THE UNIVERSITY OF DODOMA



**COLLEGE OF INFORMATICS AND VIRTUAL EDUCATION**

**CS 321: GROUP ASSIGNMENT 1**

**INSTRUCTOR: Mr. MINJA, GODBLESS G.**

**GROUP MEMBERS:**

|  |  |  |
| --- | --- | --- |
| S/N | NAME OF STUDENTS | REGISTRATION NUMBER |
| 1 | RAPHAELA MICHAEL | T/UDOM/2017/02774 |
| 2 | RENATUS SAMWEL | T/UDOM/2017/02775 |
| 3 | NICETAS SHAYO | T/UDOM/2017/02771 |
| 4 | THOMAS STEPHANO | T/UDOM/2017/02778 |
| 5 | GODFREY NICKODEM | T/UDOM/2017/02762 |
| 6 | SIGIFRIDY MUNUO | T/UDOM/2017/02777 |

**ANY FIVE REASONS FOR WHICH A BUSINESS WOULD OPT TO USE OSS.**

* Security

The security advantages on offer don’t necessarily mean that OSS is more secure than proprietary software, it just means that bugs and defects can be identified and fixed quicker. As the source code for open-source software is freely available, anyone can inspect the code for faults.

## Freedom and Flexibility

## The ever-changing business environment means that certain software solutions might not fill the desired void, or the need for them may change overtime. Businesses often have to trial a number of different solutions in order to find the one that matches their requirements. OSS doesn’t lock users in like proprietary software, therefore providing them with the freedom needed to find the right solution. Freedom from vendor lock-in also gives businesses more control and ownership over their software, allowing them to customize and apply it however they see fit.

* Customizability

While proprietary software often comes packed with features, there are always certain features or functionalities missing that cannot be added by anyone besides the company’s developers. The ability to customize open-source software not only allows businesses to tailor the software to their needs and requirements, it can also provide competitive advantages.

* Quality

In the same way developers fix bugs and defects to make OSS more secure, they also contribute by adding features to improve the user experience and the software itself. On any given open-source project, there can be hundreds or thousands of developers all adding new features or enhancing existing ones to improve the overall quality of the software.

* Cost

The financial benefits offered by open-source software have been one of the main driving factors behind the switch from proprietary (closed) software to OSS. While the word “free” in the popular term “free software” refers to freedom (libre) rather than free of charge,most open-source software doesn’t cost a dime to purchase or implement, and although this is often the first attraction for businesses, it’s the low total cost of ownership (TCO) that really seals the deal.

**DESCRIBE ANY FIVE DISADVANTAGES OF USING OSS**

* The difficulty of use

Some open source applications may be tricky to use. Others may lack user-friendly interface or features that your staff from adopting or using programs with ease.

* Compatibility issues

Many types of proprietary hardware need specialized drivers to learn open source programs which are often only available from the equipment manufacturer. Even if an open source driver exist it may not work with your software as well as proprietary driver.

* Liabilities and warranties

With proprietary software, the developer usually provides indemnification and warranty as a part of a standard license agreement.

* Hidden Cost

Software that is free up-front but a laser costs money to run can be a major burden, especially if you haven’t considered hidden costs from the outset.

* Vulnerable to malicious users.  
  Many people have access to the source code of open source software, but not all of them have good intentions. While a lot of people utilize their access to spot defects and make improvements to the program, others use this privilege to exploit the product’s vulnerabilities and create bugs that can infect hardware, steal identities or just annoy other users.

**WITH EXAMPLE DESCRIBE SHAREWARE AND FREEWARE**

Freeware is [copyrighted](http://en.wikipedia.org/wiki/Copyright) computer [software](https://www.diffen.com/difference/Hardware_vs_Software) which is made available for use free of charge, for an unlimited time. Authors of freeware often want to "give something to the community", but also want to retain control of any future development of the software.

Common examples include Internet browsers, such as Mozilla Firefox and Google Chrome, the voice-over-IP service Skype, and the PDF file reader Adobe Acrobat.

Shareware refers to commercial software that is copyrighted, but which may be copied for others for the purpose of their trying it out with the understanding that they will pay for it if they continue to use it.

Common examples include WinRAR, Any DVD, Adobe and Microsoft programs, some antivirus software

**PROVIDE A SHORT SUMMARY ABOUT THE HISTORY /EVOLUTION OF OSS**

In the 1950s and early 1960s the early software was free by definition, due to the academic nature of software development, as well as compatibility and porting requirements and the lack of a separate software business model; revenue was generated with hardware, primarily. It was also a concern that closed source software would allow for backdoors used for clandestine purposes, as it was virtually impossible to see what a software was doing while executed.

