

0. The choice of which editor one uses is very personal.
So to me any kind of not being able to chose your own editor like an IDE that does not offer the user to chose his or her own editor instead of the implemented one is unprofessional.

1. It doesn't really matter which one you particulary chose.
If you use/need it a lot, e.g. for professional usage, you'd better chose a powerful one.

2. Knowing to use your editor efficiently is way more important than the choice which one to use.

You'd better spent your time on learning an editor than to elaborate which one you chose.

3. Most efficient usage of any editor is trying to avoid usage of the mouse but using as many keyboard commands as possible.

4. Knowledge cannot be bought. Any efficient workflow always presupposes preceded learning. 2. means you have to learn its usage. And the more powerful a tool is the more learning effort is needed to use it efficiently.

But learning editors' usages is not the target.

Use one for the actual work is.

So you'd better pick one in the first place you don't need to change.

That leads to three characteristics:

a) It needs to be powerful enough to fulfil your current, and maybe future needs.

b) It's currently available on as many platforms as possible, particulary your current main production platform, which you chose for many other, more important reasons than the editor; so you don't need to change the editor everytime you switch platforms.

c) It has to be still available in the future.

Best was open source. That's almost a guarantee it's available for at least the most popular platforms, and will be also in the future. extra: you don't have to pay for it.

5. By principle there are two kinds of (visual) editors match two principle styles of workflow:

a) vi style. two separate modes for editing, and insert; reduced to the minimum needs. anything not really be done with an editor is not be done by it but with other tools outside of it (unix philosophy - small, specialized, independent but concatenateble modules)

disadvantage: hard to get into, especially when you're used to other editor styles

advantage: After reaching a certain level of learning, you may master it without much more learning effort.

b) emacs style. both modes are available at the same time. many features are already integrated, or available as add-ons to integrate as many features as wanted in one tool, everything within one monolithic suite (IDE style)

advantage: very easy to get into. almost no learning is required to start using it.
disadvantages: learning cannot be avoided. to use it efficiently you have to have some learning effort anyway.
for every other emacs like editor you have learn its usage new (almost all editors except the small vi(like) family I group as 'somehow emacs like.')

with emacs itself learning may never end; its amount of key-commands is large, way larger than vi's. and many new ones are added with each add-on.
therefor it offers other powers.
it's simply another kind of style.

However:

Everyone needs to find out for him- or herself, which style suits best (but don't make a religion out of it.)
Pick one, and learn it well is way more valuable than to search endlessly for the 'right one.'