A decade of development

This example typifies the approach of free software activists towards the concept of paying for the software. They insist that the source code of the software must be made available to everyone and those additions and insertions to the source code should provide a way of customizing the software according to the needs of each individual. Similarly, Lessig's approach to making the content free is discussed as well.

Lessig's approach is the backbone of what is called the " creative commons" license. According to this paradigm, content should be free of commercial attributions and once the source code or the content is developed, anyone should be free to add his or her code or content to the existing base and use the modified version without paying any fees. Of course, the creative commons license specifies how the original developer needs to be acknowledged for his or her work, and in the later years; this became a bone of contention as well.

The most famous example of free content and one that can be editable by anybody in the case of Wikipedia. This is an online encyclopedia that anyone can edit and add information to the existing body of knowledge. Considering the fact that this site is one of the most popular sites on the internet and the visitors to this site run into the millions, the concept of free content and open knowledge has indeed caught on.

However, as the article points out, there were several hiccups along the way as evidenced by the instances of license incompatibility and economic sustainability. These issues have to do with the fact that free software is not economically viable and the developers of the free content do not get enough by way of fees and royalties and hence they do not have an incentive to develop more of the same as happens in the case of

conventional software developers. Further, there are issues of license incompatibility that happen because of the way in which the software is used to add or modify the source code without proper attribution to the original developers.

In conclusion, whatever may be the pitfalls associated with free software, the future of free software looks bright as the next generation of products and gadgets that would be built would be based on the application of community-driven software instead of monopolized software. Hence, the software developer community must take cognizance of this changing trend and incorporate elements of the free software model instead of resisting the same.