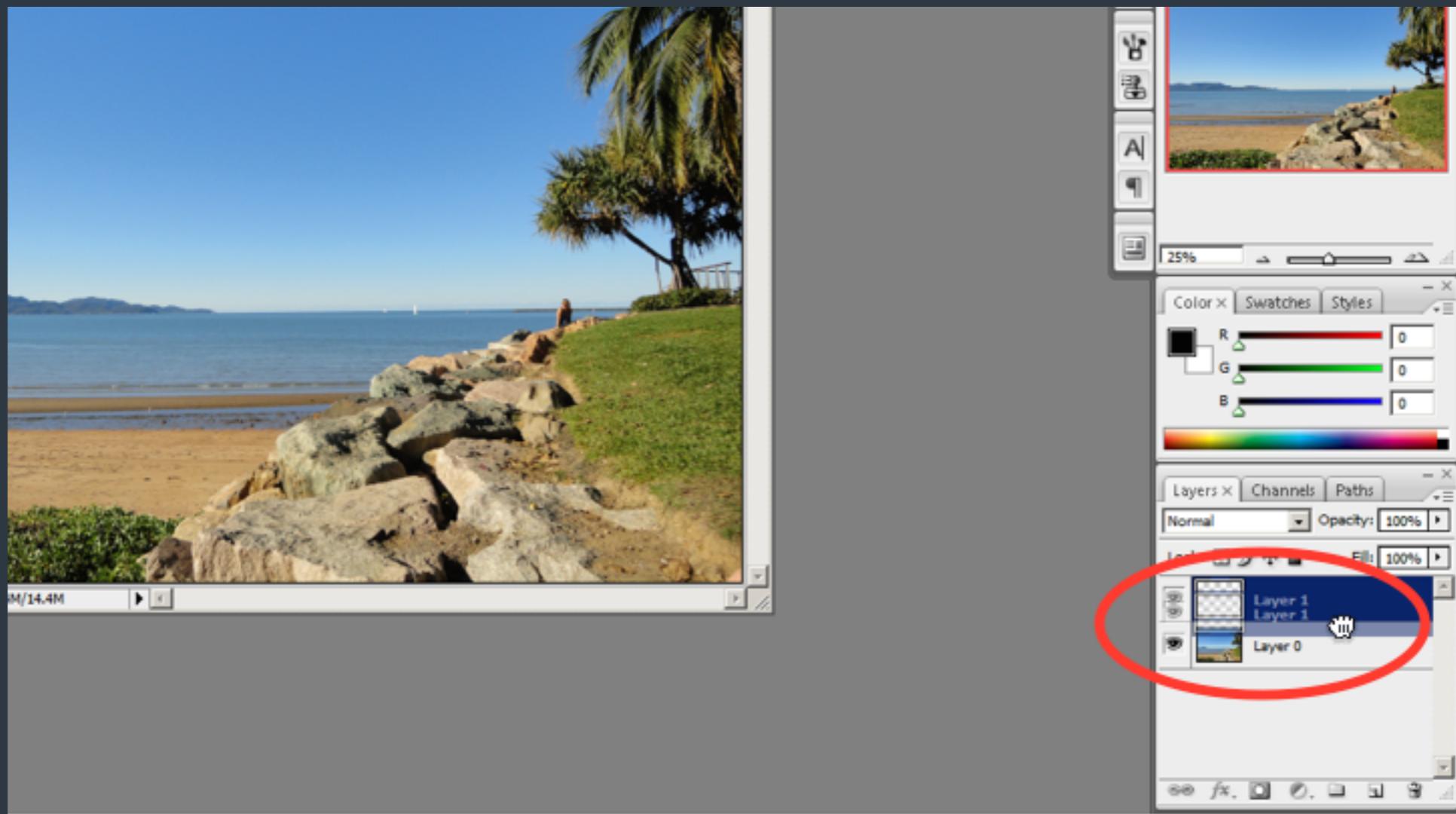
HEURISTICS

1. Visibility of system status



HEURISTICS

1. Visibility of system status



HEURISTICS

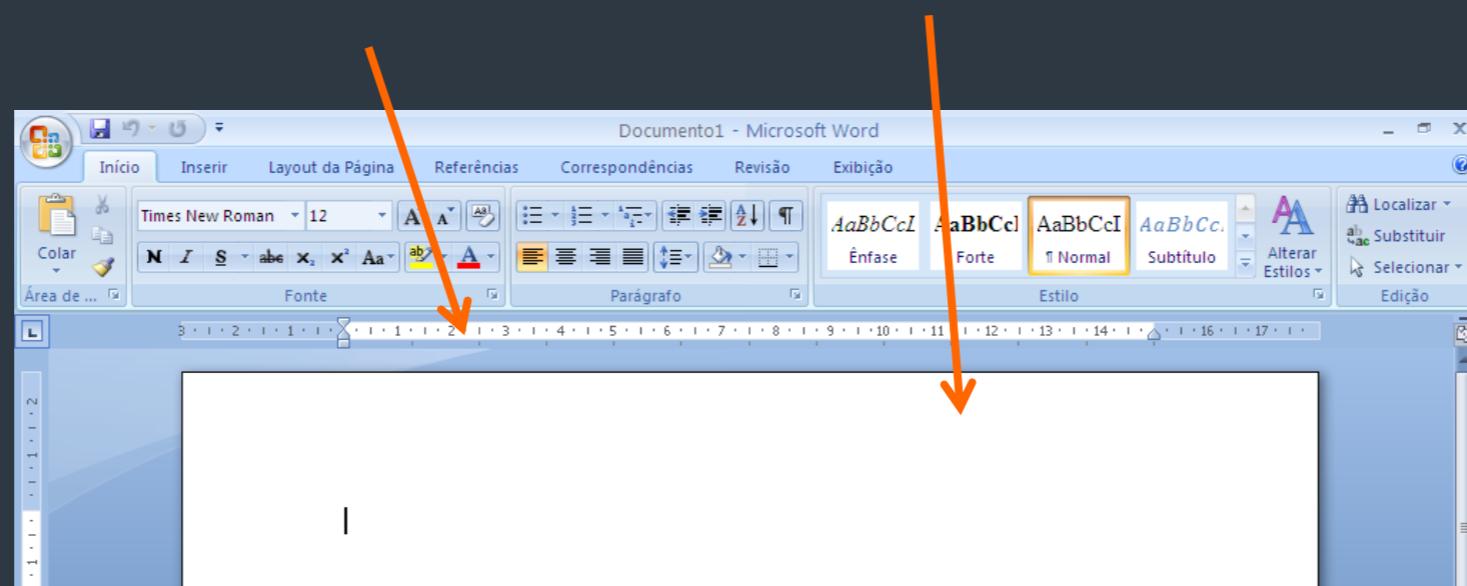
2. Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system- oriented terms. Follow real-world conventions, making information appear in a natural and logical order.



