

How to Spot a Bad UI

You can trace most things that make a UI ineffective or unattractive to one of five causes:

1. A bad UI app doesn't do the correct task or doesn't do the task correctly. Remember that the user is task oriented. The first thing the UI must do is help the user complete his or her real-world task. Everything about the UI must support this principle. This means the programmer must understand what the user needs to accomplish. It's not always easy to determine the user's requirements—programmers have struggled with this for years-but there isn't much point in programming unless you know what your program needs to do.

2. A bad UI is too slow or crashes. Performance is a UI issue. Most apps have a performance range that is fast enough and making it faster has little effect. There is also a performance range at which the application is so slow it is unusable. For apps that fall between these two extremes, it's more pleasant to use one that features better performance.

Before you can fix a slow UI, you must know what is slowing it down. It can be tough to find the slow parts of your code, but it is by fixing this code—and often only this code—that you can make performance tuning pay off. This is especially true if the user runs mainly these portions of the application. Time spent speeding up any other part of the application is wasted. Unfortunately, most apps require that you write and benchmark them before you can determine which parts perform too slowly.

Of course, nothing is more frustrating performance-wise than an app that crashes unpredictably, particularly if the user loses data. This means that good programming practices, error handling, and testing play an integral role in creating your app's UI.

3. A bad UI makes you think about how to accomplish even the simplest tasks. Flow is an important UI design concept. It describes the degree to which a user can focus without interruption on the job at hand.

Flow is so closely connected to a specific task that it's impossible to provide a recipe for achieving it. However, there are a few things that

generally break the user's flow: message boxes, inability to undo mistakes, excessive movement from keyboard to mouse or eye movement to and away from screen, anything that requires a cheat sheet, and lack of predictability. Similarly, a slow application annoys a user because it breaks the flow of his or her work.

4. A bad UI is stupid. Computers are stupid, and it is the programmer's responsibility to make them work in a way that isn't stupid. Stupid applications don't remember what they need to know or require the user to confirm operations unnecessarily.

For example, think about how quickly you would fire an assistant if you asked him or her to do a job for you, only to have the assistant ask you if you really wanted him or her to complete the task. And later, after the assistant completes the task, what if he or she were to ask you if you want him or her to keep this task completed. Yet computers do this all the time when they ask if you really want them to do something or whether you want to save certain changes.

The keys to writing a smart application are to give it a memory; let the user undo operations so you don't need to confirm what the user requests, and handle the common case without user intervention. The last key requires making assumptions the user can override. For example, consider the Save/Save As approach common to Word and other applications. Word assumes that you mean to save to the same file you opened unless you specify otherwise with the Save As command. Word only presents a dialog when the user requests the uncommon operation of saving to a new filename.

5. A bad UI is sloppy or ugly. This article shows you some specific ways to fix or avoid creating a sloppy or ugly interface. VB lets you write professional-looking applications, regardless of your programming experience. Most of these ideas take some effort at first, but some become second nature and require little effort to implement once you internalize them. Your UIs will continue to improve as you develop good habits, increase your sensitivity to the user, and learn additional skills.