Sheffield Hallam University

How can Free and Open Source software benefit the modern micro business?

Michael Ian Baker

B4012339

BSc Hon Computer Security with Forensics

Final Year Project 2018

Supervisor: Mark Townley

Table of Contents

Acknowledgements

The author would like to thank the following people for their help towards the completion of this project.

Firstly, I would like to thank the Project supervisor, Mark Townley, for their extensive support and flexibility with meetings in addition to going above and beyond to help with specific questions relating to the project.

Additionally, I would like to thank the five participants who took part in this project as they gave their time to complete questionnaires, test the program and take part in closing interviews which lead to this project being completed.

Glossary of terms

FOSS – Free and Open Source Software.
OSS – Open source software.
IT – Information Technology.
OS – Operating System.
Source Code – Any collection of computer instructions written in human readable programming language.
Proprietary software - Refers to any computer software that has restrictions on any combination of the usage, modification, copying or distributing modified versions of the software.

Abstract

Due to the nature of their classification, micro businesses will always face the issue of relying on either external IT support or small in-house support teams which each come with their own distinct issues that need to be considered when creating a solution to IT related issues. With micro businesses consisting of ten or less people, gaps in IT related knowledge will be common amongst employees and at the same time, training members of staff may have too much of a financial cost or having the employee take time off for training may result in additional issues.

This paper aims to explore whether Free and Open Source Software may be one of many solutions to help micro businesses develop their IT solutions. The author aims to show this through the use of a developed program that was tested by five participants whom are responsible for their employer's IT services and collecting their feedback to gauge the program's effectiveness. This paper will also contain an in-depth investigation into the various benefits and issues that Free and Open Source Software has in addition to examples of how larger organisations have implemented such programs.

1. Introduction

When a business needs to find software for its day to day operations, it can normally be serviced with licenced programs which are often recommended as industry standard, but in most cases, it can be considered that Free and open-source software (FOSS) can be the answer to the problem. However due to the large market position of many proprietary software companies and their increased marketing power, open source programs are not fully considered by many and their position in IT is often under rated and misunderstood.

Within this project, background research has been compiled to form a literature review focussing on the topics of the history of open source, the financial impacts of FOSS, the security of FOSS, the implementation of FOSS in larger organisations and finally the lack of existing literature around the scope of micro businesses. The aim of this piece of work in addition to the accompanying program is to make use of FOSS as simple as possible whilst still achieving the full potential of such software. This paper will focus on UK businesses with one to ten employees that may or may not have a dedicated IT member of staff. This type of businesses is defined by the EU as micro due to them having "fewer than ten employees" and "an annual turnover below €2 million" (European Commission, 2003). This scope was selected due to access the author has to such businesses in addition to being a demographic that may benefit the most from saving money on licences for software. The program will be given to a select group of local business staff in order to test its effectiveness in a live workplace. The results chapter of this work will be based on the comments made about using the program. This will be in the form of interviews in addition to evaluating the initial questionnaire that was used to decide the types of software that should be included within the program. Screenshots from the program will also be included and commentated in order to evaluate the authors work. This will be further assessed in other sections such as Methodologies and Discussion all resulting in possible future development of the program concept.

At the start of the project, a research question in addition to a list of aims and objectives needed to be created in order to gauge the success of the project. The research question that needs to be answered within this report is "Are free and open source tools the solution to IT based issues in Micro Businesses" and the success of whether or not this question has been answered will be if the following aims where met by the author:

-To discover the main IT issues micro businesses, have and which FOSS tools can be used to help with these issues

-To learn how each program works in order to make a program that will teach people how to use these programs

-To collect meaningful feedback on how the created program performs in fixing their issues

In addition to the three aims mentioned above, an additional aim of creating this report to a high standard is also outlined within the project specification. Although this aim will not be assessed within this work, it is a fundamental are that will be considered when assessing the quality of the project.

2. Literature Review

2.1 History of open source

Open source software refers to any program whose source code is made available for use or modification and users and developers see fit. Collaborative software creation had been around since the 1950's and 60's where the majority of software was produced by researchers in both the academic and corporate sectors and released into the Public-domain which allowed other people to use it and edit it as they saw fit. Most software developed during this era was distributed with both the machine code and source code due to users needing to modify the software themselves for compatibility with different hardware and Operating Systems (OS). The first example of Free and Open Source Software is largely believed to be the A-2 System created by Remington Rand in 1953 which was released to customers with its source code (Ceruzzi, P. E. 2003). In addition to this, the company actively encouraged users to send back any changed and improvements they had made to the Universal Automatic Computer (UNIVAC) division of Remington Rand. Improvements were then added to the software and shipped to new customers and thus the A-2 system became a hybrid consisting of contributions from both the corporate world and the IT community and is considered an early example of open source software. From this initial experience companies quickly realised how useful the IT community could be in evolving their products and lead to almost all IBM mainframe software being distributed with its source code. This lead to the formation of groups such as SHARE which was dedicated to the improvement of the IBM 701 machine through various forums and the Digital Equipment Corporation (DEC) formed a group known as DECUS which was similarly, an independent user group dedicated to the evolution of DEC products through group collaboration (Walli, S. 2016). Arguably one of the largest contributions to the Open Source community came in 1969 when the "Advanced Research Projects Agency Network" (ARPANET) was constructed. ARPANET was a transcontinental high-speed network that allowed the fast exchange of software code thus meaning people from research facilities, universities and the military where able to access modified code even easier. ARPANET eventually evolved into being one of the early subnets of the Internet and was eventually shut down in 1990. However, the open source community was about to be dealt one of its largest blows, companies monetising their software instead of just shipping it with applicable hardware. Due to the rising complexity of computing systems, creating software became a more time intensive effort and thus caused companies to have to invest more money in creating software for their products. Large companies where now voicing their opinions on the efforts and costs associated with software production and wanted to be allowed to protect their work. One such company was Apple which in 1983 managed to form a case against a company called "Franklin Computer Corp" (US Court of appeals. 1983). This case was a landmark case in the history of computer software due to a ruling body deciding that even though the code was only readable by the computer it was built in, it would now be classed as a "Literary work" which is one of the seven categories of copyrightable "Works of authorship" (Nussbaum, J. 1984). The wording of the act which defined what classed as a "literary work" was also redefined to include "copyright protection subsists in works that can be perceived, reproduced, or otherwise communicated either directly or with the aid of a machine or device". From there software companies began protecting their work through copyright and the "free software" movement as well as the open source community where beginning to face the concept of licenced software that a person would need to pay for and yet, never actually own. Licencing was and still is the work around big tech companies use to show that their customers never actually own the software they're paying for, but merely leasing it for a period of time and thus don't have the ability to edit and improve the software as a community.

2.2 Financial impacts

Total cost of ownership is defined as "a measure of the total cost of owning and operating a piece of software, including both the initial monetary cost of purchasing it and any associated costs of implementing it" (Lerner, Josh & Schankerman, Mark. 2010) meaning that the financial impacts of not purchasing or licencing software from a common vendor will last beyond the initial implementation of FOSS. The additional costs involved in switching from proprietary software have been outlined in a study for the UK Cabinet office (Shaikh, Maha & Cornford. 2011). They conducted a two-stage study involving a total of 52 people involving an online printable form and in-depth interviews with industry professionals and found that over half of the respondents had replaced proprietary software with Open Source Software (OSS) and there was a strong agreement that overall it saved their organisations money. They also outline the reasons why the employee's companies hadn't fully adopted OSS and these where displayed in the following table:

Table 1: Summary of comments on factors hindering adoption of open source

OS related issues
Understanding Licences and license compliance
Availability of specific apps
Some OSS is very immature, inferior user interfaces
Sometimes proprietary alternatives are simply better
Feature completeness
[Lack of a] community backing the open source project
Product related issues
Poor coverage in ERP arena ; Lack of availability of open source software for our industry
Incomplete implementations; Not working correctly
Very complex code bases (and communities)
Organisation related issues
Procurement policy
Value to money
Misinformation among upper level management; Lack of knowledge of key technical decision makers; Time availability
Support issues
Lack of in-house support; lack of in-house knowledge;
Understanding by staff; Poor support of open standards by our business partners; Support worries; Requirements for external support contracts
Environment issues
Desire to have specific software; SAP Legacy
Compatibility with Microsoft proprietary file formats
Perceived Lack of acceptance of OSS for Public sector solutions
Proprietary standards used by environment (govt & clients)

Many of these responses are based around issues such as a "lack of in-house support" and "Requirement for external support contracts" which to be resolved would either require an inhouse support member or hiring an external support group such as Red Hat which will require some level of financial investment either in the form of an external support contract or the hiring of an IT professional to work within the business.