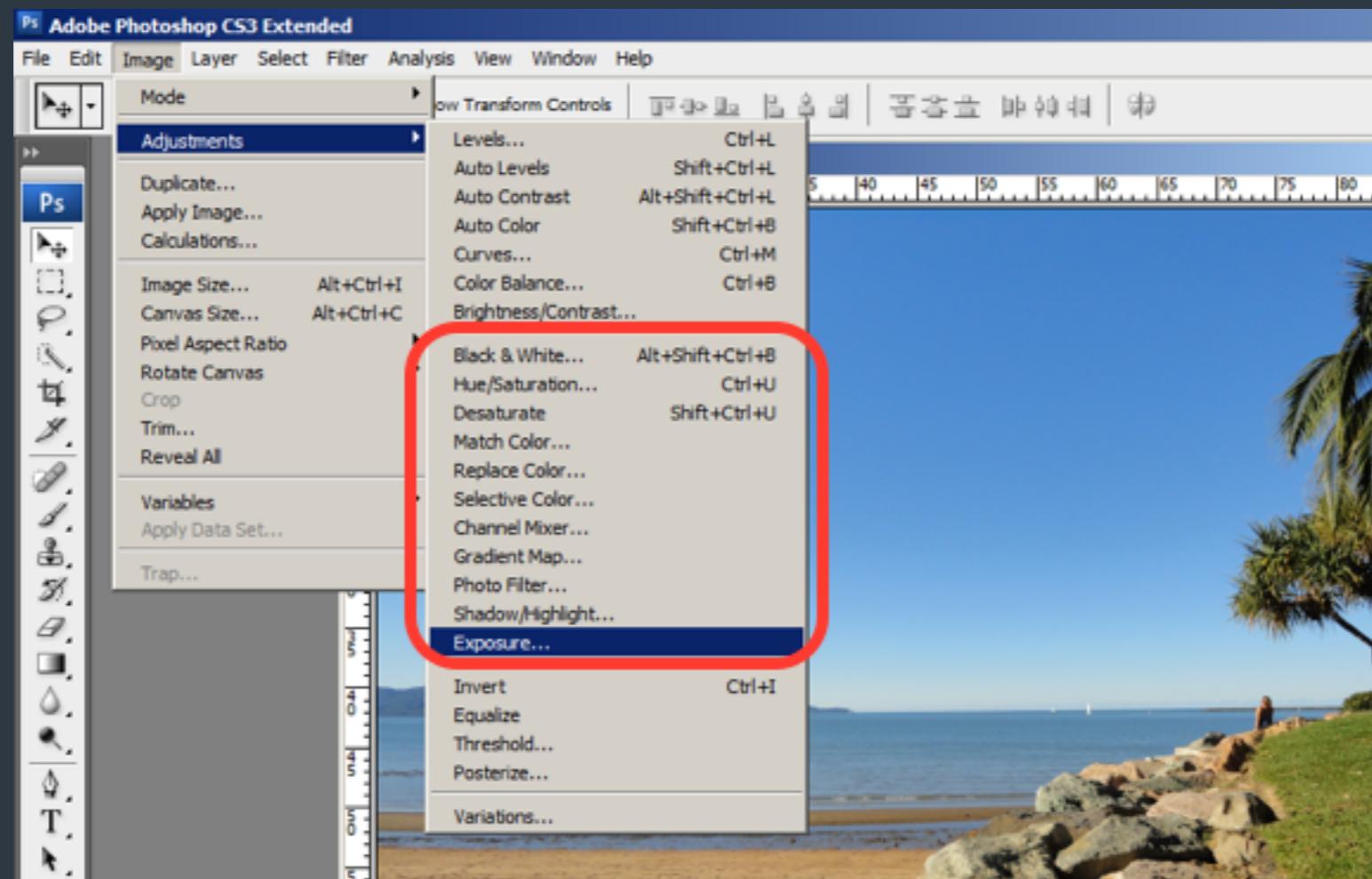
HEURISTICS

2. Match between system and the real world



HEURISTICS

2. Match between system and the real world



HEURISTICS

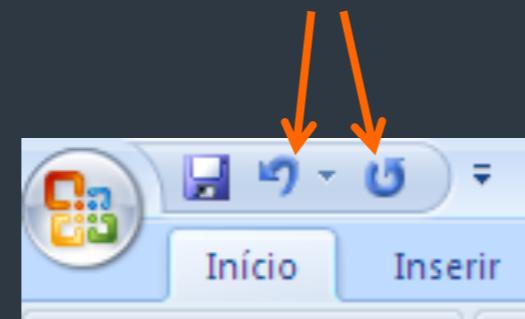
3. User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.



