# A user interface checklist

Vlad Golovach and Alexander Belyshkin. Translated from Russian; a copy of the original is on romver.ru. The checklist describes desktop software of its time.

---

Checklists are an effective and economical way to raise the quality of software products. Applied to the interface, they do not require expensive usability testing, and a fair number of interface checklists can be found freely on the internet.

Naturally, each case calls for a checklist of its own, because it has to take into account the particular software being developed and what the development tools can do. This checklist is therefore more of a template that sets out the main sections, all the more so because the items carry no scores. Even so, for companies that do not check ergonomics and consistency at all, even a list like this can be a serious help in improving the ergonomics of a software interface.

This checklist is absolute: none of its items calls for a subjective answer such as “the navigation is done well”. Anyone can therefore check an interface against it, and the checker needs no special training. The other side of an absolute list is, of course, that it is bound to be incomplete.

It also matters that no checklist can make an interface good. At best a list guarantees the absence of gross errors. That, on the other hand, a checklist does very well.

## Buttons

- Every button that starts an action is labelled with a verb in the infinitive (“search”), not another part of speech or verb form (“done”). A button may be labelled “OK” only if no verb fits.
- The clickable size of a button matches its visible or logical size.
- Adjacent buttons are separated by empty space where a click does nothing.
- No two different button states look the same.
- Unavailable commands do not disappear from the screen; they are disabled.
- Frequently used buttons carry both text and an icon; rarely used buttons carry text only.
- Modal dialogs have no Apply button.

## Input fields

- Input fields are prefilled with the most likely values.
- If a field takes a number, the limits of the range are shown in a tooltip.
- If a field takes a number from a limited range, it has a spinner.

- A field is no shorter, and where possible no longer, than the data entered in it.
- A field meant for a noticeable amount of text is multi-line.
- Multi-line fields are as tall as they can be; there is no room left to enlarge them.

## Lists

- Lists are preset to the most likely values.
- A list of more than 50 items has a filter or a search mode.
- There are no frequently used short lists (fewer than five items); these are shown as groups of radio buttons or checkboxes.
- A list is no narrower than the items in it.
- List items are sorted: by structure (shared attributes), alphabetically, or by frequency (only in lists of fewer than seven items).
- In an alphabetical list of more than 50 items, the first three items are the most frequent ones. They also appear in their alphabetical places.
- Multi-line multiple-selection lists have a checkbox beside every item (there are no old-style lists).
- Multi-line lists are at least four lines tall.
- Where there is room, extended combo boxes are used, not single-line ones.

## Checkboxes and radio buttons

- A group of more than ten checkboxes has one more that sets or clears them all.
- In a group of radio buttons one is always selected by default.
- Checkboxes and radio buttons are arranged vertically within their groups.
- If a window contains, apart from its terminating buttons, only a set of radio buttons, double-clicking a radio button selects it and closes the window.

## Interaction

- When a long operation (more than a minute) finishes, the system beeps through the computer’s built-in speaker.
- If the interface uses no direct manipulation, the system has no cursors of its own. If it does, custom cursors are used only where the operating system has no equivalent.

## System messages and error handling

- In input forms, values are validated as they are entered. If the data is incorrect, the system tells the user at once, without waiting for the whole form to be filled in.
- A message about incorrect data appears next to the control that holds it.

- The text of such a message does not say that an error has been made. It only tells the user what type and format of data is acceptable.
- A problem message has three parts: the first briefly describes the problem, the second says how to solve it, and the third says how to avoid it in future.
- Status messages (“Synchronisation completed successfully”) appear only in the status bar.

## Keyboard

- In input forms, pressing Tab moves through the form in the correct order.
- A form is submitted not only by its terminating button but also by pressing Enter in its last field.
- The most frequently used controls, menus included, have keyboard shortcuts.
- Every menu item has an Alt combination (shown by underlining).
- Alt combinations and hot keys are the standard ones.
- If there are more than 40 hot keys, the interface offers a way to change them.
- Pressing Tab moves from control to control within a form from top to bottom and left to right.

## Visuals

- Shadows on all controls fall in the same direction: down and to the right.
- Colour is never the only indicator; where it is used, the system provides another indication as well.

## Icons

- Within a group of icons, no two are similar in colour and shape.
- No icon gives a standard meaning a non-standard picture.
- Icons contain no text.
- Icons with the same meaning in different sizes share the same features or picture.

## Windows

- Resizable windows have a resize indicator.
- A window’s title matches the name of the element that opened it. If that element has no explicit name, the title gives the name of the screen form.
- The type of each window (modal or modeless, whether it can be minimised or maximised) was chosen deliberately, to suit the users’ tasks.
- Dialog windows have no menus or toolbars.

- Apply buttons are used only in palette windows (in place of OK buttons).

## Status bar

- The status bar shows only the current state of the system and buttons (not looking like buttons) for functions meant for experienced users only.
- Progress indicators appear in the status bar. The exception is wizards, where they may appear inside the window.

## Menus

- Menu items begin with a capital letter.
- Every top-level menu item opens a drop-down menu.
- There are no more than two levels of submenu.
- If a menu has icons, only the most frequently used items carry them.
- Items that open submenus look different from terminal items.

## Context menus

- Every object visible in the interface has a context menu specific to it.
- A context menu has no more than ten items.
- Items in a context menu are ordered from most to least frequently used.
- Every item in a context menu is also available elsewhere in the interface; no command can be called from a context menu only.

## Structure of forms

- A group of interactive elements (form fields, menu items and the like) has no more than seven of them.
- The Cancel button is always the rightmost.
- Multi-page forms say that they are multi-page; the user always sees how many screens remain (“Screen x of y”).
- If a form has several buttons, one is the default. If it has only one button, that button cannot be the default. Buttons dangerous to the user are never the default.
- If there is room in the window, the most frequently used terminating button is larger than the others.
- Buttons sit in the section they act on directly.
- Terminating buttons, the ones that control the window, are placed in a row at the bottom or in a column on the right.
- Buttons that apply to a whole set of tabs are placed outside the set.

- If a window or tab fills up automatically, for example with incoming messages, the name of the element that opens it shows the number of objects in it and, separately, the number of new ones. Example: Documents (8/3).
- Menu items and buttons that lead to further action by the user end in an ellipsis (…). “Save as…” needs one, because the user has to choose a file name. “About” does not, because the window it opens has no controls of its own.
- Labels of interface elements are placed consistently.
- Interface elements unavailable at the moment are disabled, not hidden.

## Input forms

- Every form that collects information has options such as “Other” and “Not applicable”.
- All required fields are marked, and the marking is explained.
- Every form that collects information says why the data is collected and what will and will not be done with it.

## Text

- All main interface elements have tooltips whose text states the result of using the element.
- The interface contains no jargon.
- The interface contains no negative wording. A checkbox “Do not show notes” is unacceptable; it should be “Show notes”.
- No element has different names in different places (the interface glossary is not only written down but verified).
- Every confirmation names the object on which the action is being confirmed.
- For readability, long numbers are split into groups of three digits with a non-breaking space: 1 234 567.
- List items follow one rule for capitals and full stops. All items begin with a lower-case letter and end without punctuation, except the last, which ends with a full stop. If even one item contains more than one sentence, all items begin with a capital and end with a full stop.
- Every list is preceded by at least one paragraph of text.
- In tables, all columns of numbers are right-aligned.
- There is no full stop at the end of a heading (if it is set apart from the text), a figure caption or a table entry.
- Labels of interface elements begin with a capital letter and end with a colon.