

MP 01B0000048

MITRE PRODUCT

A Business Case Study of Open Source Software

http://www-1.ibm.com/linux/viewaudience_42_nodeId~5265:typ

July 2001

Carolyn A. Kenwood

Sponsor: US Army
Dept. No.: W803

Contract No.: DAAB07-01-C-C201
Project No.: 0700M520-AA

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

Approved for public release; distribution unlimited.

©2001 The MITRE Corporation. All Rights Reserved.

MITRE
Washington C3 Center
Bedford, Massachusetts

MITRE Department Approval:

Francis M. Dello Russo, W803
Department Head

MITRE Project Approval:

Paul R. Garvey, W800
Chief Scientist

Abstract

This paper was prepared as part of The MITRE Corporation's FY00 Mission-Oriented Investigation and Experimentation (MOIE) research project "Open Source Software in Military Systems." This paper analyzes the business case of open source software. It is intended to help Program Managers evaluate whether open source software and development methodologies are applicable to their technology programs. In the Executive Summary, the paper explains open source, describes its significance, compares open source to traditional commercial off-the-shelf (COTS) products, presents the military business case, shows the applicability of Linux to the military business case, analyzes the use of Linux, discusses anomalies, and provides considerations for military Program Managers. The paper also provides a history of Unix and Linux, presents a business case model, and analyzes the commercial business case of Linux.

KEYWORDS: Linux, Open Source, Open Source Software, OSS, Software

Acknowledgments

The author would like to acknowledge and thank the other individuals who contributed to this document. In particular, thanks to Terry Bollinger, Francis Dello Russo, David Emery, Paul Garvey, Robert Giallombardo, Michael Macpherson, Frank McPherson, James Moore, and Audrey Taub for their insights and helpful reviews. I would like to thank Rosemarie Mauriello for her assistance in producing this document. Also, thanks to Janice Ballo and Theresa Dillon for the materials they contributed through their extensive search efforts.

Table of Contents

Section	Page
Executive Summary	xi
What is Open Source?	xi
Significance of Open Source	xiii
Benefits and Risks of Open Source Software Compared to Traditional COTS	xiii
The Military Business Case	xix
Applicability of Linux to the Military Business Case	xix
Use of Linux	xxi
Discussion	xxii
Considerations for Military Program Managers	xxiii
Federal Linux Award	xxiv
Conclusion	xxv
1. History of Unix and Linux	1
2. Business Case Analysis Model	3
3. Commercial Business Case Analysis of Linux	5
3.1 Environmental Scanning	5
3.1.1 Strengths	6
3.1.1.1 Massive Programming Expertise	6
3.1.1.2 R&D Covered by Volunteer Labor	6
3.1.1.3 Accepted Leadership Structure	6
3.1.1.4 Quick Release Rate	7
3.1.1.5 Parallel Development and Debugging	7
3.1.1.6 Maturity of Code	7
3.1.1.7 Culture of Sharing	8
3.1.1.8 Long Term Accessibility	8
3.1.2 Weaknesses	8
3.1.2.1 Lack of “Ownership”	8
3.1.2.2 Hard to Originate	9
3.1.2.3 Less User-Friendly	9
3.1.3 Opportunities	11
3.1.3.1 Internet Connectivity	11
3.1.3.2 Many Distributors	12
3.1.3.3 Competitive Support Structure	15
3.1.3.4 Influx of Start-up Companies	16

Section	Page
3.1.3.5 Garnering Support	16
3.1.4 Threats	17
3.1.4.1 Risk of Fragmentation	17
3.1.4.2 Lack of Compatible Applications	17
3.1.4.3 Need for Version Control	18
3.1.5 Other	18
3.1.5.1 Importance to Many	18
3.1.5.2 Trained Staff	18
3.1.5.3 Competition	18
3.2 Analysis of Strategic Factors	19
3.2.1 Market Viability	19
3.2.2 Market Segments	26
3.2.2.1 Servers Market	29
3.2.2.2 Desktop Market	34
3.2.2.3 Embedded Devices	36
3.3 Evaluating Feasibility of Business Opportunity	39
3.3.1 Direct Costs	43
3.3.1.1 Software and Hardware	43
3.3.1.1.1 Software	43
3.3.1.1.2 Hardware	43
3.3.1.2 Support	43
3.3.1.2.1 Internal Support	43
3.3.1.2.2 External Support	43
3.3.1.3 Staffing	44
3.3.1.3.1 Project Management	44
3.3.1.3.2 Systems Engineering/Development	44
3.3.1.3.3 Systems Administration	44
3.3.1.3.4 Other Administration	44
3.3.1.3.5 Training	44
3.3.1.4 De-installation and Disposal	45
3.3.2 Indirect Costs	45
3.3.2.1 Support Costs	45
3.3.2.1.1 Peer Support	45
3.3.2.1.2 Casual Learning	45
3.3.2.1.3 Formal Training	45
3.3.2.1.4 Application Development	46
3.3.2.1.5 Futz Factor	46
3.3.2.2 Downtime	46
3.3.3 Benefits and Risks	46
3.3.3.1 Ability to Customize	47

Section	Page
3.3.3.2 Availability/Reliability	47
3.3.3.3 Interoperability	48
3.3.3.4 Scalability	48
3.3.3.5 Design Flexibility	49
3.3.3.6 Lifetime	49
3.3.3.7 Performance	49
3.3.3.8 Quality of Service and Support	49
3.3.3.9 Security	49
3.3.3.10 Level of Difficulty/Ease of Management	50
3.3.3.11 Risk of Fragmentation	50
3.3.3.12 Availability of Applications	51
List of References	55
Glossary	59
Distribution List	61

List of Figures

Figure	Page
ES-1. OSS Provides Several Maintenance and Support Options	xviii
ES-2. Military and Commercial User Benefits of Linux	xx
ES-3. Worldwide Success of Linux in the Marketplace	xxi
ES-4. Server and Client OS Market Share in 1998 and 1999	xxii
1. Business Case Analysis Framework Applied to Open Source Products and Processes	4
2. Key Elements of SWOT Analysis	5
3. Snapshot of Linux Code	10
4. Example of Linux Screenshot	11
5. Motivations for Linux Interest	19
6. Satisfaction with Linux	20
7. Worldwide Success of Linux in the Marketplace	22
8. Percent of Companies that Use the Linux Operating System	23
9. Percent of Operating Systems that Are Or Will Be Linux	24
10. How Long Companies Have Been Using Linux	25
11. Worldwide New Linux Shipments (Client and Server)	26
12. US Linux Server Sites by Industry, 1999	27
13. Use of Linux (Datapro Survey)	28
14. Use of Linux in Server Based Applications (Information Week Survey)	29
15. Server OS Market Share in 1998	30

Figure	Page
16. Server OS Market Share in 1999	31
17. US Linux Server Shipments and Customer Spending, 1998-2003	32
18. US Server Workload Spending by Operating System, 1999	33
19. Client OS Market Share in 1998	35
20. Client OS Market Share in 1999	36
21. Most Significant Weaknesses of Linux	51
22. US Linux Users' Ratings for Server Quality by Operating System	52

List of Tables

Table	Page
ES-1. OSS Cost Element Taxonomy	xvi
ES-2. OSS Taxonomy of Benefits and Risks	xvii
1. Common Distributions of Linux by Vendor	12
2. Market-Specific and Niche Vendors of Linux	14
3. Free Support Resources for Linux by Vendor	16
4. Desktop Application Suites for Linux by Vendor and Product	34
5. Cost Element Taxonomy for OSS and Linux	42
6. OSS and Linux Taxonomy of Benefits and Risks	46
7. Comparison of Operating Systems	53

Executive Summary

What Is Open Source?

Open source, by definition, means that the source code is available. Open source software (OSS) is software with its source code available that may be used, copied, and distributed with or without modifications, and that may be offered either with or without a fee. If the end-user makes any alterations to the software, he can either choose to keep those changes private or return them to the community so that they can potentially be added to future releases¹. An open source license is certified by the Open Source Initiative (OSI), an unincorporated nonprofit research and educational association with the mission to own and defend the open source trademark and advance the cause of OSS. The open source community consists of individuals or groups of individuals who contribute to a particular open source product or technology. The open source process refers to the approach for developing and maintaining open source products and technologies, including software, computers, devices, technical formats, and computer languages.

Although OSS has recently become a hot topic in the press, it has actually been in existence since the 1960s and has shown a successful track record to-date. Examples of popular open source products include Emacs, GNU toolset, Apache, Sendmail, and Linux. The development of Perl is an example of the open source process.

Emacs was one of the first open source products. It is a text editor that is widely used for software development. As a software tool, many developers (including defense contractors) use Emacs to develop their (non-open source) applications.² The success of Emacs led to the GNU program. GNU stands for “Gnu’s not Unix.” The GNU project consists of an operating system kernel and associated Unix tools. The GNU tools have been ported to a wide variety of platforms, including Windows NT. Again, they are widely used by software developers to produce both open source and proprietary software.³

The Apache web server is a freely available web server distributed under an open source license. Apache developers form a voting committee, and votes from this committee set the direction for the project. The Apache Software Foundation provides organizational, legal,

¹ There are several licensing models for Open Source. Some require that all changes made to the source must be freely distributed with the modified product. Other licenses permit an organization to make changes and keep the changes private.

² For more information on Emacs, see <http://www.gnu.org/software/emacs/emacs.html>.

³ For further information on GNU, visit the GNU Project web server at <http://www.gnu.org/>.

and financial support for Apache projects. Apache web servers are known for their functionality and reliability. They form the backbone infrastructure running the Internet. Today, Apache comprises over 60 percent of the web server market and continues to grow.⁴

Sendmail is a platform for moving mail from one machine to another. The Sendmail Consortium, a nonprofit organization, runs the open source program and maintains a website to serve as a resource. Sendmail is estimated to carry nearly 90 percent of e-mail traffic.⁵

Linux is an open source Unix-like operating system (OS). The kernel is maintained by the Linux community, led by Linus Torvalds, the creator of Linux.⁶ Torvalds has appointed delegates who are responsible for managing certain areas of the project and, in turn, these delegates have a team of coordinators. Linux has multiple uses; it can be used as an OS for a server, desktop, or embedded environment. There are over ten million Linux users worldwide. According to an InformationWeek survey, Linux comprises about 4 percent of all operating systems, and that number is expected to rise to 15 percent in two years.⁷ Linux is the fastest growing server operating environment, increasing from 16 percent of the market in 1998 to 25 percent in 1999.⁸ In the embedded market, Linux is also expected to play a significant role.⁹ (An embedded device is a piece of microprocessor-based computing hardware, usually on single circuit board, which has been built to run a specific software application. The term *embedded* refers to the fact that these devices were originally used as building blocks in larger systems.)

While Emacs, GNU toolset, Apache, Sendmail, and Linux are examples of open source products, the Practical Extraction and Reporting Language (Perl) is an example of an open source process. Perl is a system administration and computer-programming language widely used throughout the Internet. It is the standard scripting language for all Apache web servers, and is commonly used on Unix. Perl is managed on a rotating basis by the ten to

⁴ O'Reilly, Tim, Linux eSeminar Series, 1999. For more information on Apache, refer to the Apache Software Foundation at <http://www.apache.org/>.

⁵ O'Reilly, Tim, and Ether Dyson, "Open Mind, Open Source." For more information on Sendmail, see <http://www.sendmail.org/>.

⁶ Linus Torvalds' homepage can be found at <http://www.cs.Helsinki.FI/u/torvalds/>.

⁷ Ricadela, Aaron, "Linux Comes Alive," InformationWeek, January 24, 2000.

⁸ "The Future of Linux," CNet 2000, cites IDC data, no date provided.

⁹ For further information on Linux, visit The Linux Home Page <http://www.linux.org/>, Linux International <http://www.li.org/>, and MITRE Linux Resources Page <http://w030nt.mitre.org/users/terry/pub/linux/>.

twenty most active programmers. Each takes turns managing different parts of the project. There are an estimated one million Perl users today.¹⁰

Significance of Open Source

The open source development process differs sharply from the traditional commercial off-the-shelf (COTS) model. Eric Raymond likens the corporate or traditional COTS model, whereby a corporation produces and sells proprietary software, to a cathedral and the open source model to a bazaar.¹¹ In the corporate model, individuals or small groups of individuals quietly and reverently develop software in isolation, without releasing a beta version before it is deemed ready. In contrast, the open source model relies on a network of “volunteer” programmers, with differing styles and agendas, who develop and debug the code in parallel. From the submitted modifications, the delegated leader chooses whether or not to accept one of the modifications. If the leader thinks the modification will benefit many users, he will choose the best code from all of the submittals and incorporate it into the OSS updates. The software is released early and often.

Benefits and Risks of Open Source Software Compared to Traditional COTS

Due to the different development models, Program Managers can achieve many benefits over traditional COTS by using OSS. Popular open source products have access to extensive technical expertise, and this enables the software to achieve a high level of efficiency, using less lines of code than its COTS counterparts. The rapid release rate of OSS distributes fixes and patches quickly, potentially an order of magnitude faster than those of commercial software. OSS is relatively easy to manage because it often incorporates elements such as central administration and remote management. Because the source code is publicly available, Program Managers can have the code tailored to meet their specific needs and tightly control system resources. Moreover, Program Managers can re-use code written by others for similar tasks or purposes. This enables Program Managers to concentrate on developing the features unique to their current task, instead of spending their effort on re-thinking and re-writing code that has already been developed by others. Code re-use reduces development time and provides predictable results. With access to the source code, the lifetime of OSS systems and their upgrades can be extended indefinitely. In contrast, the lifetime of traditional COTS systems and their upgrades cannot be extended if the vendor does not share its code and either goes out of business, raises its prices prohibitively, or reduces the quality of the software prohibitively. The open source model builds open

¹⁰ For more information on Perl, visit <http://www.perl.com/pub>.

¹¹ Raymond, Eric, “The Cathedral and the Bazaar,” O’Reilly Associates, 1999.

standards and achieves a high degree of interoperability. While traditional COTS typically depends on monopoly support with one company providing support and “holding all the cards” (i.e., access to the code) for a piece of software, the publicly available source code for OSS enables many vendors to learn the platform and provide support. Because OSS vendors compete against one another to provide support, the quality of support increases while the end-user cost of receiving the support decreases. Open source can create support that lasts as long as there is demand, even if one support vendor goes out of business. For government acquisition purposes, OSS adds potential as a second-source “bargaining chip” to improve COTS support.

OSS can be a long-term viable solution with significant benefits, but there are issues and risks to Program Managers. Poor code often results if the open source project is too small or fails to attract the interest of enough skilled developers; thus, Program Managers should make sure that the OSS community is large, talented, and well-organized to offer a viable alternative to COTS. Highly technical, skilled developers tend to focus on the technical user at the expense of the non-technical user. As a result, OSS tends to have a relatively weak graphical user interface (GUI) and fewer compatible applications, making it more difficult to use and less practical, in particular, for desktop applications (although some OSS products are greatly improving in this area). Version control can become an issue if the OSS system requires integration and development. As new versions of the OSS are released, Program Managers need to make sure that the versions to be integrated are compatible, ensure that all developers are working with the proper version, and keep track of changes made to the software. Without a formal corporate structure, OSS faces a risk of fragmentation of the code base, or code forking, which transpires when multiple, inconsistent versions of the project’s code base evolve. This can occur when developers try to create alternative means for their code to play a more significant role than achieved in the base product. Sometimes fragmentation occurs for good reasons (e.g., if the maintainer is doing a poor job) and sometimes it occurs for bad reasons (e.g., a personality conflict between lead developers). The Linux kernel code has not yet forked, and this can be attributed to its accepted leadership structure, open membership and long-term contribution potential, GNU General Public License (GPL) licensing eliminating the economic motivations for fragmentation, and the subsequent threat of a fragmented pool of developers. Ninety-nine percent of Linux distributed code is the same. The small amount of fragmentation between different Linux distributions is good because it allows them to cater to different segments. Users benefit by choosing a Linux distribution that best meets their needs. Finally, there is a risk of companies developing competitive strategies specifically focused against OSS.

When comparing long-term economic costs and benefits of open source usage and maintenance to traditional COTS, the winner varies according to each specific use and set of circumstances. Typically, open source compares favorably in many cases for server and embedded system implementations that may require some customization, but fares no better than COTS for typical desktop applications. Indeed, some literature sources generalize that

open source products are no worse than closed source, but our findings indicate that the scale measuring the value derived from open versus closed source software can be heavily tipped in one direction or the other depending on the specific requirements and runtime environment of the software.

A decision between OSS and traditional COTS is based on three factors: (1) costs – both direct (e.g., price of software) and indirect (e.g., end-user downtime); (2) benefits (i.e., performance); and, (3) other, more intangible criteria (e.g., quality of peer support). Direct costs are largely understood and have traditionally comprised most of the total lifecycle costs of a system. However, indirect costs as well as operational and performance benefits (e.g., scalability, reliability, and functionality) play a most influential economic role in today’s more mature software market. Other, more intangible criteria are difficult to quantify, but can also impact the effectiveness of open and closed source software. Because indirect costs and operational and performance benefits play a much larger role in OSS compared to traditional COTS products, traditional lifecycle cost models and other COTS software tools can no longer be relied on for optimal mission-oriented and IT investment decision-making involving a choice of OSS.

To understand how indirect costs should be incorporated into the analysis, Program Managers must understand what these costs mean to their programs. Since the salary and other labor costs associated with an employee are direct costs, only the labor costs that are “wasted” and could be used in more productive ways should be included as indirect costs. In other words, although there is no additional direct cost to the organization, not as much output was received from the employee due to inefficiencies in the process or system. To a profit-making organization it would be hoped that this improved productivity increases profits. For example, time wasted could be spent bringing in more business. Within a Department of Defense (DOD) organization, the concepts of bringing in more business and increasing profits do not apply, and these lost productivity costs could be viewed as justification for force structure cuts. If, for example, an organization migrates to a new solution and experiences improved productivity, the organization could perform the same job with fewer people.) Data collection efforts to understand these metrics are viewed negatively by employees for this reason. Unless a direct cause-and-effect link can be established, it may be that some indirect influences are best viewed as relative costs rather than as absolute costs in support of IT investment analyses.

Program Managers need a complete taxonomy of lifecycle costs, benefits, and other, more intangible criteria to account for hidden costs and benefits that they might otherwise have overlooked. With this taxonomy, Program Managers can make software-purchasing decisions being fully aware of their economic, performance, and mission implications. The following table represents a cost element taxonomy for OSS developed by this research investigation.

Table ES-1. OSS Cost Element Taxonomy¹²






Direct Costs	
Software and Hardware	
Software	Purchase price
	Upgrades and additions
	Intellectual property/licensing fees
Hardware	Purchase price
	Upgrades and additions
Support Costs	
Internal	Installation and set-up
	Maintenance
	Troubleshooting
	Support tools (e.g., books, publications)
External	Installation and set-up
	Maintenance
	Troubleshooting
Staffing Costs	
	Project management
	Systems engineering/development
	Systems administration
	Vendor management
	Other administration
	Purchasing
	Other
	Training
De-installation and Disposal	
Indirect Costs	
Support Costs	
	Peer support
	Casual learning
	Formal training
	Application development
	Futz factor
Downtime	

¹² Futz factor is included by GartnerGroup as an indirect cost. GartnerGroup describes this term as the labor expense when the end-user exploits corporate computing assets for his own personal use during productive work hours.

In addition to a taxonomy of lifecycle costs, Program Managers also need a taxonomy of benefits and risks along with an example rating scale to compare the costs, benefits, and other, more intangible criteria of OSS and traditional COTS software. This research developed a taxonomy of benefits and risks for OSS and an example rating scale, and these are presented in Table ES-2 below.

Table ES-2. OSS Taxonomy of Benefits and Risks

Qualitative Attributes	
Ability to customize	
Availability/reliability	
Interoperability	
Scalability	
Design flexibility	
Lifetime	
Performance	
Quality of service and support	
Security	
Level of difficulty/ease of management	
Risk of fragmentation	
Availability of applications	

Example Rating Scale	
Very Strong	
Strong	
Neutral	
Weak	
Very Weak	

The above taxonomy comprises a list of qualitative attributes. For each attribute, Program Managers should compare the relative strength or weakness for OSS versus traditional COTS products. A relative strength would indicate a benefit, and a relative weakness would indicate a risk. An example rating scale is shown above for comparing the relative value of OSS versus traditional COTS. This example scale presents five ratings – very strong, strong, neutral, weak, and very weak. Since the ratings will differ depending on the specific use and environment of the software, Program Managers should customize their ratings according to their particular circumstances.

Compared to traditional COTS products, OSS provides more options to Program Managers for life-cycle supportability. The maintenance burden of OSS can be similar to

pure COTS (“buy”), custom code (“build”), or lie somewhere in between. Unmodified OSS can be considered similar to pure COTS. Thoroughly modified and owner-maintained OSS is comparable to custom code. “Modifiable COTS,” or OSS that relies on short-term modifications yet attempts to re-merge with newly released OSS updates, takes advantage of the benefits of both pure COTS and custom code. The following diagram illustrates this spectrum and points out differences between the above scenarios.