2.3 Security of FOSS

When it comes to discussing the idea of "Is FOSS more or less secure than proprietary software" no one answer can be given due to the varying types of FOSS and the commitment of the community into keeping it secure. Literature tends to be swayed one way or the other due to biased towards one style of development. The main argument towards FOSS being a safer option is outlined as:

"Bugs including security vulnerabilities may be spotted by the many eyes – both the experts and novices alike – on the code." (Russell, Clarke & Dorwin, David & Nash, Rob. 2005)

This is a version of Linus's Law (Raymond, Eric S. 1999) and is used by Russel Clarke et al to argue that if a program that is overlooked more often will be more secure, even if not everyone is a professional involved in the creating or maintaining of the software. They also argue that large companies that use FOSS will have a vested interest in its security so will put resources into helping the community make it secure. They also argue that once found, a vulnerability can also be fixed and submitted by anyone to the main development team for review, which is a generally considered to be a faster method than a company sealing itself of to fix an issue itself. Another advantage of having more people review code is outlined as the lack of possibility to insert a backdoor which people may use to potentially harm users of the software which may go unnoticed in proprietary software (Payne, Christian. 2002). Although it can also be argued that the only people who could put a backdoor into proprietary software are the programmers developing it, which normally wouldn't risk their jobs for the chance to possibly harm users. Payne continues to elaborate on the increased transparency of open source discussing the fact that an organization can carry out their own in-house security audit of any open source software in use due to having access to the source code but also outlines how this kind of behaviour is not common practice due to the complex nature of searching through the entirety of a programs source code. The risk of back doors is further developed upon as in the case of proprietary software, the only people who are able to access source code are the development team and people who may be intentionally looking for vulnerabilities (Farooq-i-azam, Muhammad. 2005) With this being the case, when a vulnerability is revealed its chance of becoming public knowledge is very low due to the fact that the developers won't want to publicise a flaw in their program and people studying it for malicious reasons also won't reveal it to the public due to their wish to exploit it. However, as Farooz-i-Azam also outlines there are also drawbacks to the security of FOSS, one of which being that even though people can evaluate the source code of a given program, there is a possibility that nobody has carried out a thorough security evaluation. This may be due to lack of motivation to help assist the community within users, assumption others have already carried out an evaluation or a possible lack of understanding of the programming language by the community. The example given is the programming language python which is fairly new compared to other languages and thus programs developed in python won't have as wide a review pool as another conventional programming language community. The UK Government has also stated that they support both the use of and careful observation of Open Source software in government IT (UK Home Office, 2012). They also go on to outline the fact arguments both for and against the use of Open source agreeing that both sides are inherently true in different cases and reiterate the current GESG Guidance recommends that when adopting Open Source that 'no one particular type of software is inherently more, or less, secure than the other and does not favour one type over the other. Each must be approached on a case-by-case basis.'

2.4 Examples of FOSS implementation

Due to the nature of the scope chosen, there is a lack of documentation on the success of micro businesses adopting FOSS other than examples such as Apache Web Server which is already an industry standard. Due to this, looking to larger scale examples can help to gauge the effectiveness of a switch to FOSS within a work environment. One such example was an analysis based on whether or not the health sector could benefit from FOSS and includes various examples of current implementation (Arslan, M. Oğuz 2014). Within their work one such example is the Jamaican Ministry of Health adopting GNU Health, a Free/Libre Health and Hospital information system. The Jamaican Public Health Care System decided to begin to phase GNU Health into their facilities (Ministry of Health Jamaica 2013) with the help of experts from the National Health Information Service who then worked with GNU Solidario to change GNU Health into a system tailored made for them. This became known as the electronic Patient Administration System (ePAS) and being able to tailor make their software brought about advantages such as "universal access to the record at any facility a patient attends, improvements in the quality, timeliness and completeness of health reports to stakeholders and improvements in personal health information protection through controlled and audited access to each patient's record". Another example of large scale implementation of Open Source discusses the issues faced when the city of Munich, Germany attempted to have a tiered implementation of Open Source Software beginning in 2003 and ending in 2013 with 15,000 PC's migrated to LiMux OS (Silic, Maro & Back, Andrea 2017). Within the article they detail the issues faced during implementation including legal issues related to over 50 European patents, the transfer of over 21,000 templates from proprietary software to their Open Source alternative, lack of training within employee's and sustainability of the OS. Although a much larger scale than a micro UK business the lessons learnt from the LiMux project can be applied to the chosen scope. Areas such as lack of training as mentioned before are key in order to help migrate a business operation to alternative software unless with the help of trained professionals and with more technical programs is impractical due to the possibility of no access to trained IT professionals. In relation to the LiMux project, a vote occurred in 2017 to move back to windows 10 as its standard operating system by 2020. In a report published in 2015, the issues outlined ranged from a lack of full scale implementation due to software not working on Linux, people simply not putting in the effort to learn how to use the new software and slow updates being rolled out to fix common bugs such as printing errors (Accenture. 2015).

2.5 Literature within the scope of micro businesses

As mentioned previously, the limited literature around the scope of micro businesses means that much of the existing information around the advantages and disadvantages of FOSS is based around either larger businesses or government agencies. One of the many reasons for this being the case is due to the face that unlike governments, businesses are normally under no obligation to disclose changes in their operations and financial information. This leaves large gaps within the literature around the FOSS community as members are also under no obligation to share the work they have done, in addition to each example of FOSS possibly having numerous versions all created by different individuals within the community. One example is the security aspect of open source, which in the scope of micro businesses reveals very little current research due to these businesses being under little obligation to inform the general public of any security breaches in software they are currently using. An additional aim for this work is to hopefully help fill this gap in literature around the scope of micro businesses as doing so will encourage others to take on similar investigations.

3. Methodology

3.1 Program development options

At the beginning of the project the decision to create a program based around Open Source was first decided and the method of development was the next priority. Once the project specification had been submitted the next step was to decide which programming language would fit best for the given situation and after thorough review, python was selected due to the authors prior knowledge of the language in addition to its ease of use and active community on forums such as Reddit and Stack Overflow. C++ and Java where considered as alternative languages at the beginning of the project but due to the aim of the program eventually becoming open source, it was deemed that python would be a more appropriate language. This decision was made due to python being considered an easier language to learn for people who do not have prior programming experience, and thus developing the program in python would appeal to a wider range of possible future contributors who may wish to learn a programming language and help contribute towards the FOSS community. Another major deciding factor to choosing Python as the programming language for the project was its current stance as the second most popular language on GitHub in 2017 with 40% more pull requests opened than last year showing how it is an increasingly more popular language for people to collaborate in (GitHub, 2017). Once python had been decided as the development language, a structure for development needed to be implemented.

The development models researched where the Waterfall, Agile and Lean methods, all of which have their advantages and disadvantages for a single person project. The Agile software development model consists of twelve principles outlined within a manifesto (Beedle, Mike et al) and are focussed around delivering frequent results to the client and working well as a team. Due to the program being designed for a concept and for an undergraduate dissertation, regular updates between the author and the sample pool or supervisor would have been unnecessary in addition to steps such as choosing the best people for each job being redundant as it is not a team project. Although such steps could have been skipped it was ultimately decided that the Agile method would not fit the project as a primary development method, although areas such as regular reflection on the work being done where utilised within the chosen development method. The second model that was researched was the Lean software development method adapted from the Toyota Production system (Poppendieck, Mary. 2007). The Lean system is one based primarily on the identification and reduction of waste, which consists mainly of unrequired features, project churn and crossing organizational boundaries (Ambler, Scott. 2016). Due to the lean method being focussed on larger groups much like the agile method, the author deemed it an ill fit for the project as it additionally had a large focus on how the development team would deal with groups such as stake holders and management, both of which are not present in an undergraduate project. After researching the three possible development styles it became apparent that for a small project the Waterfall method would be the most suitable development model due to its simplicity to follow and implement within a one-person project. The following section will detail how the waterfall development method was utilised by the author for the project.

