Computer Science and Society: Is a New Social Contract Possible?

Alessandro ROSSINI
University of Bergen
P.O. Box 7803, N-5020 Bergen, Norway

Adrian RUTLE
Bergen University College
P.O. Box 7030, N-5020 Bergen, Norway

ABSTRACT

Researchers in the field of social studies advocate the need for scientific and social collaboration in the production of knowledge. In this article, the authors analyse factors preventing social engagement in the joint production of knowledge in the field of computer science. Deficient computer literacy is considered a primary obstacle to this joint production, as it leads to irresponsible computer usage and low quality software development. This proposal is discussed using two examples. Firstly, how the public reacted to the virus Sasser in 2004; and secondly, how the University of Bergen neglected accessibility requirements for its web site deployed in 2008. The authors argue for the need for education in computers and laws regulating software development, and propose free software communities as a successful model for joint production of knowledge.

Keywords: Computer Science, Software Engineering, Society, Social Contract, Computer Usage, Quality Software Development, Free Software.

1 INTRODUCTION

In [1] Michael Gibbons writes that "the prevailing contract between science and society was set up to sustain the production of 'reliable knowledge'; a new one must ensure the production of 'socially robust knowledge'. [...] A new contract will be based upon the joint production of knowledge by society and science.". It is easy to say that science should enter the Agora, but Michael Gibbons does not give concrete suggestions as to how to establish a productive and beneficial communication between science and society. The authors believe that, in computer science at least, "socially robust science" is not enough; a "scientifically robust society" is also necessary; i.e. a society which is sufficiently educated to provide feedback to science.

Although software systems have made in-roads into all walks of society, most mainstream software do not meet minimum quality standards. In [2] Gary McGraw, expert in software security, writes that "much of today's software is so fragile that it barely functions properly when its environment is pristine and predictable. If the environment in which our fragile software runs turns out to be pugnacious and pernicious, software fails spectacularly.". Although software engineering is a relatively young discipline, as researchers in computer science we do not believe that this excuses low quality software. The reasons should be sought elsewhere. Indeed, research in computer science has advanced considerably, but its outcome is not fully adopted in software engineering.

In this paper, we argue that a minimum level of computer literacy is necessary to avoid irresponsible computer usage and promote high quality software development. Education in computers, and laws regulating software development, are proposed as a solution to these problems. This is a prerequisite in the field of computer science for achieving the joint production of knowledge by science and society. This scenario is depicted in Fig. 1.

The remainder of the paper is organised as follows. Section 2 discusses the general lack of awareness of risks in computer usage. Section 3 analyses the reasons behind low quality software development. Section 4 presents free software communities as a model of joint production of knowledge in the field of computer science. Finally, in Section 5 some concluding remarks are offered.

2 IRRESPONSIBLE COMPUTER USAGE

It could be argued that present-day society has a lazy approach to technologies in general, and computers in particular. It is very seldom that people read manuals or take courses before using a computer for the first time; they just use it. However, computer misuse can have a negative impact on both economy and health considering the technologically integrated structures in which most members of society operate. This is something which is unfortunately ignored by both society and its politicians.

By way of analogy, let us discuss driving licenses. The need for a driving license for vehicles is unquestionable, at least in developed countries, because if you do not drive a vehicle properly and responsibly you may end up threatening economy and health. However, the need for a license to operate computers seems pointless to many. Why is there such a difference? Because society perceives car misuse as being far more dangerous than computer misuse. The truth is, however, that with software systems a part of our

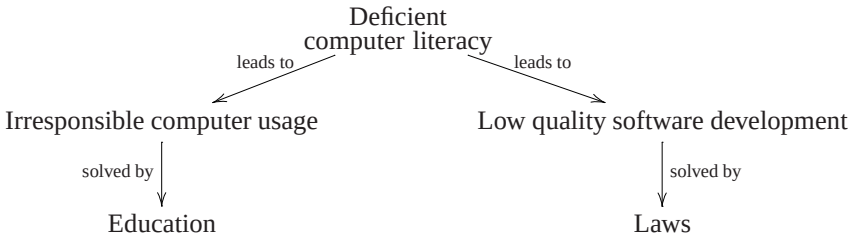

Figure 1: The lack of computer literacy and its consequences

every-day life, computer misuse can also have adverse consequences.

