Impediments to Ubiquitous, Free Computing[*]

Collin Anderson
Department of Computer Science
University of North Dakota
Grand Forks, ND 58203
collina@gmail.com

Tom Stokke
Department of Computer Science
University of North Dakota
Grand Forks, ND 58203
tstokke@cs.und.edu

Abstract

While the Free/Open Source Software movement has been significantly concerned with user-accessibility and compatibility in order to compete with commercial applications, little progress has been made in the widespread adoption of the products of this philosophy by the average user. The Kiwi Project, a student organization located at the University of North Dakota, has sought to improve access to technology by distributing modern computers to socially-disadvantaged communities. This paper will argue, within the context of the project's experiences, that the largest impediments to widespread adoption of these projects may be nontechnical problems of an organizational and sociological origin. Additionally, we will examine areas in which alternative software fails to meet the expectations of our target demographic. Finally, we will encompass this discussion around the use of such efforts as an extension of the computer science curriculum to create awareness within the student body of the social implications of technological development.

Divisions in Access

Often the societal implications of technological progress are lost in computer science and information systems educations in favor of theoretical concepts of the profession and its material application. In part, this is likely because the impact of computing and information access on areas such as the political process and economic welfare seem to be abstractions compared to the tangible principles of programming or the rigor of mathematics. On the part of students, it is easy to fail to appreciate these implications as they appear only tenuously linked to professional goals or affected populations. In doing so, they have taken a passive role in the most important issues of their profession, the socioeconomic and political definition of proliferation of the Internet and technology. In spite of this lack of concern, or perhaps to counteract it, there appears to be a common understanding on the part of computer science departments that there exists a meaningful sociological component to the field that warrants exploration. What often remains is a confusion on the part of faculty on how to bridge these subjects to motivate class interests.

Technical education and creating a social awareness amongst students need not be mutually exclusive; nor must we assume that shortcomings in programs are the result of an intrinsic dispassion afflicting the major. Despite the current cognitive disconnect between computer science students and the computing ecology, university departments are privileged in a manner that upstart non-profit groups and student organizations are not. Concerned departments are capable of providing the nominal services of office space, name credibility, 501(c)(3) status and academic incentives in exchange for the extraction of exponentially greater utility. In its role as an incubator of ideas, a university stands to gain from experimentation in such matters. By creating outlets in which students can apply their diverse array of technical knowledge and interests, programs can introduce these same sociological concepts in a form that is pertinent and creates a lasting interest in participants.

The concept of the Kiwi Project at the University of North Dakota is largely a result of two interested individuals, but it was the embrace of the forward-thinking Computer Science department which enabled it to succeed. One facet of the group's mission is to promote the adoption of open-source software and Creative Commons material on campus by calling attention to its existence, making it available and supporting its use. The more visible role, recycling old computers with Linux into low-income, and otherwise disadvantaged, households, is not new. In Portland, Oregon an organization called "Free Geek," which the project is based on, has had measurable and sustainable success in this effort. However, the Kiwi Project has developed not only as a resource mobilization project, but an effort containing non-partisan, political ramifications with the hope of bridging gaps in socioeconomic development and legislative inclusion by increasing access to communications technology and freely-licensed media.

We know extensively that, contrary to some lofty predictions[2], the domestic use of technology has not proliferated flatly across all social and racial strata. Much has been said about the digital divide in terms of age, but not enough has been done within the United States about socioeconomic divisions in technological access. While the allegory of the rising tide raising all ships alludes to some enablement of lower classes, the bulk of any political or economic mobilization that has come about as a result of technology has been and will likely continue to be focused on the educated, middle class and youth.[1] Since, historically, disenfranchisement and a state of arrested development disproportionately perpetuates in those alienated from the political system, without a concerted effort to incorporate those classes into the benefits of modernity, concepts such as enablement, otherwise known as 'emergent democracy,'[3] are meaningless and only perpetuate fragments of the antiquated social structure.

