



Free & Open Source Software Off The Grid

---

A Master's Thesis Project

Presented to

School of Arts and Science

State University of New York

Institute of Technology

Utica, New York

---

In Partial Fulfillment

Of the Requirements for the

Master of Science Degree

---

By Benjamin Besemer

July 2011

**SUNYIT**

**DEPARTMENT OF INFORMATION DESIGN AND TECHNOLOGY**

**CERTIFICATE OF APPROVAL**

Approved and recommended for acceptance as a project in partial fulfillment of the requirements for the degree of Master of Science in Information Design and Technology.

---

Date

---

Dr. Steven Schneider  
Thesis Advisor

---

Dr. Russell Kahn  
Thesis Reader

---

Dr. Kathryn Stam  
Thesis Reader

## **ABSTRACT**

This thesis project is an autoethnographic look at personally adopting Free & Open Source Software (FOSS) for a period of 8 weeks. During this period I made a switch to entirely FOSS both professionally and personally in hopes of exploring whether FOSS can be effectively used in everyday life. Furthermore, the project looks at what is involved in adopting FOSS and what benefits and/or drawbacks can be expected for individuals looking for alternatives to proprietary software. The daily experiences written about include those from my personal life and that of being a high school teacher performing daily tasks and instruction using only FOSS. A comparative analysis of the data collected is made between my prior experiences with proprietary software and that of FOSS used during this period.

## **ACKNOWLEDGEMENTS**

I would like to thank professors Dr. Russell Kahn, Dr. Kathryn Stam, and Dr. Steven Schneider for their support and continued guidance throughout this project and the Information Design & Technology graduate program.

## TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
TABLE OF CONTENTS.....	v
INTRODUCTION.....	7
Vignette.....	7
Research Questions.....	9
The Free & Open Source Software Movement in K-12 Environment.....	10
LITERATURE REVIEW.....	13
Software in Education.....	13
Free/Open Source Software as an Alternative.....	15
Free/Open Source Software in Education.....	19
METHODOLOGY.....	25
Autoethnography.....	25
Subject.....	27
Data.....	29
Setting.....	30
Expectations.....	30
Analysis Techniques.....	31
DATA ANALYSIS.....	32
Elevation.....	32
Piracy.....	32
Locating.....	38
Modifying.....	40
Ease of Use.....	42
Portability.....	48
Retooling.....	51
Compatibility.....	52
Lowering.....	53
Learning Curve.....	53
Barriers to Entry.....	56
Ease of Use.....	58

Glitches .....	59
Prepping for FOSS .....	63
CONCLUSIONS.....	65
REFERENCES .....	67
VITA .....	70

## INTRODUCTION

### Vignette

Eric is a teacher at the local high school and prides himself on integrating technology both in the classroom and his personal life. Any new software being advertised, Eric must have and places the appropriate calls within his district to make the requests. His most recent request for software included Microsoft Office Professional 2010 and Adobe Photoshop CS5. He isn't concerned about the costs and knows that his district receives educational discounts for the software they purchase, and his requests have yet to be denied.

Just prior to Eric's next classroom project called Snapshot, the software is approved and installed in his computer lab to be piloted. For this project, Eric plans on having students create a presentation using PowerPoint, Sound Recorder, and Photoshop showing a snapshot of their individual lives.

Several days into the project, Eric begins to hear students complaining that they were not able to work on the projects at home because of compatibility issues and lack of appropriate software. It turned out that many students had earlier versions of Office that were not compatible without an installed converter plug-in or by saving their in class work to an earlier version. Eric was going to recommend that students use the library computers or another open lab during their free periods, but soon realized that his lab was the only one licensed to run the new software. Students also had some difficulty within



PowerPoint and had to be coached in using the new ribbon features found with some newer software.

While the projects were completed and done well, Eric felt rushed in class and wondered if a different approach would have worked better and allowed students to spend more time working outside of class. His chief concerns with the software purchased had to do with the lack of availability outside of class, difficulty adjusting to new layouts within the programs, and the costs associated with the purchase of the software.

Paul is also a teacher for a neighboring district and prides himself on integrating technology into the classroom and his personal life, but thoroughly explores all of his options before committing to anything. When he sees advertisements for new software, Paul compares them with preexisting software and searches for any comparable free and open source software that is currently available. Only when all of the software is thoroughly vetted does Paul use it in the classroom and recommends it to his district technology committee. Even though his district has been good about providing funding for new software, he realizes that budgets are becoming even more strapped for cash and any small part he can do to help could go a long ways.

For an upcoming project, Paul would like students to also create a presentation on a snapshot of their lives and incorporate pictures, audio, and other media where appropriate. He realizes that this project will involve more time than is available during the class period and would like to encourage students to work on the project at home, but realizes not every student has the appropriate software outside of class. A solution he

thought of would be to use freely available and open source software that students could use both in the classroom and at home. The three primary software titles Paul recommended were Open Office Impress for the presentation of the project, Audacity to incorporate audio, and GIMP to acquire and edit pictures.

The projects were a huge success and exhibited a high degree of craftsmanship; Paul was also very excited to be able to allow parents the opportunity to work with their child on the projects at home. The general consensus from the students was that the software used was familiar to them, easy to learn, easy to use, and allowed for greater portability than they are accustomed. As a result of the success of these projects, Paul has decided to incorporate additional free and open source software both in his development of curriculum and how that curriculum is used within the classroom. Students have been reporting back to Paul on the many free and open source software titles that they have found and used at home; all of which was a result of his initial classroom project introducing the students to free and open source software.

### Research Questions

The primary research question for this thesis project explores whether free and open source software (FOSS) can be effectively used in everyday life? Furthermore, the project will provide an autoethnographic account of what could be expected for individuals adopting FOSS.

## The Free & Open Source Software Movement in K-12 Environment

While the public may be more willing to download and use free and open source software (FOSS) on their personal computers, there still exists some hesitation to adopt it in educational institutions. Hepburn and Buley (2006) saw the lack of awareness on FOSS as being one of the larger barriers to overcome before greater acceptance is seen in education (Getting Open Source Software into Schools: Strategies and Challenges). They felt that there needs to be more research done, so those who lack knowledge and understanding about FOSS could become more aware of its possible role in education. This continued lack of awareness fuels the apprehension to adopt and incorporate greater usage of FOSS in education.

Today's students have an arsenal of technological gadgets such as Smartphone's, iPods, laptops, and tablets; as well as membership to various social networking sites such as MySpace and Facebook. In many of today's schools there are policies in place to stop these gadgets from seeing the light of day and according to Tom Blacksell (2011) of *FT.com* this can put not only schools at a disadvantage, but also the students they serve. A study published in *Professional Educator* found that students use technology they value, are comfortable with, and when given a purpose these devices should be used in schools (Anderson J. , 2005). This idea of having greater portability between home and school with regards to electronic gadgets can also be translated into the software they utilize. If given the opportunity FOSS has the opportunity to create a greater bond between technology available at school and that at home. For example, students could use their cell phones to check course grades using the Engrade App available on iPhone and

Android OS's, use Ubuntu One to sync files between school computers and their cell phones, or use their cell phones as storage devices to run portable apps from these devices.

These are just a few possibilities available to schools who embrace technology, including FOSS, and why they should not shy away from unfamiliar territory. They must rather embrace it as a means of creating, sharing, and receiving the knowledge and skills that are vital for being successful in tomorrow's world. For schools to embrace a diverse and well structured technology plan there must be a strong desire to do more than purchase the latest and greatest hardware; whether that is interactive whiteboards, computers, remote polling devices, etc. These hardware devices are truly amazing when implemented correctly and utilize appropriate software for curriculum development and delivery. But in some instances these high priced devices go unused or not used to their full potential. Alison Leithner (2010) wrote an article for *Change.org* that raises the point that many individuals and companies pushing these devices "assume that classroom technology should engage students and improve their performance just by being in the classroom.". Anyone who has taught in the classroom knows that while these tools can help, the real engagement and increases in performance are dictated by the actions of the teacher. This is where FOSS can be combined with existing classroom technology to assist both teachers and students in getting the most out of their instruction with minimal to no monetary costs.

Hardware and software are the yin and yang of today's technology, without one the other is rendered useless. It has been my personal experience that schools provide

teachers with laptops and students with computers that lack the tools, or more specifically the software to promote curriculum development and usage. Upon inspecting a teacher's laptop you may find a fairly modern laptop without much, if any, software other than Microsoft Office. This could be looked at two different ways, either that teachers only need Microsoft Office or that they could install additional software if they chose to do so. I'll admit that Microsoft Office was and still is on some fronts the go-to program for me when I want to create a quick document, but I also know that there are countless free software titles available that can provide similar functionality. I also know that depending on the school district and their technology policy, the opportunity to install outside programs on school provided laptops may not be an option. They may also be reluctant or dismissive to outside suggestions for alternatives to currently used proprietary software.

I have found that either through a lack of Information Technology (IT) support or not knowing any better, that teacher laptops are locked down to prevent changes from being made. Software titles like Faronics DeepFreeze which is primarily intended for student computers are finding their ways onto teacher computers. Instead of IT departments seeking to support and promote alternatives, they are creating barriers for teachers in the name of security and consistency.

As a result of my experiences and perceived view that FOSS is not taken seriously as an alternative to proprietary software, and that school IT departments are not aware of its capabilities, I have elected to conduct my thesis project on this subject. For a period of 8 weeks I will document my sole use of FOSS both in teaching and personal usage. It is my hope that through this process I can better understand the role the FOSS can play in

education, personal usage, and how it can become a go-to option for school districts. Stakeholders that could benefit from this project are administrators, teachers, students, and parents; all of whom could benefit just through knowing that FOSS alternatives exist.

## **LITERATURE REVIEW**

### Software in Education

Budgeting for school expenses each year is undoubtedly a difficult and tricky task to undertake for all those involved. Part of that budgeting process involves technology expenditures that are typically outlined in a districts technology plan. The Massachusetts Software Counsel (2004) (as cited in the North Central Regional Educational Laboratory website) recommends that “the technology budget include equipment and software costs, charges for setup and upgrades, network access fees, insurance, operating costs such as phone lines and utilities, security, professional development for staff, and consultant fees” (NCREL, para. 1) (NCREL, p. para. 1). To provide some context of costs associated with technology in education we will look at three different sized school districts within the same geographical area for the 2009-2010 year. Fabius-Pompey Central School District had an enrollment of 838 students (The New York State District Report Card - Fabius-Pompey Central School District, 2011); Jamesville-DeWitt Central School District had an enrollment of 2868 students (The New York State District Report Card - Jamesville-DeWitt Central School Disitrcet, 2011); and West Genesee Central School district had an enrollment of 5017 students (The New York State District Report Card - West Genesee Central School District, 2011).

Fabius-Pomey Central School District technology plan budgeted for the 2010-2011 school year \$16,995 for NYS Aid Computer Software and \$2,894 for additional software with a \$253,298 total technology budget (p. 6). Jamesville-DeWitt Central School District technology plan budgeted for the 2010-2011 school year \$12,550 for software purchases with a total of \$1,018,515 going to their acquisition plan (2009, p. 20). West Genesee Central School District has laid out in their 2010-2013 technology plan of \$98,000 yearly for general software purchases with a \$1,508,561 total proposed projects and general purchases for technology (2011, p. 11).

Every year school districts perform a lengthy process for determining what their current budget will allow and what they may need to forgo because of budget constraints. Large proprietary software companies realize these constraints and attempt to offer schools substantial discounts on their products in hopes of hooking students at a young age into using their products (Pfaffman, 2008, p. 27). While many school districts abide by licensing requirements for their software, that software can only be used on school computers and districts can face hefty fines for improperly licensed software (Hepburn, *Open Source Software and Schools: New Opportunities and Directions*, 2005). This restrictive use can create problems for teachers, students, and parents who wish to use the software on home computers. Resulting is several options for them include pirating the software, purchasing the software, or forgoing the software all together. According to Hopper (2000) (as cited in Teston, 2008, p. 67) found “almost half (48%) of elementary and middle school students believe software piracy is legal”.

### Free/Open Source Software as an Alternative

Free/Open Source Software (FOSS) is a title for software that has been liberated from many of the common restrictions given to proprietary software. FOSS can be used, modified, and redistributed freely for any purpose, by any individual (Wheeler D. A., 2007). FOSS is a term that is synonymous with both free software and open source software with minor differences in how people perceive their usage and development. Free software is seen by many in terms of the freedom given to users and open source is seen more in terms of the strengths of the software (Free and open source software, 2008). FOSS also gives anyone the right to distribute the software in any way they want without restriction from software licenses (Pfaffman, 2008, p. 26).