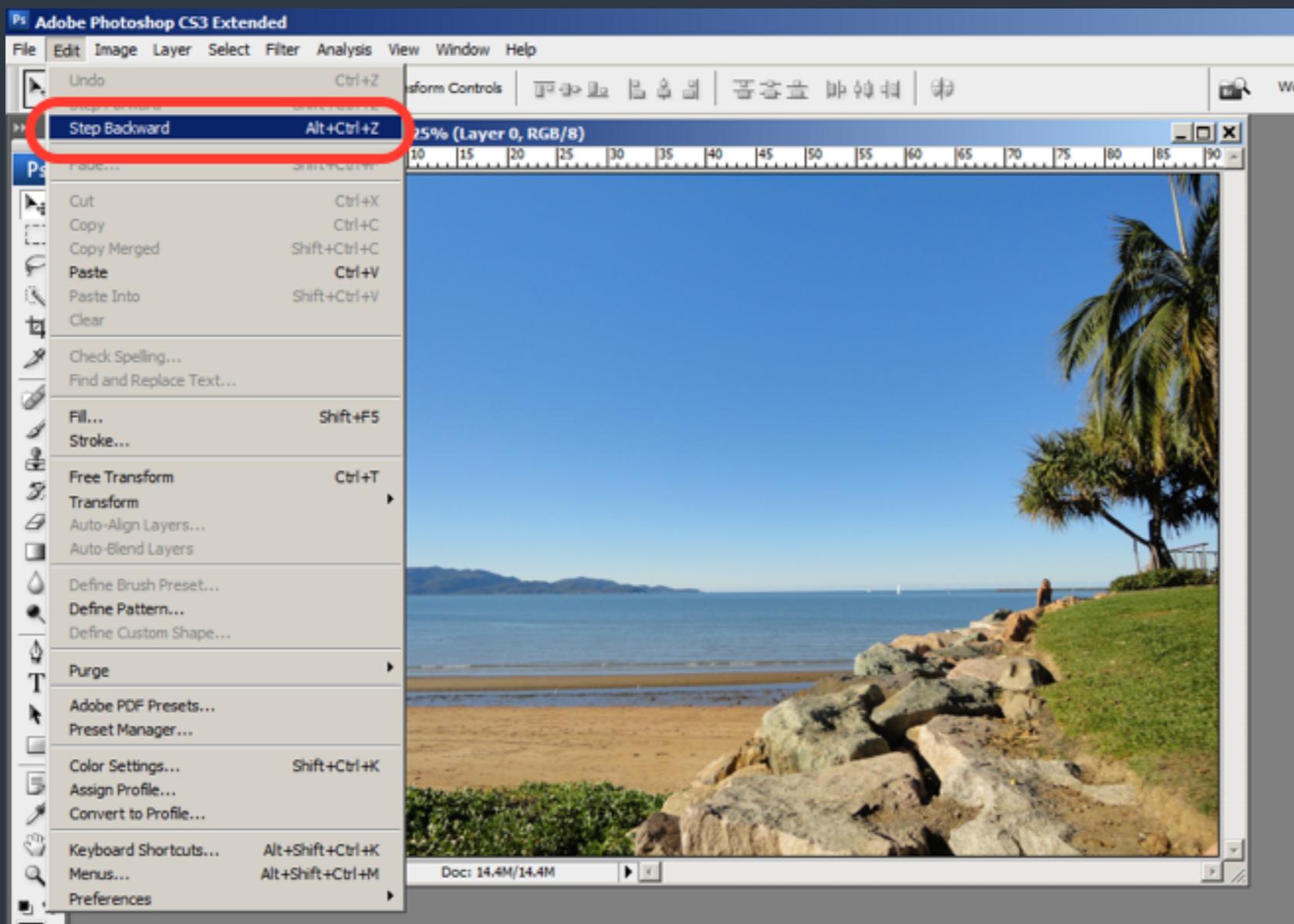
HEURISTICS

3. User control and freedom



HEURISTICS

3. User control and freedom



HEURISTICS

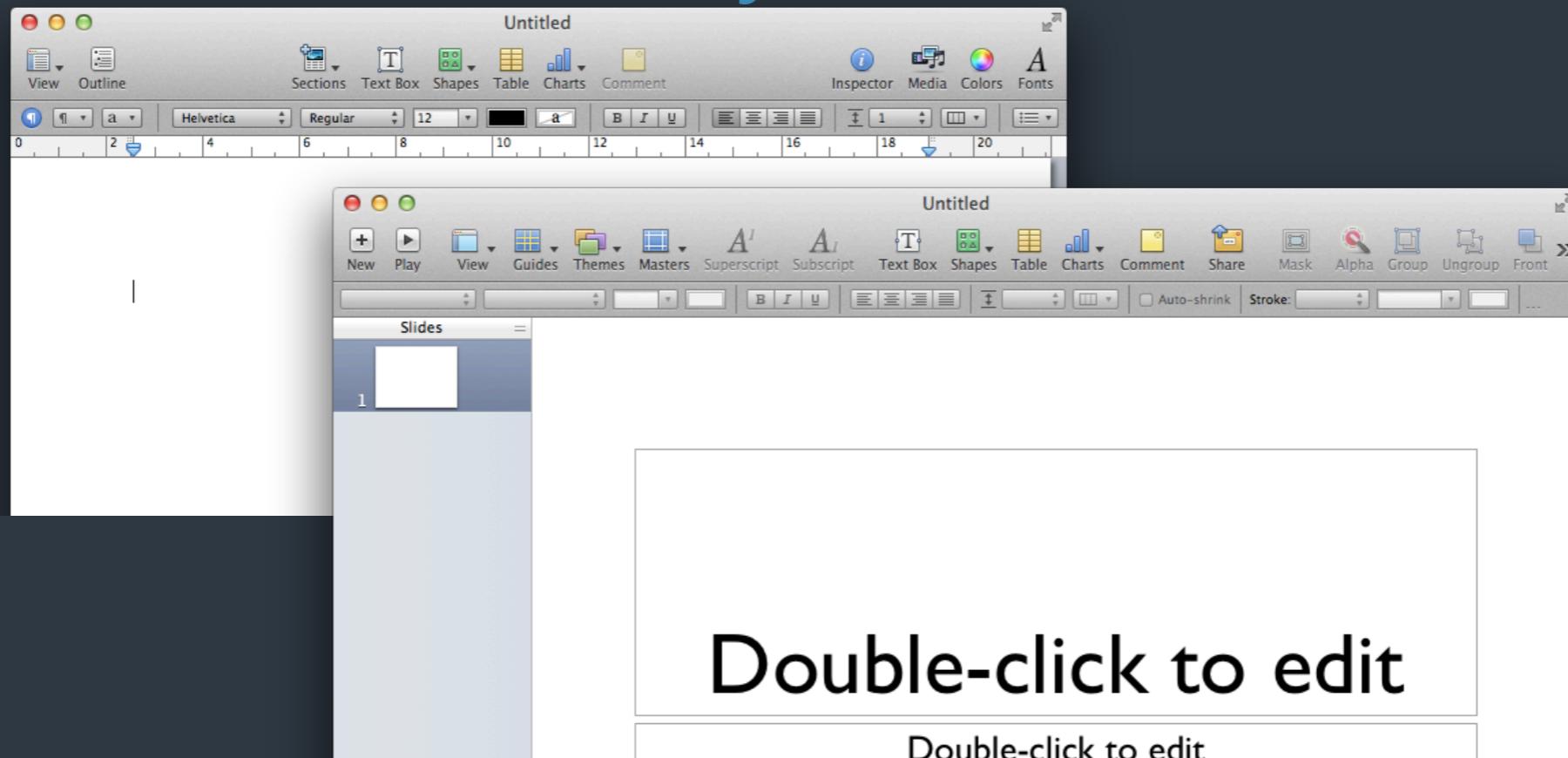
4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing.