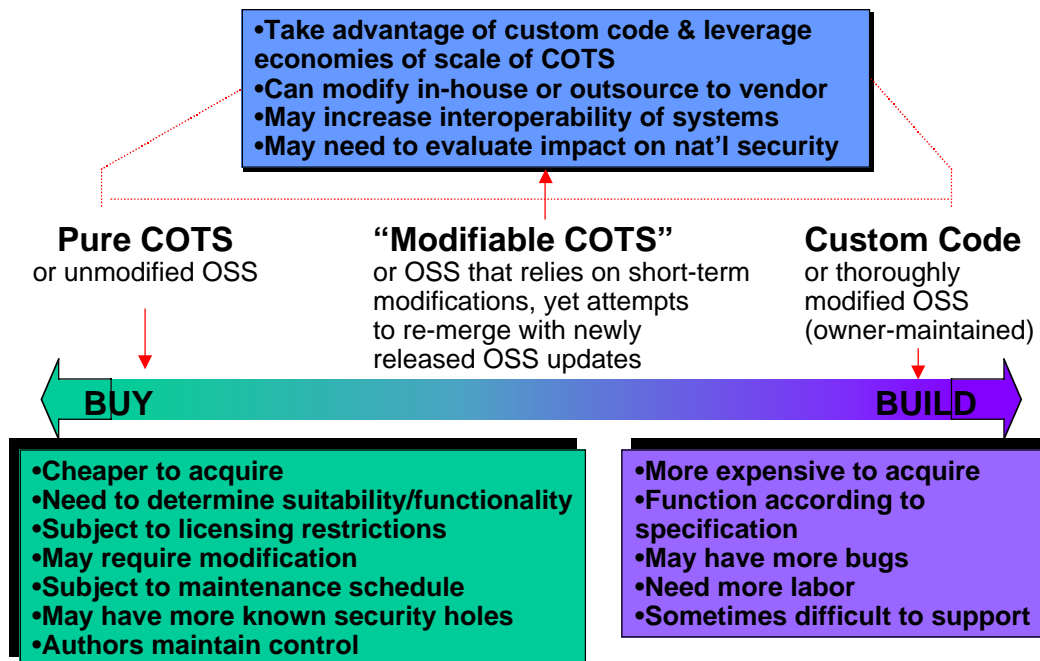


Figure ES-1. OSS Provides Several Maintenance and Support Options

Program Managers should evaluate the relative advantages and disadvantages of the pure COTS, “modifiable COTS,” and custom code maintenance models for their specific use and set of circumstances. Pure COTS is advantageous because it is cheaper to acquire. However, Program Managers need to assess the suitability and functionality of the software to their specific needs. The software may require modification, and Program Managers are subject to licensing restrictions and set maintenance schedules. Pure COTS may have more known security holes, and control is maintained by the authors of the software. “Modifiable COTS” takes advantage of customer code while leveraging the economies of scale achieved by COTS products. The software can be modified in-house or by a vendor. The interoperability of systems may be increased with “modifiable COTS.” The impact on national security may need to be evaluated. Custom code is more expensive to acquire, functions according to specification, may have more bugs, requires more labor, and is sometimes difficult to support.

Open source will benefit the government by improving interoperability, long-term access to data, and ability to incorporate new technology. Interoperability increases because open source enables the same code, documentation, and data formats to be used in every system component. (However, the downside risk of exposure should be evaluated; if the security of an open source system is compromised, interoperability could also be compromised.) Long-term access to data gives the user full access to its own systems. It is possible to contract out maintenance development work to support vendors, who have the same information as the original supplier. Open source can allow the government to more easily adopt new technology because it reduces the cost and risk of change. Open source projects tend to be evolutionary and less disruptive to operations.

The Military Business Case

The military has different software needs than the commercial sector because of its unique mission and environment. Software attributes most important to the commercial sector include application choice, ease of use, service and support, price, reliability, and performance. Most operationally significant attributes for software used in the military include reliability, long-term supportability, security, and scalability. Additional attributes of highest programmatic significance to the military include cost or price, availability or multiple distribution sources, and popularity or brand/reputation.

While both the commercial and government sectors are concerned about price and reliability, certain commercial customers generally have less stringent requirements for security, availability, and long-term supportability. However, these features are becoming more important in the private sector. E-commerce companies must have high levels of security to protect personal financial information and transactions. Availability of software from multiple sources increases competition, resulting in higher quality at low prices. Long-term supportability is important to businesses needing to access legacy data. If a commercial product or process, such as open source, is deemed suitable and offers the required functionality, the military can take advantage of these to achieve significant cost savings. There are other potential benefits to leveraging commercial products or processes, including faster deployment time, improved quality and reliability, reduced development risks, and a support system already in place.

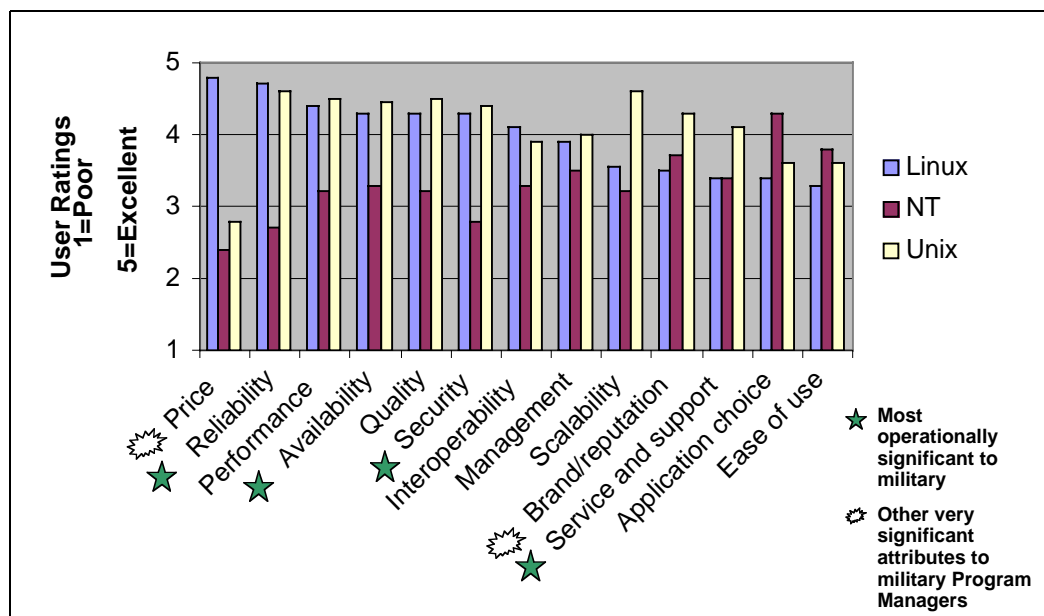
Applicability of Linux to the Military Business Case

Linux has attracted a large group of highly trained developers, and “given enough eyeballs, all bugs are shallow.”¹³ Over 120,000 programmers contribute to Linux,

¹³ Raymond, Eric, “The Cathedral and the Bazaar,” O’Reilly Associates, 1999.

volunteering about 2 billion dollars worth of labor.¹⁴ This massive amount of technical expertise could not be afforded by providers of traditional COTS products. As a result of the open source process, highly reliable and stable software is produced. This comparative advantage, along with its perceived low price, enables Linux to attract a large user base worldwide.

The following graph compares user ratings of Linux, NT, and Unix.¹⁵ While Linux is used because of its perceived low price and reliability, NT is preferred for its choice of applications and ease of use. Users select Unix for its performance, availability, quality, security, management, scalability, brand/reputation, and service and support.



Source: US Linux user ratings by server OS from Michelle Bailey, Vernon Turner, Jean Bozman, and Janet Waxman, “Linux Servers: What’s the Hype, and What’s the Reality,” IDC, March 2000.

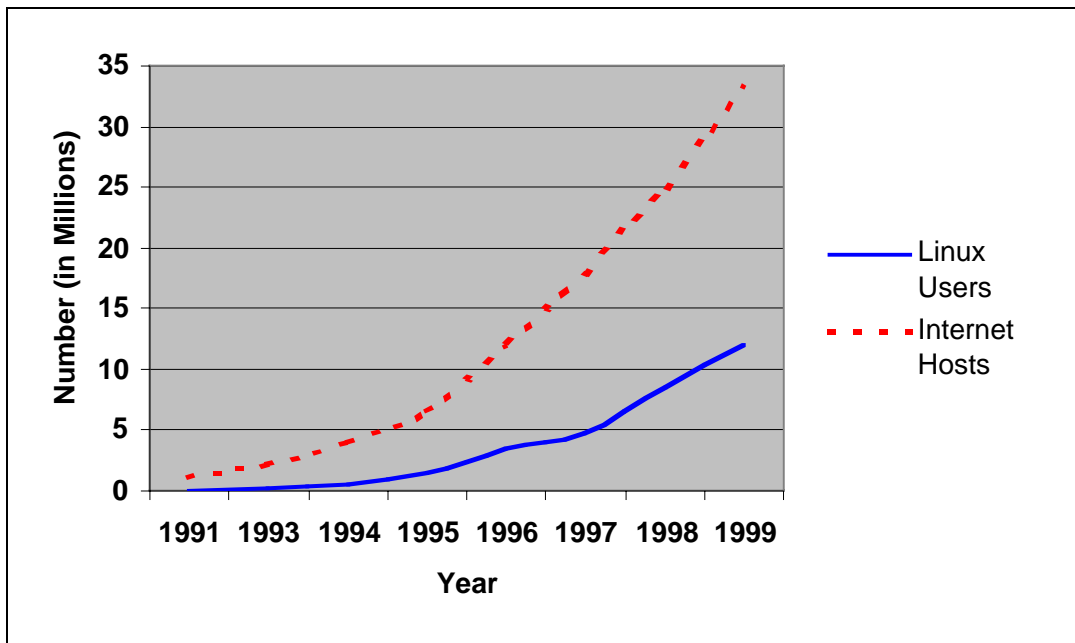
Figure ES-2. Military and Commercial User Benefits of Linux

¹⁴ Orzech, Dan, “Linux and the Saga of Open Source Software,” Datamation, February 1999 and Dan Kaminsky, “Core Competencies: Why Open Source is the Optimum Economic Paradigm for Software,” March 2, 1999.

¹⁵ US Linux user ratings by server OS from Michelle Bailey, Vernon Turner, Jean Bozman, and Janet Waxman, “Linux Servers: What’s the Hype, and What’s the Reality,” IDC, March 2000.

Use of Linux

The number of Linux users worldwide has grown from 1 user (Linus Torvalds) in 1991 to an estimated 12 million users in 1999. The following graph plots the number of Linux users worldwide against the number of Internet hosts worldwide, and shows that the number of Linux users has been growing with the number of Internet hosts. As the Internet expands, the number and productivity of open source development teams increase and attract more users.¹⁶



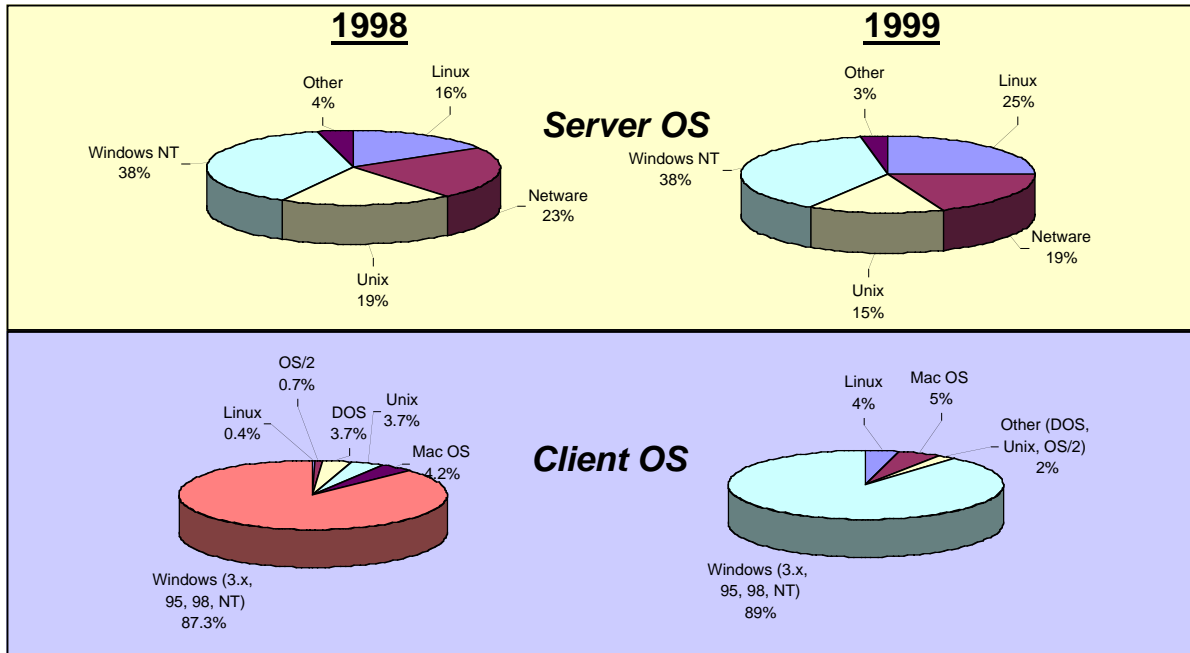
Source: Linux estimates derived from GartnerGroup, IDC, and Red Hat market research. Internet estimates based on research from Bruce L. Egan, 1996. Data based on year-end estimates.

Figure ES-3. Worldwide Success of Linux in the Marketplace

Most Linux installations are expected to be in servers. Significant investments in areas such as ease of use and configuration are needed for Linux to achieve success on desktops.

¹⁶ Linux estimates derived from GartnerGroup, IDC, and Red Hat market research. Internet estimates based on research from Bruce L. Egan, 1996. Data based on year-end estimates.

The following pie charts shows the Linux market share for the server and client OS market in 1998 and 1999.¹⁷



Source: "The Future of Linux," CNet, 2000 cites IDC.

Figure ES-4. Server and Client OS Market Share in 1998 and 1999

Although Linux deployments are widening, they are not deep. Between 1998 and 1999, the Linux server OS market share grew from 16 percent to 25 percent and the Linux client OS market share grew from 0.4 percent to 4 percent. It appears that most of this growth came from Unix users who switched to Linux.

Discussion

Although the open source development process offers many benefits over traditional COTS, Microsoft Windows continues to dominate the market. There are several reasons for this. First, Microsoft has invested significantly in marketing Windows to developers. Second, NT is a very broad platform that enables servers from different vendors to work on NT. In fact, there are over 100 NT server vendors.¹⁸ Third, users often choose Windows

¹⁷ "The Future of Linux," CNet, 2000 cites IDC, no date provided.

¹⁸ Deate Hohmann, GartnerGroup, phone conversation, December 2000.

because of the large choice of compatible applications and its ease of use. There is an affinity between the desktop and server environments when Microsoft products are used. Fourth, Windows NT has historically had a much lower initial cost of entry compared to Unix. Hardware and software costs are lower when using NT because the system runs on commodity components and standard chipset and storage devices. For the above reasons, Windows is perceived as a less risky choice by IT management. Industry analysts further add that “no one ever got fired for buying Microsoft.”¹⁹

Despite these pro-Microsoft observations, GartnerGroup has concluded that one cannot generalize whether NT or Unix offers the least expensive long-term support. Instead, the least expensive choice depends on the specific application, environment, and current skill base of the organization.²⁰ It should also be noted that Windows does not scale as well as Unix, and this can turn the tables on the relative total costs of Windows versus Unix. NT is not as powerful as Unix and, according to GartnerGroup tests, NT can only support up to 1,000 concurrent users.²¹ Smaller organizations that grow into larger ones must correspondingly add more boxes to support its larger user base. In some instances, five-times as many boxes of NT may be required to get the same performance as a Unix box. Organizations that do not plan for growth often choose Windows for its low initial cost of entry, while organizations that plan for aggressive growth upfront may choose Unix. Therefore, the optimal choice of Windows versus Unix depends on the number of users the system supports. As the number of users increase to over 1,000, Unix becomes the most effective platform, or optimal platform choice.

Since the recent surge in online use that has helped to fuel the maturation of Linux, there have been small migrations to Linux. Some users of Unix have shifted to Linux, a Unix-like OS. In addition, some start-up businesses with little capital choose Linux because it runs nicely on older computers. If more Program Managers compared OSS to traditional COTS for their specific business case, it is likely that there would be many more users of OSS today.

Considerations for Military Program Managers

OSS provides more options than traditional COTS for life-cycle supportability, particularly for long-lived systems. It can be used in the form of pure COTS, “modifiable COTS,” or custom code. Program Managers’ requirements for operating systems differ considerably depending on their particular environmental and mission requirements.

¹⁹ Garvey, Martin J., “The Hidden Cost of NT,” *InformationWeek*, July 1998.

²⁰ Hohmann, Deate, GartnerGroup, phone conversation, December 2000.

²¹ Hohmann, Deate, GartnerGroup, phone conversation, December 2000.

Command and Control (C2) Program Managers are operationally-driven. For these managers, the cost of failure is very high. Reliability and performance are essential. C2 Program Managers use traditional COTS unless the system requires more customization, and system upgrades tend to be frequent. C2 Program Managers should consider using Linux because it provides the highest level of reliability with good performance. NT is weakest for both of these metrics.

Information System (IS) Program Managers are driven by costs, quality of support, and application choice. Systems are generally replaced every five to seven years. If application choice is important, IS Program Managers should consider NT. Otherwise, Program Managers may find more service and support options with Unix and Linux. Tapping into the “modifiable COTS” option with Linux could provide very valuable additional features without the added maintenance burden associated with them.

Embedded/Weapon System Program Managers are driven by portability, ruggedness, and hard real-time requirements. System upgrades are typically expensive endeavors. Embedded/Weapon System Program Managers will likely find Linux most appealing. Its design flexibility enables the kernel to be either pared down to eliminate unnecessary features or expanded to include additional features. Linux is portable to many central processing units (CPUs) and hardware platforms. It is stable and scalable over a wide range of capabilities and easy to use for development. The software can dynamically reconfigure itself without rebooting. Linux can isolate faults and processes. Processes can load and remove kernel modules, device drivers, and custom modules based on available resources and dynamic application needs. The applications are also modular with well-defined interfaces. Furthermore, hard real-time capabilities are available from the Linux kernel extension RTLinux.

Federal Linux Award

The MITRE Corporation recently received a Leadership Award from the non-profit Potomac Forum for showing that OSS can provide substantial advantages over commercial software, particularly when reliability and long-term support are key requirements. The award was recently presented jointly to MITRE and the Office of the Secretary of Defense at the first Federal Linux Users Group conference at Crystal City, Virginia. MITRE earned the award for investigating the technology and economics of OSS in its research project, “Open Source for Military Systems.” According to Mark Norton, Office of the Assistant Secretary of Defense, “This MITRE study is the first study of Linux and other OSS that addresses both the technical advantage and the business case for using open source in Department of Defense.” The MITRE research team included technical staff members Frank McPherson, David Emery, Terry Bollinger and Carolyn Kenwood. MITRE's work included demonstrating the use of Linux in embedded systems such as the Abrams Tank and for information assurance within Army Tactical Operations Centers. MITRE also analyzed the case for federal use of OSS to help Program Managers evaluate its suitability for their

technology programs. For more information on this award, visit the website http://www.mitre.org/news/articles_00/linux12_5_00.shtml.

Conclusion

OSS is a viable long-term solution that merits careful consideration because of the potential for significant cost, reliability, and support advantages. However, these potential benefits must also be carefully balanced with a number of risks associated with OSS approaches and products. The optimal choice of OSS versus traditional COTS varies according to the specific requirements and runtime environment of the software. OSS is often a good option for products relevant and interesting to a large community with highly skilled developers. It typically compares favorably for server and embedded system implementations that may require some customization, but fares no better than traditional COTS for typical desktop applications. When making a decision about whether to use OSS or traditional COTS, it is recommended that Program Managers follow the five steps presented below.

1. **Assess the supporting OSS developer community (e.g., Linux, Apache).** Look for communities that are large, talented, and well organized.
2. **Examine the market.** Is there a strong and increasing demand for the specific OSS product? To what extent have vendors and service providers emerged in the commercial marketplace to provide complementary services and support not available from the community?
3. **Conduct a specific analysis of benefits and risks.** The MITRE effort has developed a taxonomy of OSS benefits and risks (see Table ES-2) that can be used to compare candidate OSS products to your specific economic, performance, and mission objectives.
4. **Compare the long-term costs.** Use the MITRE-developed OSS Cost Element Taxonomy (see Table ES-1) to compare the long-term costs associated with usage and maintenance of OSS versus traditional COTS relative to your specific objectives.
5. **Choose your strategy.** Following the previous four steps will provide enough information and detail to choose the most effective option combination of OSS, traditional COTS, and proprietary development to support objectives.

In conclusion, open source methods and products are well worth considering seriously in a wide range of government applications, particularly if they are applied with care and a solid understanding of the risks they entail. OSS encourages significant software development and code re-use, can provide important economic benefits, and has the potential for especially large direct and indirect cost savings for military systems that require large deployments of costly software products.

Section 1

History of Unix and Linux

Many of the cooperative development efforts in the 1970s focused on building an operating system that could run on multiple computer platforms. The Unix operating system emerged as the most successful of these efforts. The process of sharing code rapidly accelerated with the emergence of Usenet, a computer network begun in 1979 to link together the Unix programming community. Up to this point, the cooperative software development efforts were informal and did not attempt to define property rights or restrict use. This informality became problematic in the early 1980s, when AT&T claimed intellectual property rights related to Unix.