Closed source software is any software that the source code is hidden from the public and cannot be duplicated or modified without consent from the creators (Review, 2008). While some free software is closed source, it has few if any restrictions on use and distribution. Closed source software has allowed some large software companies, such as Microsoft to develop what is perceived as monopolies in the software market and the use of anticompetitive practices (NetAction, para. 8). As a result of these companies perceived monopoly status, freelance programmers have joined together to create software that is free and open to the public. While the name recognition associated with Microsoft, Adobe, and others are absent from these free and open programs, they can provide users with a free alternative to expensive name brand software.



According to Ken Coar (2006) of the Open Source Initiative, software deemed to be open source must meet the following 10 criteria and go beyond just the access to the source code.

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Fields of Endeavor
6. Distribution of License
7. License Must Not Be Specific to a Product
8. License Must Not Restrict Other Software
9. License Must Be Technology-Neutral

FOSS are programs and operating systems that are continually improved upon through a joint effort by programmers. These programmers use their time and skills to collaborate and improve software which is in turn given to the public free of charge. In May, 2003 SourceForge.net, which hosts open source software, had approximately 60,000 open source projects (Lineweaver, 2003). As of April, 2011 those projects have quadrupled to 260,000 open source projects (Sourceforge.net, 2011). This steady increase shows devotion by programmers to create and support FOSS and in turn a larger FOSS user base.

Compatibility is not something mentioned much by proprietary software manufactures, but some have adopted strategies to make their software non-compatible

with FOSS alternatives. This however is not the case when looking at FOSS software, which in many instances has the capabilities to open competitive proprietary documents (Zhaoli Meng, Sang-Yong Tom Lee). Many of the open source software packages available today have the ability to read and manipulate files that were created by a different program (Hill & Gaughan, 2006, p. 58). For example, no longer is there a need to fuss with getting a Word Perfect document to open in Microsoft Word, which can be easily overcome by using a software suite similar to Microsoft Office called OpenOffice.

Brad Wheeler (2007) sees large proprietary software companies as giving too little attention to the needs of institutions and the students they serve. There is a general feeling that many large proprietary software makers give institutions little, if any, flexibility to adapt the software to their needs (p. 1). With open source software any individual or group can add or modify a program to meet their individualized needs, which is in contrast to that of larger software developers who create a program to meet everyone's general needs.

Rob Lineweaver raised the question in his article *Cost Savings of Open Source Software in the Server Room*, "If open source software saves people so much money, why isn't everyone using it?" Lineweaver (2003) puts it simply as being a "learning curve" that many individuals and organizations are not willing to undertake. Certain FOSS lacks the common navigation and expectations that users are accustomed to in proprietary software, which might be one reason why everyone isn't using it. There is also a mentality by those not familiar with FOSS that it is less secure than proprietary software because the source code is freely available. Deb Shinder (2005) wrote an article

explaining which brand of software is more secure and ended up concluding that FOSS has both advantages and disadvantages over proprietary software, but “whether or not the source code is open is probably one of the least important factors”.

Many school districts have invested heavily in brand names like Microsoft and as a result created a barrier for change (Hill & Gaughan, 2006, p. 58). In many ways these companies have succeeded in branding their software as being secure, robust, and universal; which is another hurdle for FOSS to overcome. What may seem surprising for some is the reluctance to adopt free/open source software because of the feeling of guilt for getting something, for nothing. The idea that people create and distribute software freely can make it appear inferior to the costly proprietary software already available (Pfaffman, 2008, p. 26).

Despite the attractiveness of FOSS as an alternative to proprietary software, there are some shortcomings. There are certain applications that currently do not have an open source equivalent. An elementary school program called Accelerated Reader is widely used, but there is yet no free equivalent (Hill & Gaughan, 2006, p. 58). This lack of alternative software will force school districts to continue purchase some content specific software from proprietary vendors. Despite the steady and strong growth of FOSS, there will remain gaps in coverage. FOSS is not in the position of replacing all proprietary software and will likely never reach that point, but is a viable alternative to some proprietary software currently available.

IT staff would not only need to be trained on FOSS, but there would also need to be a major undertaking to train teachers, staff, and students. There are many tutorials and

manuals associated with proprietary software from both the manufactures and third-parties. However, as a result of the infancy and limited use of newer FOSS title these additional resources may not yet be available. FOSS wizards and tutorials will undoubtedly be available in the future, but this lack of additional help deters the adoption of open source in some circumstances (Hill & Gaughan, 2006, p. 58).

### Free/Open Source Software in Education

Free & Open Source Software (FOSS) is considered by some to be an effective and viable substitute for proprietary software in education, and according to Lakhan & Jhunjhunwala's (2008) article *Open Source Software in Education* it has already "gained considerable currency in the realm of higher education". Despite the greater availability of resources describing the use of FOSS in higher education, one could presume that K-12 educational institutions could also see the potential benefits of incorporating FOSS.

The choice to use certain software in educational environments has a trickledown effect; not only do those choices force teachers and students to adopt the usage of the software at school, but it also carry back into the home (Pfaffman, 2008, p. 26). A result of strictly using proprietary software in schools is that teachers, students, and parents can feel pressured into purchasing or attaining the software through other means for use outside of school. This is not the case with FOSS in which teachers, students, and parents can easily create, edit, and move documents between school and home without spending a dime or circumventing piracy laws.

Despite the grade school lessons about sharing and cooperating with your peers, proprietary software companies are far less willing to share for the common good. Large proprietary software companies have been overwhelmingly successful in pushing their products into schools through intense marketing and steep educational discounts (Hepburn, *Open Source Software and Schools: New Opportunities and Directions*, 2005). These companies would gain nothing and lose a great deal from opening up their source code to the public and other software developers, which is why Microsoft has actively perused legal action against FOSS vendors and has stated that many of these software programs are successful because they infringe on over 200 Microsoft patents (Parloff, 2007). Despite warnings from proprietary software giants, FOSS developers have continued to create and improve upon the next versions of FOSS. Which has the potential to erode away some of the proprietary software industries market that was estimated in 2008 to be \$136.6 billion in the US alone (*Software Industry Facts and Figures*, 2009).

According to Pfaffman (2008) the software that is used most frequently both at home and school are software suites like Microsoft Office (p. 29). These suites include individual programs like word processing, spreadsheet, database, and presentation software similar to those found in Microsoft Office. Currently, there are FOSS projects that focus on specific categories of software development. These projects bundle together related FOSS that can be used in a particular setting. The OpenScience Project is a focused initiative by scientists and programmers to create science specific software which currently includes molecule modeling and planetarium software (p. 30). Through the collaboration of programmers and content specialists these projects create a valuable addition or alternative to proprietary educational software.

There is also a sense of freedom and independence that is associated with FOSS, which is not always seen with proprietary software. The concept of FOSS extends beyond its use in the workplace and is slowly being accepted for personal and educational uses. Schools that can easily afford proprietary software may also find comfort and convenience when it comes to free/open source software (Pfaffman, 2008, p. 27). A study commissioned by the British Educational Communications and Technology Agency found free/open source software easier to use than their proprietary equivalent among middle and high school students (p. 30).

School districts and their teachers continually seek innovative and creative ways of improving classroom instruction. Implementing and using technology effectively within the classroom can aid in creating an engaging environment for students. However, there are no illusions to the high cost many school districts face when implementing technology. Looking back at the \$1.5 million dollar technology budget for West Genesee Central School district, this boils down to about \$300 per child, or Fabius-Pomey Central School Districts \$302 per child, or Jamesville-DeWitt Central School Districts \$355 per child. When these numbers are broken down further to examine the cost of software per child the numbers seem less impressive with Fabius-Pomey Central School District spending \$20 per child, Jamesville-DeWitt Central School District spending \$4.38 per child, and West Genesee Central School District spending \$19.53 per child. Despite the small portion that software purchases consume in overall school budgets, there are still opportunities that exist to save money through adopting FOSS.

There are several reasons why the usage of FOSS would benefit K-12 education. The most apparent is the fact that it is free; free to use and free to distribute to anyone. Not only is the software free to use in schools, but it can also be used at home or in the workplace (Hill & Gaughan, 2006, p. 58). No longer will students or teachers be restricted in creating or editing their documents only at school. There can now be a consistent and free flow of documents from school, to home, and back again.

School districts would also be able to freely upgrade as needed without waiting for money to become available. School districts would no longer need to budget for software maintenance and customer support fees, or budget very little for the remaining proprietary software they still have. (Hill & Gaughan, 2006, p. 58). FOSS provides an alternative where there are no longer costs associated with upgrading to the latest and greatest software versions. Individuals have the ability to save upwards of \$2,000 on software by switching to FOSS programs similar to those created by Microsoft, Adobe, and others. The costs savings could be in the thousands of dollars for educational institutions (Schrock, 2007, p. 36).

The information technology field is a very specialized field; school districts require a certain level of competency from their information technology (IT) staff that may not be available for open source. The majority of school IT staffers were trained in proprietary software and may not have the skills needed to implement and support FOSS (Hill & Gaughan, 2006, p. 58). There would need to be a major retooling for IT staff on how to use and deploy FOSS in an educational environment. This would undoubtedly create additional costs for school districts and other businesses looking towards FOSS.

According to Pfaffman (2008), the usage of computers in the classroom has yet to show a significant increase in learning (p. 28). This lack of increased learning could potentially create hesitation to adopt FOSS that is untested in education. Research has also shown that FOSS options are rarely explored in distant learning courses (Reilly, 2006, p. 1). This hesitation to adopt some form of FOSS initiative in education has created a “who’s first” scenario for many educational institutions. In which they look around to see what others are doing and fail lead the change themselves.

There is however some progress in incorporating FOSS at national levels in some countries; FOSS has a promising future in Russia based on recent initiatives. By the summer of 2008, Russia would have begun a three year project to computerize schools (Arutunyan, 2008, p. 1). Part of this project included providing schools strapped for cash open-source Linux software (p. 2). If their plan is successfully it could be used as a model for other schools and countries to adopt.

There are certainly several options for educational institutions when it comes to implementing FOSS. The final decision to switch will greatly depend on whether the current software is meeting the needs of the staff and students (Hill & Gaughan, 2006, p. 58). There will be less incentive for wealthy school districts to implement FOSS over traditional software, and the real question that will be looked at is whether students can maintain or increase their skills and knowledge from the introduction of FOSS. It does appear that FOSS can provide school districts with a low cost alternative to their software needs. Completely transforming from an all paid to FOSS is unlikely within education for several reasons. However, by making minor changing or substitutions with FOSS can



save money for districts, teachers, and students. Every penny saved through implementing FOSS within education could be directed to other programs that are finally strapped.

There is undoubtedly no single or simple solution for school districts when it comes to implementing FOSS. Each must decide what skills and knowledge their students should have prior to leaving. They will need to look at whether the skills and knowledge developed using software is only met through the use of proprietary software, or can those needs be met through the use of FOSS. Research is still lacking on whether the substitution of FOSS will affect student outcomes or influence the skills and knowledge they leave with after graduation at the K-12 levels. But if the research and literature describing FOSS usage in higher education is any indicator, then there are high hopes in seeing greater usage of FOSS throughout education. As more IT specialists become comfortable with FOSS there will also be a gradual push to integrate more of these programs within education. Whether this integration will be seen at the local or national levels has yet to be determined, but it is apparent that FOSS will continue to gain acceptance as a viable alternative to proprietary software.

## METHODOLOGY

### Autoethnography

The qualitative research method called autoethnography will be used to collect, organize, and analyze data through personal experiences. Autoethnographic approaches to research are very similar to that of conventional ethnographic research such as data collection, data analysis/interpretation, and report writing with the intended goal of understanding culture (Chang, 2008, p. 49). In Chapter 12 of Qualitative Research Methods for the Social Sciences, Berg (2009) laid out some common features found in most methodology sections which include subjects, data, setting, and analysis techniques (pp. 390-391). In the following sections I have included those common features and association with the autoethnographic method and how the data will be analyzed and interpreted.

Autoethnography is a research method that has taken many different forms since being introduced in 1975 by anthropologist Heider (Chang, 2008, p. 46). The term is not new, but maybe one that is unfamiliar to some outside the social sciences.

Autoethnography might be better understood by looking at the terms individual components as is quoted from (Wall, 2006) (as cited in Ellis & Bochner, 2000) “auto- (self), -ethno- (the cultural link), and -graphy (the application of a research process)”. The primary data used through autoethnographic writing is that of the “self” in which the writing uses “personal experiences as primary data” (Chang, p. 49). Culture can be considered the “product of interactions” one experiences with the individuals around

them (p. 23). The research process as presented by Ellis and Bochner (2000) (as cited in Chang, 2008, p. 45) can take many different forms such as autobiographies, first-person accounts, lived experiences, personal essays, among others. This data can be heavily fragmented in the form of field notes and journal entries, which are then later coded to find common themes (p. 115).