Certainly, that the elderly report to have never used a computer 45% more often than the aggregate[1] should be of concern to the goal of creating an inclusive Internet, but its lifestyle implications are not extremely severe to the group or the encompassing society. The vast majority of the American population has access to a computer and access to the Internet (75% & 74%, respectively). From there, they routinely take advantage of services such as email and the well of information on topics ranging from entertainment and sports to news and academics. They also begin to acquire the job skills that are necessary for a modern business environment, specifically the use of office applications. Contrapositively, those segments of society left out of technological education are likely, absent these modern prerequisites, to have a hard time finding employment that can significantly improve their socioeconomic position. From this scenario, it is clear that we should gravely fear a paradigm where technological integration into society perpetuates socioeconomic division instead of fulfilling its promise of ameliorating it.

Statistics that illustrate that non-Asian minorities are 15% less likely to have used a computer than their peers foreshadow potential complications. Racial minorities are not the only social groups at risk. Our particular concern is the extent to which access and marketable technical skills drop as income and education levels decrease. While there is a deficiency of data at the income levels associated with Kiwi clients, it is clear from existing work that when moving towards the poverty threshold, use of the Internet drops much more substantially than delineations based on ethnicity.[1] Examining our core audience, we can delineate two demographics, low-income adults who need to acquire marketable job skills and children that need to acquire the ability to learn with computers and fulfill the expectations of school. Those that cannot immediately afford a computer are stuck with poor options, the most frequent in our user group being rental companies that predatorily charge multitudes more than the computer's real value. Alternatively, public access computers, while generally free, create obvious issues of privacy, rationed access, limited availability and an unfriendly environment for training. Neither of these common solutions address the underlying issue in a manner that is both financially and operationally sustainable.

In this demographic sense, the problems concerning the Kiwi Project are classic social issues of income equality and meritocracy, for which mature governmental and non-governmental organizations already exist. It should not be incumbent on concerned efforts to reinvent a distribution network. As computer scientists, we are generally not well-equipped to tailor programs to the greatest social benefit, as opposed to relying on the expertise of partnerships with groups like low-income housing associations. One such example of leveraging institutions is our continued relationship with the LaGrave Learning Center, a public job

[1] This is derived from Pew behavioral studies on specific use of the Internet by various demographics.

training and housing group, to distribute computers based on a reward system that encouraged clients to participate in job training and community events.

Shifting Expectations

The evolution of the Kiwi Project has wholly been a product of its successes and its shortcomings, while a bulk of the effort has been merely matters of delegation and managing expectations. As a result, the technical aspect that exists is often relatively minor and straightforward. This is beneficial because the project is less likely to be held hostage by deficiencies in knowledge on particular topics. Instead of teaching procedural, or specific job skills, such efforts teach the abilities of critical thinking and being responsive to one's environment that are the traits of a successful computer scientist.

The difference between causes of adaptation, due to technical versus operational reasons, has been evident in the fluctuation of the output of the organization. After a time, it became necessary to establish a baseline of minimum requirement for accepting donated hardware. While the public is very willing to give, it often does so when their computers have lost value according to their own perceptions, which is generally far beyond reality. This can lead to a surplus of unusable, old computers and create a glut on daily operations. Any nascent program will understandably have low expectations, however business and educational relationships, which have come with even minor success, progressively moved our standards incrementally higher. Moreover, when managing expectations it has been necessary to focus attention to overarching issues raised by actual users. As computer enthusiasts, overly-involved solutions to minor problems, such as maintaining alternative repositories to install additional software like a customized dock and webmail applications, seem attractive areas of experimentation and curiosity. As this came to be understood, the software involvement of the Kiwi Project devolved immensely from a heavily customized suite of tailored applications to a fairly vanilla operating system installation. For the most part, we found existing solutions to be adequate enough for end-users considering the potential for over-complication. These issues were heavily rooted in concepts of management, not hardware.