As an example, let us consider the computer virus Sasser (`W32.Sasser.Worm` [3]). This virus spread in April 2004 and, within a couple of days, tens of millions of computers running Microsoft Windows were infected worldwide. The virus was particularly virulent, in that it could spread without user intervention. The major effect of the virus was the shut-down of the infected system. Security solutions supplier mi2g claimed that all the Sasser variants put together caused between 10.0 and 12.3 billions EUR of estimated damages worldwide [4]. The virus could in fact have simply been stopped with a properly configured firewall.

Several months passed before the situation was normalised. Unfortunately, we would propose that the public did not learn a lesson from this technological disaster. When responsible usage of computers is encouraged, the typical reaction is "who cares?". In general, only virus writers are blamed for such software disasters. However, the responsibility must also lie elsewhere. Firstly, we argue that software vendors should develop operating systems which are not vulnerable to viruses. Secondly, we argue that people should take responsibility for protecting their computers. Even without security protections such as anti-virus or firewall, people store personal information on hard drives and make economical transactions on the web. Eventually, they become aware of risks only after their computers have been compromised. With a proper education in computers, similar problems may be avoided in the future (see [5] for a case study on how to make a campus aware of such risks).

3 LOW QUALITY SOFTWARE DEVELOPMENT

Deficient computer literacy does not only lead to irresponsible computer usage; it leads also to more subtle problems. To make a further analogy, let us compare civil engineering with software engineering. In order to ensure the quality and safety of a building, construction companies are obliged to obey all the laws regulating civil engineering and construction. If a building collapses, the construction company is liable. However, laws regulating software engineering are scarce, and a minimum standard of software quality and safety, rare. Moreover, the intangible nature of software means that individuals or organisations require the consultancy of an expert in order to verify whether a software solution is appropriate. This is such a well known-known fact in the software industry, that many providers are delivering low quality solutions, developed with the minimum of resources. This approach does not only minimise the development costs, but will guarantee the software provider additional support and maintenance income.

As an example, let us consider the case of a low quality software solution, which was delivered to the University of Bergen. Despite having a Department of Informatics, the University was not able properly to evaluate the product of the software provider. In January 2008, a new web site for the Department of Informatics was deployed. However, on analysis, it appeared to have major problems. In particular, the "new" web site did not support universal accessibility; i.e. facilitating access to contents according to the varying capabilities and characteristics of users. Unfortunately, it appears that this is not an isolated case. A recent web accessibility report for international universities has shown that accessibility is still a problem for many of the top universities worldwide [6].

There are standard guidelines that explain how to make web contents accessible to people with disabilities. The primary goal of these guidelines is to promote accessibility. "Following them will make web contents more available to all users, whatever user agent they are using (e.g., desktop browser, voice browser, mobile phone, auto mobile-based personal computer, etc.) or constraints they may be operating under (e.g., noisy surroundings, under- or over-illuminated rooms, in a hands-free environment, etc.)." [7].

Web accessibility naturally follows the principle of many developed countries that people with disabilities have the same rights to access (on-line) information. Some governments are slowly introducing laws that demand public institutions to provide accessible web sites. At the time of writing, Norway seems to have only "recommendations" about web accessibility, but not any specific law. However, the University of Bergen has internal rules, among which universal accessibility. Despite these rules, in January 2008, none of the University Departments' web sites was compliant with these rules.

Our first step in investigating the problems we encountered was to send questions and feedback to the development team of the web site, using the "Comments" form provided. Three months passed without an answer being received. The first author informed the head of the Department of Informatics about the problems in a e-mail that stated:

"I think that it is not wishful thinking to expect that a new web site of a Department of Informatics is built using the state-of-the-art of programming practices and web technologies. Unfortunately, this is not the case for us. [...] I am quite confident to say that our 'new' web site seems to be made with web development practices of 10 years ago. This way, the entire Department might loose reputation, since in the computer science/software engineering world many other researchers/technicians might notice the same faults I noticed. You will probably agree with me that this is not a good 'visiting card' for us. [...] The web site is not accessible. Most of the western countries recommend public institutions to have accessible web sites; i.e. web sites that are designed to allow people with disabilities to access its contents. [...] Therefore, it is very probable that the inaccessibility makes this web site irregular".