Heewon Chang (2008) views autoethnography as an approach that “combinescultural analysis and interpretation with narrative details” (p. 46). In the article *Analytic Autoethnography* (2006) published in the *Journal of Contemporary Ethnography* by Leon Anderson, he lists five key features to analytic autoethnography: 1) complete member research (CMR) status, 2) analytic reflexivity, 3) narrative visibility of the research’s self, 4) dialogue with informants beyond the self, and 5) commitment to theoretical analysis (p. 378). These features can then be used to develop a working autoethnography using various writing styles such as: descriptive-realistic, confessional-emotive, analytical-interpretive, imaginative-creative, and there is even the opportunity to create your own style of writing (p. 143).

It is evident through reading about autoethnography that two opinions exist regarding the validity and place that it should have within the social sciences. The autoethnographic method will allow my behaviors, experiences, and setting to be documented and provide context to the data collect. Some researchers view this method as being “useful” and “powerful” in settings that involve “educators, social workers, medical professionals, clergy, and counselors” (Chang, 2008, p. 52). Several benefits include being more interesting and engaging to read, allows the researcher to discover

their role in the culture, and can lead to a transformation or awakening in the researcher (p. 52). There also exist several areas that if not carefully navigated during the research process can lead to poor autoethnographic writing. Chang (2008) listed several of those areas to avoid as being isolation of self, excessively narrative, lack of data, and not maintaining ethic standards during writing (p. 57).

### Subject

The subject for this research will be myself, and I am a willing participant who has given my consent to research and collect data on my personal experiences. There is no expectation that this research will result in harm to myself or others and an IRB is not necessary. The research collected and presented in this thesis project encompasses an autoethnographic look at the use of Free and Open Source Software (FOSS) for a period of 8 weeks in my life.

The following is a brief background on my role in this culture and how I am qualified to write about my experiences with FOSS. I am a 30 year old male employed as a high school technology teacher at a suburban school district in Central New York. This school year will conclude my 5<sup>th</sup> year of teaching technology education. I've always had a keen interest in computers and technology in general, with a desire to know how and why things work. I started using computers at the age of four when my mother purchased the original Mac for, what I was told, around \$3,000. This price tag boggles my mind because the computer had no storage and programs had to be run from 3 ½ inch disks. I used Mac's until I was 17 years old and then switched over to PC's running windows. The switch was primarily a result of my father's job servicing these types of computers

where he was employed, and the money I made from summer jobs made the PC more within my budget.

Throughout this period in my late teens I was always installing, uninstalling, and trying any software that was available to me. This wasn't because I needed the software, but more because I could get access to it and wanted to always see what was available. This was especially easy as I moved off of AOL dialup and began using DSL and then later cable internet service providers. I was able to download programs for trial periods and if I liked the software, I could purchase it or possibly continue the trial by reinstalling.

During college the dorm rooms provided me with high speed internet and countless network shares from other students. The campus LAN made available a new source of MP3's, but there were also commercial programs readily available on the college LAN. It wasn't until graduating college and not having a job that I began looking online and discovered many freely available programs. I thought this concept of someone providing free software pretty neat, but never really looked into the concept or cult movement behind it.

Eventually, I found a job as a public school teacher and never really had issues receiving the latest and greatest software for my classes. The school district appeared very supportive and receptive to software suggestions. They also provided enough licenses in which teachers were able to sign-out installation disks such as Microsoft Office from the library to install on their home computer.

A year later, I took a position in a new school district and soon realized the other end of the spectrum when it came to support and availability software. As a result of hitting what seemed to be a brick wall, I began looking online for solutions or at least alternatives to the big name software. It didn't take long for FOSS to be mentioned in my online quest for alternatives and I soon became a proponent of FOSS. I've gone so far as to create and submit FOSS suggestions to the appropriate individuals within my district, but never received acknowledgments that these suggestions were received. In one meeting in particular regarding my computer labs, I was told that the FOSS could not be installed because of virus and spyware concerns. This threw me off a little because this concern was never raised with proprietary software and no viable solution was provided to allow me to use the FOSS software.

It wasn't until a year or two later that I was able to install some FOSS in my computer labs by insisting they were needed and a part of some proprietary software being updated. I feel that I'm making progress in using and promoting FOSS in my school district, but at snail's pace. As a result of budget and other concerns, I now find myself in a unique position to promote FOSS alternatives to my colleagues, students, family, and friends.

### Data

The data collection will involve an autoethnographic look at my exclusive adoption of FOSS for a period of 8 weeks. The initial field notes and journal entries will be collected through a paper journal and then through FOSS. A foreseen challenge of collecting data during this research will be the continued role change between subject and

researcher. To better deal with this role changing, my initial notes throughout the day will be concise and expanded upon later.

The recorded data will include context leading up to various situations and include personal memory, self-observation, self-reflection, and external data (Autoethnography as Method, 2008). Where appropriate screen shots will be included to provide a better understanding of the context or show the particular software being analyzed. Entries of field notes may include date, time, setting, behavior, context, computer logs, and any other interactions with FOSS that are deemed significant.

### Setting

The two primary settings for the data collection during this research will be my place of employment and residence. The residence setting should be the easiest to manage and record data because of my full control over the technology within that setting. The employment setting on the other hand will require some creativity and being diligent about remaining on FOSS for the 8 weeks. As a technology teacher who uses computers daily for both content delivery and student assignments, I will need to assist students on non-free and open source operating systems such as Windows XP. To address these possible situations, I will allow myself to assist others only through verbal directives and will not assist in means that requires my personal use of non-FOSS.

### Expectations

There are three general expectations during this autoethnographic research including:

- o Commit to only using FOSS for a period of 8 weeks.
- o Continue daily computer use and activities similar to that prior to adopting FOSS.
- o Individuals (mainly students) may ask and receive help from me on using non-FOSS only through verbal directives.

### Analysis Techniques

Berg (2009) stated that ethnographic data analysis is not an “exact science” (pg.228); things seem to get even less exact when it comes to autoethnographic analysis. There are a handful of writing styles that can be used in autoethnographic narratives and the option of creating your own style, along with countless ways to encode and analysis data. Throughout this process I will organize, code, and analyze the collected data in a way that justifies and provides meaning to this experience. The collected data highlighting both the high’s and low’s of using FOSS will be looked at in a way that will provide context to both the situations that arise throughout the research and the overall understanding of what it means to be “off the grid” with FOSS. I will provide both analyses on how things work and interpret cultural meanings with regard to the data collected (Chang, 2008, p. 127). This will be done through short narratives in a descriptive-realistic writing style, which will provide analysis and interpretation either leading up to or after the narrative excerpts. The narratives will be single spaced, indented, and italicized to better differentiate between the analysis and narrative portions.



## **DATA ANALYSIS**

During this 8 week period, I had the opportunity to try several of the better known FOSS titles, along with several titles that I stumbled across. On an almost daily basis I was searching through the Ubuntu software repositories for anything that looked interesting. This provided me opportunities to integrate this software both in my personal life and introduce new titles to individuals at my work. In the following two sections: Elevation and Lowering, I provide several short narratives and a brief analysis on each.

Throughout this process my experiences with FOSS consisted of experiences that elevated and others that lowered my view of FOSS. These experiences were grouped together to include themes that elevated my position on FOSS such as piracy, redistributing, locating, and others during the 8 weeks. There were also themes that presented themselves during this period which lowered my view of FOSS which included learning curve, barriers to entry, ease of use, and glitches.

### **ELEVATION**

#### Piracy

Hopper (2000) (as cited in Teston, 2008, p.67) found that “almost half (48%) of elementary and middle school students believe software piracy is legal”. I personally believe most individuals could not see themselves walking out of a store with a box of software they hadn’t purchased, but do so through other means because they don’t physically view themselves as stealing. I was able to have a different vantage point through this process, in which the term piracy was not associated with FOSS. It no longer represented an act in which I could be a part of or be associated with while using FOSS.

This was evident one day in a computer networking course consisting of a smaller group of six students, which I taught. Having a class so small allows me have more personalized discussions and hear more of what was discussed between students on a daily basis. I was able to see how piracy manifested itself and how FOSS was able to play an alternative role in several circumstances. These events and actions by my students got me thinking more about the culture of FOSS and how the act of piracy was nonexistent within this culture; a foreign term. There wasn't a need to worry about license keys or having the right to use particular software, all of which appeared to make FOSS more attractive.

### *Offering software*

*One of the students sitting in front of his black Dell tower and LCD monitor in my networking class shifted his body to the side of the chair to face the rest of the class. He started out saying as a general announcement that he was having issues with his PC running Windows XP. He continued saying that after downloading and watching a movie online that he began receiving all types of pop-ups and his computer became exceptionally slow. Another student sitting about four tables away looked up over his monitor and responded, "You're still running Windows XP?" The student looking up only seemed to pick up on that the other student was running an older version of Windows and not that he was having issues because of a virus or malware that he received online. The student having the issues responded by saying, "I'm still running XP, because I don't want to pay for Windows 7." It wasn't that he couldn't afford it, but rather he didn't want to pay for something that may not benefit him, but rather stick with what he was familiar with and already had. As if it was not that big of a deal and that he was providing a service, the student who was surprised to even hear XP mentioned said, "I have 7 and can give it to you, you won't be able to update it because the program I use disables the authentication check and also the updates." The student in need of the fix looked excited and responded with, "really?" The two students spoke a little bit longer about Windows 7 and described the features they liked, such as the window snap feature and others. The conversation seemed to piddle out after that and each student returned to their work on the computers. The conversation never progressed to the point of either agreeing to share the software that day.*

### *Forgotten USB*

*The same student, who quickly offered up a pirated copy of Windows 7 left his USB drive in a computer he was using the previous day. This was not the first time this has happen to him and was almost expected given the miniature size of the drive. Because it was a shared classroom, another teacher had found it this morning and pulled it out of the front USB port it was plugged in to. She then plugged it into the side of her laptop and examined the contents to determine who it belonged to hoping to find a name on a document. About an hour after finding the USB drive, I came into the room to drop off my bag and hangout a few minutes before going to my hall duty. Sitting at the teachers desk, she held up the USB drive and said that if **\*\*Student Name\*\*** asks for his drive it is right here, she then placed it in the top drawer of the desk. As the desk was closed she began to mention that the student had lots of programs such as Microsoft Office 2010 and others; all of which appeared to be illegal copies. It didn't surprise me at all to hear this given that the student had already offered pirated software. Later in the day as I sat at the teacher's desk the student came into class and stood next to me. He proceeded to ask if I had found a thumb drive, which I started to point out in the drawer I had opened. I never inquired with the student about the contents he had on his USB drive, when he picked up the drive from the open drawer. He then walked to his usual class seat and threw out a figure on how many times he has forgotten his USB drive, as he plugged it back into the front of his computer.*

The conversations these students had indirectly dealing with software piracy does make me wonder how prevalent it is in high school and how it is morally viewed. I couldn't image any of these students physically walking away from a store without paying for an item, but there conversations made this seem like a common occurrence. These two events took place shortly after starting to use Ubuntu, so the notion that there was a free alternative to Windows may not have yet taken root. They were using the software daily, but not making the connection between school and home. So where was the disconnect? Could a better awareness of FOSS lessen the use of pirated software? A little further into using Ubuntu, one of my students decided to put the OS to work on his home PC in place of his malfunctioning Windows OS, which he had not correct yet. The student who was offered a pirated version of Windows 7 never followed through with the offer and instead began using Ubuntu as his primary operating system.

### *Ubuntu resources*

*A little over a week after initially being offered the copy of Windows 7, he turned around from his computer in class and made a general announcement, “Oh, I installed Ubuntu on my home computer.” It was very common for this particular student to stop what he was doing, come to attention, and speak to the entire class. I followed up with his comment by asking how he liked it and whether he did a clean install or installed it along side Windows. He said, “I did a clean install because my version of windows was messed up, and the installation was easy, I just had to plug the computer directly into our cable modem during the installation.” He continued by saying, he was installing more software from the built in software center, and wanted to know if there were any other places to find programs. I told him that there are repositories that can be added or activated in the software center and that he should Google “Ubuntu repositories” to find more possible resources. He sat there for a moment as if he was processing the response, or maybe thinking of another question; eventually he turned back around to his computer*

As a teacher, I try to encourage students to go beyond just the assignment and always wonder what happens outside of the 43mins I seen them daily. It was somewhat rewarding to know that this student had made a switch with very little prompting from me to FOSS. For the next several days he would come into class and ask questions about Ubuntu, some of which I didn't have answers to and tried to directed him to online forums. I felt the need to include him, the best I could, in the FOSS culture and encourage him to explore further. The FOSS culture is really something that you and only you can decide to accept and become a member. This was evident with my students, all of them used FOSS, but without knowing and understanding FOSS they could not necessarily be a part of the culture FOSS has created. There were times that I was worried about students that were interest in FOSS losing sight of the bigger picture and would occasionally throughout a fact or something interesting about FOSS.