In 1986 developers attempted to build a free version of the Unix operating system. This project, called GNU, allowed individual programmers, regardless of individual or commercial interests, to contribute to the development effort. GNU stands for “Gnu’s not Unix.” In the end, users were not charged for the operating system.

The GNU General Public License,²² also known as a copyleft agreement, includes the following key points²³:

- Software licensed under GNU General Public License can be copied and distributed under this same license.
- Products obtained and distributed under this license may be sold.
- Users may alter the source code, but if they distribute or publish the resulting work, they must make the software available under the same licensing terms.
- Ancillary technology can be developed, and as long as such products do not include code licensed under the GNU General Public License, they need not be licensed or made available under the terms of the GNU General Public License.

The full GNU license is available at <http://www.gnu.org/copyleft/gpl.html>.

The Linux kernel grew out of an educational project at the University of Helsinki in 1991. Linus Torvalds, a young student, created the Linux operating system and gave hackers his code so they could contribute to the development. Many programmers analyzed his code

²² The acronym *GNU* stands for GNU’s Not Unix.

²³ Gillen, Al, and Dan Kusnetzky, “Linux Overview: Understanding the Linux Market Model,” IDC, February 2000.

and wrote improvements that Linus incorporated into Linux. Linux grew and expanded into an advanced and powerful, multi-use operating system.

Conformance to an open standard was always an important goal of Torvalds. Linux aims toward Portable Operating System Interface (POSIX), a standard application programming interface (API) commonly used by Unix and Unix-like systems, but it does not conform to all the POSIX specifications contained in mainstream Unix operating systems. Using POSIX makes it easier to write source code that can be compiled on different POSIX systems. It gives Linux developers a well-defined API to share so that they do not have to track most kernel changes as long as the kernel follows POSIX. Using POSIX enabled Linus and other early Linux developers to use existing free programs written by the GNU Project, the BSD operating system, and many other free programs based on POSIX. However, Linux has not been tested by the Open Software Foundation, the owner of the Unix trademark and, therefore, is not considered a Unix implementation.

Section 2

Business Case Analysis Model

The business case analysis model was applied to Linux products and processes, and the adapted framework was followed to analyze the viability of Linux software products and design methodologies to government Program Managers.

First, this approach scanned the environment. The external environment including customers, competitors, barriers to entry, substitutes, suppliers, and distributors – was examined. Opportunities and threats were identified from the external scanning. The internal Linux community – including experience, technical skills, management, financial health, culture, organizational structure, and products and services – was assessed. Strengths and weaknesses were recognized from the internal scanning.

The second step analyzed strategic factors. The internal and external investigations were integrated to form the strengths, weaknesses, opportunities, and threats (SWOT) analysis. Internal expertise was compared to the competition to identify distinctive competencies. To analyze the distinctive competencies, the research team examined the ability of the competitive advantages to fulfill an unmet market need, and the strategy for competing on price, quality, and/or customization. In addition, new products and services were evaluated relative to existing ones. The current and potential market demand was then assessed. Questions were asked, such as: How much are customers currently spending? What percent are willing to purchase the new product(s) and/or service(s)? How much are they willing to spend? The team analyzed the position of Linux in the market, identifying short- and long-term opportunities and highlighting risks.

Finally, the MOIE team assessed the feasibility of Linux based on information obtained in Step 1 (environmental scanning) and Step 2 (analysis of strategic factors).

The following figure illustrates the business case analysis framework. This includes the above three steps as well as the numerous inputs into the process.

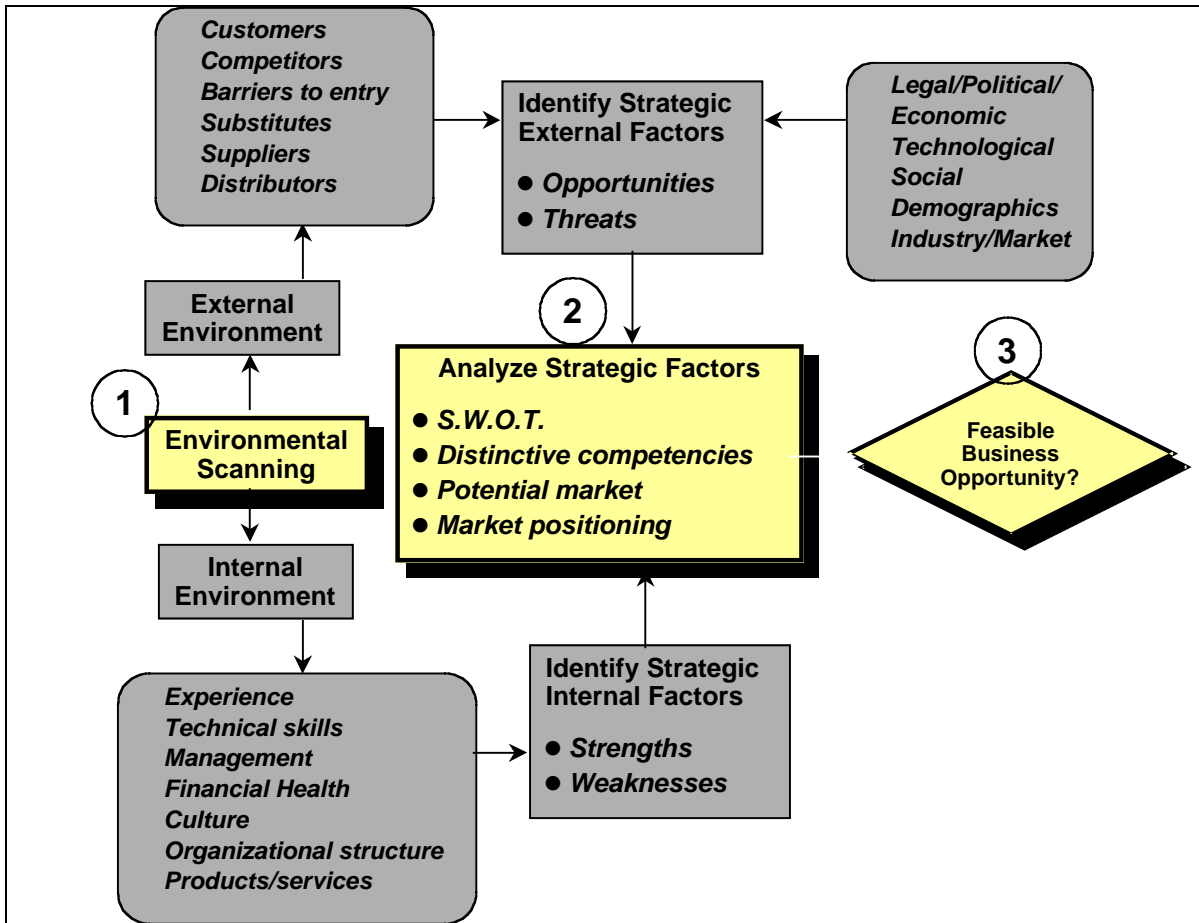


Figure 1. Business Case Analysis Framework Applied to Open Source Products and Processes

Section 3

Commercial Business Case Analysis of Linux

The business case analysis model was followed to assess the commercial viability of Linux. Results indicate that there is a business case for Linux.

3.1 Environmental Scanning

The following figure summarizes the key elements of the SWOT analysis. Under the internal environment heading, strengths and weaknesses are indicated by green and red, respectively. Under external environment, green specifies opportunities and red threats. Yellow indicates both a strength and weakness or an opportunity and threat.

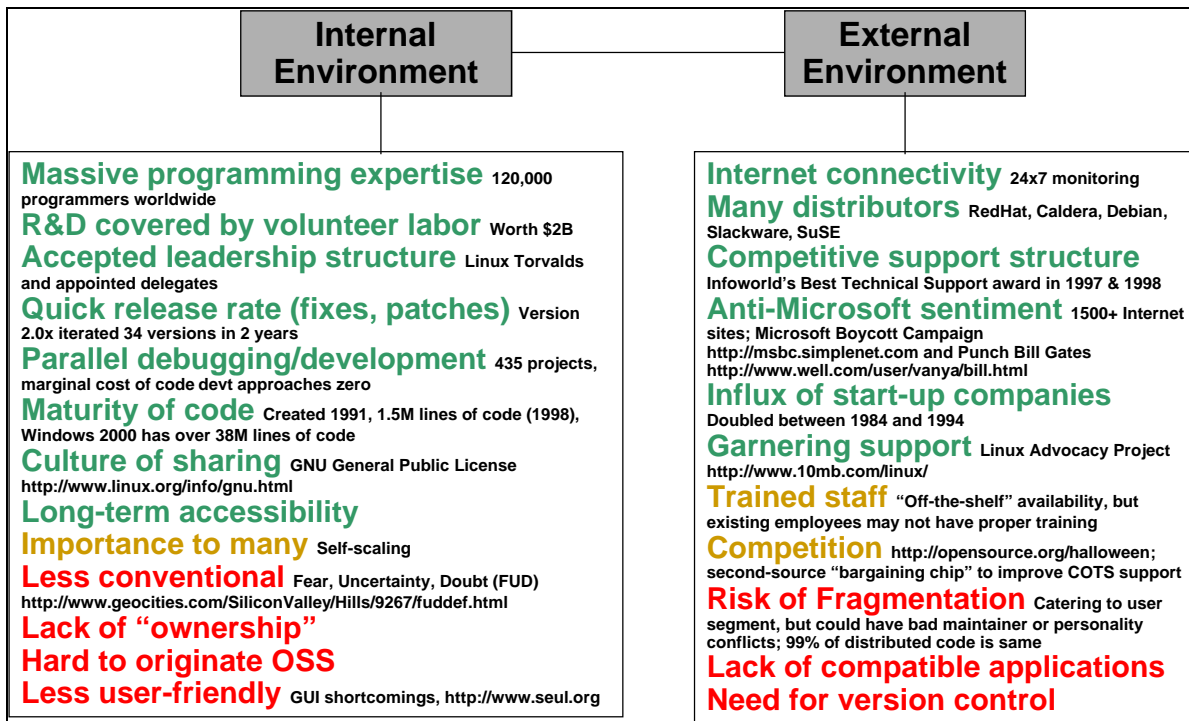


Figure 2. Key Elements of SWOT Analysis

3.1.1 Strengths

3.1.1.1 Massive Programming Expertise

Linux has a massive pool of programming expertise, over 120,000 programmers worldwide.²⁴ Linux developers are distributed internationally, and many foreigners support Linux as a means of reducing US technical domination. Open source development is self-scaling; the more valuable a project is, the more programmers will want to join. It is estimated that only 5-10 percent of the Linux kernel remains written by Linus Torvalds.²⁵

3.1.1.2 R&D Covered by Volunteer Labor

Research and Development efforts are covered by volunteer labor that is worth about two billion dollars.²⁶ Companies that build their own operating systems spend about \$80-100 million per year to play in the market.²⁷ Programmers contribute to Linux code on the side as a hobby or personal interest, usually falling outside of their professional responsibilities. However, as new commercial versions are emerging in the marketplace, this is beginning to change somewhat; Linux distributors often hire paid, full-time developers to improve the code and contribute to the growth of the Linux market. Developers are motivated to contribute their time and without monetary reimbursement. They sometimes fix a bug or customize a program for their own benefit (and, therefore, for the benefit as others as well). Others contribute to the open source code to receive ego gratification and a reputation among other hackers. Like in gift cultures, “social status is determined not by what you control but by what you give away.”²⁸ Delayed and unexpected rewards, such as future job offers, shares in commercial open source-based companies, or future access to the venture capital market, have been received by Linus Torvalds and other open source programmers.

3.1.1.3 Accepted Leadership Structure

Linux has an accepted leadership structure, similar to that found in a corporate organization. The Linux community is headed by Linus Torvalds, a well-respected manager

²⁴ Orzech, Dan, “Linux and the Saga of Open Source Software,” *Datamation*, February 1999.

²⁵ Kaminsky, Dan, “Core Competencies: Why Open Source is the Optimum Economic Paradigm for Software,” March 2, 1999.

²⁶ Kaminsky, Dan, “Core Competencies: Why Open Source is the Optimum Economic Paradigm for Software,” March 2, 1999.

²⁷ Kusnetsky, Dan, IDC, and Greg Weiss, DH Brown & Associates, *Linux E-Seminar Series*, 1999.

²⁸ Raymond, Eric, “Homesteading on the Noosphere,” no date provided.

who has achieved celebrity status, and his decisions are considered final. Torvalds has appointed delegates who are responsible for managing certain areas of the project and, in turn, these delegates have a team of coordinators. However, this leadership structure only applies to the Linux kernel; it does not pertain to supporting areas like the Graphical User Interface (GUI), system utilities and servers, and system libraries.

3.1.1.4 Quick Release Rate

Linux releases fixes and patches quickly, potentially an order of magnitude faster than commercial software. For example, version 2.0x iterated 34 versions in two years.²⁹

3.1.1.5 Parallel Development and Debugging

Open source projects utilize multiple small teams of individuals that work independently to solve specific problems. Since open source developers work on a volunteer basis, the parallel development process is not prohibitively expensive like it often is in the commercial sector. Open source fosters creativity, since developers are not mandated to work within particular limitations. The parallel development process makes it possible for 435 Linux projects to be concurrently underway.³⁰ Since developers are essentially unpaid for their contributions to open source products, the marginal cost of development approaches zero. Parallel debugging and development efforts enable the coordinating developer to choose the best potential implementation from the many choices offered.

Parallel debugging, according to Eric Raymond, improves efficiency nearly linearly with the number of individuals working on the project. Little, if any, management costs exist with the debugging of OSS. Parallel debugging typically results in quicker fixes than traditional processes. For example, the Linux community developed a fix for the TearDrop IP attack in less than 24 hours after it first surfaced on the Web. Organizations do not need to rely on a commercial provider's schedule for fixing a work-stopping bug but can, instead, opt to fix the problem themselves.

3.1.1.6 Maturity of Code

The Linux code was created in 1991, and in 1998 it comprised 1.5 million lines of code. Windows 2000 has over 38 million lines of code. Frederick Brooks claims that "complexity

²⁹ Valloppillil, Vinod, and Josh Cohen, Microsoft, "Linux OS Competitive Analysis," Halloween II, August 11, 1998.

³⁰ Orzech, Dan, "Linux and the Saga of Open Source Software," Datamation, February 1999.

is a function of the square of the number of lines of code” and, thus, Windows 2000 contains much more complexity than does Linux.³¹

3.1.1.7 Culture of Sharing

The GNU General Public License has fostered a culture of sharing, which is pervasive throughout the community. OSS eliminates the economic loss associated with duplicated work. About 75 percent of all code is written for a specific task by a single organization and is never used for any other purpose.³² Many problems in computer engineering apply across a wide range of fields and applications. This economic waste hurts US productivity.

3.1.1.8 Long Term Accessibility

Unlike proprietary companies, the open source movement cannot be driven out of business in the near-term. As long as sufficient interest and skills exist from the development community, the life of the open source product will continue. Furthermore, since the code is publicly available, the user is not entirely dependent on one organization to maintain and support the software. The user always has the choice to provide in-house maintenance and support to continue the product’s life indefinitely.

3.1.2 Weaknesses

3.1.2.1 Lack of “Ownership”

Users want accountability. A proprietary company is more tangible than the “open source process,” and can position itself to potential and current customers as a trusted caretaker. Microsoft, for instances, resorts to Fear, Uncertainty, Doubt (FUD) tactics. FUD is a marketing technique generally used by companies with a large market share (i.e., Microsoft, IBM) that are unable to respond with hard facts to a competitor product that is better and less expensive. This technique creates gossip and casts doubt to potential users, such as “Hey, it could be risky going down that road, stick with us and you are with the crowd. Our next soon-to-be released version will be better than that anyway.”³³

To reduce this lack of “ownership” weakness, Eric Raymond proposes higher quality documentation that promotes authors and publishers for customers to trust. Unlike OSS,

³¹ White, Walker, “Observations, Considerations, and Directions,” Oracle, May 8, 2000 cites Federick Brooks in “The Mythical Man Month.”

³² Stoltz, Mitch, “The Case for Government Promotion of Open Source Software,” A NetAction White Paper, 1999.

³³ Raymond, Eric, “What is FUD?” <http://www.opensource.org>, no date provided.

proprietary companies can guarantee backward compatibility and represent an entity to sue if promises are not fulfilled. There is also concern that open source projects lack a strategic direction, although innovators do continually work to add new features for added functionality, “coolness,” and reputation. As summed up by Martin J. Garvey in InformationWeek, “no one ever got fired for buying Microsoft.”³⁴

3.1.2.2 Hard to Originate

In order for an open source project to be viable, it must be able to amass a large enough community of highly skilled and interested developers to concentrate on a problem. One of Eric Raymond’s rules is that “every good work of software starts by scratching a developer’s personal itch.”³⁵ The open source project must be relevant and interesting to a large group of developers. The larger the project, the more development and debugging the code receives. Developers must share in a common goal that is clear and well-defined, analogous to an organization’s mission statement. Linux succeeded in surmounting this potential weakness because the Linux community had (over 25) years of shared experiences working on other forms of Unix; it had already adopted a common Unix skill set.

3.1.2.3 Less User-Friendly

Although Linux is working to improve its user-friendliness, its Graphical User Interface (GUI) is weak relative to Microsoft and other software products. Linux was developed for the programmer, rather than for the non-technical user. The following figure shows a snapshot of Linux code, part of the kernel source code that handles process forking (a basic operation of any Unix-like kernel).³⁶

³⁴ Garvey, Martin J., “The Hidden Cost of NT,” InformationWeek, July 1998.

³⁵ Raymond, Eric, “The Cathedral and the Bazaar,” O’Reilly Associates, 1999.

³⁶ McPherson, Frank, The MITRE Corporation, 2000.

```

/*
 * For SMP, we need to re-test the user struct counter
 * after having aquired the spinlock. This allows us to do
 * the common case (not freeing anything) without having
 * any locking.
 */
#ifdef __SMP__
#define uid_hash_free(up)  (!atomic_read(&(up)->count))
#else
#define uid_hash_free(up)  (1)
#endif
void free_uid(struct task_struct *p)
{
struct user_struct *up = p->user;
    if (up) {
        p->user = NULL;
        if (atomic_dec_and_test(&up->count)) {
            spin_lock(&uidhash_lock);
            if (uid_hash_free(up)) {
                uid_hash_remove(up);
                kmem_cache_free(uid_cachep, up);
            }
            spin_unlock(&uidhash_lock);
        }
    }
}

```

Source: Frank McPherson, The MITRE Corporation, 2000.

Figure 3. Snapshot of Linux Code

Unlike the kernel which is maintained by Linus Torvalds, the Linux GUI has not been singularly maintained and, thus, has a highly forked tree. The software does not have a consistent GUI look or feel so users must adjust to the differences. GNOME, KDE and CDE initiatives are working to improve the GUI. KDE integrates the browser, shell, and office suite for Unix desktops. KDE screenshots can be found at <http://www.kde.org/ksscreenshots.html>. An example screenshot of Linux GUI is depicted below.³⁷

³⁷ [Http://www.gnome.org/screenshots/index.html](http://www.gnome.org/screenshots/index.html), no date provided.



Source: <http://www.gnome.org/screenshots/index.html>.

Figure 4. Example of Linux Screenshot

Applications and middleware components exist to improve the ease of server deployments. Samba, for example, is an open source file server package that enables a Linux server to support Microsoft Windows desktop clients with print and file serving services.

3.1.3 Opportunities

3.1.3.1 Internet Connectivity

The number and productivity of open source development teams expands with the Internet. The technology of the Internet enables Linux development and support to continue 24 hours a day, 7 days a week around the world. The growth of the Internet will continue to expand Linux and other open source projects by making them accessible to a larger number

of people. Collaborative technologies, such as e-mail lists, newsgroups, and websites have fostered the growth of open source.

3.1.3.2 Many Distributors

There are some 204 unique distributions of Linux on the market.³⁸ Distributors offer Linux software packages with integrated tools. They act as an intermediary between the fast-paced Linux development process and customers who do not care about the day-to-day changes to the kernel, and remove many of the hassles of a downloaded operating system. Vendors are also providing sales, support, and integration, emphasizing services rather than the software product itself. These complimentary services are often not sufficiently supplied by the community. As the Linux distributors compete against one another, they escalate the evolution of the operating system by adding features, improving its packaging, and advertising for its use. The open source community provides an ongoing “service” evolving with the user to meet emerging needs, rather than a “product” that remains static and eventually requires replacement.

While there will likely be some market consolidation over the next couple of years, regionalized and niche versions of the software will continue to exist to meet the market’s demand. The following table lists some of the most common Linux distributions.

Table 1. Common Distributions of Linux by Vendor

Vendor	Distribution Name	URL
Caldera Systems	OpenLinux	www.caldera.com
CoolLogic	Coollinux	www.coollogic.com
Corel	Corel LINUX	www.corel.com
Debian Project	Debian	www.debian.org
LinuxPPC	LinuxPPC	www.linuxppc.com
MandrakeSoft	Linux-Mandrake	www.linux-mandrake.com/en/
CLE Project	CLE	cle.linux.org.tw/CLE/e_index.shtml
Red Hat	Red Hat Linux	www.redhat.com
Slackware Linux	Slackware Linux	www.slackware.com/
Stampede GNU/Linux	Stampede GNU/Linux	www.stampede.org
SuSE	SuSE Linux	www.suse.com
TurboLinux	TurboLinux	www.turbolinux.com

³⁸ “Getting to Know Linux,” Colorado Business, July 2000.