3.2 Use of the Waterfall Method

The waterfall method is defined as 'a static model and it approaches systems development in a linear and sequential manner, completing one activity before the other' (Adenowo, Adetokunbo & Adenowo, Basirat. 2013), so once each step is outlined then a clear path from beginning to end is created. These steps are outlined as Requirement Analysis, Design, Implementation, Testing and finally Operation and Maintenance (Pfleeger, Shari & Atlee, Joanne. 2006) with each step including its own distinct goals that once met, lead on to the next stage of development. The first step to using the Waterfall method is to realise the specific requirements that the product needs to meet, which were decided upon through the answers given in the initial questionnaires between the author and the sample group. The response to certain questions such as "In what areas of work do you find yourself having issues with software?" gave the author an understanding of what problems the developed program needs to help solve to answer the research question. This research in addition to personal experience from the author helped outline which program areas should be addressed within the created program and after talks with the project supervisor it was decided that the program would address two areas that the scope addressed as primary issues. Data recovery and transferring data between hard drives where the two areas that would be addressed and the system design section of the project could begin. The system design step of the waterfall method is traditionally where the main code of the program is written in small sections referred to as units, which will then subsequently be pieced together in the next stage to form a full program. The program was broken down into functions which were developed separately from one another and tested as individual units which was advantageous due to remaining focussed on one area of the program until it was working as intended. This also meant that deadlines can be set for each unit meaning if the clients within the scope asked for a progress update, a detailed review could be given to them about how much work has been completed in addition to what is needed to finish the program. If the project was to be completed by a larger group, units could be distributed throughout the development team which would reduce the time it would take to finish the project and the associated costs. Traditionally in the waterfall method, units are all created separately from one another and then put together to make the final program. This is where the author decided to adapt the waterfall method to meet their needs better and made the decision to incorporate each unit into the main code when the unit was finished rather than waiting for the completion of all units. The author found that being able to test the units to make sure they work together to be advantageous over the traditional method due to issues being found earlier and therefore being fixed earlier and because of this, the integration section of the method lasted throughout the implementation period. Once all units had been finished successfully and put into the main code, post integration tests were carried out in order to test the entire system for any faults or failures. Testing was carried out on various operating systems including Windows 10 Pro, Windows 7 Pro, macOS High Sierra Version 10.13.3 and Ubuntu 17.10. In addition to testing the programs compatibility with operating systems, it was also tested on various devices such as a MacBook Pro 2015, a windows based PC and a HP 15-bs158Sa laptop. After testing was completed in a controlled environment, the program was distributed to the volunteers within the testing scope for them to get to grips with in preparation for the closing interviews that where conducted by the author.

3.3 Sample Pool and Ethics

When approaching the project, the choice of participants was limited to those accessible and those whom may benefit the most from the program created in addition to those whom would be willing to participate in beta testing of the program in addition to agreeing to be interviewed by the author in order to assess the work completed. This method of sampling is referred to as convenience sampling. The sample pool chosen contained five participants who are all working with the IT industry. Micro businesses are being defined in the context of this project as an organisation consisting of one to ten people as mentioned in the introduction of this work.

This sample size was used due to larger business possibly encountering more significant disruption in testing the product in a work place environment. Due to the program being primarily a concept that requires more work, the areas decided to develop for testing where decided through initial questionnaires given to the five participants. The two areas that were fully developed within the program where chosen due to their lack of disruption if completed in a work environment and the lack of existing program knowledge in the specified field e.g. Data Recovery. Although the ideal scenario would have been for all participants within the scope to test the knowledge gained from the accompanying program within a live workplace, some of the participants chose to test the Data Recovery tool Rufus in addition to the programs teachings in a home environment. All participants that carried out such testing agreed with the author that the knowledge learnt from the program could be applied to a live work environment.

The only ethical issue that was encountered throughout the project was the use of external organisations. The participants were consulted and made aware that if they were to test a program such as MiniTool Partition Wizard to not use vital data in the event of an error. This warning was also reiterated within the created program to make sure the author would not be liable for any damages caused to data belonging to the businesses.

3.4 Expected Limitations

During the submission the project proposal various limitations became apparent and where discussed between the author and the project supervisor Mark Townley. The first of these expected limitations came from the given time that was available to complete the project. In an ideal world such a project would have continuous growth and program instructions would be added by members of the community in order for it to become an Open Source program itself, however due to the time restrictions of an undergraduate dissertation only some instructions were able to be completed, the amount of which was decided between the author and supervisor.

Another expected limitation was on the source pool chosen as addressed in the previous section. Although agreed with the supervisor, the choice of only 5 participants meant that the results pool would be slightly restricted due to the participants being from the same geographical area and relative age range. If expanded upon, the author would spend an increased amount of time sourcing willing participants but due to the time restrictions of an undergraduate dissertation this wasn’t accomplished within this project.

Deciding on what programs would be best to fit the need of a proprietary software replacement was one of the largest areas of research during the project and throughout the investigation it became apparent that in some areas there is only one or two viable alternatives which greatly reduces the user's ability to choose from a wide variety of alternatives. This in addition to the lack of support some of these programs have in 2018 led to many categories not being considered for their FOSS alternatives.

4. Findings

Due to the accompanying program being a concept piece that requires further development to be fully deployed, the results discussed will be based on the effectiveness of the aforementioned program within the beta testing environment, the data acquired from questionnaires during the initial investigatory period and the minutes from the closing interviews with the sample pool at the end of the project. The participants were given the option to remain anonymous at the opening stage of the project and all five participants opted in to anonymity, meaning throughout the following sections they will be named as participant one through five which correspond to their responses in the appendix.

4.1 Questionnaires

At the beginning of the project, in order to find out which areas should be investigated by the author a questionnaire that would be given to the five members of the sample pool was created. The questionnaire involved both quantitative and qualitative aspects meaning that some answers where in the form of simple close ended answers that can be reduced to simple Yes/No responses, in addition to open ended answers which could contain anything the participants wishes. The use of both in conjunction with one another meant that the questionnaire was able to gather large amounts of information with a relatively small amount of questions. The questionnaires will be included in the appendix in full however this section will deal with some of the information that sculpted the authors decisions the most and gauge their effectiveness of gathering the data.

The first question within the questionnaire was created with the hope of finding out which programs the participants use on a daily basis to find out which areas of IT the participants generally work in and what tools they use to accomplish this. The answers to this question had similarities in some program areas such as most of them using Microsoft Word and some form of web browser. This helped to eliminate these areas when considering which program area to teach the participants as if they had existing knowledge of the area, it would diminish the level of teaching that could be achieved from the program. The second question was one based around discovering the extent of the participants involvement of their works IT operations. Due to the scope being focussed around micro businesses all five participants where either part of or the sole member of small IT teams. This helped to gauge the level at which the business could benefit from that person's involvement in the project as if the business has only one member of their IT team, them learning a near skill is highly beneficial to the company whereas if there are multiple members of an It team, them learning a new skill may not be as beneficial due to another member of the team possibly having these skills already. Due to the fact that participant four works with two other people concerning IT issues, it is a distinct possibility that learning how to operate a program such as Recuva may not benefit the business at all if another employee can already do it. The third and fourth questions of the survey where focussed around finding out if the participants use any other examples of FOSS that may not be used every day. Examples such as participant one and fives use of Linux distributions meant that it was in the authors best interest to attempt to create the program in a format that would be able to run on both Windows and Linux. This helped with the decision not to package the program as a .exe as without the use of such programs as Wine, participants may not have been able to run the created program. Questions five and eight attempted to gauge the validity of the project by finding out if there was a current need for such a program to be developed. The results indicated that there was very little knowledge of any such programs or schemes and the main area where people learn about FOSS tools where from search engines apart from participant two who informed the author that a college night class he attended was focussed on getting people to use FOSS. With these responses the author felt that the creation of the program would be able to benefit these participants and thus, had a strong chance of being able to answer the research question posed.

Questions six and nine were also grouped together in an attempt to discover which areas the program should focus on helping to teach the participants about. Four of the five participants raised concerns over their lack of knowledge around the how to handle loss of data. Due to the author specialising in forensics, the decision to make one of the two program areas to be data recovery seemed to be an area of expertise that could be shared with the participants in addition to something the author actively uses FOSS tools to help deal with. With question six focussing on existing issues rather than potential new ones, it gave the author valuable insight into what program specific problems the participants had to deal with regularly with many of the participants saying that having to request additional money for new software was a particularly problematic area. Question seven was put into the questionnaire to determine the impact that the participants have with the company as if there are more employees using computers daily, then the participants knowledge may be relied on by more people. The number of employees using computers ranged from five to nine meaning that in the event of something like data loss on a shared network storage device may affect more than half of the company's employees resulting in the potential shut down of the business until the issue is resolved by the participant or possibly an outside contractor which would could the business even more money. Based on this possible scenario the author found it particularly important to teach the participants about such a crucial area within computing.

