TEI™
The language of value

The Total Economic Impact™ of Developing and Deploying Applications on Microsoft and J2EE/Linux Platforms

Project Analysts:
John R. Rymer, Vice President
Bob Cormier, Senior Consulting Advisor — TEI

September 4, 2003

Forrester Research, Inc.
400 Technology Square
Cambridge, MA 02139
Tel: (617) 613-6000
Fax: (617) 613-5000

Table of Contents

Executive Summary

Will Linux inevitably dethrone Windows as the leading operating system? Is Linux the better choice? The press has reported that Linux has generated substantial cost savings when compared to Microsoft's Windows, and that it will revolutionize the industry. Microsoft commissioned Giga Research to examine the relative benefits of Linux and Windows by comparing the costs incurred and benefits achieved by two sets of organizations: those using Linux as the basis for their applications and those using Microsoft Windows. This report presents the results of the study.

The primary conclusion of the study is that Microsoft offers a substantial cost advantage over J2EE/Linux as a development platform *for the applications considered*. Our interviews with organizations using Linux quickly indicated that J2EE was their development and deployment platform of choice. As such, a J2EE/Linux environment has been used as the basis for comparing a Linux to Windows environment in this study.

Based on the study findings, the primary sources of Microsoft's cost advantages are:

1. The J2EE application server and Unix-based database software used in the Linux development and deployment stack drive up product costs and development complexity relative to the comparable Microsoft products.
2. Microsoft's tools simplify development of applications like those profiled in the study when compared to the J2EE/Linux products in the study. This simplification translates into lower labor costs for development.

The findings in this study are based on interviews conducted with seven organizations that use the Microsoft .NET-generation platform to develop and deploy custom applications within their enterprises, and five organizations that use Linux. The analysis extrapolates from these users' experiences to create two composite organizations — a large enterprise and a medium-size enterprise — that are developing and deploying custom applications using either J2EE/Linux or the Microsoft platform.

The comparison of the two platforms shows large to medium-size organizations that develop, deploy, support and maintain custom applications on the Microsoft .NET platform can expect to experience 25 percent to 28 percent less cost during a four-year life cycle than if the J2EE/Linux platform was used.

The following are the findings based on the model of the sample large-size enterprise (see Table 1):

- For J2EE/Linux, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $2,289,041.
- For Microsoft, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $1,643,112.
- Giga found that for the large sample organization, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $645,929 less using the Microsoft platform. Microsoft's total costs were 28.2 percent less than the total costs for J2EE/Linux. The primary driver of this difference is a shorter time to deployment for Microsoft — nine months vs. 12 months for J2EE/Linux.

The following are the findings based on the model of the sample medium-size enterprise (see Table 2):

- For J2EE/Linux, the total costs associated with the initial development and deployment, plus three years of support and maintenance were $881,455.
- For Microsoft, the total costs associated with the initial development and deployment, plus three years of support and maintenance were $661,012.

- Giga found that for the medium-size sample organization, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $220,443 less using the Microsoft platform. Microsoft's total costs were 25 percent less than the total costs for J2EE/Linux.

Each of the models is based on an application scenario that was common among the interviewees. Both are portal applications. Both composites assume the same application development scenarios to allow for comparability. The report presents both the financial and the non-financial factors evident in choices made between Linux/J2EE and the Microsoft platform by the interviewees.

Among the interviewees, Microsoft's development tools were shown to be highly productive for the target applications. In addition, Microsoft's software prices were shown to be generally lower than those of application servers, databases and related products associated with the J2EE/Linux platform.

It is only when the low prices of Linux are put into a larger IT context that their true impact on IT costs becomes evident. In a head-to-head comparison, the list price for Red Hat 9 (the Linux version chosen for this study) is lower than the price of Microsoft Windows Server 2003. However, the key cost factor in the study's Linux cases was the J2EE environment, not the operating system. Although the cost of Linux is low, the impact of that lower cost on the overall cost economics of an application development project is small. The full development and deployment environment and the labor associated with the development project are the biggest costs. Comparisons of individual elements within the stack of software products required to build and deploy a complete application tell only part of the story and can be misleading.

Despite Microsoft's economic advantages over J2EE/Linux for the applications studied, the study also indicates that many organizations will adopt Linux instead of Microsoft's alternative. The interviewees using Linux, in general, saw migration from Unix/RISC to Linux/Intel as their best opportunity to reduce costs while retaining their investments in Unix skills.

