Principles of software design

These principles are intended to support the creation of software that is easy and intuitive to use, and uses minimal resources (especially CPU cycles and RAM).

- If only several simple APIs of another big library are needed, try to extract them and add them to your program instead of depending on the whole library whenever possible. (Beware of license)
- Only use libraries from other desktops when they are small or efficient enough and have few dependencies.
- Only create a daemon if there is a really good reason.
- Basic configuration options of applications should be editable via GUI whenever possible.
- Keep the user interface simple and intuitive. Never design some geek-style GUI.
- Consider the conventions of both GNOME and Windows, and try to follow the habit of most users. Do not deliberately make the GUI quite different only because you want to be different from Windows. Usability is always the top one concern. Windows might not be good in some areas, but like it or not, most computer users in the world are used to it. Trying to fight your users is apparently unwise.
- Try to shorten the startup time since this greatly affects user experience.
- Try to keep maximal compatibility with lower versions of the languages or platforms used. Try to make features requiring higher versions optional with proper conditional compilation and compatibility macros.

This page uses content from LXDE Design Principles, written by Mario Behling. Used with permission (see talk page).