Red Hat, founded in 1994, distributes the most popular version of Linux and is preferred by 68.7 percent of US Linux users, according to IDC.³⁹ The company provides open source Internet infrastructure solutions across a wide range of applications, from embedded devices to clusters and web serving. The company contributes its software innovations freely with the community under the GNU General Public License. Red Hat has partnered with top PC and server manufacturers, including Compaq, Dell, Gateway, IBM, Hewlett-Packard, and Silicon Graphics. Red Hat has a graphical system that provides online help throughout the installation process. It has simplified the process of adding third-party software utilities to the core system with its Red Hat Package Manager (RPM) utility and package format. Red Hat also offers training curriculum and a certification program. Red Hat and the Red Hat Linux OS has received many industry awards including: Red Herring's Top 100 Companies of the Electronic Economy, Upside's Hot 100 Companies, Network World's 10 Companies to Watch, Federal Computer Week's Government Best Buy, Software Development's Jolt Award, and InfoWorld's Product of the Year for three years in a row. The company announced it would go public with a \$96.6 million stock offering in June of 1999.

Like Red Hat, Caldera's OpenLinux offers package management and graphical installation systems. It has comprehensive training programs, both in-house and through the Authorized Linux Education Center, designed to prepare students for the Linux Professional Institute Certificate.

Debian, a not-for-profit organization, built the GNU/Linux distribution on a volunteer basis. This distribution is the best choice for freeware. The distribution is actively updated and supported by devoted, well-organized supporters. The format of the package is similar to RPM. Although the initial system setup is not as straightforward as that of Red Hat, a trained systems administrator will find it more efficient. Although the Debian distribution is not available from the Debian organization, it is available through other vendors, such as Corel.

Slackware was one of the first distributions to emerge, and continues to be managed actively. However, its installation is less intuitive than that of other distributions. Slackware relies on compressed archives and does not support the popular RPM format.

SuSE is the most popular distribution among European Linux users. It is often recommended for intermediate to advanced users looking for a security-aware distribution out of the box.

CoolLogic is a leading developer of embedded operating systems. Coollinux is a real-time operating system designed for Internet appliances. It can reduce the kernel size to meet the memory footprint and functionality requirements of embedded systems. Coollinux can be reduced to 355 kilobytes.

³⁹ RedHat, <http://www.redhat.com>, no date provided.

LinuxPPC, MandrakeSoft, CLE Project, Stampede GNU/Linux, and TurboLinux also provide common Linux distributions.

In addition to the above vendors, many market-specific and niche vendors sell unique Linux distributions or versions that include no royalty for the original distribution creator. These companies are usually smaller players focused on a specific demographic sector of the market. The following table includes a list of these market-specific and niche vendors.⁴⁰

Table 2. Market-Specific and Niche Vendors of Linux

Vendor	URL	Vendor	URL
Admail Japan	ms.shi.nu	Linuxbutiken.com	www.linuxbutiken.com
Alcove	www.alcove.fr/guide-linux/logiciels/debian.html	LinuxLand International	www.linuxland.de
Alejandro Sierra	debian.ipt.com.mx	LinuxPPC	www.linuxppc.com
Armed.net	www.armed.net	LinuxStore	www.linuxstore.com.br
BeNeLinux	home1.freegates.be/linux	Livraria Tempo Real	linux.temporeal.com.br
Buchhandlung Lehmanns	www.lob.de	LordSutch.com	www.lordsutch.com/cds
CD House di CA Barwood	www.sbf.it/cdnet	LSL Australia	www.lsl.com.au
Cheap*Bytes	www.cheapbytes.com	Macmillan USA	www.mcp.com
Ciberdroide Informatica	www.ciberdroide.com	MandrakeSoft	www.linux-mandrake.com/en
Coyote Linux	www2.vortech.net/coyte/coyte.htm	MNIS	www.mnis.fr
Data-Portal	www.data-portal.com	Mr O's Linux Emporium	www.ouin.com
DLX Linux	www.wu-wien.ac.at/usr/h93/h9301726/dlx.html	Net North West	www.netnw.com
DOS Linux	www.tux.org/pub/people/kent-robotti/index.html	NetArt	www.netart.com.pl
Dr Floppy Computers	www.drifloppy.co.nz	nettstore	www.nettstore.de
Dragon Linux	www.dragonlinux.nu	NoMad Linux	www.nomadlinux.com/nomad.html
Dutch Debian Distribution Initiative	panic.et.tudelft.nl/debian/cd	Open Source project	www.webnix.com/CLEe_index.shtml
easy Information	www.eit.de/c/index.html	Peanut Linux	www.metalab.unc.edu/peanut
edv-multimedia service	www.net-operations.de	Penguin Computing	www.penguincomputing.com
Eridani Star System	www.eridani.co.uk	Philip Charles	www.copyleft.co.nz
Esware Linux	www.esware.com	Philip Hands Computing	www.hands.com/debiangold.html
Eurielec Linux	www.etsit.upm.es/~eurielec/indexuk.html	Prosa Debian Linux	www.prosa.it/prosa-debian
Everything Linux	everythinglinux.com.au	Rebel.com	www.rebel.com
Frank CDROM	www.frank-cdrom.co.at	Rock Linux	www.rocklinux.org/index.html
Greenbush Technologies	www.greenbush.com/linuxcd	rrbs Linux-Software, alpha-Systems and Network Solutions	www.rroehm.de/english/index.html
German Unix User Group	www.guug.de/linux.html	Schlittermann	debian.schlittermann.de
Hal91 Linux	home.sol.no/~okolaas/hal91.html	SGI Linux	rules.c3.hu/cgi-linux
Hypercore Software Design	www.hypercore.co.jp	Slackware Linux	www.slackware.com
IKARIOS	www.ikarios.com	Softcopy Systems	www.softcopy.onca
Indelible Blue	www.indelible-blue.com	SOFTEX	cypla.com/pc
Infomagic	www.infomagic.com	softlinux.com.br	www.softlinux.com.br
Investigacion y Desarrollo Agora	www.id-agora.com	SOT Finnish Software Engineering	www.sot.com
Iplabs	www.iplabs.ru	Step Computers & Communications	www.step.gr
IntraLinux	www.intralinux.com	Steve McIntyre	www.chiark.greenland.org.uk/~stevem/DebianCD
ixsoft Softwareentwicklung und-vertrieb Bernd Hentig	www.ixsoft.de	Stivell	www.stivell.com
Libra Computer Systems	www.libranet.com	Storm Linux	www.stormix.com
Lineo	www.lineo.com	Terra Soft Solutions	www.yellowdoglinux.com
Linpus Technologies	www.linpus.com.tw	Tomsrtbt	www.toms.net/rb
Linux Canada	www.linuxcanada.com	Tree.UK	tree.uk.com
Linux CDs	www.tienpiek.net	Trinux	www.trinux.org
Linux Central	linuxcentral.com	Tyse.Net	www.tyse.net
Linux Emporium	www.linuxemporium.co.uk	Virtuale Web	www.virtuale.com.br
Linux Mall	www.LinuxMall.com	Walnut Creek Linux	www.cdrom.com
Linux Press	www.linuxpress.com	WinLinux2000	www.wlinux.com
Linux Shop	finux.cjb.net	Yggdrasil Computing	www.yggdrasil.com
Linux Systems Labs	www.lsl.com		

Source: Al Gillen and Dan Kusnetzky, "Linux Overview: Understanding the Linux Market Model," IDC, February 2000.

⁴⁰ Gillen, Al, and Dan Kusnetzky, "Linux Overview: Understanding the Linux Market Model," IDC, February 2000.

Some conventional companies are trying to profit from the Linux trend by making some of their products available for Linux under a licensing agreement. Such vendors (and products) include: Oracle (Oracle 8I and the Oracle applications), Hewlett-Packard (OpenMail), IBM (DB2, VisualAge, MQ Series, TX Series), Lotus (Domino), Tivoli (TME-10), Transarc (AFS and DFS), Computer Associates (Unicenter TNG), Sybase (Adaptive Server), Informix (Dynamic Server), SAP (portions of R3), and WebTrends (WebTrends Enterprise Reporting Server).⁴¹ Netscape and Sun have also shown support for Linux.

IBM promotes itself as a competitive alternative to Solaris. Initiating a market strategy in support of Linux, IBM has developed close ties to the open source community and shown a willingness to invest in OSS initiatives and Linux distributors. The company has asserted that Linux will improve and mature for broad enterprise use by 2005.⁴²

3.1.3.3 Competitive Support Structure

Closed source software depends on monopoly support, one company that provides support and “holds all the cards” (i.e., access to the code) for a piece of software. This gives users the choice of either withstanding whatever support the original authors provide or switching to a different software. Since the cost of switching can be substantial, users are forced to accept monopoly support. In contrast, the publicly available source code for Linux and other open source products enables many vendors to learn the platform and provide support. Because vendors compete against one another to provide support, the quality of support increases while the end-user cost of receiving the support decreases. Open source can create support that lasts as long as there is demand, even if one support vendor goes out of business. Also, the support structure is self-scaling in that the more users that adapt OSS, the more users learn and are able to support each other. The peer review characteristic of open source products helps to ensure that an adequate base of maintenance developers are familiar with the package.

Linux support is available on the Internet for free as well as from companies providing consulting and support services. Examples of free support resources by vendor are included in the following table.⁴³

⁴¹ Gillen, Al, and Dan Kusnetzky, “Linux Overview: Understanding the Linux Market Model,” IDC, February 2000.

⁴² Weiss, G., “The Competitive Impact of IBM’s Linux Announcement,” GartnerGroup, February 8, 2000.

⁴³ Gillen, Al, and Dan Kusnetzky, “Linux Overview: Understanding the Linux Market Model,” IDC, February 2000.

Table 3. Free Support Resources for Linux by Vendor

Vendor	URL
LinuxHelp Online	www.linuxhelp.org
Linux Documentation Project	www.linuxdoc.org
Linux Support Services	free.linux-support.net
Red Hat Support Links	www.redhat.com/support/docs/tips/urls/urls.html
News groups (multiple)	comp.os.linux
News groups (multiple)	alt.os.linux

Source: Al Gillen and Dan Kusnetzky, "Linux Overview: Understanding the Linux Market Model," IDC, February 2000.

Red Hat, Caldera, SuSE, TurboLinux, and Linuxcare offer vendor support on an international basis, although some are more strong in particular regions than others. These vendors, however, may have a preference to offer support for their own Linux distribution. Hardware suppliers, such as Compaq, Dell, Hewlett-Packard, IBM, and SGI, offer primary and secondary Linux support for their desktop and server systems. In addition, hardware vendors sometimes have a support agreement with one or more Linux distributors to provide tertiary support. Mission Critical Linux, www.missioncriticallinux.com, spun off from Compaq to provide a nation-wide support option. VA Linux provides systems, hardware, support, and services to help customers in-depth with Linux software, and have a particular expertise in the Web business.

3.1.3.4 Influx of Start-up Companies

The number of start-up companies doubled between 1984 and 1994. Start-up companies represent an opportunity for Linux because they typically cannot afford a large in-house development staff. Linux direct costs are often less expensive than those for Microsoft.

3.1.3.5 Garnering Support

Linux represents the best chance for Unix to beat Microsoft. Anti-Microsoft sentiment is prevalent, particularly throughout the open source community. Open source, and Linux in particular, is often regarded as the heroic underdog. Linux has been touted as a "Windows killer."⁴⁴ Over 1,500 web sites, including the Microsoft Boycott Campaign at <http://msbc.simplenet.com>, advertise the anti-Microsoft movement. Unix vendors are rallying behind Linux to increase the Unix market share. According to Forrester Research, Linux continues to gather momentum as more vendors add product and service support.⁴⁵

⁴⁴ "The Future of Linux," CNet, 2000.

⁴⁵ Jordan, Peter, "Nibbling Away at UNIX," VARBusiness, January 14, 2000.

IBM Vice President of Internet Technology named Linux as one of the three major shifts he has seen in IBM's history (the other two technology shifts before Linux were PCs in 1989 and TCP/IP in 1991); all of these major shifts were initially dismissed by industry. The support of major hardware vendors, including IBM, Hewlett-Packard, Silicon Graphics, Compaq, and Dell, has been an enormous asset to Linux. Distributors and vendors have launched very successful marketing campaigns. Red Hat, in particular, has achieved worldwide brand name recognition. There is also a fad element to Linux. Computer professionals install Linux to keep themselves educated on the latest emerging technologies.

3.1.4 Threats

3.1.4.1 Risk of Fragmentation

Risk of fragmentation of the code base, or code forking, occurs when multiple, inconsistent versions of the project's code base evolve. This can occur when developers try to create alternative means for their code to play a more significant role than achieved in the base product. Sometimes fragmentation occurs for good reasons, like if the maintainer is doing a poor job, and sometimes it occurs for bad reasons, such as a personality conflict between lead developers. Commercial Unix implementations (e.g., SCO, Solaris, IRIX, HP-UX) are examples of fragmented code as well as open source BSD Unix versions (e.g., Open BSD, NetBSD, BSDI). However, the Linux kernel code has not yet forked, and this can be attributed to its accepted leadership structure, open membership and long-term contribution potential, GPL licensing eliminating the economic motivations for fragmentation, and the subsequent threat of a fragmented pool of developers. Ninety-nine percent of Linux distributed code is the same. The small amount of fragmentation between different Linux distributions is good because it allows them to cater to different segments. Users benefit by choosing a Linux distribution that best meets their needs.

The Linux Standard Base (LSB) project is working to standardize a subset of Linux across all distributions. This minimum standardization effort is attempting to find a balance between stifling Linux development and the possibility of fragmentation into several totally incompatible versions. It is expected to become available in the third quarter of 2000. More information on the LSB can be found at <http://www.linuxbase.org>.

3.1.4.2 Lack of Compatible Applications

The number of applications written to Linux is growing at a disproportionate rate compared to that of other mainstream operating systems. Because applications are so important to companies, many typically select the applications that meet their requirements and then select the operating system that best supports those applications. The more popular the application, the more users will already be trained for that application. Either Linux users need to be convinced that the software supports comparable applications to other operating systems (e.g., the StarOffice Suite can serve as an alternative to the Microsoft

Office Suite) or the required applications must be ported to Linux. Corel has released WordPerfect for Linux. Sun Microsystems has released the StarOffice Suite for Linux.

3.1.4.3 Need for Version Control

Version control can become an issue if the system requires integration and development. The developer must make sure that the versions to be integrated are compatible.

3.1.5 Other

3.1.5.1 Importance to Many

This can be considered both a strength and a weakness. The Linux community is self-scaling. The more users interested in Linux, the larger is the Linux community. More technical excellence is gained with a larger Linux community. This attribute is a strength when there is a very high level of interest, but a weakness when the level of interest is low.

3.1.5.2 Trained Staff

This can be considered both an opportunity and a threat. It is an opportunity because many recent college graduates are skilled with Linux, having used it as a learning tool in school. Researchers also commonly use Linux because of its wide availability. However, it is a weakness when the current skill base within an organization is not trained in Linux.

3.1.5.3 Competition

Competition can be both an opportunity and a threat. It is an opportunity during the acquisition process because Linux is potentially a second-source “bargaining chip” to improve COTS support. However, it is a threat because there is a risk of companies developing strategies specifically focused against Linux. The success of Linux has made it a competitor in the market. Microsoft has developed a formal competitive analysis and a strategy against Linux and the open source movement. Evidence of this has been seen in Microsoft internal strategy memorandum, referred to by the Linux community as the Halloween documents. The Halloween documents were obtained by the community and have subsequently been posted to the following URL: <http://opensource.org/halloween>.

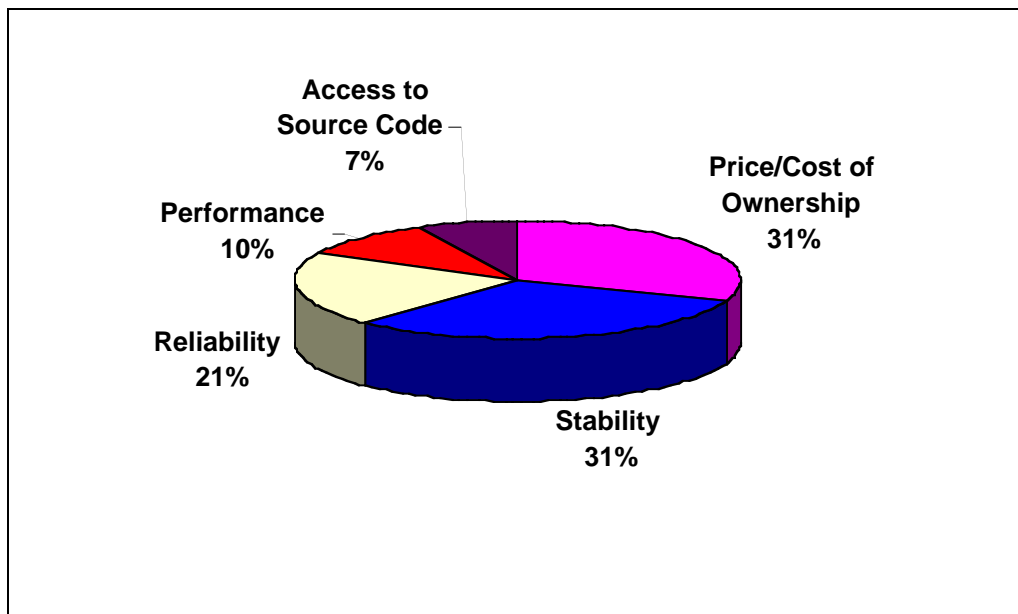
Several competitors to Linux exist on the market. Microsoft Windows NT and the recently released Microsoft Windows 2000 hold a strong share of the market. DOS is smaller and can be used for embedded systems. LynuxWorks, formerly Lynx, offers a proprietary, Unix-like system with very good real-time capabilities that can be used for embedded systems. LynuxWorks is making its software compatible with Linux. BeOS is a proprietary system with a friendly attitude to the open source movement. Within its niche market of multimedia, BeOS is gaining notoriety and easier to find people who like it. Sun Solaris is often considered the Unix hardware and operating system platform of choice in

Internet-related server deployments. However, Sun has recently received some criticism from the open source community for its community source license and the appearance of self-serving hype in the compatibility feature that enables Linux applications to run on Solaris.

3.2 Analysis of Strategic Factors

3.2.1 Market Viability

Most potential users become interested in Linux because of its price or cost of ownership and stability. A recent Datapro survey found that 31 percent of respondents chose Linux for these attributes. Other respondents named reliability (21 percent), performance (10 percent), and access to source code (7 percent) as motivations for their interest in Linux. These results are shown in the following pie chart.⁴⁶

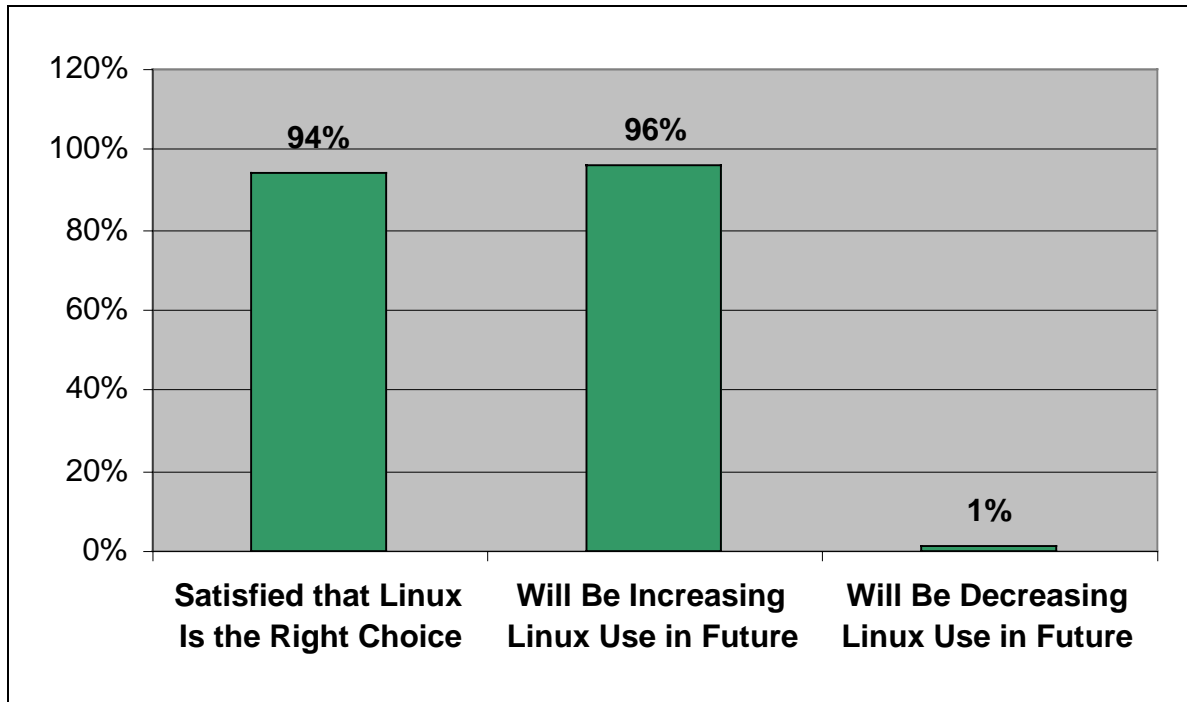


Source: Datapro, February 1999. Note that the nature of Web responses makes this a self-selecting sample; therefore, the results do not represent a scientific sample. A large number of respondents were from small organizations or in education, engineering, and software development.