This study employs Giga's Total Economic Impact™ (TEI) methodology, which measures the costs that are typically accounted for by IT as well as benefits, risks and flexibility. This study used TEI as a common business language to capture and communicate the financial and non-financial aspects of developing, deploying and supporting applications using the Microsoft and J2EE/Linux platforms.

The TEI of Developing and Deploying Applications on the Microsoft and J2EE/Linux Platforms

Introduction

In May 2003, Giga commenced work on a research project commissioned by Microsoft to provide a TEI business case comparison for developing and deploying custom business applications on Windows and J2EE/Linux platforms.

This report analyzes the costs of developing, deploying, maintaining, and supporting custom business applications for medium and large-size organizations (see Appendix A: Description of Sample Organizations). The findings in this study are based on interviews conducted with seven organizations currently using the Microsoft .NET-generation platform to develop and deploy custom applications within their enterprises and five organizations currently using J2EE/Linux. These organizations provided Giga with significant details on their actual costs, selection criteria and business goals in constructing applications.

The report extrapolates from these users' experiences to create two composite organizations that are developing and deploying custom applications using either J2EE/Linux or the Microsoft platform. Both composites assume the same application development scenarios to ensure comparability of the results. The report also presents the non-financial factors behind the choice between Linux/J2EE and Microsoft evident in the customers studied.

Microsoft selected Giga for this project because of Giga's TEI analysis methodology, which not only measures costs (areas that are typically accounted for within IT) but also weighs the enabling value of a technology in increasing the effectiveness of overall business processes. Giga employed four fundamental elements of TEI (see Appendix B: Total Economic Impact Primer) in comparing the costs of Microsoft and J2EE/Linux:

1. Cost
2. Benefits
3. Flexibility
4. Risk

Key Findings

Giga's research shows large to medium-size organizations that develop, deploy, support and maintain custom applications on the Microsoft .NET platform experience 25 percent to 28 percent less cost during a four-year life cycle than if the J2EE/Linux platform was used.

The following are the findings based on the model of the sample large-size enterprise (see Table 1):

- For J2EE/Linux, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $2,289,041.
- For Microsoft, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $1,643,112.
- Giga found that for the large sample organization, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $645,929 less using the Microsoft platform. Microsoft's total costs were 28.2 percent less than the total costs for J2EE/Linux. The primary driver of this difference is a shorter time to deployment for Microsoft — nine months vs. 12 months for J2EE/Linux.

Table 1: Financial Results — Large-Size Sample Organization

Cost Category	J2EE/Linux Platform	Percentage of Total J2EE/Linux Costs	Microsoft Platform	Percentage of Total Microsoft Costs
Computing Platform (servers, operating system licensing and maintenance)	$55,674	2.4%	$111,676	6.8%
Software Platform (application platform, database and development tools)	$208,567	9.1%	$52,591	3.2%
Software Maintenance	$160,800	7.0%	$45,845	2.8%
Development, Deployment Labor and User Support (project team only)	$1,564,000	68.4%	$1,223,000	74.4%
IT Skills Training	$300,000	13.1%	$210,000	12.8%
Total Costs	$2,289,041	100%	$1,643,112	100%

Source: Giga Research, a wholly owned subsidiary of Forrester Research, Inc.

The following are the findings based on the model of the sample medium-size enterprise (see Table 2):

- For J2EE/Linux, the total costs associated with the initial development and deployment, plus three years of support and maintenance were $881,455.
- For Microsoft, the total costs associated with the initial development and deployment, plus three years of support and maintenance were $661,012.
- Giga found that for the medium-size sample organization, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $220,443 less using the Microsoft platform. Microsoft's total costs were 25 percent less than the total costs for J2EE/Linux.

Table 2: Financial Results — Medium-Size Sample Organization

Cost Category	J2EE/Linux Platform	Percentage of Total J2EE/Linux Costs	Microsoft Platform	Percentage of Total Microsoft Costs
Computing Platform (servers, operating system licensing and maintenance)	$39,700	4.5%	$18,196	2.8%
Software Platform (application platform, database and development tools)	$41,000	4.7%	$7,158	1.1%
Software Maintenance	$17,755	2.0%	$7,158	1.1%
Development, Deployment Labor and User Support (project team only)	$713,000	80.9%	$590,000	89.2%
IT Skills Training	$70,000	7.9%	$38,500	5.8%
Total Costs	$881,455	100%	$661,012	100%

Source: Giga Research, a wholly owned subsidiary of Forrester Research, Inc.