In the late 1960s the software industry was becoming a real business. Software was getting increasingly decoupled from the hardware business, requiring a separate business model. Lawyers drafted restrictive licenses to enable this. ARPANET researchers used RfCs to develop telecom protocols. This collaboration eventually led to the birth of the Internet in 1969.

In the 1970s AT&T released early versions of UNIX. The software was free of charge, but users were not allowed to redistribute or modify it. In the late 1970s and early 1980s charging for software licenses became a dominant business model for software companies and computer vendors. Legal restrictions were imposed through copyrights, trademarks and other contracts. License enforcement via legal actions began. Software piracy was born.

In the 1980s software was shared via BBS systems. Software written in BASIC and other interpreted languages could only be distributed as source code. A lot of freeware became available. Software modding became popular and Usenet provided a good collaboration channel for programmers/Modder’s. Richard Stallman started the GNU Project and founded the Free Software Foundation. The first companies making free software as their primary business emerged.

In the early 1990s the free software community received the first complete free operating system with Linus Torvalds’s kernel combined to GNU operating system. Debian, founded by Ian Murdock in 1993, committed to the GNU and FSF principles of free software. Linux adoption by businesses and governments began in the late 1990s. Website-based companies emerged and made extensive use of free web servers, especially the Apache HTTP Server. The LAMP (Linux, Apache, MySQL, PHP) stack gained popularity over expensive proprietary solutions.

Freeware Summit organized by Tim O’Reilly brought together the leaders of free and open source projects. A developer vote decided on Open Source as a new term over Source ware. The Open Source Initiative was formed much to the disdain of Richard Stallman and the FSF, who felt that OSI was selling out on some core values. FSF and OSI remain the main schools of the movement today and so remains their philosophical discord. Fortunately, they do agree on many practical matters and are able to work together for the common cause.

In the early 2000s big software corporations began to see free software as a threat to their core business. Microsoft’s Steve Ballmer called Linux a cancer, referring to its copyleft license.

In 2003 SCO made claims that Unix IPR had been copied into Linux kernel and decided to jump straight into the deep end of the pool by suing IBM. They did not lack balls, but, as it turned out, they did mostly lack the ownership of the IPR they claimed had been violated. The case is still technically ongoing, even though SCO filed for bankruptcy in 2007, after multiple defeats in court. SCO allegedly received funding from Microsoft.

In the recent years the free software movement has seen some worrying corporate acquisitions, such as Sun Microsystems purchasing MySQL and Oracle purchasing Sun Microsystems.

**LIST AND DESCRIBE THE FACTORS THAT HAVE LED TO THE GROWTH OF OSS**

* Market Uptake  
  Even if the OSS is not mature, the market response can be a game changer in how quickly it shapes up. High adoption in a short timeframe indicates the flexibility of the software.
* Fault Recovery  
  A primary concerns for OSS is reliability, the fail-safe mechanisms put in place to avoid failures. It is important to look at the OSS features, but it is more important to evaluate the fault recovery systems that are available or can be introduced.
* Interoperability  
  Being interoperable with other well-known software is essential to leverage the flexibility of using OSS. Though it gets taken care of by the uptake in most cases, interoperability is important, as developers do not need to relearn an entirely new architecture.
* Talent AvailabilityEmploying skilled people will ensure that the software is optimal, efficient, and relatively defect-free. Assessment of software-skill match at an organizational level is required to make sure everybody does not have to quit or learn new software.
* Support Forums  
  Discussion forums are a great way to stay abreast of the latest applications. We can also repurpose some of them for our requirements. A larger discussion forum improves the probability of resolving any corner case errors efficiently.
* Hidden Cost of Ownership  
  OSS is priced way below the premium software, though it carries the additional costs of maintenance, customization, upgrades, and enhancements. Without in-house resources or an expert technology partner, “free” can turn out to be expensive.
* Commercial Alternatives  
  Desperate measures taken to market premium software are often an indication of the availability of good OSS alternatives. Availability of commercial variants also gives you a good matrix for comparison and choosing the best software to suit your needs.
* Security  
  There is an impression that OSS is vulnerable to security attacks. A good OSS system will typically have better security than a premium one*.*

**DESCRIBE THE OPEN SOURCE INITIATIVE (OSI)**

Is a non-profit corporation whose goal is to promote the use of open source software in the commercial world.

**DESCRIBE ANY THREE POPULAR OSI-APPROVED LICENSES**

* Reciprocal licenses

contain a provision that requires on relicensing the code must be  
open source. If the distributor receives the source code, then it passes it on to others.  
Example: Linux uses the GPL.