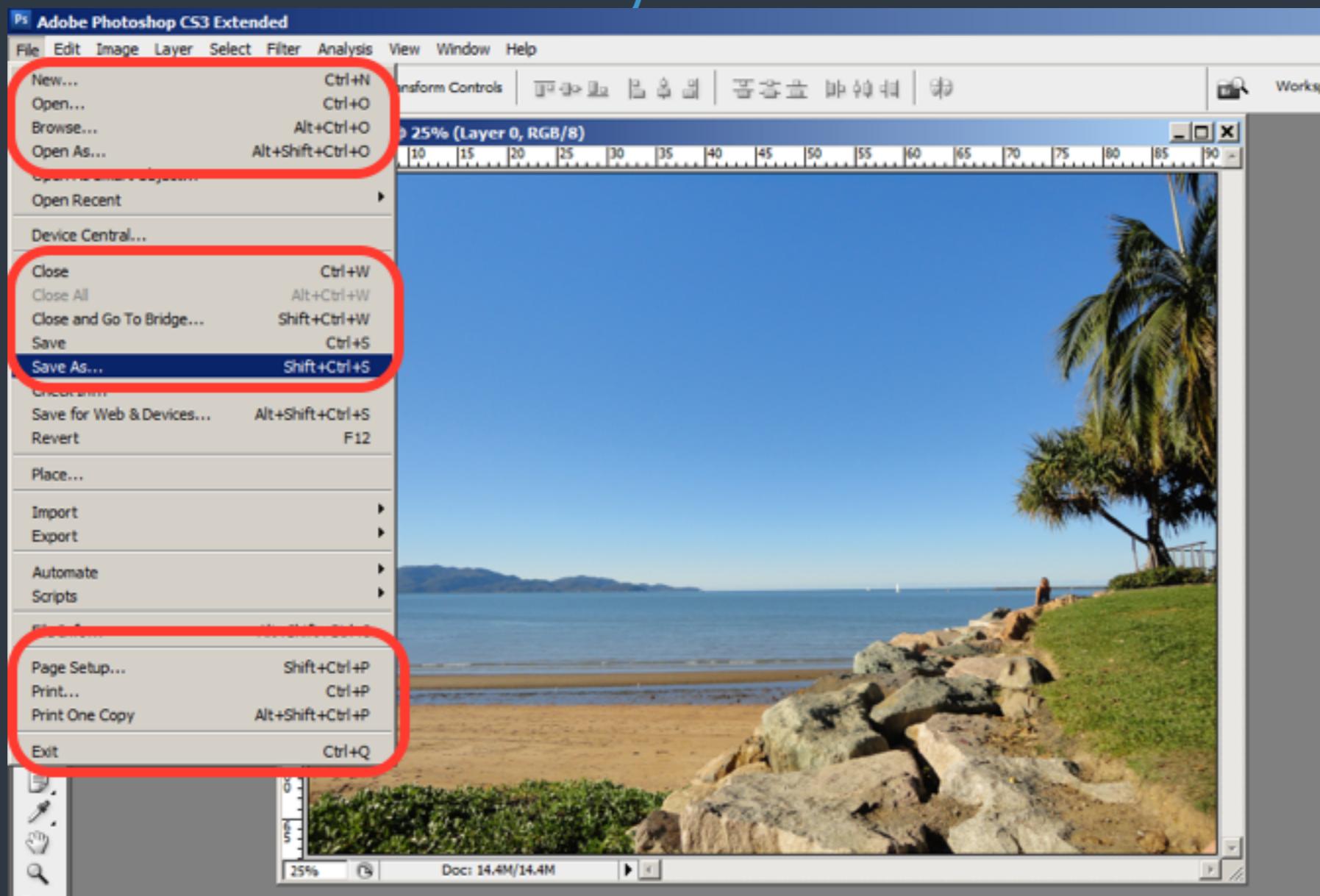
HEURISTICS

4. Consistency and standards



HEURISTICS

4. Consistency and standards



HEURISTICS

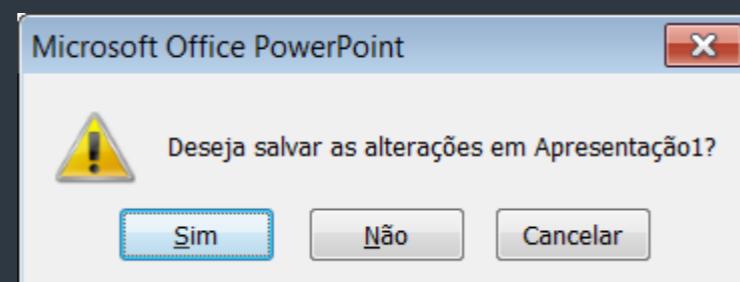
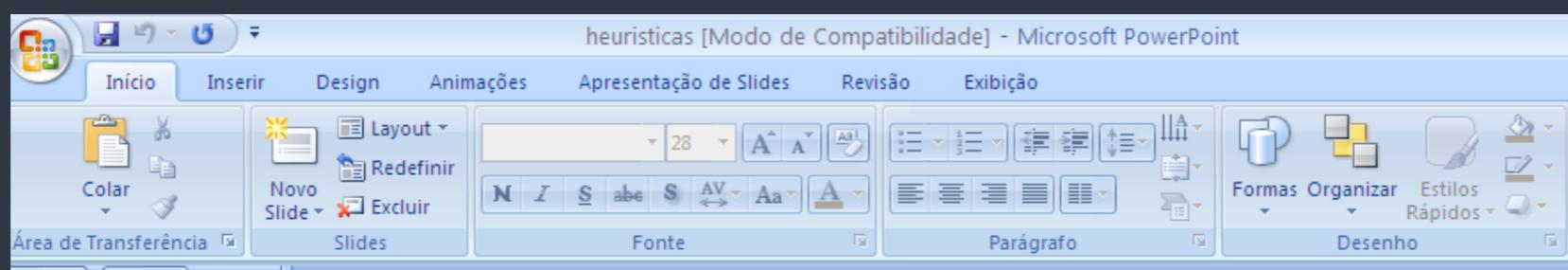
5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.