See Appendix C: Cost Comparison of J2EE/Linux vs. Microsoft Windows Platform, for a graphic portrayal of the data in Tables 1 and 2.

The assumptions and facts used to define the cost categories and the costs themselves are documented in the Cost Assumptions and Qualifications section below.

At 2.4 percent for the large organization and 4.5 percent for the medium-size organization, the cost of the Linux computing platform played only a small role in the cost economics of the sample organizations and application scenarios. By comparison, the Microsoft Platform accounts for a higher percentage of the cost economics of the large sample organization.

The majority of the cost differences between J2EE/Linux and Microsoft arose from two other factors:

- The license costs of the J2EE and database products sitting on top of Linux
- The cost of the labor required to create and deploy applications in the J2EE/Linux environment

The linkage between Linux as the operating system and J2EE as the development and deployment environment is a factor often overlooked in discussions of Linux's cost economics. The extremely low license costs of Linux appear to offer significant advantage over the higher licensing costs of Windows. This study indicates that a narrow comparison of Linux licensing costs to Windows licensing costs is misleading.

Non-J2EE development environments for Linux, such as the PHP scripting language, are likely to be less expensive than J2EE environments. However, PHP and other scripting environments are not widely used for enterprise applications, as was indicated in the interviews that support this study.

Both sets of customers were driven by the same hardware goals: to take advantage of Intel's cost economics. The interviewees using Linux viewed Intel-based servers as being broadly less expensive than any of the Unix alternatives. The interviewees using the Microsoft platform viewed the economics of Intel hardware as part and parcel of an attractive cost-economics package.

None of the customers conducted a formal return on investment (ROI) study before selecting its platform. Instead, they used informal assessments of cost economics. In addition to these informal cost assessments, cultural factors played a role in the choices of the organizations interviewed. These cultural factors were strongest in the decisions of the customers that selected the J2EE/Linux platform option.

Cultural factors that drove platform decisions in the Microsoft cases were:

- Perception of Linux as a renegade option that carries more risk than Microsoft
- View of Microsoft as an innovator

Cultural factors that drove platform decisions in the J2EE/Linux cases were:

- Concern about Microsoft's market power
- Desire to take control of the technical architecture by implementing best-of-breed products, and sometimes by implementing open source products as well
- Tradition of Unix as the core of an "open systems" strategy
- A passionate belief in open source software as the most important alternative to Microsoft's products; there was no comparable passion among the interviewees using Microsoft, although each considered Microsoft an innovative and proven product set for enterprise application development

This study does not endorse any of these cultural factors as valid reasons for choosing one or the other platform. Rather, it is important to include the cultural factors observed in the interviews to convey a complete picture of the observed behavior. Cultural factors are not the same as cost factors in a decision, although they often will have an impact on costs. For example, selection of J2EE/Linux to satisfy a desire for control over an IT architecture may drive up labor costs, based on the findings of this study.

A majority of customers were aware of switching costs and sought to avoid them when selecting platforms. Thus, the majority of Linux shops were Unix shops transferring skills to Linux. A majority of Microsoft shops were longtime Microsoft customers and were moving to the company's latest platform. There were two exceptions — a shop moving away from Microsoft to Linux and a customer that reimplemented a Linux application on the Microsoft platform.

Switching costs can be significant. The organization that reimplemented a Linux application on the Microsoft platform spent $250,000 in hardware, software and labor costs to do so. That was the extreme case of a company moving from one technical base to another in a "big bang" project. In contrast, the organization that moved from Microsoft to J2EE/Linux was accomplishing the migration gradually, as it refreshed its client and server platforms. This organization had not quantified its switching costs, but would obviously incur them over a period of years rather than in one lump sum.

The Goals and Objectives of the Large-Size Sample Organization

This report profiles the costs of developing, deploying, maintaining and supporting custom business applications for two kinds of organizations — medium-size and large-size organizations. Each of these sample organizations is a composite of the 12 organizations that provided data and is intended to represent the most common sets of requirements in the market today.