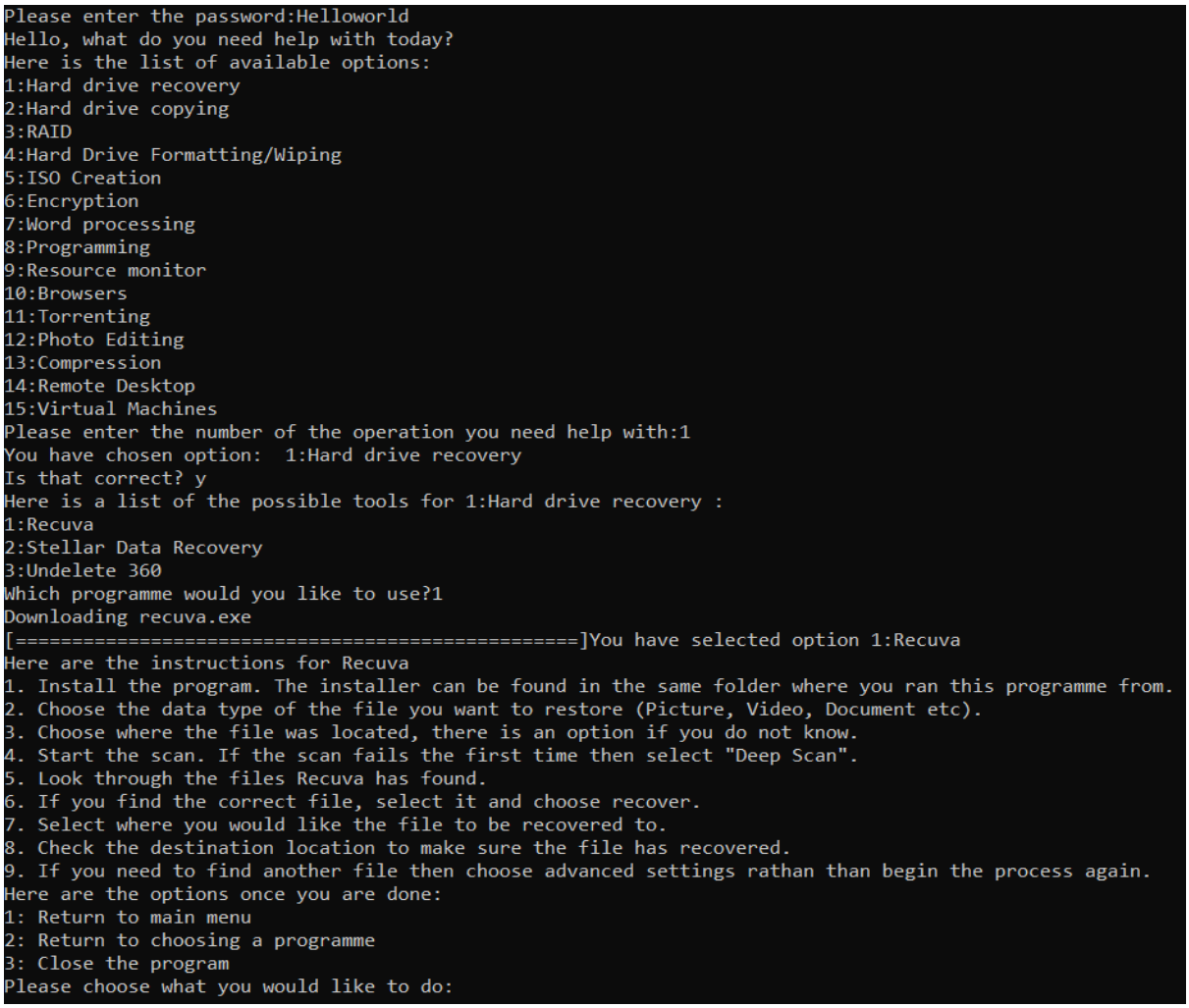
Overall the results of the questionnaires gave vital information on which direction the project should go in. It helped to sculpt both the content of the program in addition to the format of it, which lead to the program being able to benefit all participants as can be seen in the closing interviews between the author and the participants. Further discussion on specific responses to the questionnaires will be included in the following discussion chapter.

4.2 The created program running

The program created by the author uses the python programming language and as such, runs best within the python environment. This is particularly useful due to it being multi-platform with very little work being required to have it run properly other than having python installed. Converting the program to a .exe format was attempted so that the user would require no additional effort when running, however this was not possible due to various issues with creating a .exe from a python script that requires modules to be imported. The end program does require python to be installed on the user's machine in addition to them installing the requests module, however due to all participants being involved within the IT industry, the author concluded that the target audience would be able to install python and use pip to install the requests module. The following screenshots show how the program works in addition to how the program downloads the installer for Recuva. Additional screenshots are included in the appendix which show how the program works for MiniTool Partition Wizard.

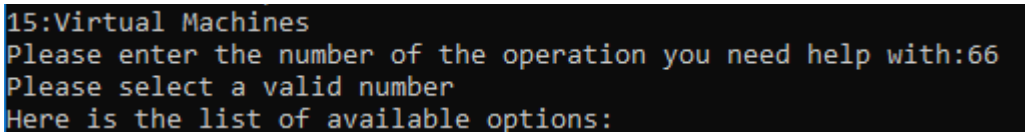
As can be seen in Fig 1, the program first runs through a simple password check to verify the user has been authorised to use the program. Once the correct password is entered the program runs through the areas that can be explored for program options. If the user where to enter any number that does not correspond to an option in this menu, it will inform the user that the option is not accepted and reprint the options for the user to choose from, as can be seen in Fig 2.

Fig 1 – The program running and displaying how to use Recuva



.

Fig 2 – The program asking users to enter a number that corresponds to an option



As discussed earlier in this work, the decision to have two program areas be completed was made. The first two responses are usable currently, although all options will result in an additional error check to make sure that the user has entered the correct area. Any response beginning with Y will result in the program going to the next step, and a result beginning with N will result in the program going back so the user can choose again. If the user where to enter any input not beginning with Y or N, the program will ask them to enter a yes or no answer, as can be seen in Fig 3. Error checks such as these have been entered due to some users believing that FOSS is often not as user friendly and having to restart the program in the event of a user error being extremely inefficient.

Fig 3 – The program asking users to enter a Yes/No response

```
Is that correct? g
Input not recognised, please put Yes or No:
Is that correct?
```

After the user has confirmed their choice, the program will offer the user the opportunity to choose which program they wish to use. Once this has been chosen, the created program will download the installer for the selected program and display instructions on how to use it. The instructions that are shown to the user are displayed in Fig 4, with the installer download being displayed in Fig 5.

Fig 4 – The instructions that are displayed to the user

```
Which programme would you like to use?1
Downloading recuva.exe
[==================================================]You have selected option 1:Recuva
Here are the instructions for Recuva
1. Install the program. The installer can be found in the same folder where you ran this programme from.
2. Choose the data type of the file you want to restore (Picture, Video, Document etc).
3. Choose where the file was located, there is an option if you do not know.
4. Start the scan. If the scan fails the first time then select "Deep Scan".
5. Look through the files Recuva has found.
6. If you find the correct file, select it and choose recover.
7. Select where you would like the file to be recovered to.
8. Check the destination location to make sure the file has recovered.
9. If you need to find another file then choose advanced settings rathan than begin the process again.
```

Fig 5 – The installer for Recuva being saved to the folder where the program is run

Name	Date modified	Type	Size
Screenshots	04/04/2018 21:50	File folder	
Mike Baker Programme	04/04/2018 21:37	Python File	11 KB
recuva	06/04/2018 15:28	Application	5,433 KB

Finally, the program will offer to let the user either return to the main menu, choose a different program or close the program as can be seen in Fig 6. This is also included to aid the user experience and minimise potential issues.

Fig 6 – The final options for the user

```
Here are the options once you are done:
1: Return to main menu
2: Return to choosing a programme
3: Close the program
Please choose what you would like to do:
```

4.3 Closing interviews

Once the programme was finished, the completed version was sent to the five participants for them to test both the program in general in addition to the content within the program. Once tested, the participants reported any bugs they had found in addition to agreeing to a closing interview for the author to gather data on whether or not the participants deemed the created program to be a good method of teaching the desired content in addition to gathering some general comments on their use of FOSS since the project began. The interviews in full are contained in the appendix and where conducted over Skype. Each interview lasted for around 15 minutes and all responses have been recorded exactly as the participant answered.

The first question of the interview asked about if the participant had done any research into FOSS since they agreed to take part in testing the program. This question lead to interesting results as a sub aim of the project was to inform people about the availability of FOSS so some participants saying they had conducted their own research means the project did help gather user interest in FOSS. Question two developed on this further by trying to find out if any new FOSS had been used since the project began, which would again show whether or not the project had encouraged the participants to implement more FOSS into their work and home lives. An interesting example of this is participants two's research into database tools leading to them trying to implement MySQL into their workplace, as this shows that implementing FOSS may benefit their business as they were currently having issues with their database solutions which MySQL may be able to fix. Question three aimed to find out whether or not the participants had used the taught programs in a real-world scenario since learning how to use them. This question would play a large part on whether or not the research question could be answered effectively and yielded some real-world examples of the programs results. Participant one and three had used Recuva and MiniTool partition wizard since learning how to use them however as both programs are only used in specific scenarios it is expected that not all participants would have found a use for them within the time period between testing and the closing interviews. Question four was focussed primarily on gauging the effectiveness of the developed program in reaching more people than just the original five participants as if the development of the program was to continue, the scope of users could be expanded to those that are mentioned in both the question and the responses. Some participants responded saying how if the program was developed further it may lessen the workload they have as instead of them having to teach people how to use certain programs, they could instead use the program that the author developed. Question five aimed to discover which of the participants would look at the source code for the developed program in the hopes that this project may encourage them to actively attempt to either view or add to the source code. Due to a secondary aim of this project being to encourage people to contribute to FOSS, participants one, three and five looking through the source code and feeling comfortable with it was a response for this aim. Participant two did look at the source code but was not confident enough to add anything to it indicating more work could be done on describing how the program works within the code through the use of comments. Due to part of the literature review being focussed on the financial impact of using FOSS, it was in the authors best interests to find out how the participants employers felt about the program concept costing no money to them but still helping their members of IT staff. The responses of question six gave an indication that all employers that where told about the project felt happy with the fact such a program existed and some showed interest in using a more developed version with other members of staff.