On the computer science side of operations, there is no other option feasible for projects such as Kiwi than the Linux operating system. Its liberal licensing and freedom of distribution are the philosophical root of the effort. While commercial competitors often offer their software for free to our client organizations, user-agreements and restrictions are burdensome to the ability of similar groups to operate. Furthermore, this offer is artificially based on business interest and not reflective of any market advantage for the end-user. Although we have consistently relied on the various flavors of Ubuntu for its focus on creating a distribution that would appeal to a wide audience and its high hardware compatibility, our standard set of applications have significantly changed along the shifts in our baseline and increase in user input. It seems evident that the popularity of Ubuntu is not going to ebb like past Linux fads, as it has found a substantial place in its environment. Initially, accommodating slower memory and lower processing capacity meant using the minimalist

XFCE interface of Xubuntu with all effects and unnecessary services turned off. With more modern computers we could spend less of an effort on making sure that a select group of applications ran in favor of wide compatibility. We then moved to the suite of Gnome applications that comes with the mainline Ubuntu and then to the Kubuntu distribution based on KDE for accessibility reasons.

Historically speaking, the growth of Kiwi has been simple and linear, in line with the progression of increasing power available. The project has grown in a short time frame from its initial batch of ten year-old, Celeron desktops running XFCE desktop environment with 192 MB of memory and eight gigabyte hard drives to a minimum baseline of at least a late model Pentium 4 or PowerPC G4 (or comparable). The growth was quick, by being able to present concrete results the project proved itself to administrators and organizations and a new level of resources was opened. This initial period is not without its merits and lessons. One lesson learned from infancy was chiefly that memory is the point of highest return in terms of investing in speed and usability. By increasing memory and cutting less-useful services, we were able to extend these first computers' usefulness. Remarkably, such a lesson is still valuable due to constraints in our supply that have pushed down our current baseline to 256 MB of memory (which will run a computer sustainably if somewhat less smoothly, although 384 MB or more is desirable).

Limitations on accessible hardware point to the substantial benefits of our software choices. As a result of commercial software's development mentality in regard to resource use, our equipment may be limited to comfortably run up to only Windows XP while the latest version of Ubuntu 8.10 with KDE 4.1 runs well. Linux's modular, open nature allows us to comfortably do what is often technologically inadvisable elsewhere, such as using an image of one particular machine to run a heterogenous environment, or situationally necessary, namely rationing memory and storage. In the former case, despite a lack of common hardware platform, we are able to generate predictable, stable installations, with exception to infrequent tweaks to the graphics configuration or OS-independent, low-level hardware failures. Under Microsoft Windows, obscure drivers and digital rights management would make such a task much more difficult, if not impossible. As a result, we are able to use virtualization applications such as *VirtualBox* to create environments for users which are then installed via PXE network booting using the free software *Clonezilla*. Clonezilla's installation method also means that we are able to install to an almost arbitrarily large set of computers at one time over multicast using an ethernet switch. We are also able to easily create and maintain branches of our default installations for situations that arise, such as PowerPC computers, smaller hard drives or locked-down environments for children.

Sources of Impediments

In the end, those skills that are crucial to the actual success of the organization are not taught in the standard set of computer science classes. From the experiences of the past year, there are three irreducible, but often overlapping, categories of problems that have affected the Kiwi Project: bureaucratic, sociological and technological, with the latter being more predictable and solvable than the first two. Much of the success of the group has been the result of not the availability of resources, but the ability of members to address these problems efficiently. Importantly, for members and sponsoring departments, these principles are universal to any field, and students who acquire them are uniquely privileged above their peers in the job market.

For the organization to grow beyond the self-directed collaboration of a few people, the effort requires managerial skills and organization. This hierarchy is antithetical to the normally flat division of labor and authority that occurs during group projects in academic courses. However, with such a narrow focus, bureaucratic issues are generally rooted in trivially differing visions of the delegation of labor and the application of effort. In most cases, solutions to problems that come up are rarely confined to one of the three categories. More importantly, the issues that arise cross a multitude of subjects, technical or social. An example of this being the artificial scarcity of hard drives created by concerns over information security. This problem has forced us to operationally rely on any drive donated (technical), but moreso to critically assess the use of storage by the end-user to understand operational baselines (sociological). We are forced to ask whether our users will indulge in the culture of piracy prevalent on the Internet (especially considering Ubuntu's default inclusion of a BitTorrent client) and whether that is much of the basis for modern storage demands. By identifying the diverse roots of the problem, we can promote broadly applicable solutions; in this specific case, cloud-based services, such as Pandora and Hulu, which cover both technical constraints and sociological vulnerabilities.