Figure 5. Motivations for Linux Interest

⁴⁶ Datapro, February 1999. Note that the nature of Web responses makes this a self-selecting sample; therefore, the results do not represent a scientific sample. A large number of respondents were from small organizations or in education, engineering, and software development.

In this same Web-based user opinion survey, Datapro assessed users' satisfaction with Linux. Ninety-four percent of respondents replied that they are satisfied that Linux is the right choice. When the user was asked whether he would be increasing or decreasing his use of Linux in the future, 96 percent said increasing and only 1 percent said decreasing. These survey results are shown in the bar graph below.⁴⁷



Datapro, February 1999. Note that the nature of Web responses makes this a self-selecting sample; therefore, the results do not represent a scientific sample. A large number of respondents were from small organizations or in education, engineering, and software development.

Figure 6. Satisfaction with Linux

⁴⁷ Datapro, February 1999. Note that the nature of Web responses makes this a self-selecting sample; therefore, the results do not represent a scientific sample. A large number of respondents were from small organizations or in education, engineering, and software development.

Companies that are cautious about converting to Linux cite compatibility with existing systems as their number one concern, followed by lack of rapid development tools and technical support.⁴⁸

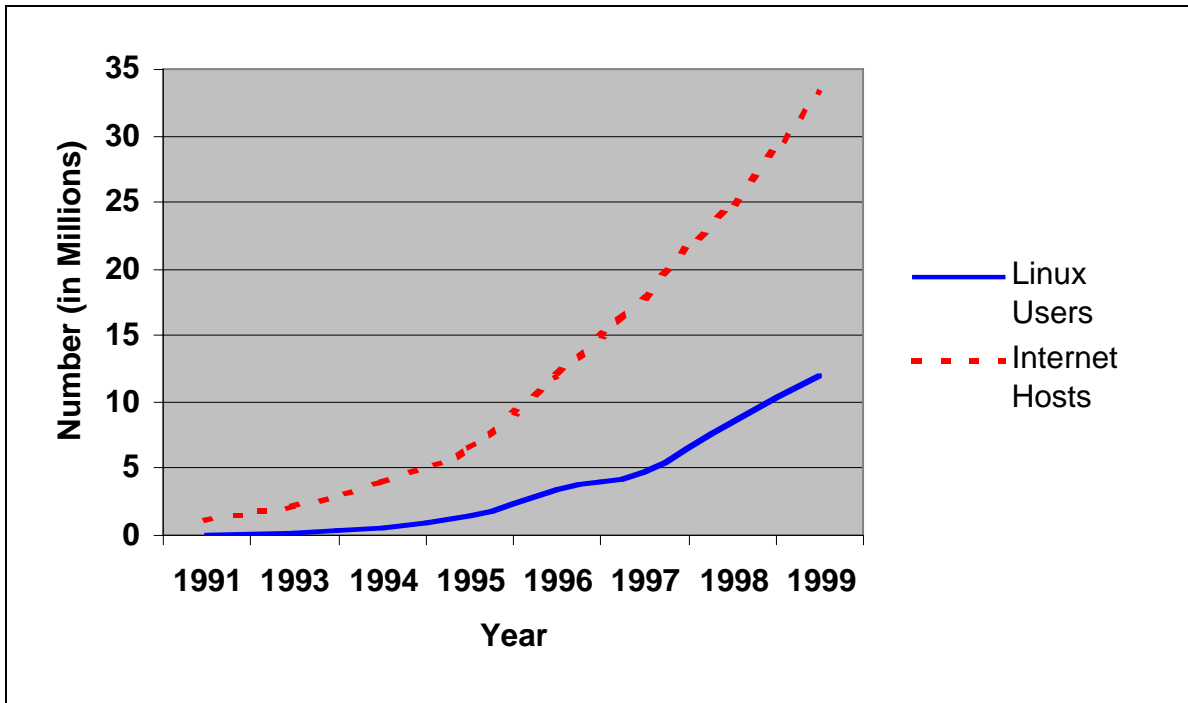
The exact market share of Linux is difficult to calculate because there are installations from anonymous FTP sites; commercial Linux purchases can be used to install multiple machines; there is a high likelihood of double counting since new versions of Linux are released often; and, there are not separate client and server distributions. Furthermore, historic estimates of the Linux market can differ depending on the methodology used to obtain those estimates. It is generally believed that there are between 4 and 27 million Linux users.⁴⁹

The research team compared the number of Linux users to the number of Internet hosts over the last nine years. The team independently derived Linux estimates based on several sources, including GartnerGroup, IDC, and Red Hat. These estimates show the number of worldwide Linux users growing from 1 user (Linus Torvalds) in 1991 to about 12 million users in 1999. The following graph plots the number of Linux users worldwide against the number of Internet hosts worldwide, and shows that the number of Linux users has been growing with the number of Internet hosts. The graph shows that as the Internet expanded, the number of Linux users also increased. This can be attributed to the Internet's effect on the open source development process; as the Internet expands, the number and productivity of open source development teams increase.⁵⁰

⁴⁸ Harmon, Paul, "Linux and Architecture," Cutter Consortium, February 9, 2000.

⁴⁹ Raymond, Eric, <http://www.opensource.org>, no date provided.

⁵⁰ Linux estimates derived from GartnerGroup, IDC, and Red Hat market research (no date provided). Internet estimates based on research from Bruce L. Egan, 1996. Data based on year-end estimates.

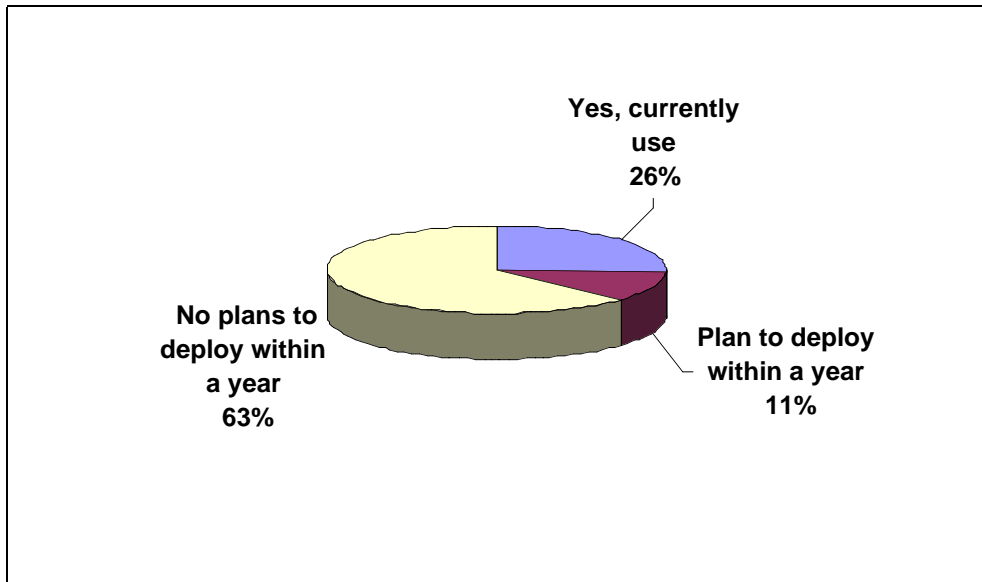


Source: Linux estimates derived from GartnerGroup, IDC, and Red Hat market research. Internet estimates based on research from Bruce L. Egan, 1996. Data based on year-end estimates.

Figure 7. Worldwide Success of Linux in the Marketplace

About 26 percent of companies currently use the Linux operating system, another 11 percent plan to deploy within a year, and 63 percent do not have plans to deploy within a year. These results were obtained from an InformationWeek survey of 300 IT managers, and the results are shown in the pie graph that follows.⁵¹

⁵¹ Ricadela, Aaron, "Linux Comes Alive," InformationWeek, January 24, 2000.



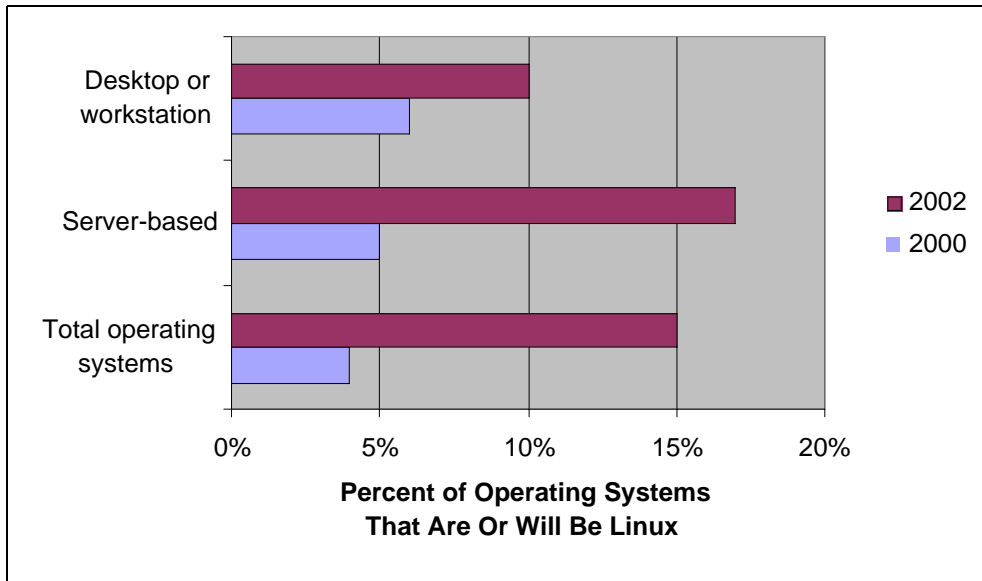
Source: Aaron Ricadela, "Linux Comes Alive," InformationWeek, January 24, 2000.

Figure 8. Percent of Companies that Use the Linux Operating System

Two-thirds of companies using the Linux platform have been doing so for a year or less, according to the InformationWeek survey.⁵² Although Linux deployments are widening, they are not deep. Respondents say that only 4 percent of their total operating system environments consist of Linux. That number is expected to rise to 15 percent by 2002, but will continue to fall short of Microsoft Windows. The following graph compares the percent of operating systems that are Linux in the year 2000 with those that are expected to be Linux in 2002.⁵³

⁵² Ricadela, Aaron, "Linux Comes Alive," InformationWeek, January 24, 2000.

⁵³ Ricadela, Aaron, "Linux Comes Alive," InformationWeek, January 24, 2000.

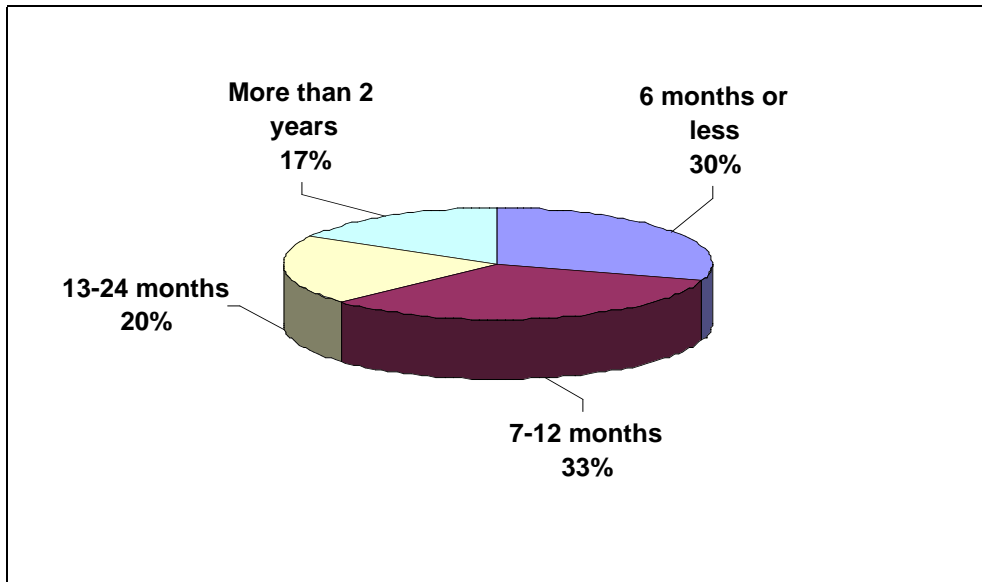


Source: Aaron Ricadela, "Linux Comes Alive," InformationWeek, January 24, 2000.

Figure 9. Percent of Operating Systems that Are Or Will Be Linux

Of those companies with Linux, most have only recently started using it. About 63 percent have been using Linux for one year or less. The following pie graph shows how long companies have been using Linux.⁵⁴

⁵⁴ Ricadela, Aaron, "Linux Comes Alive," InformationWeek, January 24, 2000.

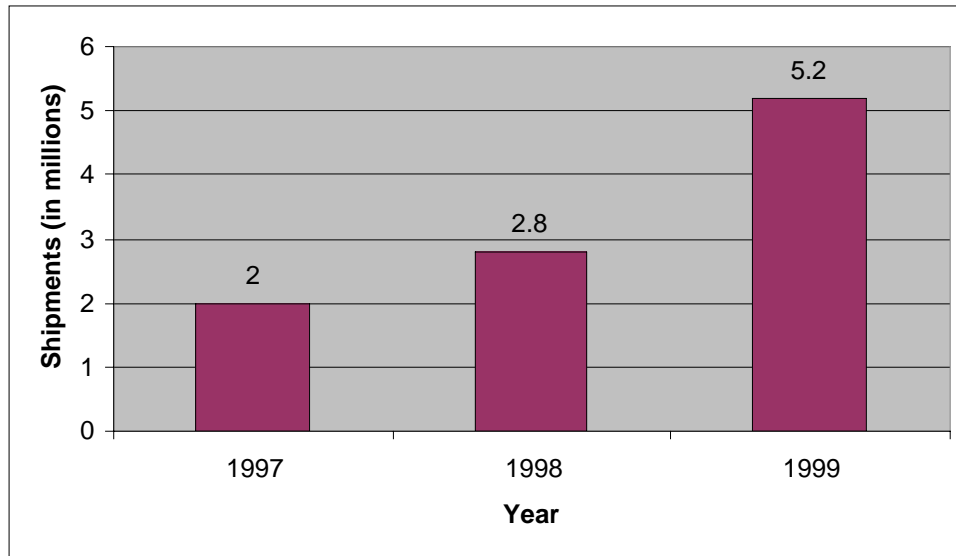


Source: Aaron Ricadela, "Linux Comes Alive," InformationWeek, January 24, 2000.

Figure 10. How Long Companies Have Been Using Linux

Worldwide new Linux shipments for client and server applications have been growing at an increasing rate over the past several years. IDC determined that shipments increased by 40 percent in 1998 to 2.8 million, compared to the previous year. In 1997, shipments increased by almost 86 percent to 5.2 million. The following bar graph shows the growing number of new Linux shipments for client and server applications worldwide.⁵⁵

⁵⁵ IDC, 2000.



Source: IDC, 2000.

Figure 11. Worldwide New Linux Shipments (Client and Server)

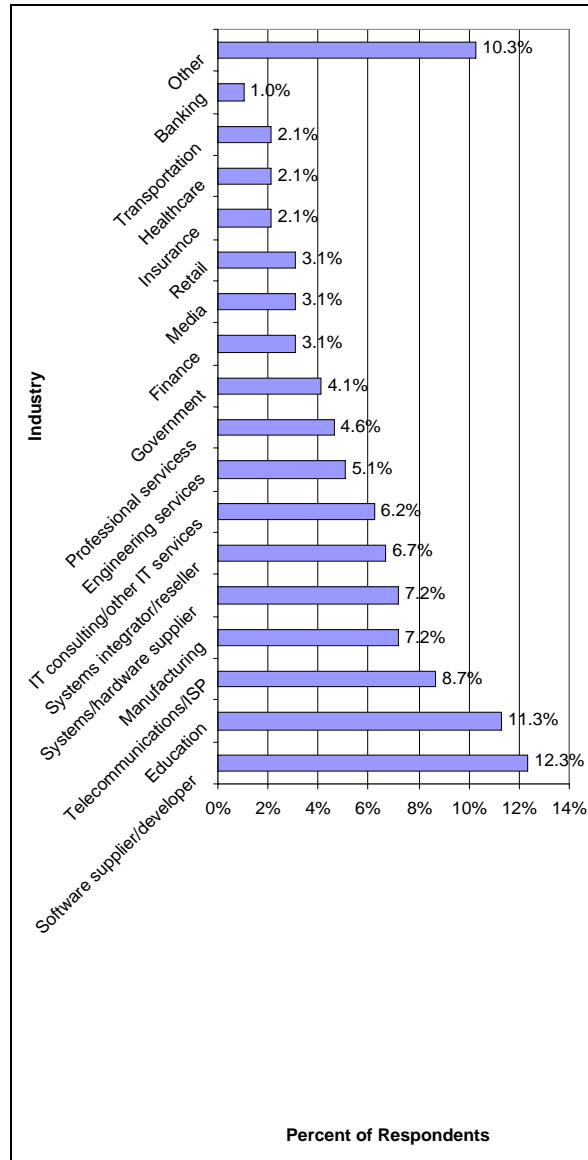
3.2.2 Market Segments

Given Linux's roots in research and educational organizations, the highest concentration of Linux users are found in the computer and IT industry (particularly ISPs and software developers) and educational institutions. ISPs are a core user base of Linux. Over 26 percent of ISPs use Linux, according to a survey from Infobeats.⁵⁶ ISPs must endure very small profit margins, and Linux's free price and wide hardware support is attractive. If something breaks, ISPs must fix the problem immediately; larger ISPs have the technical expertise to debug code breaks or install quickly available patches. Moreover, ISP system administrators commonly have a strong Unix background. Remote manageability, reliability, and scalability are also key features for ISPs.

Often highly technical users prefer Linux, and research and educational organizations like Linux's low procurement cost and ability to tinker without first needing negotiation and contractual agreements with the owner of the source code. However, Linux servers are also found across a wide range of industries and in all sizes of companies. Linux users will likely continue to broaden as the software gains more commercial appeal. The following chart

⁵⁶ "Business Use of Linux," <http://www.linuxbusiness.com/en/case.html>, no date provided.

shows the high proportionate number of Linux server users in the computer/IT and educational sectors.⁵⁷

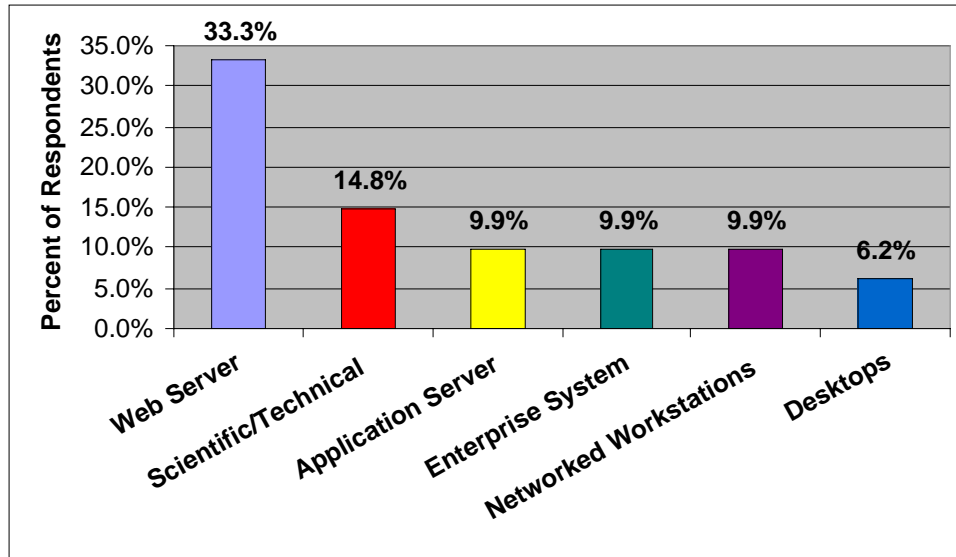


Source: IDC, 2000.

Figure 12. US Linux Server Sites by Industry, 1999

⁵⁷ IDC, 2000.