Due to the fact the project results needed to be linked to existing literature, question seven was focussed around finding out if the participants carry out security checks on the software they've implemented at work or at home. However as mentioned before, the literature around micro businesses is immensely lacking but the responses to this question may help future research around the subject of FOSS. Some participants openly admitted to not carrying out security checks on software they use, however three of the participants did say the reason for this was that they acquire software from trusted sources, so they feel it unnecessary to carry out their own investigations. One participant however did state that they vet their software choices as working with sensitive data means that a security breach would result in massive consequences for their business. As question eight only included confirmation that the participant used my program, the responses didn't yield any usable results for answering the research question or project aims. The last question was aimed at gathering any additional comments surrounding the use of the program. Four of the five participants gave overall positive feedback on the programme stating it was "simple" and "easy to understand" which helped the author immensely in deeming whether or not the program was a success. Some responses this question also gave invaluable criticism of the program that the author hopes to use during further development of the program, including expanding the areas covered and possibly looking into an option other than having the user download the requests module.

Overall the author feels that the comments made by the participants helps to gauge the success of the project immensely in addition to assessing the overall quality of the program that has been developed due to them representing the view of potential users. Taking into consideration all responses helps give a balanced view on whether or not they benefitted from learning about FOSS in addition to possible future work that the author could do around the subject area.

5. Discussion of results

5.1 Linking results to the literature review

Due to the literature around the chosen scope being limited, matching the results of the project with the literature came with various difficulties. One of the main difficulties came from the lack of time to carry out a full investigation into FOSS implementation in businesses. The literature in this work shows that not all organisations would benefit in the long run from FOSS implementation, however due to the short time frame of an undergraduate dissertation, only partial benefits can be seen in the results of the closing interviews as some participants weren't able to use either Recuva or MiniTool partition wizard however certain responses from the participants did link to the existing literature. One such case is how Farooz-i-Azam details how due to users assuming others had carried out security checks on software, users themselves wouldn't carry out their own security inspection of software. This way of thinking is evident in this project as only one participant responded to question eight of the closing interview saying they carry out security checks on software that they use. Although not unique to FOSS, such findings do indicate a general disregard for ensuring programmes used are secure and don't have security vulnerabilities, which is more impactful to FOSS due to the larger number of possible contributors that may have malicious intentions. This can further be indicated by one of the participants not looking at the source code of the program the author developed, showing how trusting the source is often a much higher priority for users as the author could have included malicious code in the program that wouldn't have been identified by that participant.

Another example of similar results comes from the closing interviews and the table included in the financial impact section of the literature review. One of the reasons given in Shaikh and Cornfords study as to why businesses weren't using FOSS is that often OSS is immature and lacks the content to be considered a full alternative to various proprietary options. These options where mirrored by the participants comments which included how they wouldn't recommend the created program in its current state but would be more likely to if it where a more complete product. Due to FOSS often having smaller dedicated development teams, finishing a product with as much content as a proprietary option is often difficult in addition to FOSS contributors often having no financial incentive to finish the project. If the author chose to continue development of the created program, it would be completed with no financial gain due to it being distributed for free on GitHub whereas if the program was being developed by a company for financial gain, the finishing of the program would be of more critical importance.

As displayed in the literature, finding the full extent of the financial impacts of FOSS takes into account many factors. As can be seen from the closing interviews, all participants that informed their employers of the project where happy that their employees where being trained in using more software at no extra cost to them. The issue of value for money raised in Shaikh and Cornford's study appears to have been mitigated within this project due to the Recuva and MiniTool partition wizard not being considered as alternatives to current software, but more as possible examples of FOSS to use in the future. If the project where to be completed again, trying to find a micro business that may allow a full replacement of a piece of proprietary software would yield results that would be more comparable with the literature around financial impacts however, such a project was not possible this time.

5.2 Program areas selected

As stated previously in this work, the decision to only include two fully implement programs into the developed program was made between the author and supervisor. When this was decided the next decision that needed to be made was the specific program areas to discuss. Due to the author being specialised in computer security and forensics it seemed appropriate to select two areas that would fit this area and that would be useful to the participants. The initial questionnaire gave valuable insight into what areas the participants were either experiencing unsatisfactory software solutions to problems or had little knowledge in that they might want to learn about. The last question of these initial questionnaires was "What areas of IT do you have little/no knowledge of that you may need to know about in the future?" and the responses are as follows:

Participant 1: "The main things are emergency processes such as data recovery or malware defence"
Participant 2: "I'd say I know very little about what to do in the event of a server dying or losing our backups"
Participant 3: "The main area I would say is network protection. Other than installing anti-virus and firewalls I know very little"
Participant 4: "I would say the biggest thing that might damage the company is if we were to have laptops break due to them containing future orders as we have no separate server available"
Participant 5: "The biggest things are emergency protocols such as if someone were to attack our network or if we were to lose data to either employee negligence or a freak accident"

After receiving these responses, the author felt that exploring options for data recovery programs would be appropriate as a computer forensics area in addition to an area that would benefit the participants if included in the developed programme. The other results that the questionnaire yielded when considering covered areas included malware/network defence and backing up data. Due to network defence being such a broad area involving multiple levels of complexity, the author deemed it an inappropriate choice as it would involve multiple programs and settings that would be too case specific to be able to write efficient instructions for which lead to the second area being looked into by the author being data back-ups and processes involved in data transfer such as transferring data from one storage drive to another.

When looking into which programs to write instructions for, a set of parameters needed to be met for a program to be a good fit. The first parameter is that it needs to be FOSS as this is a FOSS based project. It also needs to be available for the developed program to download the installer to minimise the amount of work the user needs to do and this installer needs to be downloaded from a trusted source to help minimise the security risks of each program to the user. With the literature review having a section dedicated to the security of FOSS, making sure the installers are downloaded from a trusted source is paramount to the success of the project as installing a programme with malware included would ultimately be a detriment to a micro business which is why discussing the security of FOSS was included in this works literature review. Additional support being available for the chosen program is also preferred as in some cases extra information may be required by the user although trying to use programs that require less instructions are also preferred due to complex software often not being necessary.

When looking for data recovery solutions, review websites recommended Recuva as the best example of FOSS that fits the needs of users which is an opinion shared by the author as they have used Recuva before. Knowing that the program can help with the issues mentioned such as losing data to employee negligence ultimately lead to it being included as the choice for a data recovery solution.

Even though its use is scenario specific the closing interviews identified that one of the participants had used Recuva and it worked well dealing with their issue. This helped to show that including Recuva in the project did ultimately benefit one of the participants in the short period of time between testing the program and the closing interviews. Direct examples of the program's effectiveness in this short time frame are invaluable as it massively aids in answering the research question posed.

The other area that was to be covered concerned the transfer of data from one storage option to another due to one participant specifically saying their business has no back-ups of their data and others worrying about losing data to hardware faults. Like Recuva, searching for options involved finding reviews of software from trusted sources in addition to asking colleagues and fellow students if they have any experience in FOSS options in the area. After consideration of all options, MiniTool partition wizard was chosen as the most appropriate program due to it meeting all the required parameters in addition to its additional features such as checking for file system errors, wiping specific partitions and data sanitation.

Overall the programs selected fit the specification of the project effectively and all participants agreed in their closing interviews that the information learnt about these programs from the authors developed program will benefit them immensely when the need for such programs arises and thus will also benefit the business they work for.

5.3 Comparing results to what was expected

Overall the author feels that the results gained from the initial interviews, the program testing and closing interviews went very well and gathered large amounts of useful data. When the project began, the author had various expectations and assumptions, some of which are mirrored in the actual results. When creating the initial questionnaires for the participants to complete, the aim was to gather information on both potential content of the created program and the participants opinions and experience on FOSS. The main expectation of the initial questionnaires where that around five or six potential areas would be revealed as suitable for teaching, however only around three or four where actually revealed. This may be due to the fact the scope consisted of participants who all currently worked in IT for existing companies so expected examples such as operating systems didn't come up due to the participants already having established IT systems. However, the author did expect that areas concerning the preservation of data would be mentioned by the participants due to it being such a crucial area in IT.