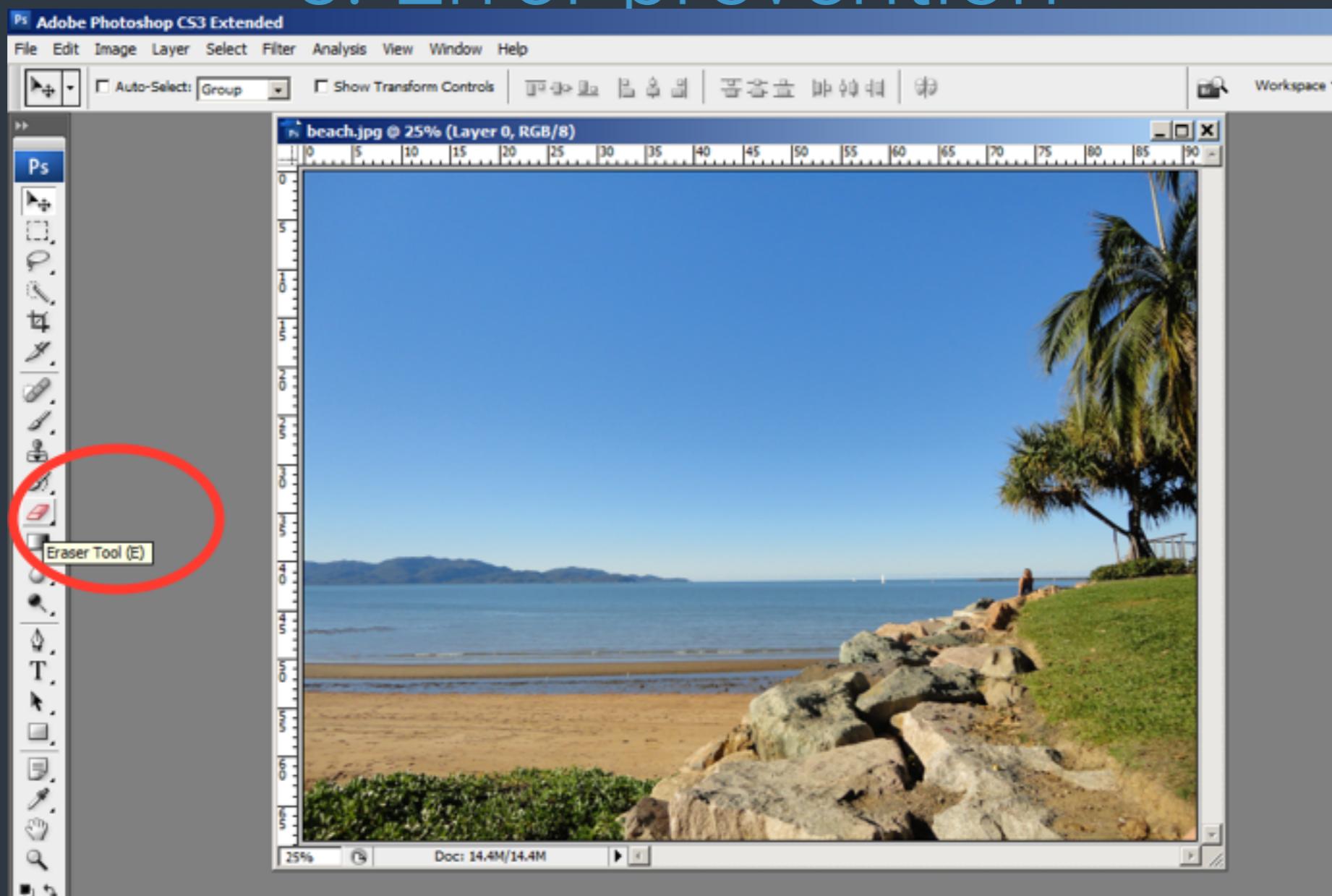
HEURISTICS

5. Error prevention



HEURISTICS

5. Error prevention



HEURISTICS

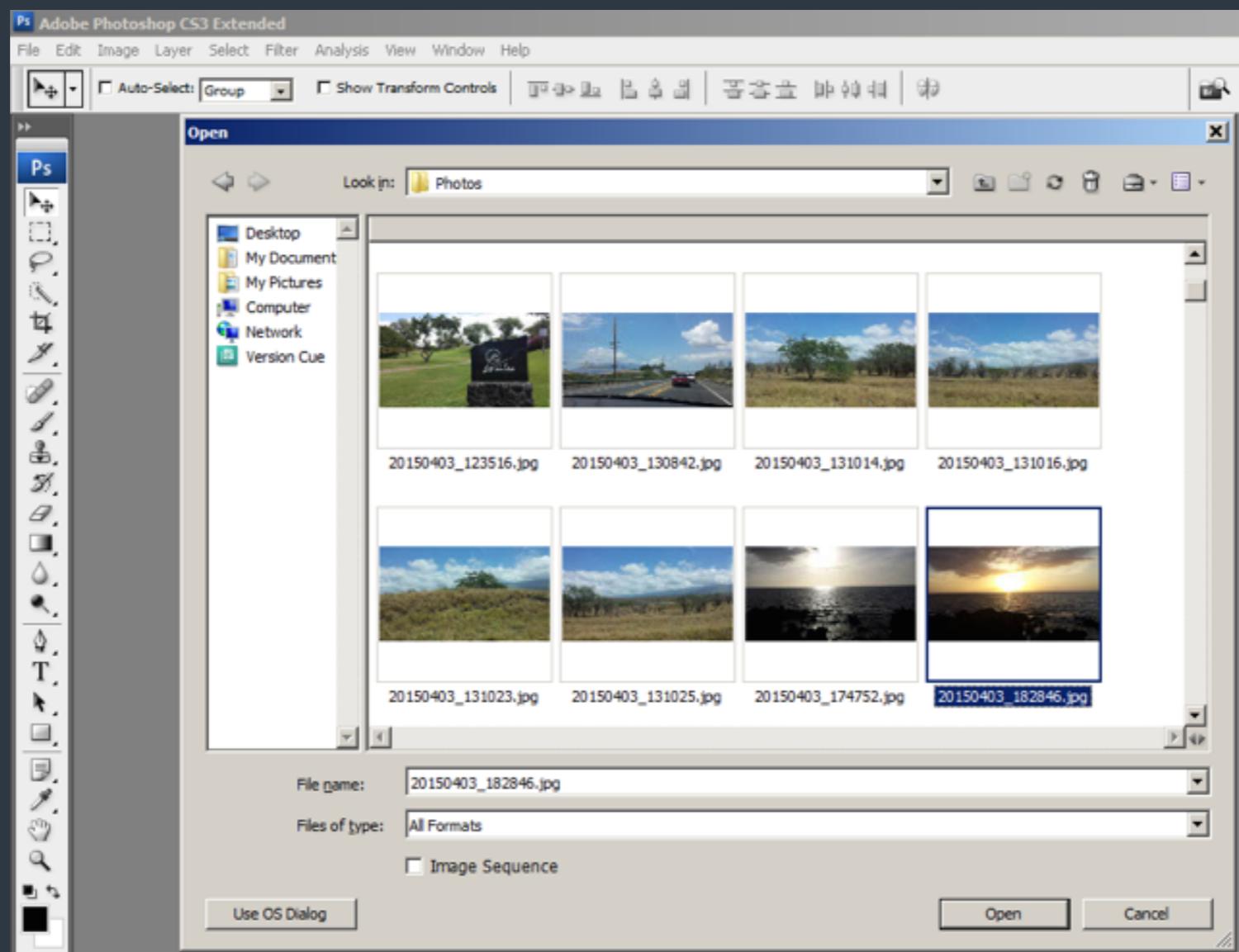
6. Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.



HEURISTICS

6. Recognition rather than recall



HEURISTICS

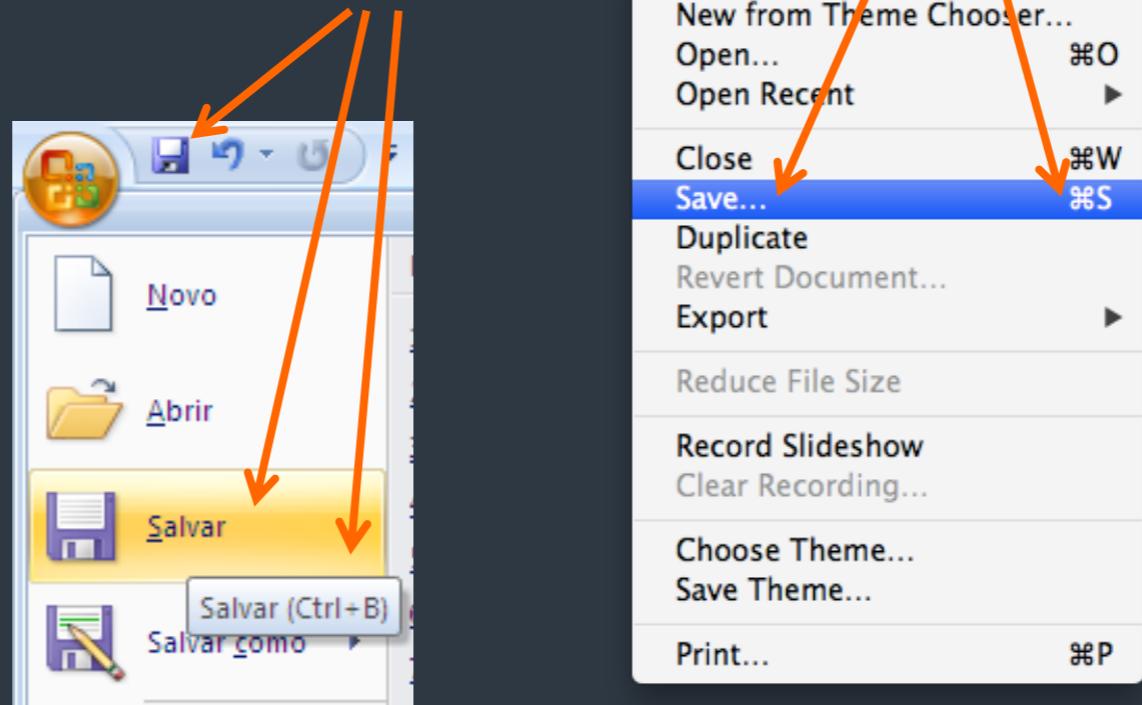
7. Flexibility and efficiency of use

Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.