According to a recent Datapro survey, Linux is used most often for web servers (33.3 percent) and scientific/technical applications (14.8 percent). It is used less often for application servers (9.9 percent), enterprise systems (9.9 percent), and networked workstations (9.9 percent), and least frequently for desktop applications (6.2 percent). The following figure illustrates these responses.⁵⁸

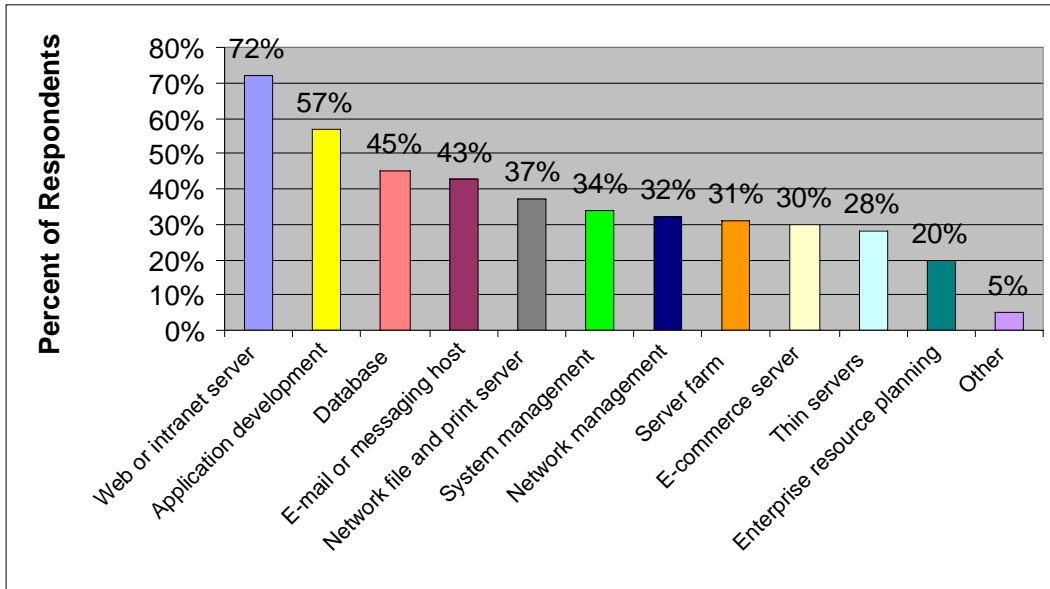


Source: Datapro, February 1999. Data was collected via a Web-based user opinion study on the Datapro home page. The nature of Web responses made this a self-selecting sample. Therefore, the results do not represent a survey or scientific sample. A large number of the respondents were from small organizations or in education, engineering, and software development. In the survey, a total of 1,841 individuals said their organizations use Linux.

Figure 13. Use of Linux (Datapro Survey)

⁵⁸ Datapro, February 1999. Data was collected via a Web-based user opinion study on the Datapro home page. The nature of Web responses made this a self-selecting sample. Therefore, the results do not represent a survey or scientific sample. A large number of the respondents were from small organizations or in education, engineering, and software development. In the survey, a total of 1,841 individuals said their organizations use Linux.

An InformationWeek survey of 300 IT managers similarly concluded that of the server-based applications, Linux is most often used for web or Intranet servers. This survey allowed for multiple responses, and the results are depicted in the following figure.⁵⁹



Source: Aaron Ricadela, “Linux Comes Alive,” InformationWeek, January 24, 2000.

Figure 14. Use of Linux in Server Based Applications (Information Week Survey)

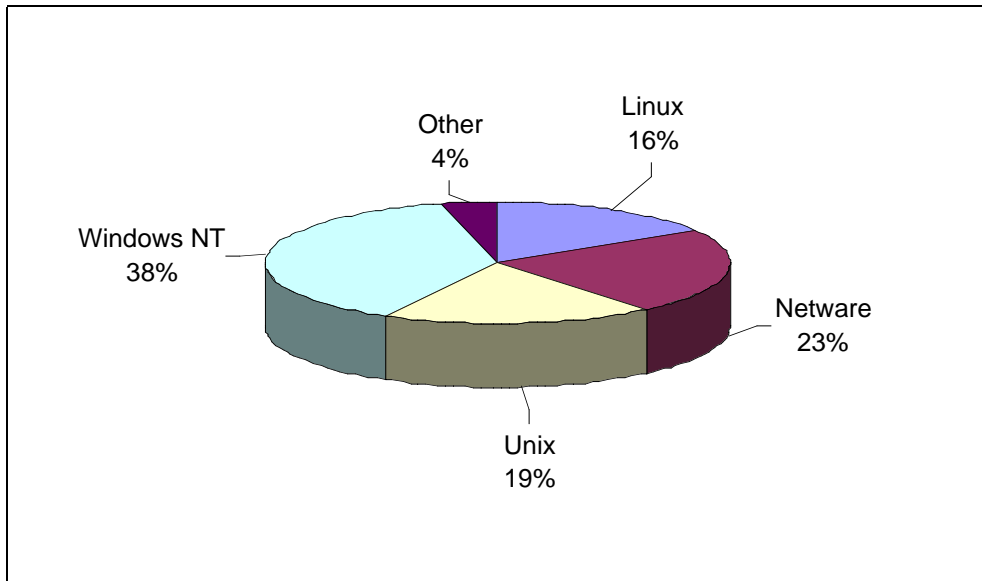
3.2.2.1 Servers Market

The majority of Linux installations is expected to be in servers. The server market is particularly strong for Linux for a couple of reasons. First, the server market, especially at the high-end, is already familiar and comfortable with Unix. Second, high-end server administrators tend to also be developers and are comfortable with the technical demands. Linux can be used for a wide range of purposes: Web, FTP, proxy, mail, or DNS servers; firewalls; or TCP/IP routers. A single server can also provide all of the above functionalities.

According to IDC research, Linux was the fastest growing server operating environment in 1998, increasing over 190 percent that year and capturing more than 15.8 percent of the

⁵⁹ Ricadela, Aaron, “Linux Comes Alive,” InformationWeek, January 24, 2000.

4.4 million revenue shipment server operating systems market segment.⁶⁰ Linux has increased its server OS market share from 16 percent in 1998 to 25 percent in 1999. Since 1998, Unix and NetWare have been losing ground to Linux. Windows NT market share has not yet been impacted by Linux growth. The following pie charts show the server OS market share for Linux, NetWare, Unix, Windows NT, and other in 1998 and 1999.⁶¹

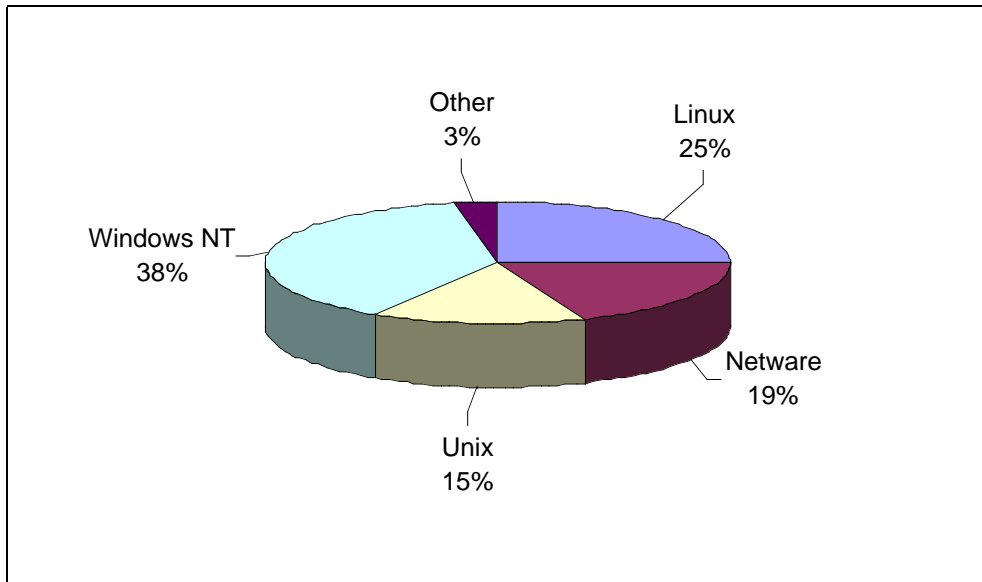


Source: "The Future of Linux," CNet, 2000 cites IDC.

Figure 15. Server OS Market Share in 1998

⁶⁰ RedHat, <http://www.redhat.com>, no date provided.

⁶¹ "The Future of Linux," CNet, 2000 cites IDC.

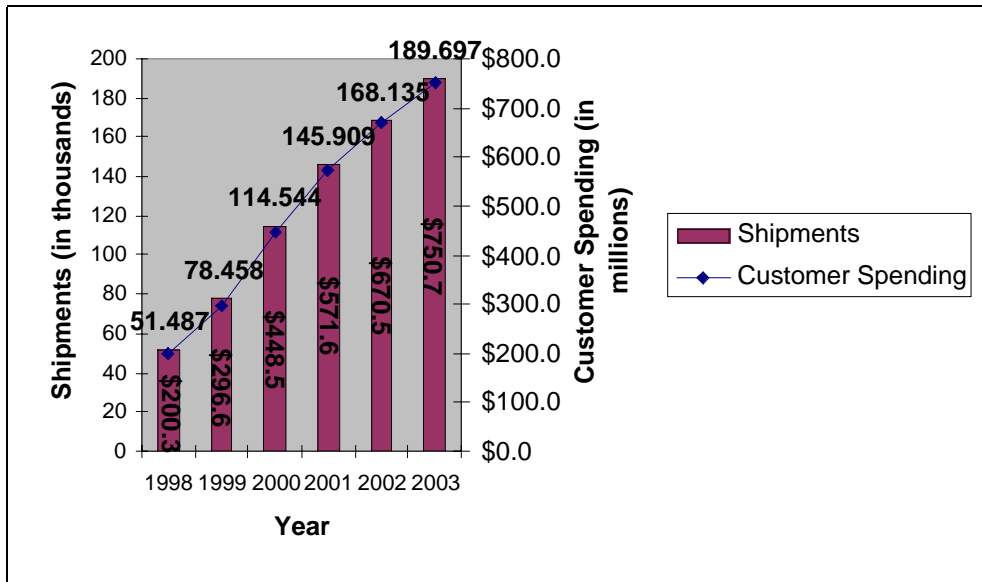


Source: "The Future of Linux," CNet, 2000 cites IDC.

Figure 16. Server OS Market Share in 1999

According to IDC estimates, customers in the US spent close to \$300 million on Linux servers in 1999. This revenue equals over 75,000 server shipments sold in the US in 1999. Included in these numbers are servers built by vendors and resellers as well as self-built systems. IDC estimates that these revenues and shipments represent just over 5 percent of the total available market for US entry-server shipments (server priced under \$100,000). The following figure graphs historic and expected Linux shipments and customer spending in the US from 1998 to 2003.⁶²

⁶² Bailey, Michelle, Vernon Turner, Jean Bozman, and Janet Waxman, "Linux Servers: What's the Hype, and What's the Reality?" IDC, March 2000.

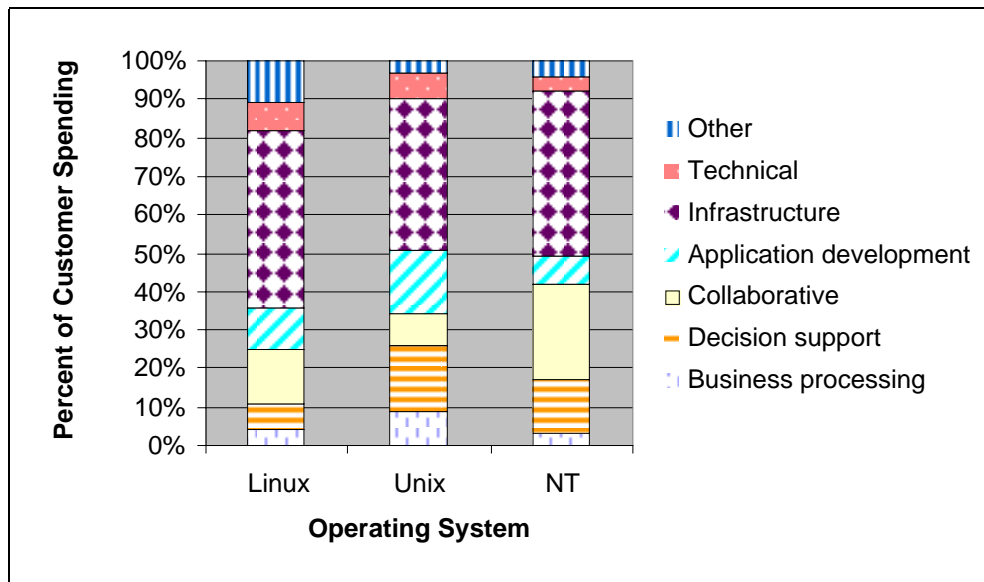


Source: Michelle Bailey, Vernon Turner, Jean Bozman, and Janet Waxman, "Linux Servers: What's the Hype, and What's the Reality?" IDC, March 2000.

Figure 17. US Linux Server Shipments and Customer Spending, 1998-2003

For its recent survey, IDC defined a workload topology of mutually exclusive workloads. All of the server workloads amounted to 100 percent of the total available server function. The topology was defined as follows: business processing (enterprise research planning, online transaction processing, and batch applications); decision support (data warehousing, data marts, data analysis, and data mining); collaborative computing (use of e-mail and workgroups); application development; infrastructure (file/print, networking, systems management, and proxy/caching/security); technical (real-time process control, numeric-intensive, and scientific applications); and, other (publications/information/education, video/audio on demand, and personal applications). Survey results showed that Linux servers are most often purchased to support infrastructure applications. Over 40 percent of all Linux server expenditures are used for file and print applications, networking functionality, systems management, and proxy, caching, and security services. Other important uses of Linux include collaborative applications such as e-mail (14.2 percent) and application development (10.9 percent). The following figure compares US server workload spending for Linux, Unix, and NT.⁶³

⁶³ IDC, 2000.



Source: IDC, 2000.

Figure 18. US Server Workload Spending by Operating System, 1999

GartnerGroup agrees that Linux is highly niche-focused with over 80 percent of deployments in web infrastructure-related applications at the present time, although it is gaining more acceptance as an enterprise system.⁶⁴ NetCraft claims that 29 percent of all public Web servers run on Linux, making Linux the most popular operating system for public Web sites.⁶⁵

Clustering micro-computers provides the power of a supercomputer at an inexpensive price. Either dedicated computer pools (e.g., Beowulf or Titanic) or an existing network during off-peak hours can be used.

Special purpose application servers are those pre-designed to meet the needs of specific type of applications, such as Internet sales or Web publishing, and are expected to grow in importance. For users with specific goals, these platforms offer high-end, exceptional performance. Special purpose application servers will target new ranges of applications in the future, including those specializing in e-commerce, content syndication, and personalization.

⁶⁴ Weiss, G., "Linux Adoption Best Practices: A 10-Point Program," GartnerGroup, February 8, 2000.

⁶⁵ RedHat, <http://www.redhat.com>, no date provided.

3.2.2.2 Desktop Market

Significant investments in software applications, ease of use, and configuration are needed for Linux to succeed in the desktop market. Most end-users do not choose their operating systems. Users typically buy their software applications based on their computing needs and then choose a computer that will run that software. Most software is developed for PCs or Macs, and most computers run Windows or the Mac operating system. Not much software is currently available for Linux. The Linux community is working to increase the amount of Linux-compatible software. The Wine Project is an effort to implement as open source the Microsoft Windows 95 and NT application programming interfaces (API) and enable them to run on Linux. Companies such as Corel are working to make applications like CorelDraw and WordPerfect for Windows run on Linux.

Another barrier to Linux's success in the desktop market is that it is not as user-friendly as Windows. Several initiatives are attempting to improve Linux's user-friendliness. Companies, including Corel and Caldera, are improving the ease of installation for their Linux distributions. However, commands still sometimes need to be typed, and most users prefer the point-and-click style of Windows and Mac operating systems. GNOME, KDE, and CDE desktop environments come bundled with basic productivity applications. Hewlett-Packard and Sun Microsystems have announced that they will begin using the GNOME desktop environment as their default Unix desktop interface. Corel and StarOffice provide full office suite products with some degree of file compatibility with Microsoft Office. A dominant desktop windowing package and more applications are also needed. The following table lists desktop application suites for Linux by vendor and product.⁶⁶

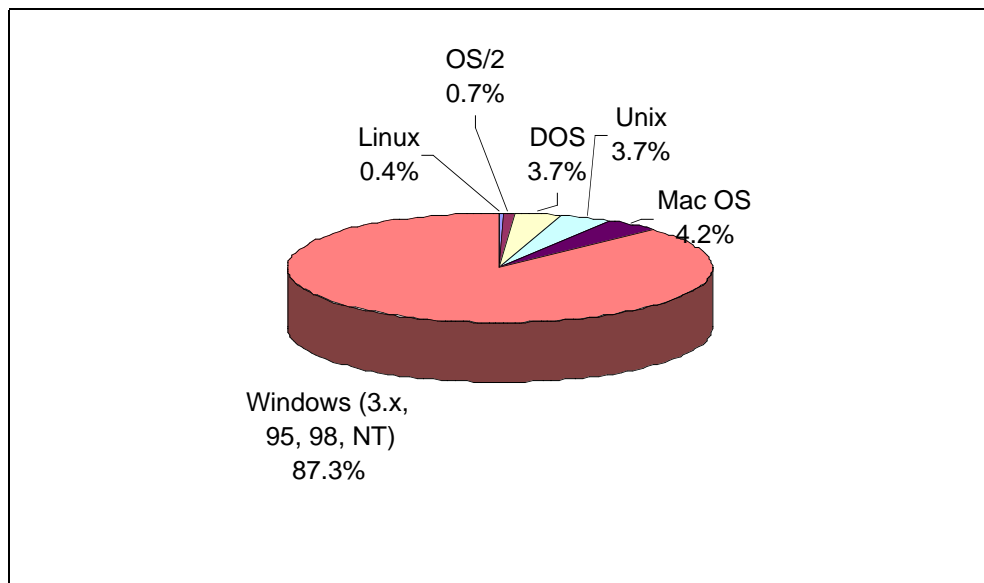
Table 4. Desktop Application Suites for Linux by Vendor and Product

Vendor and Product	URL
Applixware Applixware	www.applix.com/applixware
Corel WordPerfect	www.corel.com
Sun Microsystems StarOffice	www.sun.com/staroffice
SuSE Office Suite	www.suse.com

Source: Al Gillen and Dan Kusnetzky, "Linux Overview: Understanding the Linux Market Model," IDC, February 2000.

⁶⁶ Gillen, Al, and Dan Kusnetzky, "Linux Overview: Understanding the Linux Market Model," IDC, February 2000.

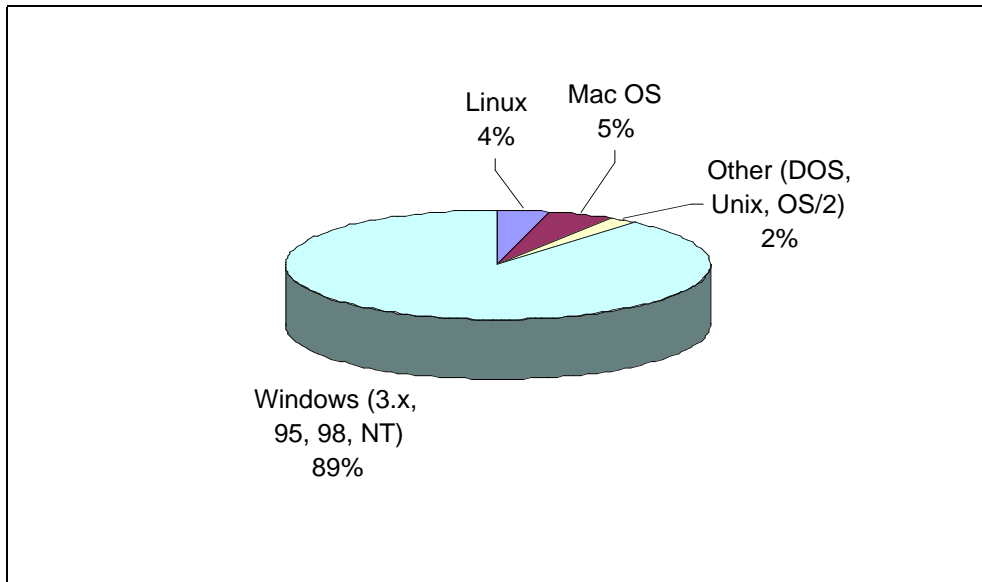
The following pie graphs illustrates the client OS market share in 1998 and 1999.⁶⁷



Source: "The Future of Linux," CNet, 2000 cites IDC data.

Figure 19. Client OS Market Share in 1998

⁶⁷ "The Future of Linux," CNet, 2000 cites IDC data.



Source: "The Future of Linux," CNet, 2000 cites IDC data.

Figure 20. Client OS Market Share in 1999

3.2.2.3 Embedded Devices

An embedded device is a piece of microprocessor-based computing hardware, usually on a single circuit board, which has been built to run a specific software application. The term *embedded* refers to the fact that these devices were originally used as building blocks in larger systems. A thin server is a computer that contains only enough hardware and software to support a particular function that users can share in a network, such as access to files on a storage device, access to CD-ROM drives, printing, or Internet access. The Linux operating system offers many advantages for thin and embedded servers. It is portable to many central processing units (CPUs) and hardware platforms, stable, scalable over a wide range of capabilities, and easy to use for development. Linux software can dynamically reconfigure itself without rebooting. It can isolate faults and processes. Processes can load and remove kernel modules, device drivers, and custom modules based on available resources and dynamic application needs. The applications are also modular with well-defined interfaces.

Embedded devices sometimes have a real-time requirement. Real-time is a relative term, and for some systems, near real-time response in the five to fifty millisecond time range is sufficient. Others are required to respond in a deterministic manner and within one microsecond. These real-time systems either respond to specific events, or process a

constant stream of information.⁶⁸ There is a real-time extension for Linux, called RTLinux. More information on RTLinux can be found at www.rtlinux.org.