### *Ubuntu and iTunes*

*He turned around from working on a computer assignment one day and asked, “do you know if iTunes would work on Ubuntu?” I responded by saying,*

*“I’m not sure, but there is a program that you can download through the Ubuntu Software Center called Wine, which allows you to run some basic windows applications.” At this point I stood up from behind my desk and began walking towards the student, and I continued by saying, “Wine works well with standalone .exe files, but anything that needs add-on software like DirectX probably wouldn’t work.” He quickly asked, “So I can download that through the software center?” Before allowing me to answer he said “I’ll try that”. He then spun back around in his seat and continued working on the computer, which really entailed him playing a dirt bike game he had downloaded through the software center earlier.*

*The next day he came to class and said about 10mins in, that he had tried Wine, but it kept giving him permission errors. I thought for a second and then remembered this happening to me and after fiddling around for awhile, I was able to fix it. I told him the solution that had worked for me by saying, “I had the exact same issue when I tried Wine and finally solved it by going into the properties of the .exe file and un-checking a permission checkbox.”*

The conversations relating to Ubuntu had slowed down after these initial conversations as the students became more focused on learning to use Joomla, which was another FOSS. I wondered if the student using Ubuntu at home had ever gotten any of his .exe files to run under Ubuntu. Would these .exe files be programs that were FOSS based, or would they include pirated software? If he had gotten wine to work it would have given him added incentive to stay with Ubuntu knowing that he had not completely given up all ties to Windows. It was my hope that through introducing a program like Wine to these students that the transition to Ubuntu would seem more inviting. There exists a comfort level with individuals; some are able to change routines without problems, while others may need to bring elements from the past to provide comfort for the future.

### *Wine program*

*Several days after the Wine software was initially discussed a student said, “Mr. B, I installed Ubuntu last night.” When I asked why he had installed Ubuntu, he said, “I got bored and decided to try it.” He had also been using Windows XP, but his change was not prompted by an event, but rather a desire. He also mentioned that knowing that a program like Wine existed made his decision a little easier, having that connection back to Windows was important to him. He had also mentioned, as he sat and spoke while facing his monitor, that he*

*downloaded the wrong Wine program, but was able to get it working on the second attempt.*

To this day the two students that had installed Ubuntu are still using it as their primary operating system, and I still hear the occasional comments about how they discovered something new in the OS. They had even begun telling me interesting features or programs they had discovered through the Software Center. I began to see them developing a fondness for FOSS that continued to build through its daily usage. It appeared to be a treat for them to have and explore something different than what the masses were using.

Just recently for a final exam grade, I assigned a take home exam that involved using various commands to Ping, Tracert, etc. among other networking related topics. No longer strictly using FOSS after the initial 8 weeks, I hadn't thought about students not having access to a Windows OS at home. I had created this exam a year earlier and all of my students at that time had been using some form of Windows OS. Despite not having a Windows OS, I was able to direct a student using Ubuntu on networking programs that were similar to those he needed for the exam.

### *Final software*

*One of the students chimed in as I discussed what they needed to do for the final in two weeks. He asked, "Do I need to use Windows to complete this? I'm running Ubuntu." I quickly asked, "Could you use another computer at home that has Windows to complete the exam?" In a somewhat slow and drawn out response he said, "ummmm, no I only have access to Ubuntu". This took me a little by surprise because I hadn't thought about students not having access to or using a different operating system at home. After his initial question, I had thought that his parents or a sibling would have a computer running Windows for him to use. Thinking about the situation for a few seconds, I remembered that Ubuntu has a networking folder that has similar programs that would perform the same tasks. I said, "Actually, if you go to the Administrator folder on the top menu, move down to the Networking folder, you should have a program or programs in there that perform the same tasks. Also, you could complete the take*

*home exam in the library or use another computer in the school.” He responded by saying that he had upgraded to Ubuntu 11 and that menu bar was no longer available, but he would look for those programs.*

### Locating

Searching for proprietary software prior to using FOSS typically involved visiting the manufactures website or possibly finding and purchasing a downloadable version off of *Amazon.com* or a comparable site. I found it a lot easier searching for software either through the Ubuntu Software Center, which allowed me to add additional software repositories, or visiting sites like Sourceforge.net. Ubuntu has created a software center, where programs are categorized into 13 different departments and these programs can be downloaded through 6 software source categories (Hopkins, 2010). Another alternative for FOSS is Sourceforge.net which had over 260,000 open source projects back in April, 2011 (Sourceforge.net, 2011). It was interesting to see services like these that did not promote software for financial gain, but rather through a community of individuals. These individuals would rate and rank software based on user input and number of downloads. In some instances, I relied on comments or recommendations by anonymous users, whose membership to this culture and this community was validation enough. The software I finally settled on for recording data was easy to located and installed without having to unzip or go through a prompted install, which I had been accustom to in Windows.

### *Ubuntu installed*

*I’m one of those individuals who tend not to read a user manual or help guide unless I’m absolutely stuck. Ubuntu wasn’t an exception and once I had the operating system installed and running, I wanted to see what this OS had to offer. Looking at the desktop I wasn’t sure where to begin; the menu that I was accustom to at the bottom of the screen was now at the top, and the desktop was*

void of any icons. The mouse cursor moved the same, a good steady pace as I pushed the mouse left and right. Moving the mouse to the upper left of the screen, I positioned it over the Ubuntu logo and clicked. It just seemed like a good starting point because Windows Vista and 7 both used a logo to open up the start menu. Sure enough a menu opened, I wasn't sure what it offered yet; but it was a start.

Once the menu opened downward, I slowly moved the mouse over the listed folders and watched as sub-menus opened. There were fewer programs in these menus than I had anticipated; a few familiar names and some unknowns appeared throughout the menus. Not wanting to waste time and remembering how much I hated recording everything by hand, I began looking for software that could be used for field and journal notes. Up until now I had been using a combination plan and record book typically used for planning out weekly teaching schedule. I could have used LibreOffice (aka. Open Office) which was already available in the menus, but I wanted to see if there was software specific to recording data. My cursor was still floating on the dropdown menu with an icon at the bottom labeled "Ubuntu Software Center". Left-clicking once on this icon/label quickly opened a new window.

With the Ubuntu Software Center window open my attention was first drawn to the Department section which had individual categories of software. I decided to use the search box in the upper right corner, because I wasn't exactly sure what Department would contain software for recording journals. In the search box I typed in "Journal" and as the letters were being entered into the search box, results were appearing in the window below. Each of the results had a very brief description below its name and icon. The first result described a "chronological log" which mentioned the word "contact". This wasn't exactly what I wanted, so my eyes wondered further down the list until stopping on an icon whose color and design made it stick out from the rest of listed programs.

Its name was Lifeograph and was described as "Keep a private journal of your life". With the exception of "private" in the description, this sounded close to what I was looking for and I selected it from the list, and then clicked on the Install icon. I watched as the status bar appeared and began to fill next to the program, which represented its installation progress. Once it finished, I didn't exactly know how to open the program, there were no icons on the desktop or little icons in the main menu bar. After navigating to the Applications menu bar at the top of the screen, I began moving through the various folders. Almost immediately I saw the Lifeograph program listed under the Accessories folder; I never really understood why it would be listed as an accessory and not an Office program.

Once I found Lifeograph, I left-clicked on it once and it opened into a somewhat uneventful main screen. With only two options to click on, I select the "Diary" menu and selected the "Created A New Diary". Rather quickly I became



*accustom to the program and how it kept a log file with all my entries and categorized based on dates. It also allowed for searching and had a built-in calendar to make navigating easier and eventually was the deciding factor with using Lifeograph.*

### Modifying

Large proprietary software companies are seen by some as giving too little attention to the needs of institutions and the students they serve (Wheeler B. , 2007). There is a general feeling that many large proprietary software makers give institutions little, if any, flexibility to adapt the software to their needs (p. 1). Through introducing a content management system called Joomla and learning to use the program in tandem with my student, I was able to see the benefits of being able to modify programs to meet individual needs. Even though we were not modifying source code, this program allowed me and my students to see the strengths of having control over how software functions. It also provided additional resources online, which users had created to supplement or modify the program based on the source code that was freely available.

My networking course was able to use Ubuntu and Joomla to create a content management system of their choosing. Initially it was a little rocky getting started and understanding how entire websites could be created using this software took a little time. The idea of creating individual articles for all of your content and assigning them to modules was the initial barrier to overcome. Several students were a little disappointed because components and extensions for this software would not always work with their templates. In some instances, students were able to find discussions online about how to work around problems they faced. This was made possible by the fact that this software was open source and individuals had developed solutions based on this openness.

## *Joomla project*

*As I stood in the middle of the room, I began to introduce a program called Joomla to my networking students. As I discussed what Joomla was and how we would be using it, I walked around and handed out a single sheet of paper. On the paper was a description of a Joomla mini project to show students a FOSS alternative to costly website content management software. I provided examples of similar software which the school uses, and said that I'm starting to see the Joomla name more and more at the bottom of websites.*

*because it was Friday and wanting to give the students a break, without actually having them goof around I told them to get onto the internet and that we would be setting up Ubuntu to utilize Joomla. I also thought this would be a good opportunity because earlier in the week at the district technology committee meeting it was mentioned that the district was redesigning their website for next year. I kind of thought it would be great if I can show, through student work, what FOSS can do in hopes of having the district view it as a feasible alternative. I also recalled seeing Joomla on my old high school's website and a church website that my wife would like us to attend. I found it interesting seeing Joomla pop-up at various times online just prior to introducing it in class. The feeling was similar to when you buy a new car, thinking it's something special, just to begin seeing your exact make and model everywhere you look.*

*It wasn't until the following Tuesday that we had all the machines up and running using Ubuntu and Joomla. A great benefit being a technology teacher and having several preps is that I'm always learning and experimenting; it also makes it difficult to stay in front of students once they take off on an assignment. I wasn't that familiar with Joomla yet and was trying to stay updated through online tutorials and the occasional YouTube video. Standing in the middle of the classroom, I attempted to coach my six students through logging into Joomla and navigating to the administration section. This task was the easy part and once there, I did what I normally do when it is either too difficult to explain or I'm not able to demonstrate on a projector; I told the students to begin clicking and exploring on their own. As they did this, I walked around the room and watched. After several minutes, I saw that their interest and attention began to fade because they were lost and didn't understand what they were clicking. In an attempt to draw them back in, I told them to visit the Template section of Joomla. Once there I described that Joomla had two types of templates, one for the front end and one for the back end. The front end template, as I described it was for visitors to see and the back end was what you as a designer would see during the modification of your site. After describing the templates I told them to go online and search for templates that could be used for either the front or back end. The interaction between students after they began finding templates was interesting to watch. They began telling each other "look over here" or "look at this template"; one student even found a template for cats.*

## Ease of Use

British Educational Communications and Technology Agency commissioned a study that found free/open source software easier to use than their proprietary equivalent among middle and high school students (p. 30). Throughout the 8 weeks using FOSS and introducing various projects to my students that required its use, I would agree with that study. There were of course situations that arose where students and I included found certain tasks difficult, but the vast majority of FOSS used was intuitive enough to allow us to work through those problems.

Starting a new marking period at the same time as I switch over to FOSS allowed me to make an easier transition over to a new grade book. There were several other options for grade book software found through open source searches online and through the Ubuntu Software Center, but I decided to go with a web based solution called <http://www.engage.com>. Engage was not my first choice, but after briefly using a grade book called SchoolTool I decided it was a better choice. This trial and error process of finding FOSS was something I had starting to become accustomed to because of the ever increasing number of available programs. This also concerns me somewhat because despite the perceived easiness of using FOSS, it only takes one bad experiences early on to remove an individual's interest in exploring FOSS further. I was fortunate through the use of Engage and other FOSS to have a very positive experience when it came to ease of use for the software.