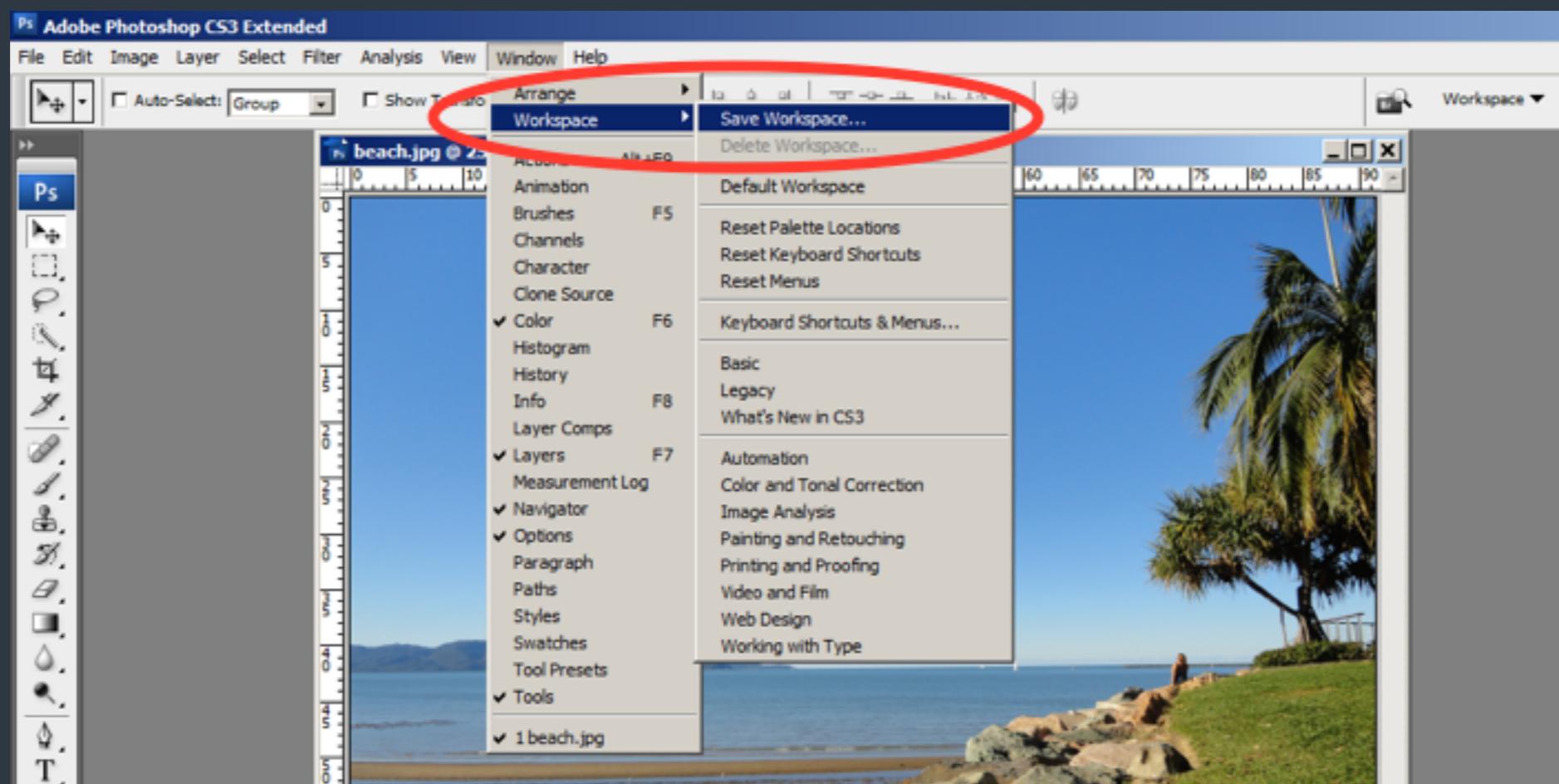
HEURISTICS

7. Flexibility and efficiency of use



HEURISTICS

7. Flexibility and efficiency of use



HEURISTICS

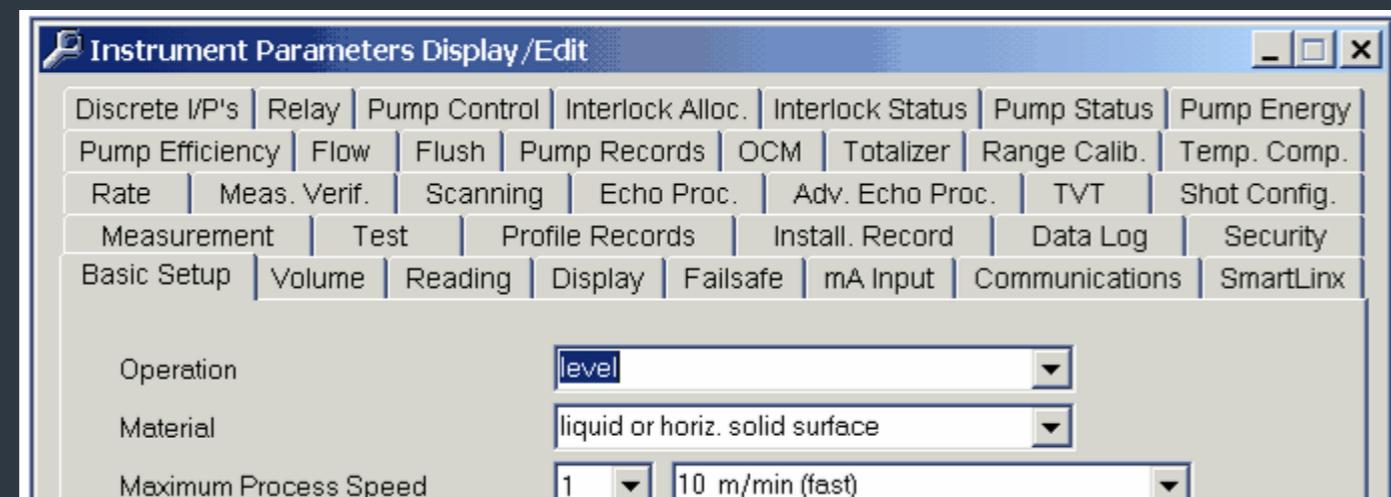
8. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.