Moreover, despite an unequivocal adoption of open source by the department in its classes and in its lab, Linux skills are not in abundance in the student body. Without these talents, the effort's support network risks disintegration and the product becomes stale or unstable. More deeply, this should be a concern for departments; in some respects, a diversity of platforms is tantamount to the diversity of ideas that is fundamental to any legitimate institution of higher education. Issues with particular companies' business strategies aside, diversity brings new opportunities and creates value. By not being open to other platforms, our educational programs teaches specific skills and not self-sufficiency or intuition. Certainly, giving students the opportunity to dig deeply and explore in a safe environment is a good first step to overcoming such deficits.

On the other hand, we have been forced on occasion to revisit the basic foundations of the Kiwi Project over basic concerns about the applicability of open-source software to those trying to learn skills that apply to the job market. We were surprised at the reaction of recipients to a program for earning donated computers that was run by a partner. By doing tasks such as attending educational workshops and community meetings, families could earn points towards, after a certain threshold, one of the first Kiwi computers. In the end, the contest brought a palpable sense of entitlement. Despite having a demo computer available and despite it being fully adequate for basic computer use, the feedback we received included significant disappointment. People expected the height of modernity: movie streaming, video games and speed; but they also expected the same interface and software from their learning labs. Any departures detracted from the product and at least a few machines have gone unused.

Some of these disappointments are understandable. The computers delivered were slow compared to their modern lab machines and the XFCE interface could have been better arranged. At the root of the Kiwi Project there is one basic question: can we deliver not-Windows to a Windows world; can Linux survive in this environment? This opens up many derivative questions. We have been pushed to ask, did the introduction of OpenOffice, after residents had received training in Microsoft Office, adversely affect their computer education by creating confusion? Thus far, we have settled on one principle: being an open-source advocate should be synonymous with fundamentalism. Certainly, graphic design students need to be professionally comfortable with software such as Photoshop and, due to a market-monopoly, all propective employees need to be familiar with Microsofts Office applications.

Essentially, Linux needs to have its own appeal and cannot rest on its laurels in regard to open-source principles. The product which we deliver has clear advantages that should be conveyed. While users cannot buy software from retail stores, the demographic which we are connected with should be the last to do so. Nor do they need to. Ubuntu's roots in Debian Linux bring the mature *apt* software management back-end. Apt allows users to easily install applications and their dependencies, as well as keep them up-to-date, with minimal intervention and little prerequisite knowledge. As a result, users of the Kiwi computers have a panoply of packages ranging from drivers to music players to enterprise-level server software at their disposal for free. In this sense, if the user is willing to take advantage of new opportunities, our system is much more accessible and inexpensive than commercial competitors. Again, however, we sense that this difference seems to haunt general impressions as a negative, despite the obvious advantages.

Lessons Learned

The overarching theme of the Kiwi Project is that the organization generally cannot fully predict the needs of the users beyond the basic usage scenarios. Unlike normal classes, this creates a scenario where the program's output slowly evolves according to user input and constraints. In this respect, the project follows the normal process of development more than the generalized software development courses. The evolution of Kiwi's operating system is a primary example of this, our end product shifted and grew according to the axioms of usability and compatibility, working towards minimizing such shortcomings. While we had initially settled on Gnome as our window manager of choice after XFCE, it is not currently what is installed on fresh Kiwi desktops. Just as we immediately neglected that our customers needed modems for dial-up internet access, we came to understand from user input that the visible layout differences between Gnome and the standard Windows interface was an impediment to adoption. Directly as a result, we moved to KDE 4.1, which we felt is more intuitive and accessible for new Linux users. This underscores two of many lessons: what is straight-forward and obvious is relative to personalities and experience, and that the intuition, open-mindedness and curiosity that is the hallmark of an ideal computer scientist, is a privilege to those who really need help.

In concert with this continuous learning process was the realization that hardware maintenance and inventory matters took up the vast majority of our time, as opposed to client relations or software issues. Without that initial baseline we acquired a large amount of unusably old hardware for which disposal is difficult and scrapping services cost significant money. In the interim, we have found minor respite by opening select hardware for the public to take for noncommercial use. We continue to question our support of PowerPC models as its architecture differences require the maintenance of additional system images and is not capable of PXE booting for the cloning service. Additionally, many of the applications useful for Linux and packaged for Kiwi users, like Skype, Picasa and Flash, are unlikely to be ported to the system.