The large-size sample organization is a service company with a single data center running applications that automate ordering, accounting, human resources (HR) and other internal operations as well as provide electronic interactions with customers, agents, employees and suppliers. This company provides services to consumers both directly through a Web site and/or indirectly through a sales channel.

Initially, the application will support 1,000 concurrent internal users, growing quickly to 4,000 by year three of our analysis. Further, the application will initially support 1,000 concurrent external users, growing to 6,000 by year three. This user growth rate assumes that the organization will roll out the application to its substantial population of internal and external users over the period included in the model. Giga observed this pattern among the interview subjects.

The business goals for the application include:

1. Reduce the cost of operations by streamlining core selling, fulfillment and accounting processes
2. Expand revenue at an efficient cost of sales and operations
3. Increase customer service and satisfaction
4. Reduce time to market for products and services

The IT goals for the new architecture are:

1. Reduce the cost of IT infrastructure via hardware with superior price performance while maintaining or enhancing reliability
2. Leverage of existing back-office systems within the new architecture
3. Implement new applications to streamline sales, ordering and service provisioning both through browser/Web clients and conventional desktops
4. Build in the ability to scale up processing workloads
5. Incorporate the ability to cost-effectively manage applications over time, including updates and extensions

The Goals and Objectives of the Medium-Size Sample Organization

This sample organization is a medium-size enterprise or departmental workgroup with multiple departmental applications that automate ordering, accounting, HR and other internal operations as well as provide electronic

customer-facing interactions with users, third-party agents and suppliers. This organization provides products and services to consumers both directly through a Web site and/or indirectly through a human channel.

Initially, the application will support 100 concurrent internal users, growing to 150 by year three of our analysis. Also initially, the application will support 250 concurrent external users, growing to 400 by year three. This application will be targeted to the general public (millions of potential visitors) using only browsers. This user growth rate assumes that the medium-size organization is growing at a fast rate and adding new users as its business grows. Giga observed this pattern among the interview subjects.

The business goals for the application include:

1. Implement Web-based channels of commerce and information for customers, employees, partners and suppliers; this requirement is being driven by the mandates of large trading partners and by legislative/regulatory requirements
2. Reduce the cost of operations by streamlining core selling, fulfillment and accounting processes, while expanding revenue at an efficient cost of sales and operations
3. Increase customer service and satisfaction
4. Hold the line on IT budgets

The IT goals for the new architecture are:

1. Streamline IT infrastructure costs by repurposing hardware and software wherever possible and employing Intel hardware for new installations
2. Implement new Web-based applications to provide access to core information and business processes with appropriate security and management controls
3. Build in the ability to scale up processing workloads
4. Incorporate the ability to cost-effectively manage changes to the new applications over time

Cost Assumptions and Qualifications

Through extensive interviews, Giga gathered information and data from 12 organizations that developed and deployed custom applications using either Microsoft's platform and tools (seven organizations) or J2EE/Linux (five organizations). Each of these organizations has unique skills, backgrounds, cultures, and workloads as well as characteristics that are typical of broad numbers of customers. In order to convey our findings, Giga created the two sample organizations described above, which aggregate the information from the 12 organizations used in this study into two uniform models. These models allow general comparisons of the two platform options.

To create uniform models representing cost economics, Giga used the vendor list prices of the products in the analysis. For example, the model assumes use of Microsoft Windows Server 2003 for the application, and uses Microsoft's list price for licensing that product. Giga recognizes that Microsoft and other vendors offer discounts on license costs under certain conditions. This study does not factor those discounts into its cost comparisons.

Giga also made assumptions about the size of the development team, the deployment effort, the costs of training and the size of the operations teams *based on the information obtained from the interview subjects*. These assumptions are described below. They represent costs in a four-year application life cycle in which year zero is the investment (development and deployment) year, with the application supporting users in years one through three.