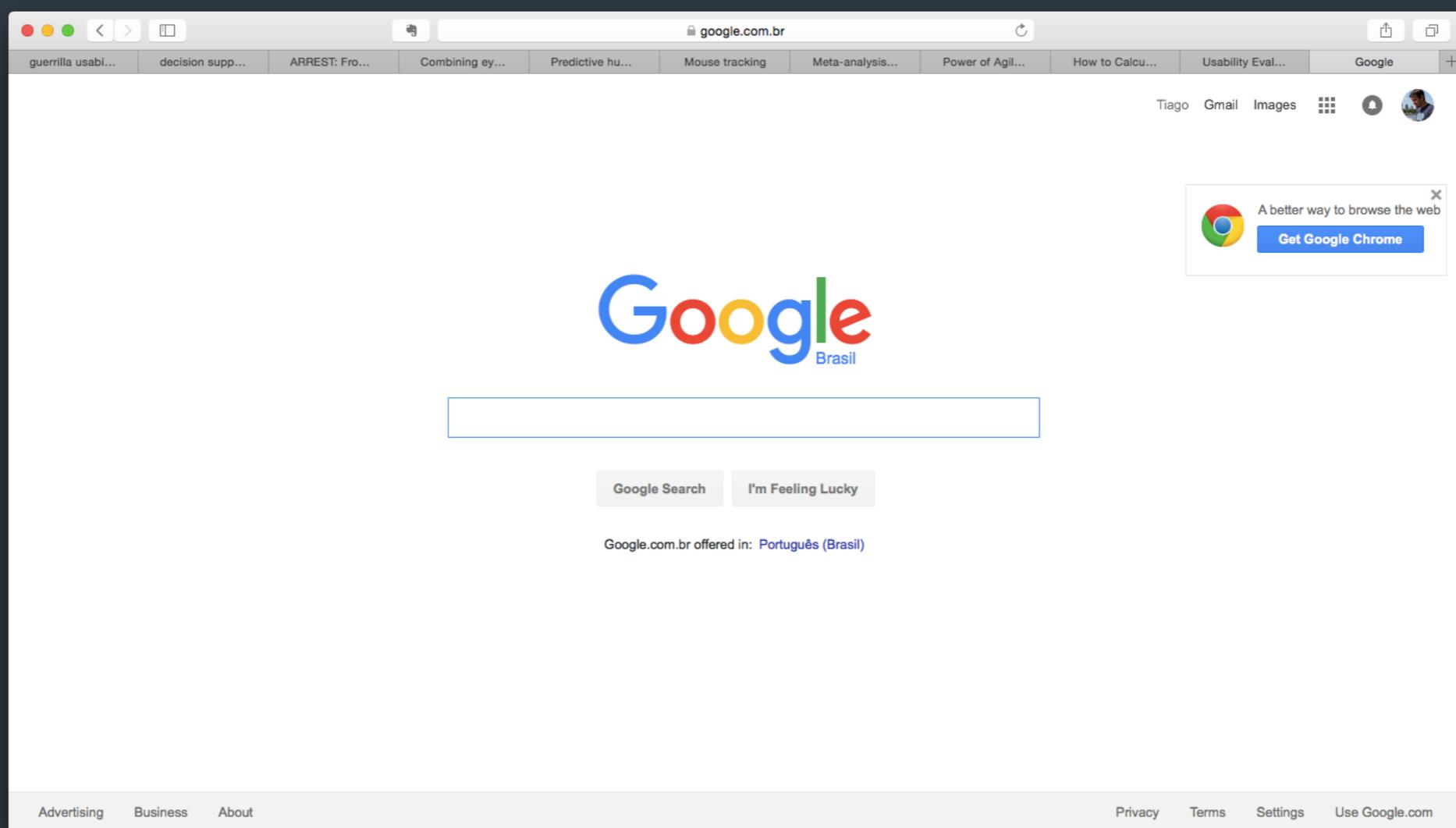
HEURISTICS

8. Aesthetic and minimalist design



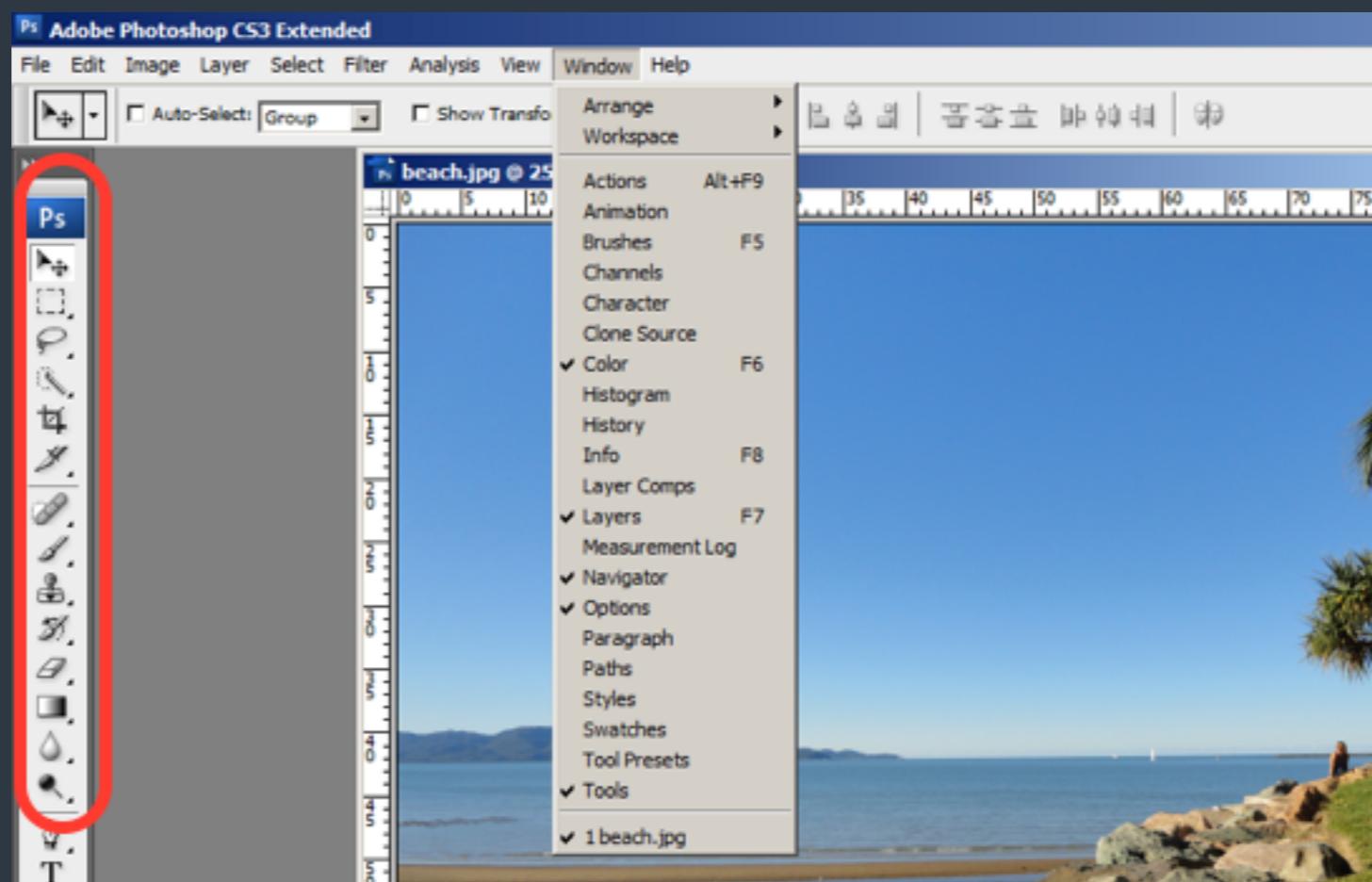
HEURISTICS

8. Aesthetic and minimalist design



HEURISTICS

8. Aesthetic and minimalist design



HEURISTICS

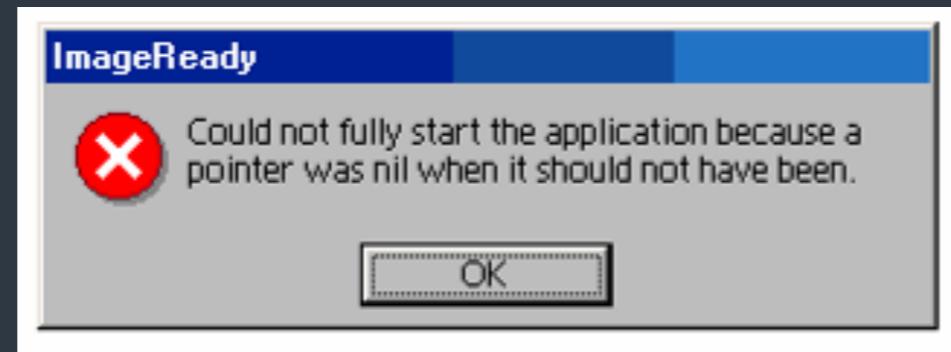
9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.