### *Finding a gradebook*

*Searching online for an open source grade book using Google brought me to SchoolTool, which looked promising from its screen shots online; a little bland visually though. It also had instructions on how to install it on a system running*

*Ubuntu, but those involved using the command line, which I was not fully accustom to yet in Ubuntu. Not being that familiar with the Ubuntu command line, I blindly followed the instructions on the SchoolTool website and hoped for the best. The process really wasn't difficult, but for those accustom to installing programs with a quick double-click of an icon, this could be a little intimidating. Once I finished the last step in the installation tutorial, I opened Firefox and typed into the URL 127.0.0.1:7080 which brought me to a login screen that needed the default username and password. Once logged in, I spent a few minutes moving around and clicking on anything I saw on the screen. This was typical of me trying something through trial-and-error, rather than reading a manual or how-to. Despite getting it up and running, I really wasn't impressed with the overall look of the program and found it to have a cheap look with everything being in very plain looking tables. I also got thinking that my students would not be able to access their grades through this program without me setting up a server, which was a problem in itself from my attempt with Moodle. For these reasons, I decided to bag this option for a grade book and continued looking online for alternatives the next day. My planning period only had a minute or two left, so I decided to come back later in an attempt to find a grade book.*

*The next day I attempted to tackle a grade book again and sat down in front of my laptop at a work bench in my DDP room. The search began online using the same method as yesterday, but I changed up a few keywords. After scrolling through some previous results, I happened to stumble upon an online grade book called Engrade which was discussed on an online review form. The review was mixed and I looked on additional sites for more reviews. Collectively the negative comments were mainly to do with server uptime and users having hit-or-miss luck accessing the Engrade server. These comments were a little old as I saw the dates they posted, so I went ahead with creating an Engrade account in hopes that these issues had been fixed. Creating an account was easy and didn't require me to provide random personal information, which I had seen on some websites. I was glad to notice "Free" in several different spots on the Engrade website, but I questioned what they really meant by free. Was it free for a trial period? Free for limited use? I really wasn't sure, but had no way of knowing without creating an account. I also wasn't able to determine whether Engrade was open source, as one web page had mentioned in its review.*

*The form required me to enter my name, email address, username, and password. There were no ads or annoying pop-up's on the site, which can convey a cheap or money driven agenda. Several minutes after creating an account by hitting submit, I received an email with additional information on how to log in for the first time. Once logged in, I began clicking and finally went to the help section to fully understand how to use Engrade. The help section was very well done, to the point, and organized in a manner that made sense to me. I created a dummy class using the instruction in the help section and added some fake students to help me decide if this would work for my new grade book. After*

*spending some time playing around with it, I found its responsiveness and ease of managing grades exactly what I was looking for and accepted it as my new grade book.*

One of the strengths of FOSS is that it could be used, modified, and redistributed freely for any purpose, by any individual (Wheeler D. A., 2007). It took awhile for my students to understand that they could download and use at home the same software that we were using in class. They hadn't quite understood yet that FOSS meant it could be used wherever and whenever, without restriction. I found myself promoting and being a cheerleader for FOSS through the mere act of utilizing it in all of my classes.

Parents also showed interest knowing their child's grades and being able to use the software at home for missing assignments. They would send emails asking about the software being used in class and how they could download it at home in order to help their child get caught up in class.

### *Using Engrade*

*At the end of every week I did a quick check of grades on Engrade to see if any student has an overall grade drastically lower than the average. This week was no different and I pulled two names of students, whose parents I begin writing a brief email to informing of their missing assignments. My emails were the generic form that I used saying, "I'm emailing to inform you that \_\_\_\_\_ is missing a number of assignments...." Replacing their names, grades, and missing assignments to match that pulled out of Engrade. If I'm luckily, I typically get a response back 50% of the time from a parent and this was no different. Within about an hour after sending the emails, one of the parents writes back asking if any of the missing assignments can be done at home? In most instances in the past I had to write back saying that the labs and assignments had to be done in class or possibly after school, but this response was a little different. Since using FOSS as the sole software for students to work on assignments, I have been able to reply to parents with a URL link and brief description on what needed to be done to complete an assignment. Not only did this parent receive their child's grades, but they were also provided URL's to the software used in class. In the two emails sent I also included log in information for Engrade, which had descriptions and additional materials for the assignments the students were missing.*

The ease of use for software I believe goes beyond just using the software, but also entails the process of getting that software up and running. I found Ubuntu to be rather intuitive and easy to install, with the only difficult part prepping my computer and creating a bootable CD. The initial barrier for new users to FOSS, myself included, comes down to the attractiveness and ease of use for various types of software. To have individual's successfully promote FOSS it requires a collective effort by all those within this culture to create and support each other in developing user friendly programs. One such development that demonstrated an easy entry into was the installation and use of the operating system called Ubuntu.

### *Installing Ubuntu*

*The day prior to installing Ubuntu on my laptop I had downloaded and burned a bootable installation disk. Sitting down on my couch I typed in <http://ubuntu.com> in the URL box of Internet Explorer, which placed me on the Ubuntu home page. The website had a clean, orange colored look to it that was inviting and simple to navigate. The menu banner had a link "Download" which I clicked on; the appropriate version (32-bit) of Ubuntu was already selected on the download web page. I clicked on the "Start download" link and a popup banner inside Internet Explorer came up asking me to confirm the download, which I agreed to and the 693mb file began to download. After several seconds the download banner leveled off with 16 minutes remaining on the download from the University of California at Santa Barbra FTP server. There was also an option to have a pre-burned CD shipped to you, but the wait could be 5-6 weeks and donations are encouraged to offset the costs of production and shipping.*

*In the meantime, as I waited for the download to complete, I walked into my basement and opened a filing cabinet which contained various blank floppy disks, CD's, and DVD's. I opened a Staples 25 CD spindle and removed a single CD and walked back to my laptop. Several more minutes passed by as I watched the time countdown to finish. Once completed, I opened a program called InfraRecorder, which was a FOSS I had downloaded earlier from the Sourceforge website. Once the program opened, I selected the Write Image button, selected the Ubuntu ISO file downloaded in the Downloads folder, placed the CD into the laptop drive, and pressed the OK button. The burning of the ISO took about 5mins to complete running at 20x burn speed; once finished I took a permanent pen off the fridge and wrote Ubuntu on the CD.*

*Not wanting or really being allowed to change the content of the hard drive already in my work laptop, I decided early on to purchase an additional notebook hard drive that I would switch for the duration of the project. This process was something I had to do, but it is unlikely that the average person would need to go through the added steps of purchasing and installing a new hard drive. It also provided me the option of working with a clean slate without the chance of losing previous files, or corrupting an install through a dual boot.*

*Typically, I order computer parts online off sites like Amazon or Tigerdirect, but I didn't want to wait several days for the shipment. Instead I drove to one of several BestBuy stores in the area which was about 10mins away. Entering the store I received the usual "hello" greeting and then proceeded to look around for the computer section. Once I saw some of the computer monitors peeking above the selves, I begin to make my way over. I ended up moving up and down two isles until coming across some desktop hard drives and eventually seeing one 250 GB Western Digital Scorpio Notebook SATA for \$44.99. I didn't really need one this large, but being the only remaining notebook drives on the shelves, I didn't have many options. Walking up to the checkout, I saw a young woman at one of the checkout counters who wasn't waiting on anyone. Walking up to the counter I pulled out my wallet from my pocket and placed the hard drive onto the counter. The young women gave me the usual Best Buy greeting as she rang up the hard drive, "Hello, do you have a Best Buy rewards card?" Just as she finished the greeting I handed her my key sized rewards card, which she scanned and said, "\$48.59". After swiping my Discover card in the reader I was handed the hard drive in a bag with the receipt.*

*Arriving back home, I headed back over to my couch and removed the purchased hard drive from the bag and placed it on the coffee table. My laptop was running, so I shutdown and unplugged the power adaptor from the back; I also removed the battery from the laptop. Flipping the laptop face down on the coffee table I picked up a small Philips head screwdriver I had taken out earlier and removed the two screws marked with a hard drive logo. The hard drive & chassis was then slid out from the side of the laptop and one additional screw was removed. The new hard drive was removed from its box and electrostatic bag and installed in the laptop. The whole process of removing and installing a new drive in my Dell Latitude D620 laptop took about 4 minutes.*

*Once everything was reinstalled, the laptop was powered on and because the burned Ubuntu installation CD was still in the laptop, it immediately booted into the installation process. The laptop was also connected to its power adaptor and plugged directly into my cable modem as suggested on an Ubuntu installation screen that appeared after booting.*

*The first screen asking for user input gave the option to "Try Ubuntu" or "Install Ubuntu". I chose the Install Ubuntu option and continued with the installation selecting the Forward button after each screen. The remaining*

*screens asked me to select the appropriate installation drive, provide username and password, and other general information seen on many operating system installations. I personally liked the availability of background information on Ubuntu during the installation. It provided some general information on what to expect with Ubuntu once the installation completed. Following the installation the computer was restarted, I logged in and accepted the recommended updates that appeared in the Update Manager.*

My overall experience with Ubuntu was enjoyable and productive, and once I became familiar with the overall layout its usage became second nature. The speed at which the programs opened and the availability to switch between desktops was great. It was easy to switch backgrounds, but I had some difficulty creating shortcuts to programs on the menu bar. Ubuntu was different, but just similar enough to Windows to make my transition easy and the overall usage was intuitive enough to make the experience enjoyable.

I also had the opportunity to remove my SATA hard drive from my laptop and use it in my desktop computer when I needed a break from the laptop keyboard and screen. Immediately after installing the hard drive into my desktop computer Ubuntu was up and running with no changes or modifications needed by me. The dual screen capabilities of Ubuntu automatically worked with my two LCD screens and the wireless card seemed to work. Because of the almost unbelievable ease of switching between my laptop and desktop, I continued to swap back and forth for the remainder of the 8 weeks. It was great to experience something that just worked; all those experiences between swapping hard drive, dual monitors, and more were successful because of the continued efforts to improve the software.

Being able to print whenever and wherever at work has been somewhat of a luxury for me because not every teacher has direct access to a printer. However, because



of incorrect drivers installed on my laptop the printing tended to be confined to a single room when printing from my laptop prior to adopting FOSS. When I tried to print to any printer outside of my primary classroom, the program I was using typically locks up and eventually crashed. Having the experiences of truly having a plug-&-play OS made me more invested in pushing the limits of the software and see if they could exceed my expectations. This experience showed me the ease at which a random device could be added to your system and Ubuntu would do its little scan and within no time the hardware would be functioning from minimal user intervention.

### Printing

*Picking up my laptop and the USB cable, I make my way to an adjacent classroom of mine which had a brand new color HP LaserJet. I seemed to recall when hooking it up earlier in the year that it had a USB connector, among others. As I walked up to the printer I could see the USB port on the back, just next to the power cable. I balanced my laptop on top of the printer and connected the cable to the back of the printer. Taking the other end of the USB cable, I plugged it into one of four USB ports on my laptop. Almost immediately, I noticed a little print icon appear in the upper right corner of the window. Never installing a printer under Ubuntu, I decided to navigate to the printer section which I was accustomed to under Windows. Navigating to the Ubuntu menu I selected System -> Administration -> Printing and anticipated having to install drivers or search for the printer. Surprisingly a printer was already listed and the name happened to be the one it was plugged into, minus the "n" at the end of its name.*

*My document was already open in LibreOffice (aka. OpenOffice), and I was able to navigate to File->Print and select the newly installed printer. I waited for a second wondering if this was going to work, and soon enough I heard the printer going through its warm up and out came my single sheet I was a little relieved knowing that not being on the school network wouldn't restrict me from printing.*

### Portability

FOSS not only can be used freely in schools, but it can also be used at home or in the workplace (Hill & Gaughan, 2006, p. 58). It was difficult at times to convey to

students that the work they are doing in class can be easily done at home through downloading the software. I tried to make it a point whenever introducing new software to show students where the software can be found online. Continued use of FOSS by my students eventually eroded away some of the misconception that assignments worked on in one location required that same location to complete.

### *Taking it home*

*Walking into my computer programming class I quickly turn on the projector, unhook its VGA cable from the back of the desktop computer it was plugged into, and attached it to the back of my laptop. Luckily I had a little bit of time for everything to power on because the student's computers took some time to power on and log into. Once everything was ready I directed their attention to me, which tends to work best when I mention grades, exams, etc. Soon enough the chatter stops and I direct their attention to a quick presentation describing a final project that was put together in LibraPresenter (aka. OpenImpress). The project itself was to create a program, of their choosing, using some type of software they had used thus far in the course. As I described the assignment, I walked around the room handing out a single sided sheet also describing the project. There was not a single question, which could be good or bad sign; I wasn't sure yet.*

*It wasn't until the last few minutes of class that a few questions popped up, one of which had to do with working on the assignment outside of class. A student who chose to use Android as his language of choice wanted to know how he could work on the assignment at home. I didn't really expect this particular question would go along with the App Inventor software because I thought the students in the class already knew that the program was accessible off of Google's App Inventor website. In response to his question I said, "Yes, you just do exactly what you've been doing in class. And because your app is saved online, it will be accessible anyplace you are online." Because I wanted to make sure everyone knew that the software we have used in the class can also be downloaded and used for free at home, I walked to the center of the room and made a general announcement. Saying, "I just want everyone to know that all the programs we have used in this class are free and can be downloaded and used at home, and your current project can also be done at home."*

*I wondered how much of what I said was actually heard and understood, because when Friday rolled around another student asked a similar question. He wanted to know if he could work on the assignment at home over the weekend, and I explained to him also that he could download the exact same software that we use in class.*

Trying to find CAD software that was FOSS and that would also be available on Ubuntu was a little harder than I anticipated. I did find one program call QCad, but I was reluctant to introduce it in fear that students would get confused since they just came from AutoCAD. Instead, I decided to introduce something new this year that involved architectural and room design using a program called SweetHome3D. This FOSS was not only available on several platforms, but was also provided through their website using a Java program that could be run right online.

