



Design principles



Software can be designed to simplify tasks and to create a positive overall experience for users. Thoroughly understanding the goals of users and stakeholders and designing software with those goals in mind are the best approaches to successfully delivering products that will delight customers.

The following checklist identifies core principles and best practices to assist software designers and developers to produce software that stands out in the marketplace.

1. **Concepts from the product's subject domain (for example, systems management) should be central and apparent in the software design.**
The relationships among user interface objects in products should be accurate, so that users can rely on their previous experience in the domain when learning and using the software.
2. **Keep it simple.**
Give easy access to the features that most users will need most of the time; features used less often or by only a subset of users are placed less prominently.
3. **Optimize the design for the most frequent or important tasks.**
Understanding how users will use the software you are designing is critically important. Designers should use that understanding to anticipate the information, task flows, and features that users require at key points within the user experience.
4. **Make the interface accessible and visible to users.**
Design your Web site or application so that users can view and easily access objects or information within the interface. Choices should be visible to users rather than hidden with cryptic key sequences. When objects and choices are immediately visible, users learn and complete work tasks efficiently.
5. **Use proper default values when supporting complex tasks.**
Use good defaults so that users can complete tasks relatively easily and quickly rather overwhelming them with choices.
6. **Be flexible.**
Let users customize the application to meet their unique needs. For example, specialized users could be given a way to make secondary choices more prominent in the product. Also, don't limit users by artificially restricting their choices to a "correct" sequence. Flexibility is also enhanced by letting users select options in various sequences and in letting them modify default values.
7. **Keep your users informed and in control by providing informative and timely feedback tailored to the current situation.**
For example, progress indicators let users know that their system is healthy and is carrying out their request. At a minimum, alert users when they take actions that will result in the loss of data. (Ideally, such choices would be disabled or even eliminated when they are inappropriate in a particular situation.) All the information included in the feedback should be meaningful to the average user.
8. **Things that look the same should behave in the same way, and an action should always produce the same result.**
Avoid modes that change the effects of actions. Where modes are unavoidable, make them clear to the user and easy to change.
9. **Provide the ability to undo and redo actions.**
Applications must provide users with the ability to freely explore applications (which includes the ability to make mistakes) without fearing permanent damage.
10. **Make your application predictable by using industry standard user interface conventions wherever possible.**
For example, users should be able to use standard selection models and keyboard equivalents like Ctrl+C and Shift+→ (to copy the currently selected object and extend the current selection one unit to the right, respectively) everywhere they work with data. Use a common set of design patterns and guidelines so that users don't have to relearn how to perform common tasks.
11. **Always keep your target users in mind as the product is designed.**
Developing personas and identifying and defining the roles your users fulfill can help you understand how various roles will use your product. Designs based on typical workflows and the other software that users might use in conjunction with yours will delight users.
12. **Avoid adding features just so they can be ticked off a list.**
Remember that every additional feature equals a set of choices added to what is likely already a wide array of choices. Too many choices can overwhelm users.
13. **Design your user interface so that it can be localized for other geographies without redesigning the interface.**
For example, don't crowd form controls too tightly so they can accommodate longer German translations.
14. **Consider persons with disabilities when designing your applications.**
Many users of your product may have impaired vision or physical limitations that affect their ability to use a mouse or a joystick.
15. **Design the application so that contextual help is available to users when they need it.**
Users should not have to refer to Help constantly to complete their tasks.
16. **Bring objects to life through good visual design.**
The goal of visual design in the user interface is to surface to the user in a cohesive manner all aspects of the design principles. Visual design should support the user model and communicate the function of that model without ambiguities. Visual design should not be the "icing on the cake" but rather an integral part of the design process. The final result should be an intuitive and familiar representation that is second nature to

users.

17. **Create user interfaces that promote clarity and visual simplicity.**

The following visual design principles help create that effect:

- **Subtractive design:** Reduce clutter by eliminating any visual element that doesn't contribute directly to visual communication.
- **Visual hierarchy:**
Understand the importance of users' tasks and establish a visual hierarchy of these tasks. An important object can be given visual prominence. Relative position and contrast in color and size can be used to convey task importance.
- **Affordance:**
When users can easily determine the action that should be taken with an object, that object displays good affordance. Objects with good affordance usually mimic actual objects.
- **Visual scheme:**
Design a visual scheme that maps to the user model and lets the user customize the interface. Do not eliminate extra space in your image just to save space. Use white space to provide visual "breathing room."