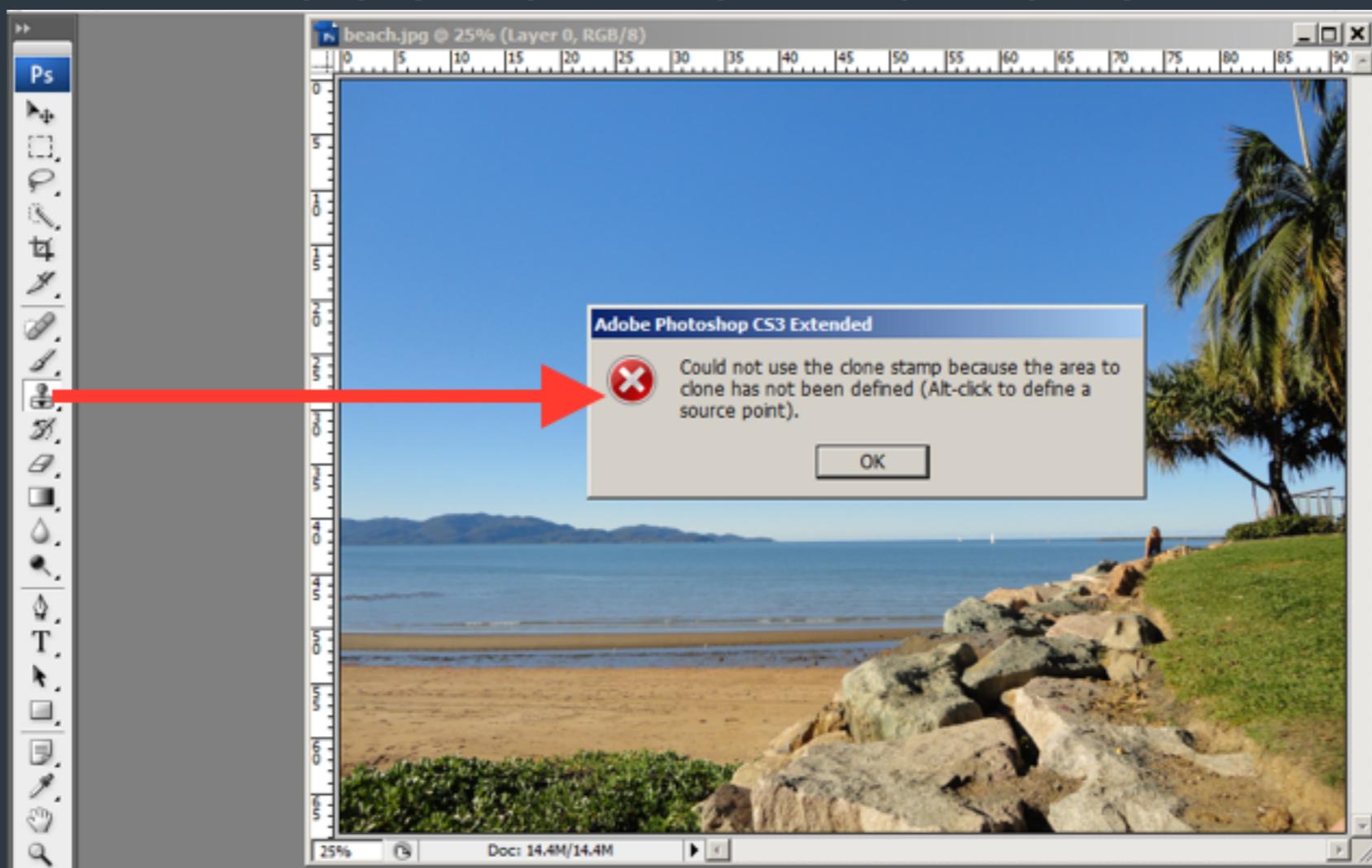
HEURISTICS

9. Help users recognize, diagnose, and recover from errors



HEURISTICS

9. Help users recognize, diagnose, and recover from errors



HEURISTICS

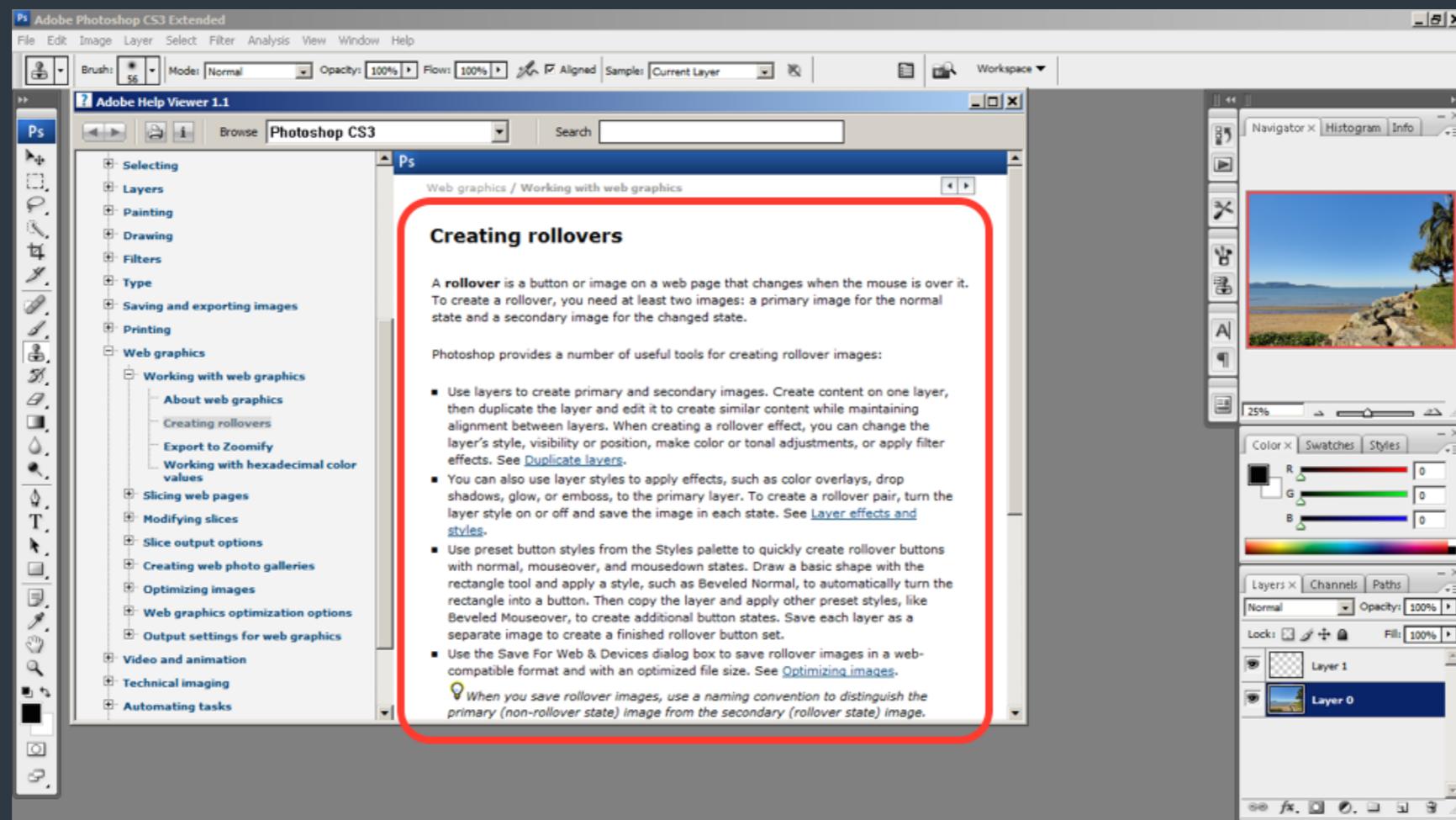
10. Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.



HEURISTICS

10. Help and documentation



HEURISTICS

Severity Rating

- Used to allocate resources to fix problems.
- Combination of:
- Frequency
 - Impact
 - Persistence (one time or repeating)
-
- Should be estimated after all problems have been seen
 - Independently first is good



HEURISTICS

Severity Rating

- 0 – don't think this is a usability problem
- 1 – cosmetic problem
- 2 – minor usability problem
- 3 – major usability problem; important to fix
- 4 – usability catastrophe; must fix



References

- **How to Conduct a Heuristic Evaluation.** <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>
- **10 Usability Heuristics for User Interface Design.** <https://www.nngroup.com/articles/ten-usability-heuristics/>