### *Finding Sweet Home 3D*

*I stumbled across a program called Sweet Home 3D while searching online using the Google search engine for FOSS software. I needed something for my students in Design & Drawing for Production to use that was either CAD or design based. The screen shots and description off the website looked promising and not above the difficulty level of the freshmen students I taught. Clicking on the Download link at the top of the screen brought me to a somewhat cluttered page, but I could see Windows, MacOS, and Linux listed. I was glad to see that all three of the big OS's were listed because even though I was using Linux, I wanted to make sure that my students were able to use the software regardless of their OS. There was also one large link that stuck out on the download page called "Launch Sweet Home 3D with Java Web Start"; I knew what Java was, but wasn't familiar with Web Start. Curious, I click on the link and was presented with a popup asking if I'd like to open the JNLP file. Clicking OK, I waited about 10 seconds and watched my hard drive LED flash away. I was a little concerned with the wait time, 10 seconds in computer time seems like forever. Soon enough a Sweet Home 3D banner presented itself and the program appeared on the screen. This was something new that I wasn't familiar with; the program could be run simply by visiting the website. This was great because it didn't require installing on the school computers, worked on Ubuntu, and appeared to be a very capable program.*

*After spending about two days playing around with the software on my laptop, I decided to introduce it as a program to create a simple room design. As cumbersome as it was attaching and detaching my laptop every period to the projector, I did it again for my Design & Drawing for Production Class. Introducing anything more than a few steps a day to this class created confusion and a slew of questions, so I started simple. Opened up Firefox and typed in the Sweet Home 3D website and began to explain that their first project was to create a bedroom that was 150 square feet with at least 15 items. I demonstrated how to click on the Java Web Start link to open the program and showed several very*

*basic commands to create walls and place items on the screen. They were then told to go over to the computers and visit the Sweet Home 3D website and start up the program. It was interesting to see how quickly they moved over to the computers on days that I spoke a little longer than average. Within a few minutes they were all logged on and visiting the website, several students needed additional reminders that they were not downloading the Windows installing of the program, but rather running it off the website. One drawback from running it off the website was quickly realized when hands began popping up in the air and students saying it wasn't loading. Quickly looking at a few screens I told the class that the program was loading, but is slow because everyone is attempting downloads at the same time.*

Several of the students took to using Sweet Home 3D quickly and began designing entire homes. One of my students asked for a library pass several days in a row to work on his room design in the library during his lunch period. The portability of this program was great because the library was only running Windows, and he didn't need to install anything directly onto the computer. Another student called me over to show an entire first floor plan of a house he created at home and saved on a USB drive using the Sweet Home 3D Java Web Start.

### Retooling

The information technology field is a very specialized field; school districts require a certain level of competency from their information technology (IT) staff that may not be available for open source. The majority of school IT staffers were trained in proprietary software and may not have the skills needed to implement and support FOSS (Hill & Gaughan, 2006, p. 58). There would need to be a major retooling for IT staff on how to use and deploy FOSS in an educational environment. This would undoubtedly create additional costs for school districts and other businesses looking towards FOSS. There are certainly several options for educational institutions when it comes to

implementing FOSS. The final decision to switch will greatly depend on whether the current software is meeting the needs of the staff and students (Hill & Gaughan, 2006, p. 58). As a member of the technology committee, the title sound a little more impressive than what we actually do, I was able to hear of situations in the district that related to IT staff experience and training with software. It was apparent through brief discussions on the topic that IT staff in the district will need to be retrained and stay updated on new software that would be implemented next year. The software wasn't FOSS, but this interaction demonstrated that regardless of whether software is proprietary or FOSS there will always be a need and cost associated with training support staff.

#### *District technology committee*

*Sitting at one of the eight tables in the library at one of the district's elementary schools, I listened as the technology director discussed upcoming changes. One of those changes was the switch from a Novel network server over to a Windows network server. He didn't really provide specifics as he stood in front of about the 12 people who showed up to the meeting, but one thing mentioned was that the switch to Microsoft would make it easier for the IT staff in the district to support. I never really liked or disliked Novel, but it was apparent through network downtime and other issues that we were not able to support what we had. Sitting there I never said anything out loud, but seriously questioned whether switching everything over to a Windows Network Server would really help the situation, because the issues seemed to be a lack of knowledge in maintaining what we had and not the software and hardware. I wondered if all this effort was going into training individual for a system that was pricey, why not explore other options along the FOSS avenue and save some money?*

#### Compatibility

Compatibility is not something mentioned much by proprietary software manufactures, but some have adopted strategies to make their software non-compatible with FOSS alternatives. This however is not the case when looking at FOSS software, which in many instances has the capabilities to open competitive proprietary documents

(Zhaoli Meng, Sang-Yong Tom Lee). Many of the open source software packages available today have the ability to read and manipulate files that were created by a different program (Hill & Gaughan, 2006, p. 58). There were numerous situations that came about where FOSS provided a great alternative for students to complete work outside of class. It also allowed me to open more file formats than I had prior, and was used to help student print and save work that was not compatible with current school software.

### *GIMP to the rescue*

*Typically I have about 2-3 students that would come in to use the computers afterschool despite having A's in my class, and today was no different. Two of the students were sitting next to one another and the third was about four computers down. I was sitting across the room and would occasionally glance up to see what they were doing and whether anyone needed help. One of the students said in a quiet voice, "Mr. B, I'm having some trouble." I stood up from my desk and walked over to where he was sitting and said, "ya, what's up?" Nothing on his screen looked familiar from class, so I waited for him to collect his thoughts and ask a question. He began by saying, "I'm trying to open an image that I was working on at home in Photoshop and I can't seem to get it to open." I told him that we don't have Photoshop in this lab; they do have it in a photo class downstairs. Seeing that his USB drive in the front of the computer was flashing I remembered that there was a portable version of GIMP that he could download and put on his USB drive. I told him to go to Google and type in "portable apps" and locate a program called GIMP. Once I walked him through how to download the program, I began describing more about GIMP and how to use it instead of Photoshop. He was rather excited and grateful to be able to open and export his photo at school, but I'm not sure if that ever translated into using GIMP at home instead of Photoshop.*

## **LOWERING**

### Learning Curve

Rob Lineweaver raised the question in his article *Cost Savings of Open Source Software in the Server Room*, "If open source software saves people so much money, why

isn't everyone using it?" Lineweaver puts it simply as being a "learning curve" that many individuals and organizations are not willing to undertake (2003). There were moments throughout this project that I felt less prepared in prepping and using software than I would have liked. For example, Moodle was a course management suite that I had a lot of difficulty implementing. It was not an issue of being inferior software, but rather my inability to configure the computer prior to installing Moodle. This difficulty was discouraging for me, but I continued to fiddle and search for solutions beyond the typical time I would have spend otherwise. There is a learning curve with the use of all software, but I found it especially difficult to tackle both the software and operating system at the same time.

### *Attempting Moodle*

*I decided to setup an additional computer in my basement running Ubuntu and using an older HP that had been sitting around for a while. Once everything was hooked back up and I performed another Ubuntu install, I decided to give Moodle a try. Initially, I went to the Software Center and attempting to install through a process that I had become very familiar with over the past week. As I began to type its name in the search box, sure enough it popped up with the install button waiting for me to click. Installation was more of a trial and error for me because Moodle needed other software and configurations done on the host computer, such as setting up an Apache server, MySQL, and other configurations. This wasn't really an issue with Moodle, but more of an issue of my inability to setup a web server and database using an operating system which was unfamiliar.*

*The installation and configuration of Moodle initially involved viewing a local webpage that walked me through the installation and setup. The installation started off good until I reached the page that wanted to check my current configuration. At the bottom of the screen was a Check button, which I clicked on and within a few seconds a new page appeared listing three color coded labels: Pass in green, Caution in yellow, and Failed in red. These labels were next to various components, some familiar and some not. The green labels appeared the most, but I had four red labels and two yellow that required my correcting before moving forward with the installation. My eyes moved to the first component labeled with Fail, I read the component name and performed a Google search with the component name, Ubuntu, and the word fail. One-by-one I searched for a solution to each of the failures and was able to follow troubleshooting sections on*

*various webpage's to solve all the problems. After implementing each possible solution, I would click Check again and if everything went correctly, there was one less error on the screen. After what seemed like an exceptionally long period of time I was finally able to view the main Moodle administration and log in screen through my Firefox web browser.*

After several attempts I was able to get Moodle installed and running on the spare computer of mine. My hope was to have that computer be a dedicated server for trying Moodle and providing students with a connection at school to my home IP address. I was able to create accounts, add content and change general settings with Moodle, but my inability to access Moodle on any computer other than the server really made any further development pointless.

As an alternative I did explore the possibility of using free only Moodle services that would store and host any content. I created several accounts at various websites, but most returned in search engines had since closed or put a freeze on allowing new courses/content to be uploaded. Others charged fees which didn't seem appropriate given the topic of this project and my desire to only use free and open resources. I felt that having two programs, Ubuntu and Moodle, that were continually updated separately made it difficult to locate an installation tutorial that supported both of the newest versions. I encountered situations similar to this in with Joomla, Alice, and other FOSS programs. There appeared to be a greater time gap with FOSS then with proprietary software with regards to those programs and helpful resources, like tutorials and walkthroughs.



## Barriers to Entry

Many school districts have invested heavily in brand names like Microsoft and as a result created a barrier for change (Hill & Gaughan, 2006, p. 58). In many ways these companies have succeeded in branding their software as being secure, robust, and universal; which is another hurdle for FOSS to overcome. The idea that people create and distribute software freely can make it appear inferior to the costly proprietary software already available (Pfaffman, 2008, p. 26). My experiences prior, during, and even after these 8 weeks have shown me that there are many barriers for FOSS to overcome in order to be viewed by the masses as a viable alternative to proprietary software. There exists FOSS that can easily compete with proprietary software and exceed consumer expectations, but the lack of awareness and understanding of FOSS has created reluctance and hesitation. I had noticed on several occasions that individuals working in the IT department of the district were unaware of FOSS that could be used and how that software could benefit not only students, but also teachers.

### *Not knowing*

*About three years ago I attempted, with little success, to have several FOSS programs installed in several of my computer labs. It seemed easy enough; I created several sheets with pictures, descriptions, and licenses in a presentation sleeve from staples and placed it in the mailbox of the individual responsible for giving the OK to install software. Needless to say, I never heard a response to my suggestions. It seemed like the effort put into suggesting the software was a waste because of the lack of an acknowledgement. As a result of not hearing anything back and not having the software I needed in my labs, I sent an email to an administrator explain my frustration. This in turn resulted in a meeting several months later with a conversation on why I was not allowed to have FOSS installed in my labs. It was explain, without making much sense that they believed it was unsafe and that it needed to be scanned for viruses. They also believed that the software currently installed was adequate and that I didn't need this FOSS. This response, along with others made with regards to my attempts to introduce*

*new software to my students rather annoyed me. I felt that I knew what was needed in my labs and that they did not understand FOSS.*

*Fast forward to the beginning of this year when I was asked to help prepare an image for several of our technology labs because the district IT department wasn't sure how to make adjustments to our CAD software. After several emails and realizing that my computer labs would not be up and run at the beginning of the school year I decided to head over to the district technology office. Once arriving I go into the main office of the elementary school and asked where I would find the technology office. The secretary behind the desk thought for a second, you could tell she was forming a plan for getting me to the correct room. Eventually, she starts off saying there are two ways of getting downstairs to the basement. Since I was already inside, she told me to go down the hall to the stair and there will be a red metal door on the left. Once I go through the door, head down the stairs through the basement till I reach the end and there I'll find the technology department. I followed her directions and sure enough I found the office, which appeared to be this newly constructed office. It stuck out with its white walls, lights coming through the glass windows when compared to the surrounding dark, damp, and dirt of the basement. Once inside, I was given permission to work on two computers which would then be images and sent out to the other computers in my rooms. During my time adjusting settings and trying to organize the already installed software, I asked one of the IT technicians if I could install some other free software that I needed. I tried not to elaborate much, thinking that he would make me get permission and eventually be turned down again. But to my surprise, he said it was fine and to also put the programs on his USB drive. He wanted to try these programs out and see if they were anything that could be used in other classrooms. I was rather taken back by this, but just went along so not to draw attention to what I was denied two years earlier. It seemed that IT guy I was working with had a better understanding and acceptance of FOSS than the administrators I had tried to work with earlier.*

### *GIMP alternative*

*The last night of teaching the Adult Ed Photoshop class I decided to introduce a photo editing program called GIMP, which is very similar to Photoshop. When I initially asked if anyone was familiar with GIMP only one individual out of 15 raised their hand. I continued with my little spiel about how this was a free and open source program that you could use at home or on a flash drive to edit photos, and was very similar to Photoshop. Up to this point I had been saying not to purchase any photo editing programs because I would show several free alternatives at the last class. Being that the last class was several days after starting my FOSS only routine, I thought this would be a great opportunity.*

*There was about a minute left before the class ended, so I told everyone that if they had any questions to email me or visit the websites I had suggested*

*throughout the class. This was the last class and I wanted to make sure they had everything they needed to venture out on their own. Several of the individuals said, “thank you”, before leaving. One said I did a good job, but suggested that I incorporate more portraits next time. The last individual to leave was a woman who worked in the IT department for the district and said, “I’ll have to tell them not to remove GIMP from the laptop images.” I personally never even realized that the program made it onto the teacher’s laptops because mine had not been re-imaged in about two year. I responded, “No, definitely leave GIMP on the computers. I use it a lot in my DDP courses and I think it could be useful to put on all the computers”. I was glad to see that some effort, besides my own, was being used to promote some FOSS titles within the district.*

### Ease of Use

Ease of use can be a difficult term or meaning to nail down when describing software. What could be easy for one individual may not be for another using the same software. It could also be used to describe a particular aspect of the software and not necessarily the program as a whole. While I found the overall ease of use for most of the programs used, I did encounter a few issues that make using a particular program difficult or frustrating. I encountered some issues with printing, program layout, and confusion caused by learned routines in Windows that we not similar to those used in Ubuntu.