There is a large and growing market for embedded devices, including information appliances and mobile devices. Commercial applications include point-of-sale terminals, digital jukeboxes, car stereos, gas-pump credit card verifiers, medical equipment, set-top boxes, personal digital assistants (PDAs), washing machines, and hotel room locks. The embedded computer market absorbs over 95 percent of all microcomputer chips minted each year.⁶⁹ The market for embedded software development tools is expected to grow at a compound annual growth rate of 25 percent or more over the short term, according to GartnerGroup.⁷⁰ Small, lightweight, inexpensive computers using embedded operating systems are expected to fill a void for those users not currently online, but interested in accessing the Internet. The market for Internet appliance users may be larger than today's entire PC user base, according to Jupiter Communications. Household penetration of Internet appliances is projected to reach 37.3 million by 2002.⁷¹

There are several software choices for embedded systems, including DOS, Microsoft Windows, and Linux. Although there are more than 100 million DOS users worldwide,⁷² DOS has well-known limitations in embedded systems. Microsoft Windows lacks a real-time capability. In the desktop and server market, new devices emerge every three months or less, and finding device drivers for traditional COTS products can be difficult.⁷³ To support the hardware chosen by their customers, traditional real-time operating system vendors typically either charge up to \$30,000 in consulting fees and retain the driver code or use Linux drivers from the open source community.

A subset of the Linux kernel, specifically the Embeddable Linux Kernel Subset (ELKS), can run in embedded devices, and the Linux operating system is real-time. Because Linux is scaleable and flexible, it can handle a wide range of embedded system functions. The user can choose only those components in Linux that are needed for the application. Linux will

⁶⁸ IT-Director.com, <http://www.it-director.com/ts/linux/embedded.html>, no date provided.

⁶⁹ Embedded Linux Consortium, <http://www.embedded-linux.org/pressroom.php3#1>, no date provided.

⁷⁰ GartnerGroup, 1998.

⁷¹ Jupiter Communications, "Internet Appliances," cited by Be, Inc., <http://www.be.com>, no date provided.

⁷² "PC DOS," IBM, <http://www-3.ibm.com/software/os/dos/>, no date provided.

⁷³ Ready, Jim, and Bill Weinberg, "Leveraging Linux for Embedded Applications," LinuxDevices.com, no date provided.

become more real-time and deterministic in performance through kernel substitution strategies and through enhancements to the standard kernel. Linux device driver code appears regularly with or prior to the release of new drivers.⁷⁴

Margins are very low in embedded devices. The free cost of Linux helps this market. Using open source for embedded systems avoids the licensing fee from closed source vendors, which amounts to a large cost savings for manufacturers producing large volumes of embedded systems. Developers of embedded systems can compress Linux and tweak it to suit the needs of their specific applications. Developers worldwide can cooperatively enhance the software and fix bugs real-time.

Linux was not originally designed for embedded systems, but has been adopted to them. Linux was designed in a non-integrated, componentized manner from the outset, so it is easy to build boxes that do not have a monitor, keyboard, etc. Embedded developers are reassured that their system can adopt new changes and fixes immediately because the Linux source code is constantly being upgraded.

There is a risk of fragmentation in the embedded Linux market. Over 100 commercial real-time operating systems (RTOSs) exist because there is a very wide range of embedded devices (from cell phones to refrigerators).⁷⁵ To scale down Linux for these devices and use the least amount of hardware as possible, different pieces of code are taken out of the common GNU/Linux code base and different device-specific extensions are added to optimize performance. In attempt to avoid fragmentation, Cygnus Solutions developed the EL/IX Application Programming Interface, a “compatibility layer,” or hardware abstraction layer, for standard Linux to drive different devices.

Embedded Linux Consortium (ELC) is a vendor-neutral, non-profit trade association dedicated to promoting and advancing the Linux operating system throughout the embedded community. More information on ELC is available at www.embedded-linux.org.

Major players in the embedded market, such as Motorola and Intel, are backing the use of Linux. Vendors offering Linux for embedded systems include Lineo/Caldera (Embedix), MontaVista (Hard Hat Linux), Cygnus (eCos), and Be (BeOS, Stinger). LynuxWorks offers products and support services for embedded software development under the Linux operating system. It is planning on offering specialized training programs and the industry’s firsts long-term support program for embedded Linux. LynuxWorks embraces the open source

⁷⁴ Ready, Jim, and Bill Weinberg, “Leveraging Linux for Embedded Applications,” LinuxDevices.com, no date provided.

⁷⁵ Matthews, John, “Cygnus to Save Embedded Linux from Fragmentation?” *Linux Today*, September 30, 1999, wysiwyg://35/http://linuxtoday.com/stories/10601.html.

model of Linux to offer customers a solution optimized for embedded applications. More information on embedded Linux can be found at www.linuxdevices.com.

3.3 Evaluating Feasibility of Business Opportunity

When comparing long-term economic costs and benefits of open source usage and maintenance to traditional COTS, the winner varies according to each specific use and set of circumstances. Costs and benefits are influenced by the platform environment, operating needs, and mission objectives. Within their specific environments and parameters, users should choose the most effective software option, the one that optimizes the benefits and minimizes the costs. The literature cites diverse quantitative data and qualitative opinions on the total cost of ownership and benefits for OSS compared to traditional COTS alternatives. Some of these opinions follow.

- GartnerGroup estimates that the open source business model in the Unix industry will result in a 20 percent savings in the IT server budget as compared with the traditional license model from 2001.⁷⁶
- GartnerGroup expects the total cost of ownership argument on behalf of Linux to disappear. Unix platform vendors such as IBM and Sun already offer their Unix operating systems at virtually no charge.⁷⁷
- The total cost of running Linux is about the same as NT, Unix, or anything else, according to Computerworld.⁷⁸
- An independently managed company with high personnel turnover can experience a cost of computer ownership that is as much as 50 percent higher than more stable industry peers because of the increased cost of training and reconfiguration of systems.⁷⁹
- Jack Bryar concludes that a total cost of Linux ownership argument is there for the making. Now someone in the Linux community has to step up and make it.⁸⁰
- For various tests, Linux showed roughly equal to 15 percent better, according to German technical computer magazine c't. Overall, Linux beat NT, but not in every circumstance

⁷⁶ Weiss, G., "How the Open Source Movement Will Affect Users," GartnerGroup, January 26, 1999.

⁷⁷ Weiss, G., "The GartnerGroup Server Operating System Forecast," GartnerGroup, March 26, 2000.

⁷⁸ Hayes, Frank, "The Secret of Linux," Computerworld, March 1999.

⁷⁹ Bryar, Jack, "How Much Does Free Cost?" The Andover News Network, March 15, 2000.

⁸⁰ Bryar, Jack, "How Much Does Free Cost?" The Andover News Network, March 15, 2000.

(in one test, NT was clearly superior), and there was definitely no overwhelming superiority for either system.⁸¹

- Bloor Research concludes that the overall winner is Linux, although the difference with NT is often small. Linux comes out on top for file, print, Web, or mail servers, as well as for mixed workloads. In a database server environment, there is little or no difference between Linux and NT. Windows NT is better for application servers because there is so much more software available for this platform.⁸²
- According to the Mindcraft benchmarking survey, sponsored by Microsoft, Windows NT is up to 2.7 times faster than Linux.⁸³ However, Mindcraft has been criticized for optimizing NT, but not Linux, for maximum performance.
- Microsoft Windows file server operates about 200 percent more efficiently under an open source operating system than under Microsoft's NT operating system.⁸⁴
- GartnerGroup, KPMG, and Forrester Research all agree that hardware and software, together, comprise less than 20 percent of the total cost of corporate computer ownership.⁸⁵
- A report by Compaq indicates that less than 4 percent of IT purchases are subject to even a preliminary total cost of ownership analysis.⁸⁶

Typically, open source compares favorably in many circumstances for server and embedded system implementations which may require some customization, but fares no better than COTS for typical desktop applications. Indeed, some literature sources generalize that open source products are no worse than closed source, but our findings indicate that the scale measuring the value derived from open versus closed source software can be heavily tipped in one direction or the other depending on the specific requirements and runtime environment of the software.

⁸¹ Schmidt, Jürgen, "Mixed Double," c't (German technical computer magazine), 1999.

⁸² Godden, Frans, "How do Linux and Windows NT Measure Up in Real Life?" GNet, January 2000.

⁸³ Weiner, Bruce, Mindcraft, "Open Benchmark: Windows NT Server 4.0 and Linux," Mindcraft, June 30, 1999.

⁸⁴ Seiferth, C. Justin, Major, Deputy Chief, Global Air Traffic Operations Division, US Air Force, "Adoption of Open Licensing," COTS Journal, November/December 1999.

⁸⁵ Bryar, Jack, "How Much Does Free Cost?" The Andover News Network, March 15, 2000.

⁸⁶ Bryar, Jack, "How Much Does Free Cost?" The Andover News Network, March 15, 2000.

An open versus closed source decision is based on three factors: (1) costs – both direct (e.g., price of software) and indirect (e.g., end-user downtime); (2) benefits (i.e., performance); and, (3) other, more intangible criteria (e.g., quality of peer support). Direct costs are largely understood and have traditionally comprised most of the total lifecycle costs of a system. However, indirect costs as well as operational and performance benefits (e.g., scalability, reliability, and functionality) play a most influential economic role in today's more mature software market. Other, more intangible criteria are difficult to quantify, but can also affect the effectiveness of open and closed source software. Traditional lifecycle cost models and other COTS software tools, therefore, can no longer be relied on for optimal mission-oriented and IT investment decision-making.

GartnerGroup, the leading vendor of IT ownership cost tools, addresses in their literature the fact that their total cost of ownership tools are not well-suited for evaluating business software application development and maintenance costs. GartnerGroup further admits that their tools are not sufficiently customizable to address software alternatives. The FY99 Army MOIE research *Integrating Total Ownership Cost Methods with IT Investment Strategies* confirmed that any work breakdown structure will need to be customized for the specifics of a customer's environment and proposed initiatives. Moreover, existing IT ownership cost tools do not consider differences in benefits between the software choices in question. Therefore, Program Managers need a complete taxonomy of costs, benefits, and other, more intangible criteria to account for hidden costs and benefits that they might otherwise have overlooked. With this taxonomy, Program Managers can make software-purchasing decisions being fully aware of their life-cycle economic, performance, and mission implications. The following table represents the MITRE-developed Cost Element Taxonomy for OSS and Linux.

Table 5. Cost Element Taxonomy for OSS and Linux⁸⁷

Direct Costs	
Software and Hardware	
Software	Purchase price
	Upgrades and additions
	Intellectual property/licensing fees
Hardware	Purchase price
	Upgrades and additions
Support Costs	
Internal	Installation and set-up
	Maintenance
	Troubleshooting
	Support tools (e.g., books, publications)
External	Installation and set-up
	Maintenance
	Troubleshooting
Staffing Costs	
	Project management
	Systems engineering/development
	Systems administration
	Vendor management
	Other administration
	Purchasing
	Other
	Training
De-installation and Disposal	
Indirect Costs	
Support Costs	
	Peer support
	Casual learning
	Formal training
	Application development
	Futz factor
Downtime	

⁸⁷ Futz factor is included by GartnerGroup as an indirect cost. GartnerGroup describes this term as the labor expense when the end-user exploits corporate computing assets for his own personal use during productive work hours.

3.3.1 Direct Costs

3.3.1.1 Software and Hardware

3.3.1.1.1 Software

Linux can be downloaded for free over the Internet or purchased from a vendor for a nominal price of about \$60. There is no licensing fee. The cost of Microsoft Windows NT is approximately \$600-\$800 for five users and \$35 for each additional user. Additional features in Microsoft, such as telnet, news server, better DNS server, and disk quotas can run about \$3,800; these features are included in Linux at no extra charge. Unix runs from \$1,000 to \$5,000 or \$15,000 for unlimited user licenses.⁸⁸

3.3.1.1.2 Hardware

Because Linux has relatively few lines of code and is highly modular, it can run on less powerful computers than can other operating systems. Users commonly run Linux on older, otherwise unusable, computers (e.g., 486 PCs) and pay nothing for hardware. For companies that have used Unix, it is often less expensive for them to switch to Linux than to NT.

3.3.1.2 Support

3.3.1.2.1 Internal Support

Annual labor costs for help desk support are about \$30,000.⁸⁹ With open source code, it is possible for problems to be fixed internally by the user's organization. For closed source, proprietary software, problems must be fixed by the external supplier or vendor.

3.3.1.2.2 External Support

External support for Linux costs about \$60-\$85 per incident or \$3,000 for a block of ten incidents. There are also enterprise support packages available for Linux for about \$60,000 a year. Microsoft NT support costs \$200 per incident or \$1,700 for a block of ten incidents.⁹⁰

⁸⁸ D. H. Brown Associates, "Linux: How Good Is It?" 1999; John Kirch, "Microsoft Windows NT Server 4.0 Versus UNIX," August 7, 1999; and Robert Lauriston, "The Un-Microsoft Office," ComputerCurrents.com, March 23, 1999.

⁸⁹ Labor costs based on data from Medzilla; Wageweb; and Chim-Net, no date provided.

⁹⁰ "Join the Freeware Revolution?" CIO, March 19, 1999 and Robert Lauriston, "The Un-Microsoft Office," ComputerCurrents.com, March 23, 1999.

3.3.1.3 Staffing

Findings from literature reviews and interviews with experts have indicated that there often is not an identifiable difference in annual labor costs per employee between OSS and traditional COTS. However, depending on the specific project and structure of the organization, the quantity of labor required could differ between OSS and traditional COTS. We recommend that staffing costs be evaluated on a case-by-case basis.

3.3.1.3.1 Project Management

Annual labor costs for a Project Manager can run approximately \$60,000.⁹¹

3.3.1.3.2 Systems Engineering/Development

Labor costs for a Systems Engineer are about \$45,000-\$90,000 per year.⁹²

3.3.1.3.3 Systems Administration

Annual labor costs for a Systems Administrator are about \$30,000-\$65,000. Per hour rates range from \$30 to \$100.⁹³

3.3.1.3.4 Other Administration

Annual labor costs for other administrative services are approximately \$21,000 to \$45,000 per person.⁹⁴

3.3.1.3.5 Training

Training courses cost about \$2,000 for a four-day course and \$2,500 for a five-day course. For Systems Administrators, Unix professionals often take one course; professionals transitioning from other operating systems typically take four courses. Developers often take one to two classes.⁹⁵

⁹¹ Labor costs based on data from Medzilla; Wageweb; and Chim-Net, no date provided.

⁹² Labor costs based on data from Medzilla; Wageweb; and Chim-Net, no date provided.

⁹³ Labor costs based on data from Jeff Covey, "A New Business Plan for Free Software," Freshmeat, January 22, 2000; Medzilla; Wageweb; and Chim-Net, no date provided.

⁹⁴ Labor costs based on data from Wageweb, no date provided.

⁹⁵ Red Hat, <http://www.redhat.com>, no date provided.

3.3.1.4 De-installation and Disposal

It is relatively easy to de-install and dispose the Linux software. However, there may be integration costs involved to make new software compatible with the system.

3.3.2 Indirect Costs

Indirect costs include “hidden” influences and causal inter-relationships that may be difficult to capture. Since they are potentially significant, they are important to identify and consider. These indirect costs can be measured in terms of lost productivity attributed to the computing environment. While the salary and other labor costs associated with an employee are captured under the direct cost category, the indirect costs represent labor costs that are “wasted” and could be used in more productive ways. In other words, although there is no additional direct cost to the organization, not as much output was received from the employee due to inefficiencies in the process or system.

3.3.2.1 Support Costs

Indirect support costs vary according to the specific use and environment of the software. These costs could differ between OSS and traditional COTS. Therefore, indirect support costs should be evaluated on a case-by-case basis.

3.3.2.1.1 Peer Support

Peer support indirect costs include labor expenses for end users supporting each other in lieu of obtaining technical support from service desk or IS personnel. A case study showed that tech support handled informally by peers in the workplace rather than IT professionals saved money for the IT department, but added up to 27 percent to overall administrative costs.⁹⁶

3.3.2.1.2 Casual Learning

This includes labor expenses of end users training and supporting themselves in lieu of formal training and support programs.

3.3.2.1.3 Formal Training

This indirect cost includes all of the course time spent by end-computing users on computer system and application training. Training courses typically last four to five days. This time should be considered an expense since it requires the attendee to forego their direct work responsibilities. Travel costs may also be involved depending on where the course is offered.

⁹⁶ Bryar, Jack, “How Much Does Free Cost?” The Andover News Network, March 15, 2000.

3.3.2.1.4 Application Development

Application development involves labor expenses of end users performing development and customization of non-business/mission critical applications.

3.3.2.1.5 Futz Factor

Futz factor is included by GartnerGroup as an indirect cost. This is the labor expense when the end-user exploits corporate computing assets for their own personal use during productive work hours.

3.3.2.2 Downtime

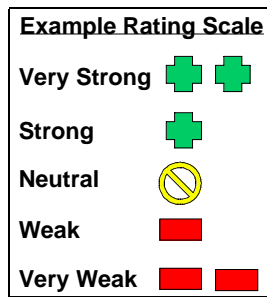
Downtime involves losses in productivity due to the unavailability of the desktop computer, servers, applications, or other tools. Cost is measured as lost wages. It can be calculated as planned and/or unplanned downtime hours times percent of productivity impact to users when downtime occurs times end user burdened salary. According to most benchmarking studies, Windows users experience greater downtime than Linux users. More information on relative uptime/downtime can be found in Section 3.3.3.2 .

3.3.3 Benefits and Risks

The following table presents the MITRE-developed taxonomy of benefits and risks for OSS and Linux as well as an example rating scale.

Table 6. OSS and Linux Taxonomy of Benefits and Risks

Qualitative Attributes
Ability to customize
Availability/reliability
Interoperability
Scalability
Design flexibility
Lifetime
Performance
Quality of service and support
Security
Level of difficulty/ease of management
Risk of fragmentation
Availability of applications



Program Managers can use this taxonomy as a basis for comparing costs, benefits, and other, more intangible criteria of OSS and traditional COTS software. The above taxonomy comprises a list of qualitative attributes. For each attribute, Program Managers should compare the relative strength or weakness for OSS versus traditional COTS products. A relative strength would indicate a benefit, and a relative weakness would indicate a risk. An example rating scale is shown above for comparing the relative value of OSS versus traditional COTS. This example scale presents five ratings – very strong, strong, neutral, weak, and very weak. Since the ratings will differ depending on the specific use and environment of the software, Program Managers should customize their ratings according to their particular circumstances. A discussion of Linux benefits and risks follows.

3.3.3.1 Ability to Customize

Because the source code is open, Linux can be tailored to meet an organization’s needs. Organizations can contribute useful ideas and expand on existing functionality to provide a totally new feature or system. Linux’s strict application, OS componentization, and readily exposed internals make it the preferred choice for customized tasks. It is often preferred over Windows NT for isolated, single task servers, such as DNS, File, Mail, and Web. Customization will grow increasingly important over time as the number of servers and their dedication to specific tasks will increase. The modularity of Linux enables it to be used in a wide range of systems, from a supercomputer to a refrigerator.

3.3.3.2 Availability/Reliability

This is the amount of time a system is up and running. It is a primary objective of the Linux community and one of the greatest weaknesses of Windows. Since so many programmers work to improve the Linux code, bugs are more likely to be discovered and fixed to improve the software’s stability. Also, the Linux kernel uses a virtual memory management system that shares memory across all active programs. It gives each program a separate virtual address space, reducing the effect of one program on another. This management system also prevents programs from overwriting critical areas of memory (i.e., areas where Linux kernel is stored). The computer usually must be restarted when Windows

NT incurs reconfiguration or software loading problems; this is usually not necessary for Linux. Benchmarking studies agree that Linux is more reliable than Windows. The Bloor Research benchmarking study measured the uptime/downtime of Linux and Windows NT over the period of one year. Over that time, Linux crashed once because of hardware fault (disk problems), and it took four hours to fix. Windows NT crashed 68 times due to hardware problems, memory, file management, and a number of miscellaneous problems, all of which took 65 hours to fix. Thus, the availability of Linux was 99.95 percent and the availability of NT 99.26 percent. In a similar benchmarking study, Giga Information Group determined the availability of Unix as 99.8 percent and the availability of Windows NT as 99.2 percent.⁹⁷

3.3.3.3 Interoperability

Every open protocol that exists has been ported to Linux. Samba enables Linux to look like NT. Linux and Microsoft Windows NT are both operable on various hardware platforms.

3.3.3.4 Scalability

Scalability is the software's ability to add users to a system while maintaining performance. Several benchmarking studies compared the scalability achieved by operating system. The German magazine *c't* showed Linux achieved better scalability. CIO Magazine reported that Unix can support 2,500 concurrent users. Microsoft NT can only support up to 850 according to CIO Magazine or 1,000 users according to GartnerGroup.⁹⁸ Frans Godden specified that Linux is better for breadth of scalability because it can run on high-end servers as well as small machines and handheld computers, whereas Microsoft must change to a different size operating system. Bloor Research indicated that the scalability performance of Linux and NT was almost equal at the low-end. The Mindcraft benchmarking survey, sponsored by Microsoft, showed that NT was more scalable than Linux.⁹⁹

⁹⁷ DiDio, Laura, cited by Derek Slater, "Deciding Factors - Operating Systems," CIO Magazine, February 1, 2000 and Frans Godden, "How do Linux and Windows NT Measure Up in Real Life?" GNet, January 2000.

⁹⁸ Hohmann, Deate, GartnerGroup, phone conversation, December 2000.

⁹⁹ Bittman, Thomas, GartnerGroup cited by Derek Slater, "Deciding Factors - Operating Systems," CIO Magazine, February 1, 2000 and Frans Godden, "How do Linux and Windows NT Measure Up in Real Life?" GNet, January 2000.

3.3.3.5 Design Flexibility

The Linux kernel can be pared down to eliminate unnecessary features or expanded to include additional features.

