Why Software Sucks

Dr. Ronald Krawiz

Ian Fordyce

05/06/2022

When product limitations restrict quantity or access, it creates an environment of exclusivity. Exclusive items can be challenging to work with and maintain, as they are not commonplace and not well known to the general population. As a result, it can be difficult to communicate with an outsider how a specific item works, or how to properly maintain it. This was definitely the case with the development of computer science, and the interactions between computer developers and computer users. During the early years of development, computer systems were completely foreign to most of the general population. They were accessible in few places, such as science labs, computer centers, and schools. At that time of early computer development, the experts who maintained the computers and wrote new programs for them, rarely had to explain how computers worked to others who did not have the same knowledge. These experts were the main resource to fix any issue. But as time progressed, computers improved and became more accessible. The general population gained access to these systems, causing software developers and manufacturers to interact more with individual users who did not have the same knowledge base. This caused serious communication gaps between the two groups, bringing two main problems to light: ease of use and security.

“Why Software Sucks…and What You Can Do About It” was written by David S. Platt in 2006. Mr. Platt has worked with computers and programing for over 20 years, running Rolling Thunder Computing, a computer education and consulting business. He also taught software development at Harvard University. In his book “*Why Software Sucks…*”, Mr. Platt explains that despite all the good that has come from modern technology, there are still things that are lacking in the world of information technology and computer science. He explains how these fields are designed in such a way that only people with extensive knowledge of computers will understand. The general public who regularly own and use computers, without the specific educational background in computer science, are at a disadvantage in effective use of computers as well as in the safety of their use.

With these issues in mind, Mr. Platt looked at the process and mindset of how computer programs are developed, and how the developers are not considering the actual user when creating these programs. When computer systems were first introduced, basic functions that we take for granted today, such as copying and pasting text or printing a document, were extremely time-consuming and complicated. As a result of this, developers were forced to learn the computer language by studying long and intricate commands, with some saying it was a right of passage, and that the developer had shown his worth. Even though technology has exponentially improved over the years, this mindset that the complicated way is the only way, has continued into how developers design their user interface. Developers think that because the program they are making is for a computer person, the user will understand their method of thinking and have the same in-depth knowledge they do. In actuality, the user has minimal-to-no knowledge of the program, and is only interested in what they want the program for. This can cause confusion and make the user not trust the developer.

Another area that Mr. Platt explains as lacking in today’s computer world, is how websites are created and how they are connected to the world wide web. Similar to programs, when internet websites were first introduced, only a very limited number of individuals had access to computers and websites. Mr.Platt explains the history of the internet from the 1990’s, when a Swiss scientist created the first link which allowed users to easily access their documents. At the time, workers asked if it was possible to connect documents from different computers. Using similar methods they had tried in the past, they were able to connect two computers together to share documents. This started a chain reaction, as computers became more popular in the general public. Computer scientists were able to connect computer systems together and allow anyone access to available information, either for free or by making users pay a fee to have access. But as they became more popular, a disconnect again continued between the general population and those who had been working with computers for decades before hand.

Now with easy access to information, more problems arose. Developers and users began to realize just how easy it was to steal information. Hackers and other criminals can take advantage of people by stealing their information. Hackers can put viruses on users’ computers to download ransomware and create destruction. As an ongoing issue today, the computer science world and the general public have continued awareness of this problem, and are developing solutions to routinely improve security. However, there are still too many loopholes that allow hackers to access someone’s computer, steal a person’s identity, and potentially ruin their lives.

In “Why Software Sucks” Mr. Platt suggests solutions for the problems he has identified. He reviews the different ways that could help improve the experience for both the developer as well as the average user. He recommends that computer science and programming education become more prominent in schools and other areas of education, so the general population can become more familiar with not only the basics, but become aware of the dangers. He recommends that programs become very specific to their users, and encourage all not to be afraid to ask very specific questions. He suggests improving the developer-user relationship by understanding what users want the program for, and be very specific in what they want the result to be. This will help customize and streamline the development process, improving communication between the world of computer science and the computer users.

In conclusion, Although there is the belief that software has improved since the late 1990s. There is still a lot of influence from that era on today's approach in how people view programming. Whether it be how a program should be written, the idea that everyone that works with computers thinks the same, and finally, the belief that the internet has become more secure as time has passed when none of these are, in fact, the case. There are several improvements that can be made in each of these areas as well as improve the security of the internet to prevent the users information to be stolen

References

1. Platt, D.S. (2006). Why Software Sucks…and What You Can Do About It. (1st edition). Addison-Wesley Professional.