#### *Attempting to print*

*It was the beginning of the week again and I needed to print my weekly lesson plans to give to my department supervisor. At work I have an HP LaserJet in both of my classrooms and decided to try and print to the older model which is in the room I typically reside in for the mornings. Printing is done through logging into the Novel server using client software, which was no longer possible because I was not on the school network and did not have client software for Ubuntu. I decided to try attaching my laptop directly to the printer with a USB cable. I pulled out a USB cable attached to one of the student computers and a Digiac electronic station and walked over to the printer. Following the power and Ethernet cable to the back of the printer didn’t see a USB port. I didn’t think much of it at first because it just seemed practical that every printer now-a-days would be USB compatible. I continued to look a little longer, moving the printer around a bit and checking out the sides for the elusive port. A little bewildered I gave up and decided to go for Plan-B which involved using a parallel cable, which I knew this printer did support. However, before becoming too investing in*

*this plan I saw that my laptop was missing a parallel port on the back. It had a COM port, which seemed even more antiquated than a parallel port; but no parallel port. At this point I had already wasted several minutes gathering the appropriate cables, which got me no closer to printing.*

*Eventually, when I was in my other classroom I plugged my laptop into that printer using a USB cable. The printer was quickly detected and was visible when I checked the Printers folder under Ubuntu. The initial print job I sent went through without any issues and the colored printout appeared on the top of the LaserJet. I then switched over to my other document I wanted to print, which was a PDF describing a CISCO course I hoped the school would adopt next year. After I clicked the print icon in the PDF viewer, I waited for the sheet to print. After several seconds I wondered if maybe I didn't press the print icon, so I did it again; nothing again. It was almost lunch, so I decided to tackle the printing problem when I had more time. I later found out that this printer work well for single print jobs, but if I sent more than one or attempted to print again nothing would print. The only work around was to unplug the printer cable and plug it back in, which would then allow for a second print job to continue.*

### Glitches

I never had the opportunity to try other versions of Ubuntu or even explore other FOSS operating systems, but a former student of mine happened to be using the Ubuntu Netbook Edition. I happened to run into him at a district technology meeting on a Tuesday afternoon. He seemed very pleased with the OS, but mentioned a feature that I used regularly was not working on this edition. There were small, but noticeable glitches with some features of the FOSS used. While these glitches wouldn't be noticed unless you used a particular feature or used the program in a way it didn't expect. I never had the opportunity to use the netbook edition, but I could relate because software like Alice would have the occasional glitch. Students would joke in my programming class about seeing the dark Alice pop on the screen, which would mean an error had occurred. The errors would provide an unintelligible reason for the error and the program would sometimes need to restart.

### *Previous student*

*Walking into one of our district elementary libraries for our monthly technology committee I see a former student of mine sitting at one of the tables. He is typing on a small Acer netbook and talking to a current student of mine. I walked over and took a seat at the table next to them, saying "Hey" as I sat down. My former student was intensely typing on his small netbook keyboard and talking to my other student about programming on Ubuntu. I mentioned that my networking student have been using Ubuntu for the past several weeks and asked him how he likes it? I also mentioned that I have been using it on my laptop for awhile. He said, "He likes using a lot and has been learning how to program using Python." I was curious if he was online with the netbook using its 3G wireless capabilities, which I recalled that both my brother and wife had on their Acer netbooks, but neither had paid to activate the service. He said he hadn't tried the 3G built in capabilities, but did try tethering it to his cell phone. I mentioned that's what I've been doing with my work laptop, and asked him how that was working for him using Ubuntu. He replied that every time he attempted to activate the tether, Ubuntu would lock up on him. I said, "Really, I haven't had many problems with it." He described that the version of Ubuntu he was using is the netbook version and that other people online have had similar issues.*

One of the most frustrating periods during this project was when my computer entered into sleep mode and my actions of forcibly shutting down resulted in a system crash. Despite my best efforts of finding a solution on various help sites online, I was never able to correct the situation and had to reinstall Ubuntu a second time on my laptop. There was a lack of potential solutions online, and those I found never corrected problem. I believe that as Ubuntu grows in popularity that greater support will become available, but it was lacking at the time I needed it the most. This glitch not only prevented me from continuing what I was doing, but also required a great amount of time resolving the issue.

### *Ubuntu frozen*

*It was time for hall duty again, one of those exciting times of standing around for 43mins seeing if students had passes to be in the hallways. The last several years I used this time to read books, or perform a balancing act of grading papers to pass the time. Most of my hall duties were far enough away from the library that the wireless signal from the library's wireless access point*

wouldn't pickup. Since rooting my phone and using Ubuntu, I have been able to take my laptop with me and do work or look like I'm doing work during my Hall duties. I have yet to evolve into one of those teachers with a mobile cart moving throughout the halls, and instead opted to exhaust my arms holding the laptop in one arm and perform pigeon typing with the other. I would receive the occasional comments from passing faculty and staff who would observe my balancing act. One custodian passing by said, "Hello" and soon after asked, "are you able to get wireless down here?" Here being a corner of the building that students like to loiter in and not known for its wireless connectivity. I said, "No I wasn't able to get on the school's wireless but I have my laptop tethered to my cell phone." Realizing that she probably didn't know what tethering was, I continued with saying that the laptop was going through my cell phone's internet access. She said, "Oh that's neat." and continued down the hall. The other comments I would receive were similar to this, but once or twice I would go out without my laptop and people would comment "no laptop today?"; as if it was a part of me that was missing. I liked the ability to stay connected to the internet during hall duty, because I didn't feel like everything had to stop for 43mins twice a day as I waited standing in the halls for something exciting to occur. It gave me the ability to keep updated on emails between parents, faculty and the ability to do work which seems ever increasing depended on the Internet.

I was getting ready to head downstairs to the corner of the building I stand in for my second hall duty of the day. I double check my phone to see how the battery is holding up and checked the time remaining on my laptop's battery. Everything looked good, so I closed the lid and curled my fingers around the sides of the laptop, similar to caring a book while walking. About two minutes later I'm in my usual standing position outside the auditorium and lift my laptop onto the left arm to balance and flip the screen up. It wasn't uncommon for the monitor to be black for a few seconds as I move the touch pad and press radon keys to get it to respond. This time seemed to be taking longer, so I look over the bottom right of the screen to see what the LED lights were doing. I still had power and the Wi-Fi light was lit up, but still nothing as I zigzagged across the touchpad and no random blinking of the hard drive light. I waited a few seconds, tried again; still nothing. I remembered this happened a lot with these laptops and that is why many teachers walk with their laptops slightly open from class to class, in hopes of them not going into standby or having the monitor turn off. I did what I always did when this happened with this laptop when it was running Windows XP, hold the power button down till it turned off and press it again to get it up and running. I hated doing this because anything I had on the screen prior to closing it would be lost; despite this hesitation I forced the laptop to power down. When I pressed the power button the LED lights flickered and the screen came up to the usual bios screen showing the usual black screen and some white text. After the bios finished loading the screen turned black and a few seconds later a GNU Grub screen appeared. It was a black background with white text, giving the options to load Ubuntu, Ubuntu with Recovery, or a memory test. I selected the standard Ubuntu boot using the arrow keys to select and the enter key to accept.

## Alternatives

Despite the attractiveness of FOSS as an alternative to proprietary software, there are some shortcomings. There are certain applications that currently do not have an open source equivalent. One of many examples is the elementary school program called Accelerated Reader which is widely used, but there is yet no free equivalent (Hill & Gaughan, 2006, p. 58). I discovered rather earlier on that some of the documents and files I can created in the past were not accessible to my while running Ubuntu. This resulted in lost time and productivity on my part, because some of those documents had to be recreated. Events similar to this had me questioning whether FOSS could really be an alternative to those individuals already using other platforms, or would it be better suited for new users who had little to lose.

### *Can't open the file*

*Sitting at my after school during interaction I thought it would be a good idea to update some posters and flyers for technology courses next year. Most of the work had already been done previously using Microsoft Publisher and the 24inch plotter and color printers in my classrooms. After plugging in my external Western Digital Passport hard drive into my laptop I navigated to the drive and began digging through the folders. I'm usually pretty organized with files, but constantly creating and saving files on various machines over the years makes my searching take longer than I'd like. I manage to find a JPG image of an Engineering Technology poster, but wasn't able to find my brochures by navigating through the folders. So I decided to try the file search feature under Ubuntu and through in some keywords into the search box like technology, .pub, brochure, and soon enough several results popped onto the screen. The two files Computer Technology Brochure.pub and Engineering Technology Brochure.pub were in the returned results and I attempted to open them by double clicking; I realized that I didn't have Publisher on the computer, but I thought that possibly Open Office would have the appropriate program or conversion.*

*Unfortunately, every attempt to open the Publisher file either resulted in garbled text or inability to open. I decided to search for a solution online, but the primary solution involved opening the document in Publisher and copy the content into another program like Scribus. This wasn't possible because I was on an only FOSS regiment, so I decided to just redo the documents in Scribus.*

## **PREPPING FOR FOSS**

The spotty wireless and inability to connect non-approved devices to my school districts network led me to work out a temporary solution to maintain internet access during this project. The temporary solution I decided on was to try was rooting my Motorola Droid device running the FOSS Android operating system. The device was already out of warranty and I had previously read online about users rooting their phones to gain access to wireless capabilities without using a USB tether. A benefit of having a phone running Android was that it is considered to be FOSS, which allowed me to maintain the FOSS theme at all ends of this project.

### *Getting everything out of 3G*

*Using Ubuntu and a program called Super One Click Root, I was able to root my phone which provided me full access to use an App called Wireless Tether, which was available on the Android Market. This program was freely available on the Android marketplace for rooted phones. In short, I was able to use my phone's 3G wireless connection as a hotspot for my laptop running Ubuntu.*

*There were several benefits of doing this including no longer being restricted by the district filter which provide similar website filtering for both students and teachers. The other benefit was being able to remain connected to my phone regardless of where I moving throughout the building. During hall duties I was able to roam up and down the halls with the laptop in one hand and my phone in my pants pocket. This was great because the wireless connectivity provided by my district was spotty at best. Once in awhile my signal would dip from being inside, but I was never disconnected.*

*There were two issues that were more of a nuisance than problems when using my phone's 3G wireless. The first was the speed, which was great for browsing the web but downloading any large files or attempting to update Ubuntu*



*would require hours. This wasn't as noticeable until about two weeks of using my 3G connection and from what I have read online could have been a result of throttling done by my wireless provider to limit my data usage. The second issue was battery life on my phone, which wasn't great to start with but would die around mid-day unless I charged it periodically throughout the day. Typically, I would plug my phone into my laptop's USB during planning periods and lunch.*

*The ability to be connected to the Internet was something that I needed both at work and at home during this endeavor. Accessing the Internet at home wouldn't be a problem because I could connect to my home wireless router, but connecting at work would be a problem. Not only was there limited wireless throughout the high school, but teachers were not given permission to connect to the school network. As a result I needed something to allow me to still get online without getting in trouble at work or spending an excess amount of money paying for one of those wireless portable hotspots.*

*I recalled seeing two apps in the Android Market a few months prior dealing with tethering your phone to a computer and use its wireless connection. The problem with these apps was that both cost money for the full blown versions and your phone had to be connected directly to the computer using a USB cable. The idea of paying would defeat the purpose of this endeavor and the idea of moving around constantly with a phone on a leash wasn't practical. There was however one other App that name alone sounded very promising called Wireless Tether, and it was free. The only problem was that my phone wasn't rooted and this app wouldn't work otherwise. Several days prior to fully diving into this FOSS endeavor I searched through and read various forums and articles on rooting an Android phone. There were even several YouTube videos specifically showcasing the rooting process, which put me a little more at ease for the entire process.*

*Several of the methods for rooting seemed a little complicated even for me and their described process of rooting were various vague warnings on performing the process in error. One program particular looked and sounded easy, and only involved a single program to download and use. The day prior to switching over to FOSS, I went back to my bookmarked sites in Internet Explorer, copied the URL addresses into an email addressed to myself, and pressed send. I would later use these URL's to download and perform a root on my phone.*

*Once Ubuntu was up and running with internet access provided through my home router, I went into me email account using Firefox and opened up the URL's pertaining to rooting my phone. Each URL: SuperOneClick software, tutorial on rooting, and a YouTube video were opened on separate tabs to make switching and following their directions easier. I was rather surprised at how easy it was following the provided information to root my phone, and I went through the process a second time thinking I missed a step along the way.*

## CONCLUSIONS

This project provided me a unique opportunity to use only FOSS for an 8 week period of time. While using FOSS during this period, I was able to experience events that elevated and also lowered by view of FOSS. By immersing myself in FOSS I was able to become part of this culture and record my growth as a FOSS user. This allowed me to develop a better understanding of what is involved in adopting FOSS and whether it can be a feasible goal for others to undertake.