We remain concerned with the influence of Linux on the overall use of the hardware we donate. The initial reaction we receive from group administrators and recipients to the operating system difference is skeptical but open-minded, concerned mostly with delivering basic connectivity. The best available evidence asserts that this devaluation is psychological and not grounded in any faults of Linux. In one study on KDE usability,[5] the average time to carry out routine tasks was found to not be significantly higher than that of Windows XP. The same group responded equally as comfortable carrying out those operations between the two platforms and equally open to continued use. On top of this, our interest in promoting Linux benefits from the difference in progression of the two systems. While we would likely be confined to the same XP if we did use commercial applications due to resource constraints, KDE has improved signifcantly in the five years since the survey.

Moreover, for the first time in a decade, Microsoft's platform position has decreased due to the success of Apple's Mac OS X platform and Linux.[4] While there is no reasonably imaginable future without Microsoft having a majority of installations, it is clear that the progression of these competitors, in addition to increasingly powerful mobile computing, is creating a pluralistic ecology built on open standards. This should create a market for those with, not just particular skills, but an intuition and comfortability that allows an individual to learn and adapt to different environments. In terms of institutional pedagogy, we feel, as a result, that the congruencies in the fundamentals between operating systems are what need to be taught. In reality, the operating system is evolving from its basic formula as increased memory and processing has opened up new opportunities in interfacing with the user. Computer users will inevitably have to accommodate changes as software technology progresses regardless of whether they use open-source software or commercial platforms, as is the case with Microsoft's introduction of Office's ribbon and interface changes in Windows 7.

For those students who are not immediately inclined to spend time working towards the goals of the organization, numerous academic and informal incentives have been used to entice wide participation. With the prior establishment of an effort called the "Open Systems Support Center," which was established to provide support for free software like Linux and OpenOffice, we inherited experiential learning courses which could then be used to offer credit for application of labor and technical skills to the project. Informally, the ability to immediately distinguish oneself and advertise technical skills is important for the employment process, especially for computer professionals. As a resumé line, students can project themselves as not only technically capable but also engaged with broader, complex social issues. For the intellectually curious, we have created a playground for experimentation with software and hardware without having to worry about mistakes that lead to data loss or hardware failure. The response from undergraduates, graduates, faculty and the public has been grounds for optimism amongst those concerned about these issues. There does appear to be an underlying desire to participate and, given the right conditions and tools of motivation, there is latent base for developing such efforts.

In the larger scheme, similar projects represent new opportunities for computer science departments and their students to interact internally and with new segments of their communities. By participating in the broader public, outreach can represent the stereotypically quiet computer scientists in a more favorable light. For the department, such projects can serve as a recruitment tool and affect the socioeconomic standing of minorities and other populations that are traditionally underrepresented in mathematics and the sciences. Within the array of added value that such projects bring to departments, there has been a clear reciprocation and constant communication between the project and pre-existing social implications courses to create a tangible image of the issues pertinent to modernity. More broadly, student and faculty project members have found numerous occassions to address classes on how particular principles relate to real conditions.

Fundamentally, the best thing that the Kiwi Project, and its related efforts, can do is facilitate an environment of experimentation in which its volunteers feel free to experiment with technological and sociological concepts without consequences, while making connections between theory and practice that their non-involved peers could not, in the end fulfilling institutional and societal goals of creating computer scientists better equipped to participate in their profession and their communities.

References

[1] Usage over time survey collections. Pew Internet & American Life Project, March 2008.

[2] Friedman, T. L. *The World Is Flat.* Farrar, Straus and Giroux, 2006.

[3] Ito, J. *Extreme Democracy.* Lulu Publishing, 2005, ch. Emergent Democracy.

[4] Peers, M. Window into microsoft's monopoly money. *Wall Street Journal* (March 14th 2009).

[5] Relevantive AG. *Linux Usability Study Report*, August 2003.