Concerning the created program, the author had hoped for it to be a more complete product by the end of the project, however as mentioned previously this was quickly realised by both the author and the supervisor at being too much of a work load for an undergraduate project and wouldn't further the authors ability to answer the research question. The feedback received from the participants however went beyond expectations as the only main issues mentioned came from the program not being finished and the need for modules to be installed for the program to run. As previously mentioned, converting the program to the .exe format was attempted however complications in addition to possible compatibility issues meant that this was abandoned early in the project. Positive feedback concerning the program included comments on it's ease of use and simple design, both of which can be issues with FOSS, so this feedback helped the author in deciding whether or not to make the program itself open source as if user feedback was negative, more work would need to be done before possible contributors felt ready to work towards its completion. This opinion was also reinforced by users saying they would feel comfortable contributing towards the development due to the layout of the source code and the included comments. The feedback given concerning the covered content in the program however was extremely positive as all participants said that they now felt comfortable using both Recuva and MiniTool partition wizard.

The closing interviews yielded results that where both helpful for the project and helpful in filling the gap in existing literature around FOSS and micro businesses. Responses to question one of the interviews yielded results that the author feels represent that a main issue with FOSS is lack of potential user knowledge. One example of which is participant two who had not only began conducting their own research into FOSS alternatives, but had actively tried to learn how to use MySQL as a replacement to their existing proprietary software solution. Three of the five participants did say they actively researched FOSS tools during the project which in itself is a positive result as the author feels the more people who take an active interest in FOSS the better as it may help the negative stereotypes around it's creation, intentions and use.

6. Conclusions and Critical Reflection

6.1 Research Question and Aims

When the project began a specification needed to be submitted and within this, the project aims and objectives where decided between the author and the project supervisor. These aims and objectives where used to gauge the success of the project and the following section will go through each of these and detail their level of success.

The first aim was to discover the main IT issues within micro business and how open source tools may be able to help aide them solve these issues. This was mainly achieved through the use of the initial questionnaire which was created to find out specific problems that the author may be able to help. Unfortunately, due to the small sample pool and the lack of time to fully develop the program only certain areas where able to be covered. If more time was available a more in-depth evaluation of IT issues would have been carried out with more participants however all participants did feel that the program did a good job at solving the issues raised so the author feels that the main issues where identified and somewhere able to be solved through the use of the created program with more being possible without time restraints. The second aim of the project was to be able to take the findings from the first aim and apply it to further research which examples of FOSS could help fix these problems. This aim was again limited by the time given for the project so not all areas where addressed in the final version of the program however extensive research was carried out into which tools would best address the main issues raised by the sample pool. The third aim was based around the closing interviews conducted by the author and whether the feedback gathered was sufficient enough to answer the research question. The author feels like the interviews conducted where thorough enough to be able to answer the research question to a good standard and where crucial in deciding whether or not the program needed additional work in order to fulfil its purpose as a teaching tool. The choice of utilising an interview rather than choosing to do an extra questionnaire also helped to answer the research question due to the answers being mainly qualitative questions and thus gave quite detailed responses.

Based on these three aims being met, the author feels that the research question can be answered with sufficient supporting evidence. The author believes that FOSS can be used to benefit micro businesses but the extent of which they can benefit is largely based on which areas the business operates in and the current complexity of their IT solutions and the full extent of these benefits may take additional time to measure. All participants agreed that the knowledge they gained from the created program will benefit hem in the future when the knowledge is needed. The research done within this project also helps to show that organisations will benefit from FOSS more or less than other organisations due to various factors such as complexity of FOSS implementation, the level of expertise of the organisations IT team and the area that the organisation works in.

6.2 Recommendations and further possible work

The following section will contain recommendations and further work that could be done to further develop the created program in addition to additional research that could be done around the scope of the project. One of the main areas that could be researched into is the benefits of long term implementation of FOSS tools within micro businesses. Due to the time limit for this undergraduate project, gauging the long-term benefits of using FOSS is difficult and could be looked into further and finding existing research was extremely difficult as mentioned in the literature review. Another area for possible work is to look into the ability for people to learn from the program who may have different computing abilities amongst different age groups. Due to the scope within this project consisting of similarly ages participants, research into the benefits to children and the elderly may yield results that may help adapt the program to be a general teaching tool for all to be able to use easily.

7. Evaluation

7.1 Evaluating research

The research carried out for this project within the literature review focussed primarily on the advantages and disadvantages of FOSS and not directly linked to the scope of the project. If the research where to be carried out again, more emphasis would have been put on sourcing literature that relates more directly to the scope of micro businesses. In addition to the content of the literature, a wider range of source types would have been advantageous as many of the current pieces of literature are based heavily on things found on the internet in the form of reports and journals however this was done primarily due to other sources being heavily based on bias opinion from either pro FOSS users or proprietary software manufacturers.

7.2 Evaluating the methodology choices

The methodology decisions within this project involved the selection and implementation of the Waterfall method. This development model had to be adapted as mentioned previously in order to be more flexible as by its nature, it is extremely restrictive due to the linear path the user is meant to use. Borrowing elements from the Agile method allowed for this increase in flexibility and ultimately lead to the method being utilised to its high effectiveness throughout the development process. Due to this being the first time the author has used such methodology models, they feel that it taught them valuable lessons on how to implement and adapt various techniques to fit the desired purpose and therefore, the next time such a model is used it can be implemented from the beginning of the project more efficiently.

The other decision mentioned in the methodology section was which programming language to use when creating the program. The decision to use Python was decided due to its ease of use and extensive support available from multiple sources. Overall the decision to use Python lead to the program being able to handle everything that the specification required of it in addition to extra features such as the installer downloader which was added with relative ease and drastically enhanced the user friendliness of the program. One of the few issues that arose when dealing with Python came mainly from user area as often, due to the system that the author was using containing both Python 2 and 3, the respective pip package manager sometimes accidently downloaded the wrong version of modules such as requests. Although this was mainly a user error, it is worth mentioning as a downside to Python as any users who may take on the challenge of further developing the program may find themselves in a similar situation.

7.3 Personal development

Throughout the project the author feels that skills developed will aide them in their future works and prospective career path. One such skill is the ability to understand and interpret people's issues and help find solutions that will aide in the understanding of seemingly complex issues. Initial questionnaires and closing interviews helped show that before the project began, 4 out of the 5 participants had very little knowledge of open source tools and where unfamiliar with solutions to issues such as data loss and after using the authors program where able to recover lost data from their computer systems using Recuva. Due to the authors plan to begin teacher training in the following months after this project ends, learning such skills is crucial to their ability to successfully fulfil their career ambitions. The project also taught invaluable time management skills in addition to thorough project planning which are also transferrable within a teaching environment due to the emphasis on structured development throughout an academic year.

8. References

Accenture. (2015). *Interim report and external investigation of IT in the city of Munich*. Retrieved from https://www.ris-muenchen.de/RII/RII/DOK/SITZUNGSVORLAGE/4027760.pdf

Adenowo, A., & Adenowo, B. (2013). *Software engineering methodologies: A review of the waterfall model and object oriented approach*. International Journal of Scientific & Engineering Research, 4(7), Page 427.

Ambler, S. (2016). *The principles of lean software development*. Retrieved from http://www.disciplinedagiledelivery.com/lean-principles/

Apple computer, inc. v. franklin computer corp., 1983

Arslan, M. O. (2014). *Economic benefits of free and open source software: An evaluation for health sector*

Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., . . . Thomas, D. *Principles behind the agile manifesto*. Retrieved from http://agilemanifesto.org/principles.html

Ceruzzi, P. E. (2003). *A history of modern computing* (Second ed.) MIT Press.

European Commission. (2003). *Commission recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises.*

Farooq-i-azam, M. (2005). *Role of free and open source software in computer and internet security.*

Github.com. (2017). *The state of the octoverse*. https://octoverse.github.com/

Lerner, J., & Schankerman, M. (2010). *The comingled code: Open source and economic development*. Massachusetts, USA: MIT Press.

Ministry of Health Jamaica. (2013). *National health information system strengthening and e-health strategic plan 2014 to 2018.* (). Retrieved from http://www.moh.gov.jm/wp-content/uploads/2015/07/MOH_NHIS-eHealth_StrategicPlanFINAL.pdf

Nussbaum, J. (1984). *Apple computer, inc. v. franklin computer corporation puts the byte back into copyright protection for computer program*.*14*(2), Page 281.

Payne, C. (2002). *On the security of open source software*. Information Systems Journal, 12, 61.

Pfleeger, S., & Atlee, J. (1998). *Software engineering: Theory and practice* (4th ed.) Prentice Hall.

Poppendieck, M. (2007). (2007). *Lean software development*. Paper presented at the 29th International Conference on Software Engineering (ICSE'07 Companion), Minneapolis, USA.

Raymond, E., S. (1999). *The cathedral and the bazaar.* O'Reilly Media.

Russell, C., Dorwin, D., & Nash, R. (2005). *Is open source software more secure?* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=BD98AC3FD7478592ECA969683E8D0657?doi=10.1.1.460.4134&rep=rep1&type=pdf

Shaikh, M., & Cornford, T. (2011). *Total cost of ownership of open source software: A report for the UK cabinet office.* (). London, UK: Open Forum Europe. Retrieved from https://eprints.lse.ac.uk/39826/1/Total_cost_of_ownership_of_open_source_software_(LSERO).pdf

Silic, M., & Back, A. (2017). *Open source software adoption: Lessons from Linux in Munich*. January/February 2017*,* Page 42.