I discovered many great FOSS titles and how they can be used on a daily basis as alternatives to proprietary software. Since concluding my 8 week stretch of only using FOSS, I have continued to use various titles both at work and home. Even though I have since begun using again some proprietary software such as Microsoft operating systems, Office, Quicken, and some other titles, I continue to use what I have learned through this project to make educated decisions on the usage of software. This has allowed me to maintain and promote FOSS in my classroom and expand not only my understanding and acceptance of FOSS, but also those I introduce to the culture.

The ability to install Ubuntu alongside a Windows operating system has allowed me to continue using both operating systems on as needed basis. Continuing to use Open Office was an option, but I found it added time for me in creating documents because of my lack of familiarity. Attempting to write this thesis project using only FOSS titles was what finally pushed me back into using Microsoft products after the 8 week period. It became difficult to maintain consistence and reference materials using Open Office. A switch back to Quicken was also done because of the easy of importing banking information directly into the program with a single click update. I still continue using

several FOSS titles such as: Engrade, GIMP, Alice, Firefox, Sweet Home 3D, VLC Player, and others. This has allowed me to balance the best of both worlds, while still giving me the opportunity to discover new FOSS titles as they become available.

Despite switching back to several proprietary software titles, I do believe from this process that FOSS can be effectively used in everyday life. There are shortcomings with various software titles, but the concept and strong support by users will only continue to help elevate and promote this culture of free and open source software. Individuals who have not yet developed strong connection to various software titles are those most likely to succeed in the FOSS culture. Those rooted in proprietary software have the opportunity to expand and incorporate tenets from the FOSS culture and possibly influence their own cultures through the process.

## REFERENCES

- Anderson, J. (2005, 08 22). *New Study Recasts Cell Phones as Effective Teaching Tool*. Retrieved 04 10, 2011, from ergoweb.com:  
<http://www.ergoweb.com/news/detail.cfm?id=1180>
- Anderson, L. (2006). Analytic Autoethnography. *Journal of Contemporary Ethnography* , 35 (4).
- Arutunyan, A. (2008, February 1). Schools to Get New Software. *Moscow News (In English)* , p. 3.
- Berg, B. L. (2009). Qualitative Research Methods for the Social Sciences. In B. L. Berg, *Qualitative Research Methods for the Social Sciences*. Boston: Pearson Education, Inc.
- Blacksell, T. (2011, 04 18). *Schools must embrace digital revolution*. Retrieved 04 18, 2011, from FT.com: <http://www.ft.com/cms/s/2/cd88e0a6-6779-11e0-9138-00144feab49a.html#axzz1JsntPWRJ>
- Carolyn Ellis, T. E. (n.d.). *Autoethnography: An Overview*. Retrieved from <http://www.qualitative-research.net/index.php/fqs/article/viewArticle/1589/3095>
- Chang, H. (2008). *Autoethnography As Method*. Walnut Creek: Left Coast Press, Inc.
- Coar, K. (2006, 7 24). *The Open Source Definition*. Retrieved April 12, 2008, from Opensource.org: <http://www.opensource.org/docs/definition.php>
- Comprehensive Technology Plan - Jamesville DeWitt Central School District 2009-2012*. (2009, 02 11). Retrieved 04 10, 2011, from CNYRIC:  
[http://docspot.cnyric.org/CNYRIC/PublicDocs/ITD/J\\_D\\_TechnologyPlan2009-2012BOCESedition.pdf](http://docspot.cnyric.org/CNYRIC/PublicDocs/ITD/J_D_TechnologyPlan2009-2012BOCESedition.pdf)
- Fabius-Pompey Central School District Technology Plan 2008-2012*. (n.d.). Retrieved 04 10, 2011, from CNYRIC:  
<http://docspot.cnyric.org/CNYRIC/PublicDocs/ITD/Fabius%20Tech%20Plan.pdf>
- Free and open source software*. (2008, April 12). Retrieved April 13, 2008, from Wikipedia: <http://en.wikipedia.org/wiki/FLOSS>
- Group Account Pricing*. (2011, 04 01). Retrieved 04 01, 2011, from MyGradeBook: <http://www.mygradebook.com/pricing.cfm>
- Hepburn, G. (2005). Open Source Software and Schools: New Opportunities and Directions. *Canadian Journal of Learning and Technology* , 31 (1), pp. 1-8.
- Hepburn, G., & Buley, J. (2006). *Getting Open Source Software into Schools: Strategies and Challenges*. Retrieved 12 04, 2010, from innovateonline.info:  
[http://innovateonline.info/pdf/vol3\\_issue1/Getting\\_Open\\_Source\\_Software\\_into\\_Schools\\_-\\_Strategies\\_and\\_Challenges.pdf](http://innovateonline.info/pdf/vol3_issue1/Getting_Open_Source_Software_into_Schools_-_Strategies_and_Challenges.pdf)
- Hill, A., & Gaughan, S. (2006, November/December). Open Source Software: Is It a Solution for Schools? *Library Media Connection* , pp. 58-59.
- Hopkins, B. W. (2010, 09 01). *How to Install Apps With Ubuntu Software Center*. Retrieved 05 20, 2011, from PCWorld:  
[http://www.pcworld.com/businesscenter/article/202571/how\\_to\\_install\\_apps\\_with\\_ubuntu\\_software\\_center.html](http://www.pcworld.com/businesscenter/article/202571/how_to_install_apps_with_ubuntu_software_center.html)

Leithner, A. (2010, 06 17). *change.org*. Retrieved 04 10, 2011, from Are Classroom Gadgets Worth the Money?: <http://news.change.org/stories/are-classroom-gadgets-worth-the-money>

Lineweaver, R. (2003, August 22). *An informal case study in K-12 education*. Retrieved March 30, 2008, from Cost savings of open source software in the server room: <http://staff.harrisonburg.k12.va.us/~rlineweaver/>

NCREL. (n.d.). *Develop a Technology Budget*. Retrieved 04 15, 2011, from North Central Regional Educational Laboratory: <http://www.ncrel.org/sdrs/areas/issues/methods/technlgy/te3lk76.htm>

NetAction. (n.d.). *The Case for Government Promotion of Open Source Software*. Retrieved 04 10, 2011, from [netaction.org](http://www.netaction.org/opensrc/oss-advantages.html): <http://www.netaction.org/opensrc/oss-advantages.html>

Parloff, R. (2007, 03 14). *Microsoft takes on the free world*. Retrieved 07 14, 2008, from CNN Money: [http://money.cnn.com/magazines/fortune/fortune\\_archive/2007/05/28/100033867/index.htm](http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/index.htm)

Pfaffman, J. (2008, Feb/Mar). Transforming High School Classrooms with Free/Open Source Software: It's Time for an Open Source Software Revolution. *91* (3), pp. 25-31.

Reilly, C. A. (2006, March). The price for free software: Labor, ethics, and context in distance education. *Computers & Composition* , *23* (1), pp. 68-90.

Review, C. P. (2008). *SO10 Explore Open Source Alternatives*. Retrieved April 12, 2008, from California Performance Review: <http://cpr.ca.gov/report/cprprt/issrec/stops/it/so10.htm>

Schrock, T. (2007, March). Applications, Open-Source Programs vs. Fee-Based. *Smart Computing in Plain English* , *18* (3), p. 36.

Shaheen E. Lakhan, Kavita Jhunjunwala. (2008). *Open Source Software in Education*. Retrieved 04 10, 2011, from Educause Quarterly: <http://www.educause.edu/EDUCAUSE+Quarterly/EDUCAUSEQuarterlyMagazineVolume/OpenSourceSoftwareinEducation/162873>

Shinder, D. (2005, 04 06). *Is Open Source Really More Secure?* Retrieved 04 10, 2011, from [windowsecurity.com](http://www.windowsecurity.com): [http://www.windowsecurity.com/articles/Open\\_Source\\_Secure.html](http://www.windowsecurity.com/articles/Open_Source_Secure.html)

*Software Industry Facts and Figures*. (2009). Retrieved 04 10, 2011, from <http://www.bsa.org>: [http://www.bsa.org/country/Public%20Policy/~media/Files/Policy/Security/General/sw\\_factsfigures.ashx](http://www.bsa.org/country/Public%20Policy/~media/Files/Policy/Security/General/sw_factsfigures.ashx)

Sourceforge.net. (2011). *Sourceforge.net - About*. Retrieved 04 10, 2011, from Sourceforge.net: <http://sourceforge.net/about>

Teston, G. (2008). *Software Piracy among Technology Education Students: Investigating Property Rights in a Culture of Innovation*. Retrieved 04 10, 2011, from VirginiaTech Digital Library and Archives: <http://scholar.lib.vt.edu/ejournals/JTE/v20n1/teston.pdf>

*The New York State District Report Card - Fabius-Pompey Central School District*. (2011, 02 05). Retrieved 04 10, 2011, from nySTART: <https://www.nystart.gov/publicweb-rc/2010/eb/AOR-2010-420601040000.pdf>

- The New York State District Report Card - Jamesville-DeWitt Central School District.* (2011, 02 05). Retrieved 04 10, 2011, from nySTART:  
<https://www.nystart.gov/publicweb-rc/2010/d5/AOR-2010-420411060000.pdf>
- The New York State District Report Card - West Genesee Central School District.* (2011, 02 05). Retrieved 04 10, 2011, from nySTART:  
<https://www.nystart.gov/publicweb-rc/2010/a0/AOR-2010-420101060000.pdf>
- Wall, S. (2006). *An Autoethnography on Learning About Autoethnography*. Retrieved 05 02, 2011, from University of Alberta Libraries Journal Hosting:  
<http://ejournals.library.ualberta.ca/index.php/IJQM/article/viewFile/4396/3522>
- WGCS D. (2011, 03 21). *An Updated Plan for the Application of Instruction Technology*. Retrieved 04 10, 2011, from cnyric.org:  
[http://docspot.cnyric.org/CNYRIC/PublicDocs/ITD/Approved\\_WG\\_TechPlan%202010-13.doc](http://docspot.cnyric.org/CNYRIC/PublicDocs/ITD/Approved_WG_TechPlan%202010-13.doc)
- Wheeler, B. (2007, Jan/Feb). Open Source 2010: Reflections on 2007. *EDUCAUSE Review* , 42 (1), p. 48.
- Wheeler, D. A. (2007, April 16). *Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!* Retrieved April 5, 2008, from David A. Wheeler's Personal Home Page: [http://www.dwheeler.com/oss\\_fs\\_refs.html](http://www.dwheeler.com/oss_fs_refs.html)
- Zhaoli Meng, Sang-Yong Tom Lee. (n.d.). *Competition and Compatibility with Open Source Software*. Retrieved 04 10, 2011, from <http://csrc.lse.ac.uk>:  
<http://csrc.lse.ac.uk/asp/aspecis/20070112.pdf>

## VITA

Benjamin Besemer was born in Ithaca, New York on April 4<sup>th</sup>, 1980. He attended Ithaca College majoring in Organizational Communication Learning and Design after high school, later transferring to SUNY Oswego to pursue a degree in Information Science. In 2003, he received a BA in Information Science and a minor in Computer Science from SUNY Oswego. A year later he returned to SUNY Oswego and took courses in their Technology Education department, which led to a New York State technology education teaching certificate in 2006. From 2006-2007 he taught high school technology courses at Bethlehem Central School District in Delmar, NY. In 2007, he took a new position at West Genesee High School in Camillus, NY where he is presently employed. In 2008, he enrolled in SUNYIT's Information Design & Technology graduate program.

