

Quote:

Since U++ is greatly focusing on less time and code needed to implement a project, I feel like sharing a few thoughts.

Here is a short list of what the majority of people like me are looking for (in no particular order):

- 1) Good looking apps
 - 2) Data Visualization controls for Quick and easy data handling in forms, grids, reports. Things like, sort, filter, search, column moving, grouping, report design etc. etc.
 - 3) Ability to use external libraries (like DLLs) for special stuff, e.g. encryption
 - 4) re-usable forms, windows, controls
 - 5) Exporting/Printing to various types Excel, PDF etc.
 - 6) Auto resizing & docking of controls and windows.
 - 7) Transactions support
- Cool SQL SQL SQL
- 9) Stored procedures support
 - 10) Easy string, date, manipulation
 - 11) Easy filesystem handling
 - 12) Easy networking stuff (ftp, smtp, http) etc.
 - 13) XML support
 - 14) Auto complete
- etc. etc.

I strongly believe that this would greatly improve U++ popularity even though it uses the scary for many C++.

It just seems to me that U++ is closer than anything I have seen until now to generic & still time effective philosophy, I feel that U++ has the potential to stand next to anything like .Net, Powerbuilder and the likes as long as it provides the "Widgets for Dummies" or "The tools for non-geeks" if you prefer.

Either we like it or not, many people out there are VS addicts and you just can't make them get rid of some of their habits.

For example, "double-click" a control and code an event.

And last but not least, all these don't have to be FREE, actually there are many GPLed projects out there that offer additional modules or functionality for a price.

I also thought a similar manner, but over time I began to understand how to develop leaders U + + are doing.

First, I sorrow for the absence of an abundance of modules and plugins 3D-Party developers.

But then even accustomed to the fact that you can simply open the source code for all modules

and learn how to operate different things.

And this was much more valuable.