* Non reciprocal licenses

do not contain a relicensing provision, so they allow derivative  
works from open source code to revert to closed. This is nonreciprocal in the sense that a distributor can receive source code but may not necessarily pass it on.

Example, Apple uses FreeBSD code as part of Mac OS X without needing to distribute the Mac OS X source code.

* Dual licenses.

Some products are dual licensed. They are available with either an open source license or a commercial license.

The dual license allows these companies to offer open source products to those who are developing open source software, or individual end users. Others may be required to pay for a commercial license.  
Examples of such products are:  
– Qt, from TrollTech, the GUI toolkit used by KDE  
– MySQL, from MySQL AB, the database server  
– Berkeley DB, from SleepyCat Software, the embedded database program

**DESCRIBE HOW HELPFUL CAN THE OPEN SOURCE INITIATIVE BE TO A B**U**SINESS:**

* The OSI helps businesses understand, adopt, contribute to and benefit from open source software, development practices and communities.
* The open source model offers lower total cost of ownership to the business world, both as consumers and contributors.
* The open source model enables increased reliability; Participating in open source projects and communities is a way to build open standards as actual software, rather than paper documents. It's a way for companies and individuals to collaborate around shared needs on a product that none of them could achieve alone or, in and of itself, does not constitute a key business differentiator. It's a better development model where bug-fixes and new features identified by your customers are done more quickly and with higher quality, yielding increased reliability.
* The open source model also enables increased security; because code is in the public view it will be exposed to extreme scrutiny, with problems being found and fixed instead of being kept secret until the wrong person discovers them. And last but not least, it's a way that the little guys can get together, innovate and have a good chance at beating a monopoly
* In general, these are the advantages of open source software for business
* lower software licensing costs, no supplier lock-in, freedom to do what you want with the software, open standards that support collaborative development and freedom to upgrade software as it suits your business

**DESCRIBE FREE SOFTWARE**

Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software.

**IDENTIFY THE FOUR ESSENTIAL FREEDOM THAT USERS HAVE AS FAR AS FREE SOFTWARE IS CONCERNED**.

Freedom 0: The freedom to run the program, for any purpose

Freedom 1: The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition to this

Freedom 2: The freedom to redistribute copies so you can help your neighbor.

Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general to the public, so that the whole community benefits. Access to the source code is a precondition for this

**DESCRIBE GNU**

* Is a mass collaborative initiative for the development of free software.
* Founder Richard Stallman was a programmer at MIT AI Lab.
* The original purpose of GNU project was the creation of a free operating system. Free, in a software context does not necessarily mean free of cost.
* The freedom referred to is the ability of anyone who wishes to run, copy, distribute, study, change and improve the software.

**DESCRIBE GNOME**

* Is a free and open-source desktop environment for Unix-like operating systems
* GNOME is developed by the GNOME project which is composed of both volunteers and paid contributors, the largest corporate contributor being Red Hat.
* GNOME: provide a consistent set a libraries and applications for an easy to use and friendly desktop environment.
* GNOME allows the user to select one of the several desktop appearances and address usability by non-technical people.

**DESCRIBE THE RELATIONSHIP BETWEEN GNU PROJECT AND GNOME**

* GNOME is the official desktop environment of the GNU Project and is therefore an official GNU Project and a part of GNU operating system.
* Historically, it was created by GNU in response to KDE’s dependence on Qt. In fact, the G in GNOME stands for GNU.

**DESCRIBE THE RELATIONSHIP BETWEEN FREE SOFTWARE AND FREE SOFTWARE FOUNDATION**.

The Free Software Foundation (FSF) is a [non-profit organization](https://en.wikipedia.org/wiki/Non-profit_organization) founded by [Richard Stallman](https://en.wikipedia.org/wiki/Richard_Stallman) on October 4, 1985, to support the [free software movement](https://en.wikipedia.org/wiki/Free_software_movement), which promotes the universal freedom to study, distribute, create, and modify [computer software](https://en.wikipedia.org/wiki/Computer_software).

From its founding until the mid-1990s, FSF's funds were mostly used to employ software developers to write [free software](https://en.wikipedia.org/wiki/Free_software) for the [GNU Project](https://en.wikipedia.org/wiki/GNU_Project). Since the mid-1990s, the FSF's employees and volunteers have mostly worked on legal and structural issues for the free software movement and the free software community.

FSF aims to use only [free software](https://en.wikipedia.org/wiki/Free_software) on its own computers

**DESCRIBE HOW HELPFUL CAN FREE SOFTWARE FOUNDATION AND THE FREE SOFTWARE FOUNDATION BE TO;**

* **Normal user**
* Available at minimal cost

Free software does not mean zero cost software. Free software lowers the production cost. Making a system like MS-Windows costs millions of dollars. But if you produce the same kind of system using a free POSIX version, it would probably cost less than a hundred dollars.