Fortunately the head of the Department of Informatics appreciated this input, and redirected the feedback to the project leader of the web site development at the Communication and Media Centre of the University. The project leader answered that the web site deployed in January 2008 would be replaced by a brand new one in the beginning of 2009. The reader might be puzzled by the choice of deploying a temporary web site when a new one was already in the pipeline. This administrative choice seemed inexplicable; but it was not the only problem.

The web site was replaced in February 2009 by a new content management system called EksternWeb [8], which was adopted by the entire University. Despite the feedback given one year before, the new web site was not fully compliant to accessibility standards. The development of EksternWeb costed 12.8 million NOK (1.5 million EUR) and relied on a well-known Norwegian software provider for the its technical implementation. Oddly enough, however, the Department of Informatics was not involved. Compared to the solution of 2008, the EksternWeb solution represents a big leap forward, but the authors find the cost unjustified, in light of the system's lack of important features.

This example is illustrative of the problems extant in the interaction between computer scientists, software engineers and users. Science provides elegant solutions to enable web accessibility, but engineers do not always adopt them [9]. Even top universities with departments of computer science and software engineering are not able to guarantee the quality of their software solutions. The introduction of laws regulating software development is necessary to oblige software developers to meet certain quality standards, e.g. web sites which are fully compliant with accessibility requirements.

4 FREE SOFTWARE COMMUNITIES

For many years, communities of computer scientists, software engineers and users have been collaboratively developing free software. "Free software is software that gives you the user the freedom to share, study and modify it. [...] To use free software is to make a political and ethical choice asserting the right to learn, and share what we learn with others. Free software has become the foundation of a learning society where we share our knowledge in a way that others can build upon and enjoy." [10].

Users of free software are typically more curious and keen to read documentation, and several groups provide free education and support to users. This guarantees a "scientifically robust" user base, with the sufficient skills to provide feedback to scientists and engineers developing new technologies and software solutions. Moreover, a free software project allows unrestricted access to its technical documentation, e.g. software models and source code. This guarantees peer review, and hence quality; a process integral to scientific development. In addition, features may be introduced as a result of community feedback. Managers of free software projects may have to accept choices that go against their personal preferences, on the basis of shared community opinion. The joint production of software solutions is, then, in fact a concrete reality, but only where the public involved has the necessary education to provide feedback.

5 CONCLUSION

One of the goals of computer science research is to improve software quality. However, it will not be feasible to prevent software providers from delivering sub-standard solutions, until feedback from the general public can be considered valuable. Once the users are comprehensively educated in computers, and laws regulating software development are enforced, a closer relationship between computer scientists, software engineers and users may guarantee the joint production of knowledge in the field of computer science.

REFERENCES

[1] M. Gibbons, "Science's new social contract with society," *Nature*, vol. 402, no. 2, pp. C81–C84, 1999.

[2] B. Chess and J. West, *Secure Programming with Static Analysis*. Addison-Wesley, 2007.

[3] Microsoft, *Security Bulletin MS04-011*, August 2004. http://www.microsoft.com/technet/security/bulletin/ms04-011.mspx.

[4] mi2g, *Sasser's colossal damage makes it 5^{th} worst malware of all time*. http://www.mi2g.com/cgi/mi2g/press/100504.php.

[5] N. P. Kutner, "Viruses, updates, and security: making the campus aware," in *SIGUCCS 2004: 32^{nd} Annual ACM SIGUCCS Conference on User Services* (J. S. Whiting, J. Ashworth, and D. Mateik, eds.), pp. 322–326, ACM, 2004.

[6] S. K. Kane, J. A. Shulman, T. J. Shockley, and R. E. Ladner, "A web accessibility report card for top international university web sites," in *W4A 2007: 2007 International Cross-Disciplinary Conference on Web Accessibility* (S. Harper and Y. Yesilada, eds.), vol. 225 of *ACM International Conference Proceeding Series*, pp. 148–156, ACM, 2007.

[7] W3C, *Web Content Accessibility Guidelines*, May 1999. http://www.w3.org/TR/WCAG10/.

[8] University of Bergen, *EksternWeb Wiki*. https://wikihost.uib.no/ewwiki/index.php/Main_Page.

[9] J. Lazar, A. Dudley-Sponaugle, and K.-D. Greenidge, "Improving web accessibility: a study of webmaster perceptions," *Computers in Human Behavior*, vol. 20, no. 2, pp. 269–288, 2004.

[10] Free Software Foundation, *Web Site*. http://www.fsf.org.