UK Home Office. (2012). *All about open source: An introduction to open source software for government IT* (2nd ed.)

Walli, S. (2016). *My history of free and open Source Software*. Retrieved from https://medium.com/@stephenrwalli/my-history-of-free-and-open-source-software-1bac0a75a6ba

9. Bibliography

This bibliography contains a list of websites the author used for additional information

https://www.reddit.com/r/Python/

https://stackoverflow.com/questions/tagged/python

www.python.org

https://opensource.org/

https://stallman.org/

https://www.techradar.com/news/the-best-free-data-recovery-software

https://www.lifewire.com/free-disk-partition-software-tools-2624950

10. Appendix

10.1 Initial questionnaires

Participant 1	
What computer programmes do you use at work daily?	Normally the only programmes that are used daily are Microsoft word, Google Chrome, Windows Defender for regular anti-virus checks, Wireshark, Putty and Active domain stuff
Who manages the IT side of your business?	I'm the person who implements all IT related matters in the business, but all decisions are made between me and the owner
Do you currently use any free/open source software at work? If so what software do you use?	Notepad ++ and Linux distributions are the main ones I use at work
Do you currently use any free/open source software at home? If so what software do you use?	At home I'm a lot more unrestricted and don't have the same budget for software as my work so I try to use as much as possible. The main ones I use are open office and thunderbird for my emails
Would you be interested in a programme that teaches you how to use free/open source software?	Yes, very much so, the main issues with free and open source is that sometimes the tutorials on how to use them are very confusing
In what areas of work do you find yourself having issues with software?	There haven't been many prominent issues with software at work, but I find there's certain things that I could learn how to do with open source rather than going straight for paid software such as data recovery
How many employees does your business have that work with computers?	Everybody at my work has a laptop so 6 people
Are you currently aware of any programme or scheme that encourages people to use free/open source software? If so which ones are you aware of?	Apart from sourcing them yourself I've not heard of anything that advertises open source alternatives
What areas of IT do you have little/no knowledge of that you may need to know about in the future?	The main things are emergency processes such as data recovery or malware defence

Participant 2	
What computer programmes do you use at work daily?	Wireshark, Spotify, Internet explorer, Notepad, Access, Excel and Word
Who manages the IT side of your business?	Nobody is the dedicated member of the “It Team” but I normally get asked to do things because I'm good with computers
Do you currently use any free/open source software at work? If so what software do you use?	At work not really, due to the fact we’re a family run bakery
Do you currently use any free/open source software at home? If so what software do you use?	At home I tend to use more due to the high price of things like word, so I use open office
Would you be interested in a programme that teaches you how to use free/open source software?	Yes, very much so, being able to give that to my family would take a lot of burden off me
In what areas of work do you find yourself having issues with software?	I myself don’t have many issues but my family often have issues that I have to explain 3 or 4 times
How many employees does your business have that work with computers?	We have 5 employees including myself that use computers
Are you currently aware of any programme or scheme that encourages people to use free/open source software? If so which ones are you aware of?	A college night class I attended tried to get people to be aware of them but apart from that no
What areas of IT do you have little/no knowledge of that you may need to know about in the future?	I’d say I know very little about what to do in the event of a server dying or losing our backups

Participant 3	
What computer programmes do you use at work daily?	Angry IP Scanner, Dameware, Call Agent, Notepad, Terminal, Google chrome
Who manages the IT side of your business?	Me
Do you currently use any free/open source software at work? If so what software do you use?	Apart from those mentioned I use notepad ++ when it's installed on a machine and Wireshark
Do you currently use any free/open source software at home? If so what software do you use?	Not really because all the licences from my work are on my laptop so I just use those. I use Firefox, but I tend not to spend too much time on the pc at home
Would you be interested in a programme that teaches you how to use free/open source software?	Yes
In what areas of work do you find yourself having issues with software?	Trying to get support from some companies is an absolute nightmare and it doesn't look very professional having to go on YouTube to figure out bug fixes
How many employees does your business have that work with computers?	We all use computers but I'm the only person that uses anything other than Microsoft office
Are you currently aware of any programme or scheme that encourages people to use free/open source software? If so which ones are you aware of?	Not really, I normally have to just google things and find reviews
What areas of IT do you have little/no knowledge of that you may need to know about in the future?	The main area I would say is network protection. Other than installing anti-virus and firewalls I know very little

Participant 4	
What computer programmes do you use at work daily?	Microsoft Office, Internet Explorer, GIMP, Windows Remote access
Who manages the IT side of your business?	Nobody is necessarily "In charge" of IT but three people do all the general IT stuff
Do you currently use any free/open source software at work? If so what software do you use?	We use GIMP at work but that's it
Do you currently use any free/open source software at home? If so what software do you use?	I use GIMP at home too due to the high price of photoshop
Would you be interested in a programme that teaches you how to use free/open source software?	Very much so, my boss is always complaining when we have to ask for more money for software
In what areas of work do you find yourself having issues with software?	I often find I must use two programmes to do something where I feel one programme could be designed to do both tasks
How many employees does your business have that work with computers?	10 people work at the company and all but the driver tend to use computers every day
Are you currently aware of any programme or scheme that encourages people to use free/open source software? If so which ones are you aware of?	No and I would be using it if there was such a thing
What areas of IT do you have little/no knowledge of that you may need to know about in the future?	I would say the biggest thing that might damage the company is if we were to have laptops break due to them containing future orders as we have no separate server available

Participant 5	
What computer programmes do you use at work daily?	SQL-Ledger, MySQL, Samba, Apache web server, Microsoft office, Wireshark
Who manages the IT side of your business?	Solely me
Do you currently use any free/open source software at work? If so what software do you use?	Apart from those mentioned I also use programmes such as PostGrey, Evolution and GIMP when needed
Do you currently use any free/open source software at home? If so what software do you use?	The main ones I use at home are GIMP, Open office and Firefox. I also use ubuntu OS on one of my laptops
Would you be interested in a programme that teaches you how to use free/open source software?	IF it covers specialist areas yes as these often do not have clear instructions
In what areas of work do you find yourself having issues with software?	The main issues we have is cost of licencing. Getting extra software normally involves a small battle to get the funding
How many employees does your business have that work with computers?	All 9 of us use computers to various extents
Are you currently aware of any programme or scheme that encourages people to use free/open source software? If so which ones are you aware of?	Not really apart from people telling me about good software they got for free
What areas of IT do you have little/no knowledge of that you may need to know about in the future?	The biggest things are emergency protocols such as if someone were to attack our network or if we were to lose data to either employee negligence or a freak accident

10.2 Closing interviews

Participant 1

From when we last spoke concerning free and open source software, have you done any research into its uses? If so did you learn anything that you have then used in the workplace?

I've looked into more software that might be able to solve possible issues rather than having to learn when they come up. Just stuff like RAID options for future use.

Have you started using any additional free and open source software since we last spoke at the start of the project?

Since we last spoke I haven't used much else other than what I already have done.

Have you found a situation at work or home where you have used either of the taught programs (Recuva or MiniTool Partition Wizard) yet?

Yes, my little brother was complaining that he accidently deleted some things and Recuva helped restore them.

Would you recommend the use of the created programme to your friends, colleagues and employers if it were to be developed further?

If it were to be a more finished product I could see myself telling people about it. It would be nice for them to be able to learn for themselves rather than relying on me.

Did you look at the created programmes source code? If so, would you feel confident in editing it and contributing towards its development?

I did have a look. I felt like it would be pretty simple to expand the programme to add more to it as it's laid out and commented pretty well.

How did your employers feel about you being able to learn extra knowledge at no cost to them?

They loved it. I think when an employer hears you're putting extra time into learning things to do with work they're always going to be pleased.

Do you look for reviews on the security of software before using it? If so is that for all software or just certain ones?

Honestly no. I normally download the software either from the manufacturer or from trusted websites I've used before, so I've never really done much of a thorough security check.

Can I confirm, you tested the programme designed by me? Either at home or at work?

Yes I did.

Do you have any general comments on using the programme? E.g. How was it to use and was it helpful?

Apart from having to download the requests module it was very simple and easy to use. The fact it downloads the installer for you is a definite bonus.

Participant 2

From when we last spoke concerning free and open source software, have you done any research into its uses? If so did you learn anything that you have then used in the workplace?

Yes I've been looking into alternative database tools for our inventory management as the one we currently use is a bit restrictive.

Have you started using any additional free and open source software since we last spoke at the start of the project?

I've been attempting to use MySQL so that we can streamline our inventory management as mentioned before

Have you found a situation at work or home where you have used either of the taught programs (Recuva or MiniTool Partition Wizard) yet?

Not yet but in the event I do need to use them I feel like I now can. The worry of losing all our information is definitely less of an issue now I can use the two programs.