* Provides full freedom

Originally, computer manufacturers aimed only for hardware innovation and didn't consider software as a business asset. This is because most computer users were scientists and technicians who could modify the software themselves, so hardware was distributed with the software pre-installed. Later, high level programming languages were introduced which were compatible for almost every kind of computer. That meant even less efficient hardware design could be made to work better.

* No imposed upgrades

Free software never disappears like proprietary. If proprietary vendors stop supporting a product, users have two options: either use an unsupported version of the software, or go for an (unwanted) upgrade. Imposed upgrading never happens for free software.

* No spying on users

If users have no control over the software they are using, it can easily spy on their activity. The company behind proprietary software often installs features that restrict users from sharing it with others. Since anyone who buys proprietary software must sign a licensing agreement before using it, they are agreeing that the vendor has the right to inspect hard drive content without warning. This violates our privacy because our computers hold our personal information and daily activities.

* Auditability

When closed-source software manufacturers claim that they made improvements in the program, improved security and restricted backdoors, users are forced to believe it. Since the source code is not provided, there is no evidence for such claims.

* Provides better security

It is a commonly known fact that proprietary software threatens users' security. There is a long history of security vulnerabilities. Proprietary software doesn't necessarily stop the spreading of viruses and letting hackers to take over people's computers for sending spam. Because the software is secret, all users are dependent on the corporation to fix these kinds of problems

* No monopolies

It is easier and cheaper to switch from one free software to another compared to switching proprietary software. Free software does not tie you down to any corporation

* Truly user-oriented

One of proprietary software's major claims is that free software is not user-oriented. A proprietary vendor used to listen to its customers' needs and respond and develop accordingly. But companies like Red Hat and IBM are crafting strategies that rely on all end users having the same needs.

* No lock-in standards

Proprietary vendors lock-in proprietary standards to ensure that their users will become returning customers. Free software works in open standards

* Part of social movement

Free software is not just for the individual user's sake. It promotes social solidarity and represents society as a whole through sharing and cooperation. Since our activities are progressively digitized, free software is becoming an even more essential part of our culture and life activities

* **Business.**
* flexibility and agility

IT leaders must fundamentally provide flexibility and [agility](https://enterprisersproject.com/taxonomy/term/351) for their business. If you can’t compete on agility, you’re going to get left behind by the competition. Open source enables technology agility, typically offering multiple ways to solve problems. Open source helps keep your IT organization from getting blocked because a particular capability isn’t available from a vendor. Instead of waiting for the vendor to deliver that capability, you can create it yourself

* speed

Your business will soon be competing on speed, if it isn’t already. [Open source enables speed](https://opensource.com/article/17/8/enterprise-open-source-advantages?intcmp=701f2000000tjyaAAA). A great advantage of open source is the ability to take the community versions, get started, understand whether they can solve your business problem, and begin to deliver value right away. Once you make that determination, professional support and services are increasingly available for open source products, especially those supported by Red Hat.

* cost-effectiveness

Open source is generally much more cost-effective than a proprietary solution. Not only are open source solutions typically much more inexpensive in a business environment for equivalent or superior capability, but they also give businesses the ability to start small and scale (more on that coming up). Given that businesses are often budget challenged, it just makes financial sense to explore open source solutions.

* ability to start small

With open source, you can start small and quickly with community versions, and then migrate to a commercially-supported solution as your business requirements drive you there. If the project doesn’t require support, you can continue on the community version indefinitely. You have the option to try the various alternatives, pick the one that’s going to work, and then [scale up](https://enterprisersproject.com/node/7361) with a commercial solution

* solid information security

Commercial open source has a solid information [security](https://enterprisersproject.com/taxonomy/term/166) record in a dangerous world. Obviously, it’s difficult to claim security superiority for any solution and it’s a challenging environment for all of us, but the responsiveness of the open source community and vendors relative to information security problems has been very good.

* attract better talent

Open source gives business the ability to attract better [talent](https://enterprisersproject.com/taxonomy/term/21). Most professional technologists are well aware of open source and many believe it’s where the industry is headed. Many enjoy creating their own projects and having the ability to interact with others outside their business to develop solutions. Giving developers flexibility and freedom can be an important tool in [attracting better talent](https://enterprisersproject.com/node/8466)

* share maintenance costs

You can solve your business problems while effectively sharing some of the maintenance costs. One of the fundamental advantages of open source is community involvement. Rather than writing an application and having to sustain it yourself, you can share the cost of maintaining and sustaining applications among multiple parties