3.3.3.6 Lifetime

The lifetime of open source licensed systems can be extended indefinitely since the source code and documentation are freely available. Information requiring long-term access will not become buried in obsolete or non-functional, undocumented, proprietary formats every few years. With open source licensing, the user can develop the software in-house, outsource to the original vendor, or outsource to an aftermarket support vendor.

3.3.3.7 Performance

Performance is the ability to use computer resources (e.g., processors, memory, and disk) efficiently. On a typical single-processor PC, Linux will perform better than a Windows family operating system, since Windows systems tend to use up far more of the processor, memory, and disk resources that would otherwise be available for use by application programs.

3.3.3.8 Quality of Service and Support

As covered in the SWOT analysis, the competitive service and support structure of OSS enables Linux vendors to offer higher quality support and a lower price than traditional COTS vendors.

3.3.3.9 Security

For security audits, consultants prefer Linux over Sun Solaris by a ratio of 50:1, according to Network Associates.¹⁰⁰ An ad hoc working group comprised of DARPA, GSA, NIST, and NSA concluded that the use of OSS can have both positive and negative effects on the security of federal systems.¹⁰¹

The security benefits of OSS include:

¹⁰⁰ Clark, Tim, "Network Associates Adds Linux Product," CNET News, February 8, 1999.

¹⁰¹ "Report on Open Source Code and the Security of Federal Systems," DARPA, GSA, NIST, and NSA, prepared for the National Coordinator for Security, Infrastructure Protection, and Counter-Terrorism, June 1999.

- An opportunity for interested parties to apply static analysis tools to detect the presence of malicious code or undocumented features. Automated tools can be applied to reduce the effort involved in looking for vulnerabilities.
- Flaws and bugs found in the software can be quickly removed through the creation and distribution of software patches.
- Since the source code is widely available, “white box” testing methods can be performed by interested parties other than just by the developer.
- The wider availability of source code makes it more likely that negative or unexpected consequences of component modifications on the rest of the system will be uncovered. This is due to the size and diversity of interested parties able to assess the impacts.
- Individual user organizations can modify the source code to meet their own specific needs. For example, the government could develop tailored versions of the OSS by incorporating extra security features into the standard release to produce a government-approved version.

However, OSS often lacks key security features that are needed to protect critical information and processing. Whether or not the system is open source, poorly configured and managed operating systems are generally insecure.

3.3.3.10 Level of Difficulty/Ease of Management

As discussed in the SWOT analysis, many find Linux less user-friendly and, therefore, more difficult to use than traditional COTS products. However, compared to Unix and Windows NT, Linux is the easiest to manage because it is more centralized and enables features such as remote management, disk quota support, remote security, and diskless booting; with Linux a network administrator is not needed at every site. NT is most difficult to manage and particularly not good with remote management.

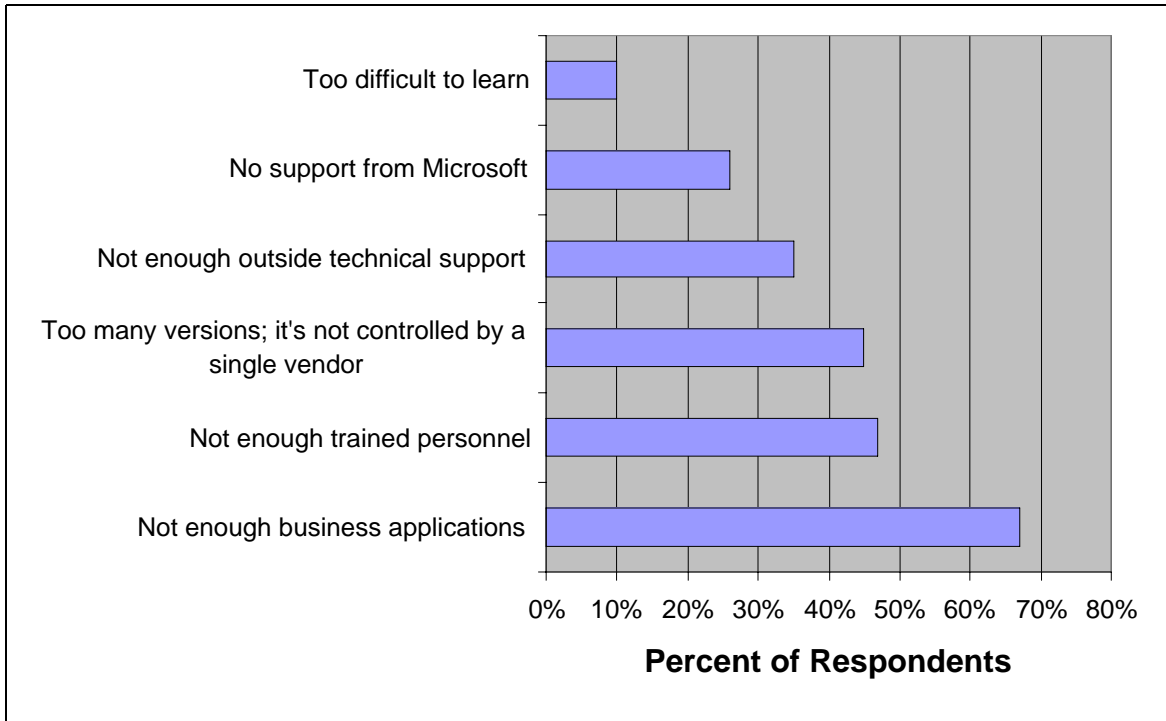
3.3.3.11 Risk of Fragmentation

As noted in the SWOT assessment, the Linux kernel code has not yet forked. However, OSS products face a greater risk of fragmentation of the code base than do traditional COTS products. A few incompatibilities exist among the many unique Linux distributions that are available. This fragmentation can cause in-fighting among vendors, as they strive to dominate the market. However, fragmentation enables vendors to address different markets. Linux can be used in a wide range of systems, and the software’s modularity lends itself to varying versions.

3.3.3.12 Availability of Applications

As described in the SWOT analysis, the number of applications written to Linux is growing at a disproportionate rate compared to that of other mainstream operating systems.

The following graph depicts the most significant weakness of Linux, according to the InformationWeek survey.¹⁰²



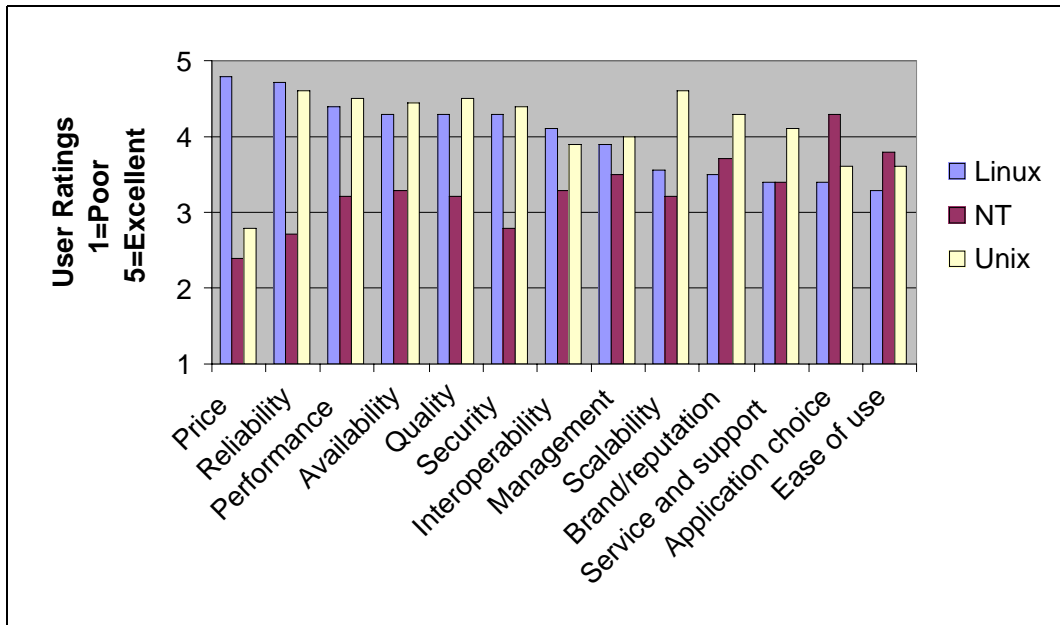
Source: Aaron Ricadela, "Linux Comes Alive," InformationWeek, January 24, 2000.

Figure 21. Most Significant Weaknesses of Linux

Linux users in the US were asked in the IDC survey to rate the server quality of Linux, NT, and Unix based on several performance metrics. These metrics included price, reliability, performance, availability, quality, security, interoperability, management, scalability, brand/reputation, service and support, application choice, and ease of use. Users rated the price of Linux servers much better than the price of NT or Unix servers. Users further rate Linux servers higher for quality compared to NT on the most important server

¹⁰² Ricadela, Aaron, "Linux Comes Alive," InformationWeek, January 24, 2000.

characteristics: performance, availability, quality, security, interoperability, and management. Unix is rated comparable to Linux on these important server characteristics. Yet Unix servers rate higher than Linux servers for scalability, brand/reputation, and service and support. These higher ratings for Unix can be attributed to its more mature platform. The following graph illustrates the performance ratings for server quality by operating system.¹⁰³



Source: Michelle Bailey, Vernon Turner, Jean Bozman, and Janet Waxman, “Linux Servers: What’s the Hype, and What’s the Reality,” IDC, March 2000.

Figure 22. US Linux Users’ Ratings for Server Quality by Operating System

Many of these findings are consistent with a recent GartnerGroup evaluation of Linux, NT, RISC Unix, and Windows 2000 operating systems. GartnerGroup compares current functionality (in 1999 or 2000) ratings with projections for year 2003. The evaluation concludes that Linux is making technical progress, but will continue to face challenges at the high end as Unix and Windows 2000 advance. By 2003, Linux will attain an acceptable rating for most categories, but few excellent ratings. High marks go to Linux for stability, clustering, security, and pricing. NT gets excellent marks for plug-and-play drivers and

¹⁰³ Bailey, Michelle, Vernon Turner, Jean Bozman, and Janet Waxman, “Linux Servers: What’s the Hype, and What’s the Reality,” IDC, March 2000.

independent service vendor/value added reseller (ISV/VAR) support. Unix is praised for its stability, symmetric multiprocessing (SMP) scaling, clustering, high availability, relational database management system (RDBMS) size, technical support, ISV/VAR support, system management, and security. Windows 2000 excels at SMP scaling, plug-and-play drivers, technical support, ISV/VAR support, and system management. These ratings are displayed in the table that follows.¹⁰⁴

Table 7. Comparison of Operating Systems

	Linux 99/03	NT 99	Unix 99/03	Win 2000 00/03
Stability	++/++	-	++/++	-/+
SMP scaling	-/+	+	++/++	+/++
Clustering	+/++	-	+/++	-/+
High availability	-/+	-	+/++	-/+
RDBMS size	-/+	+	+/++	+/+
Ease of use	-/-	+	-/-	+/+
Plug-and-play drivers	-/+	++	-/-	++/++
Technical support	+/+	+	++/++	+/++
ISV/VAR support	-/-	++	++/++	+/++
System management	-/+	+	+/++	+/++
Security	+/++	-	+/++	-/+
Pricing	++/+	+	-/+	+/+
<p style="text-align: center;"> ++ Excellent + Acceptable - Deficient </p>				

Source: G. Weiss, "Updated OS Evaluation: Linux vs. Unix and Windows 2000," GartnerGroup, July 25, 2000.

¹⁰⁴ Weiss, G., "Updated OS Evaluation: Linux vs. Unix and Windows 2000," GartnerGroup, July 25, 2000.

List of References

1. Bailey, Michelle, Vernon Turner, Jean Bozman, and Janet Waxman, "Linux Servers: What's the Hype, and What's the Reality?" IDC, March 2000.
2. Be, www.be.com.
3. Bryar, Jack, "How Much Does Free Cost?" The Andover News Network, March 15, 2000.
4. Caldera, www.caldera.com.
5. Clark, Tim, "Network Associates Adds Linux Product," CNET News, February 8, 1999.
6. Chime-Net; Medzilla, 1999; and Wageweb, 2000.
7. CoolLogic, www.coollogic.com.
8. Corel, www.corel.com.
9. Covey, Jeff, "A New Business Plan for Free Software," Freshmeat, January 22, 2000.
10. Datapro, February 1999.
11. Debian, www.debian.org.
12. D. H. Brown Associates, "Linux: How Good Is It?" 1999.
13. Embedded Linux Consortium, www.embedded-linux.org.
14. Epplin, Jerry, "Linux as an Embedded Operating System," October 1997, www.embedded.com/97/fe39710.htm.
15. "French Ministry Adopts Open-Source Culture, Linux," InfoWorld.com, February 8, 2000, <http://www.infoworld.com/articles/ec/xml/00/02/08/000208eclinparis.xml>.
16. "The Future of Linux," CNet, 2000.
17. "Getting to Know Linux," Colorado Business, July 2000.
18. Gillen, Al, and Dan Kusnetzky, "Linux Overview: Understanding the Linux Market Model," IDC, February 2000.
19. Gutfraind, Alexander, "Introductory into the World of Linux," The Linux World, <http://www.tht.net/~gutfrnd/linux/intro/linworld.htm>, 1998.
20. Harmon, Paul, "Linux and Architecture," Cutter Consortium, Feb. 9, 2000.

21. Hohmann, Deate, GartnerGroup, phone conversation, December 2000.
22. Hontañón, Ramón J., UUNET, "Building a Robust Linux Security Solution, Network Magazine, 2000.
23. "Join the Freeware Revolution?" CIO, March 19, 1999.
24. Jordan, Peter, "Nibbling Away at UNIX," VARBusiness, January 14, 2000.
25. Kaminsky, Dan, "Core Competencies: Why Open Source is the Optimum Economic Paradigm for Software," March 2, 1999.
26. Kirch, John, "Microsoft Windows NT Server 4.0 Versus UNIX," August 7, 1999.
27. Lauriston, Robert, "The Un-Microsoft Office," ComputerCurrents.com, March 23, 1999.
28. Lerner, Josh and Jean Tirole, "The Simple Economics of Open Source," National Bureau of Economic Research, March 2000.
29. LinuxDevices, <http://www.linuxdevices.com>.
30. "Linux Is Biggest Focus Shift Since TCP/IP, Says IBM," Network News, August 30, 2000.
31. "National Security Agency Selects Secure Computing to Provide Type Enforcement on Linux OS," January 14, 2000.
32. "Open Source Code and the Security of Federal Systems," Report of the Ad Hoc Working Group, DARPA, GSA, NIST, and NSA, prepared for the National Coordinator for Security, Infrastructure Protection, and Counter-Terrorism, June 1999.
33. O'Reilly, Tim and Ether Dyson, "Open Mind, Open Source."
34. O'Reilly, Tim, Linux eSeminar Series, 1999.
35. Orzech, Dan, "Linux and the Saga of Open Source Software," Datamation, February 1999.
36. "PC DOS," IBM, <http://www-3.ibm.com/software/os/dos/>.
37. Quinlan, Daniel, "The Past and Future of Linux Standards," Linux Journal, June 1999.
38. Raymond, Eric, "The Cathedral and the Bazaar," O'Reilly Associates, 1999.
39. Raymond, Eric, <http://www.opensource.org>.
40. Ready, Jim and Bill Weinberg, "Leveraging Linux for Embedded Applications," LinuxDevices.com.
41. RedHat, <http://www.redhat.com>.

42. Ricadela, Aaron, "Linux Comes Alive," InformationWeek, January 24, 2000.
43. Schmidt, Jürgen, "Mixed Double," c't (German technical computer magazine), 1999.
44. Secure Computing Corporation, "Reconsidering Assurance for Open Source Software," November 30, 1999.
45. Seiferth, C. Justin, Major US Air Force, Deputy Chief, Global Air Traffic Operations Division, "Opening the Military to Open Source," COTS Journal, November/December 1999.
46. Seiferth, C. Justin, Major, US Air Force, Deputy Chief, Global Air Traffic Operations Division, "Adoption of Open Licensing," COTS Journal, November/December 1999.
47. Seiferth, C. Justin, Major, US Air Force, "Open Source and the United States," Air Command and Staff College, Air University, June 11, 1999, http://ceu.fi.udc.es/GPUL/articulos/varios/US_DoD_and_OSS.txt.
48. Slackware, www.slackware.com.
49. Slater, Derek, "Deciding Factors - Operating Systems," CIO Magazine, February 1, 2000.
50. Stoltz, Mitch, "The Case for Government Promotion of Open Source Software," A NetAction White Paper, 1999.
51. SuSE, www.suse.com.
52. United States Air Force Scientific Advisory Board, "Ensuring Successful Implementation of Commercial Items in Air Force Systems," SAB-TR-99-03, April 2000.
53. Valloppillil, Vinod, Microsoft, "Open Source Software," edited by Eric Raymond as Halloween I, 1998.
54. Valloppillil, Vinod, and Josh Cohen, Microsoft, "Open Source Software," edited by Eric Raymond as Halloween II, 1998.
55. Vaughn-Nichols, Steven J., "TripWire Delivers Open Source DDoS and Security Answer," Sm@rt Reseller, March 1, 2000.
56. Weiner, Bruce, Mindcraft, "Open Benchmark: Windows NT Server 4.0 and Linux," Mindcraft, June 30, 1999.
57. Weiss, G., "The Competitive Impact of IBM's Linux Announcement," GartnerGroup, February 8, 2000.
58. Weiss, G., "The GartnerGroup Server Operating System Forecast," GartnerGroup, March 26, 2000.

59. Weiss, G., "How the Open Source Movement Will Affect Users," GartnerGroup, January 26, 1999.
60. Weiss, G., "Linux Adoption Best Practices: A 10-Point Program," GartnerGroup, February 8, 2000.
61. Weiss, G., "Updated OS Evaluation: Linux vs. Unix and Windows 2000," GartnerGroup, July 25, 2000.
62. White, Walker, "Observations, Considerations, and Directions," Oracle, May 8, 2000 cites Frederick Brooks in "The Mythical Man Month."
63. Williams, Tom, "Linux Catches the Embedded Wave," February 2000, <http://www.embeddedsystems.com>.

Glossary

API	Application Programming Interface
B	Billion
C2	Command and Control
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
DNS	Domain Name Server/Service
DOD	Department of Defense
ELC	Embedded Linux Consortium
ELKS	Embeddable Linux Kernel Subset
FTP	File Transfer Protocol
FUD	Fear, Uncertainty, Doubt
GNU	Gnu's Not Unix
GPL	General Public License
GUI	Graphical User Interface
IS	Information System
ISP	Internet Service Provider
ISV/VAR	Independent Service Vendor/Value Added Reseller
IT	Information Technology
LSB	Linux Standard Base
M	Million
MOIE	Mission-Oriented Investigation and Experimentation
OS	Operating System
OSI	Open Source Initiative
OSS	Open Source Software

PC	Personal Computer
PDA	Personal Digital Assistant
Perl	Practical Extraction and Reporting Language
POSIX	Portable Operating System Interface
R&D	Research and Development
RDBMS	Relational Database Management System
RTOS	Real-Time Operating System
RPM	Red Hat Package Manager
SMP	Symmetric Multiprocessing
SWOT	Strengths, Weaknesses, Opportunities, and Threats
TCP/IP	Transmission Control Protocol/Internet Protocol
US	United States

Distribution List

Internal

D480

B. N. Obando

F063

D. C. Hite

G040

G. L. Hollis

J. P. Root

W010

S. D. Huffman

W030

C. C. Howell

W032

T. B. Bollinger

W110

J. W. Moore

W118

D. E. Emery

M. A. Macpherson

W800

R. C. Evans

P. R. Garvey

R. Haller

S. E. MacReynolds

W802

R. T. Arace

S. A. Castro

J. M. Deems

J. C. Ellenbogen

T. L. Hoffman-Boswell

M. A. Janiga

B. J. Jasper

P. A. Kelley

M. W. Kilgore

H. W. Loomis

J. E. Manring

W. B. Reading

T. J. Restivo

B. M. Rolfe

D. A. Smith

R. C. Tepel

J. R. Valaika

D. L. White

W803

D. A. Crawford

F. M. Dello Russo

W803 (Continued)

R. J. Giallombardo
A. M. Goldberg
E. S. Goyette
C. A. Kenwood (10)
R. A. Moynihan
D. H. Plummer
W. F. Schaefer (2)
A. E. Taub
J. A. Vitkevich

External

Program Manager, Abrams Tank System
Attn: Kevin Houser
6501 E. 11 Mile Rd.
Building 229
Warren, MI 48397-5000

R. C. Seay
Department Chief of Staff for Programs
Attn: DAPR-FDT
700 Army Pentagon
Washington DC 20310-0700

Stan Levine, DOI
C/o R. C. Seay
Department Chief of Staff for Programs
Attn: DAPR-FDT
700 Army Pentagon
Washington DC 20310-0700

External (Continued)

Program Manager, Abrams Tank System
Attn: LTC Robert Lovett
6501 E. 11 Mile Rd.
Building 229
Warren, MI 48397-5000

Barry Mareiro
ESC/ACW,
50 Griffiss Street (2M-104)
Hanscom AFB, MA 01731-1625

Frank McPherson
1415 Andover Ct.
Evans, GA 30809

Colonel (P) James Moran, Commandant
Defense Systems Management College
9820 Belvoir Rd., Suite G3
Ft. Belvoir, VA 22060-5565