Would you recommend the use of the created programme to your friends, colleagues and employers if it were to be developed further?

If it were to cover a larger range of programs, then yes certainly. Being able to learn something like a new program by yourself would be very useful

Did you look at the created programmes source code? If so, would you feel confident in editing it and contributing towards its development?

I did have a look however due to me not being overly technical I didn't feel too confident in meddling with it. I would like to learn however as it seems very interesting

How did your employers feel about you being able to learn extra knowledge at no cost to them?

I told my mum (the boss) about it and she thought it was really good, just as long as it doesn't interfere with my day to day work she encouraged me to learn as much as I could

Do you look for reviews on the security of software before using it? If so is that for all software or just certain ones?

We get everything from places like Microsoft so not really because if something was wrong they would fix it

Can I confirm, you tested the programme designed by me? Either at home or at work?

Yes at home

Do you have any general comments on using the programme? E.g. How was it to use and was it helpful?

I feel like the program was simple to use and if expanded upon, would make an amazing teaching tool for people who are bad at IT. It seems like something Which magazine would do but then again, I doubt they ever will.

Participant 3

From when we last spoke concerning free and open source software, have you done any research into its uses? If so did you learn anything that you have then used in the workplace?

I've investigated into how it became a thing as it seems odd that free and open source programs used to be the norm and now you have to pay for literally everything. I had a look at one of our servers and deemed it more appropriate to be running Linux rather than windows, so I plan to change that

Have you started using any additional free and open source software since we last spoke at the start of the project?

I've been getting the hang of more Linux distributions such as ubuntu so that I can do the work

Have you found a situation at work or home where you have used either of the taught programs (Recuva or MiniTool Partition Wizard) yet?

I used MiniTool to transfer my home data to a new hard drive which is double the size

Would you recommend the use of the created programme to your friends, colleagues and employers if it were to be developed further?

Yes definitely, I've already given it to a few friends to see if they like it as it seemed something they would be interested in

Did you look at the created programmes source code? If so, would you feel confident in editing it and contributing towards its development?

Yes I had a look. Although I've never used python before it seemed to be quite simple to grasp so I might learn in the future and test what I learn with your created program

How did your employers feel about you being able to learn extra knowledge at no cost to them?

I haven't actually told them yet. Since I run all the IT services it hasn't come up yet

Do you look for reviews on the security of software before using it? If so is that for all software or just certain ones?

I do yes. Before I implement any software into our work system I always vet it for known security flaws as we regularly work with sensitive data.

Can I confirm, you tested the programme designed by me? Either at home or at work?

Yes I did, I used it at home. Coincidentally a few days after I used it I needed to use Recuva to restore some deleted files at work

Do you have any general comments on using the programme? E.g. How was it to use and was it helpful?

It was very useful but a bit lacking in content but that's understandable given your time frame

Participant 4

From when we last spoke concerning free and open source software, have you done any research into its uses? If so did you learn anything that you have then used in the workplace?

Because of where I work, we only need to use GIMP and Microsoft office for most things so any extra research I do is normally when a problem arises

Have you started using any additional free and open source software since we last spoke at the start of the project?

No

Have you found a situation at work or home where you have used either of the taught programs (Recuva or MiniTool Partition Wizard) yet?

Not yet but my boss is hoping to expand our NAS so using MiniTool might be useful in the future

Would you recommend the use of the created programme to your friends, colleagues and employers if it were to be developed further?

If they had a problem with IT then yes but it wouldn't really come up in conversation otherwise

Did you look at the created programmes source code? If so, would you feel confident in editing it and contributing towards its development?

I didn't. I don't really do much programming at work or at home so I thought it might be a bit difficult for me to handle

How did your employers feel about you being able to learn extra knowledge at no cost to them?

They were really happy. As I've said before, it's always a battle getting money from my boss for additional software so the thought of me becoming a better employee that can offer the business more, without any additional cost to them was a welcome idea

Do you look for reviews on the security of software before using it? If so is that for all software or just certain ones?

Not really

Can I confirm, you tested the programme designed by me? Either at home or at work?

Yes I did

Do you have any general comments on using the programme? E.g. How was it to use and was it helpful?

Installing the module you told us to install was a bit tricky as I have never used command line before, but I did find help on YouTube which guided me through it

Participant 5

From when we last spoke concerning free and open source software, have you done any research into its uses? If so did you learn anything that you have then used in the workplace?

I already know a fair bit about FOSS so I didn't feel it necessary to do any more research into it

Have you started using any additional free and open source software since we last spoke at the start of the project?

Again, not really because I use a lot of open source tools at work and home

Have you found a situation at work or home where you have used either of the taught programs (Recuva or MiniTool Partition Wizard) yet?

Not yet but they're both more emergency tools so I might use them in the future

Would you recommend the use of the created programme to your friends, colleagues and employers if it were to be developed further?

I told my work colleagues about the program and that I was testing it and they seemed interested to see if it was something they could find useful

Did you look at the created programmes source code? If so, would you feel confident in editing it and contributing towards its development?

I did. I would feel confident because it's structured nicely and you can see how the program flows from the start to finish

How did your employers feel about you being able to learn extra knowledge at no cost to them?

I honestly haven't told them but I'm sure they will be happy when I can instantly solve an issue rather than having to look how to fix it

Do you look for reviews on the security of software before using it? If so is that for all software or just certain ones?

Due to us using a lot of well known open source tools it's more about where we download it from rather than what we're downloading so because of this I don't really look to see how secure each programme is

Can I confirm, you tested the programme designed by me? Either at home or at work?

Yes and i tested Recuva at home and work and MiniTool at home because of the data loss warning

Do you have any general comments on using the programme? E.g. How was it to use and was it helpful?

I think what you made is a really cool idea but it does need some more work before it could be released to the public. The command line interface is easy to understand but other users who may not be as used to it may find it a bit odd.

10.3 GitHub link to the created program

The program that has been created by the author has been posted to GitHub so that others who wish to contribute to its development can do easily. The program can be found at the following link:

https://github.com/mokeybacon/Dissertation-Program

10.4 Program running with Recuva chosen

```
Please enter the password:Helloworld
Hello, what do you need help with today?
Here is the list of available options:
1:Hard drive recovery
2:Hard drive copying
3:RAID
4:Hard Drive Formatting/Wiping
5:ISO Creation
6:Encryption
7:Word processing
8:Programming
9:Resource monitor
10:Browsers
11:Torrenting
12:Photo Editing
13:Compression
14:Remote Desktop
15:Virtual Machines
Please enter the number of the operation you need help with:1
You have chosen option:  1:Hard drive recovery
Is that correct? y
Here is a list of the possible tools for 1:Hard drive recovery :
1:Recuva
2:Stellar Data Recovery
3:Undelete 360
Which programme would you like to use?1
Downloading recuva.exe
[================================================]You have selected option 1:Recuva
Here are the instructions for Recuva
1. Install the program. The installer can be found in the same folder where you ran this programme from.
2. Choose the data type of the file you want to restore (Picture, Video, Document etc).
3. Choose where the file was located, there is an option if you do not know.
4. Start the scan. If the scan fails the first time then select "Deep Scan".
5. Look through the files Recuva has found.
6. If you find the correct file, select it and choose recover.
7. Select where you would like the file to be recovered to.
8. Check the destination location to make sure the file has recovered.
9. If you need to find another file then choose advanced settings rathan than begin the process again.
Here are the options once you are done:
1: Return to main menu
2: Return to choosing a programme
3: Close the program
Please choose what you would like to do:
```

10.5 Program running with MiniTool partition wizard chosen

```
Please enter the password:Helloworld
Hello, what do you need help with today?
Here is the list of available options:
1:Hard drive recovery
2:Hard drive copying
3:RAID
4:Hard Drive Formatting/Wiping
5:ISO Creation
6:Encryption
7:Word processing
8:Programming
9:Resource monitor
10:Browsers
11:Torrenting
12:Photo Editing
13:Compression
14:Remote Desktop
15:Virtual Machines
Please enter the number of the operation you need help with:2
You have chosen option:  2:Hard drive copying
Is that correct? y
Here is a list of the possible tools for 2:Hard drive copying :
1:Ease US Backup Home
2:Minitool Partition Wizard
3:Clonezillar
Which programme would you like to use?2
Downloading minitool.exe
[==================================================]You have selected option 2:Minitool Partition Wizard
Here are the instructions for Minitool Partition Wizard
1. Install the program. The installer can be found in the same folder where you ran this programme from
2. Make sure all important data is backed up due to the original data being deleted during the transfer
3. Select the drive or partition you wish to transfer
4. Right click the drive or partition and choose copy
5. Select the destination drive or partition
6. Allocate partitions to the new drive if needed
7. Read and understand the BIOS information, for more information visit http://www.boot-disk.com/boot_priority.htm
8. Click apply
Here are the options once you are done:
1: Return to main menu
2: Return to choosing a programme
3: Close the program
Please choose what you would like to do:
```