Large-Size Sample Organization — J2EE/Linux

- Hardware: $50,676 — Purchased two four-way HP ProLiant DL740 Servers with 2GB addressable memory at a cost of $25,338 each.
- Server operating system license: $4,998 — Purchased Red Hat Enterprise Linux AS license at $2,499 for each of two servers and includes 24x7 support for years zero through three.
- Application platform license: $68,000 — Purchased four BEA WebLogic Enterprise Edition application server licenses at $17,000 each.
- Database software: $80,000 — Oracle Database Enterprise Edition (Processor perpetual license) at a cost of $40,000 for each of two servers.
- Development tools: $60,567 — Includes BEA dev2dev Subscription Tools Edition at $4,659 per developer (seven developers in year zero, two developers thereafter) and includes software maintenance for each year.
- Software maintenance: $160,800 — Represents 25 percent annually of list price for BEA WebLogic Enterprise Edition application server license and 29 percent annually for Oracle Database Enterprise Edition.
- Development team: $1,364,000 — Includes 9.25 full-time equivalent (FTEs) for one full year at an average fully loaded cost of $105,000 each. Team includes: one project manager, one architect/lead technical, one database administrator (DBA), four software developers for development and integration, two developers for testing and documentation, and 0.25 FTE for deployment. Thereafter, 1.25 FTEs for annual maintenance and integration in years one through three.
- System administration: $200,000 — Includes 1.0 FTE at a fully loaded cost of $80,000 for system administration during development year zero and 0.5 FTE annually thereafter.
- IT staff training: $300,000 — The project team of nine will each take 10 days of J2EE and WebLogic training at $2,000 per day (including travel) for a year zero cost of $180,000. Thereafter, two developers will take seven days of training annually for a cost of $40,000 for years one through three.

Large-Size Sample Organization — Microsoft

- Hardware: $50,676 — Purchased two four-way HP ProLiant DL740 Servers with 2GB addressable memory at a cost of $25,338 each.
- Server operating system license: $40,000 — Windows Server 2003 Enterprise Edition (32-bit version) at $4,000 each (perpetual), plus one CAL per new user at $40 each. Giga also assumed that the sample organization purchased Microsoft's Software Assurance maintenance program. The price of Windows Server 2003 also includes two Terminal Services licenses for remote administration, which was commonly used by the interview subjects.
- Server operating system maintenance: $21,000 — Twenty-five percent per year of the price of Windows Server 2003 Enterprise Edition license plus and associated CALs purchased during the four years covered by the model.
- Application platform license: $0 (zero cost) — Microsoft's application platform is embedded in the server operating system license, which is described above.
- Database software: $39,998 — SQL Server Enterprise Edition at $19,999 each of two processors (license includes unlimited users).
- Development tools: $12,593 — Includes VisualStudio .NET 2003 Enterprise Developer at a cost of $1,799 each (perpetual) for the seven developers using it.
- Software maintenance: $45,845 — Twenty-five percent of list price annually for SQL Server Enterprise Edition database and VisualStudio .NET developer tools (maintenance for seven tools in year zero and only two tools/developers in years one through three).
- Development team: $1,023,000 — Includes 9.25 FTEs ($728,250) for *nine months* at an average fully loaded annual cost of $105,000 each. Team includes: one project manager, one architect/lead technical, one DBA, four software developers for development and integration, two developers for testing and documentation, and 0.25 FTE for deployment. Thereafter, 0.90 FTEs for annual maintenance and integration in years one through three.
- System administration: $200,000 — Includes 1.0 FTE at a fully loaded cost of $80,000 for system administration during development year zero and 0.5 FTE annually thereafter.
- IT staff training: $210,000 — The project team will each take seven days of Microsoft training at $2,000 per day (including travel) for a year zero cost of $126,000. Thereafter, two developers will take seven days of training annually for a cost of $28,000 for years one through three.

Medium-Size Sample Organization — J2EE/Linux

- Hardware: $4,400 — Purchased two Dell PowerEdge two-way SMP systems at $2,200 per system.
- Server operating system: $300 — Red Hat Linux 9.0 professional server operating system for $149.95 each.
- Server operating system maintenance: $35,000 or $8,750 annually — For Red Hat Linux our sample organization is self-supporting using one-twelfth of an FTE at a fully loaded cost of $105,000.
- Application platform license: $11,000 — BEA License of $10,000 per CPU. Our sample organization's application will share one-tenth of a license for the development server and one full license for deployment for a total cost of $11,000. The rest of the application platform licenses are at no cost and include: Tomcat, Apache, ZOPE, Plone, PHP Admin and Webem.
- Database software: $15,000 in year three only — MySQL initially (at no cost) and migrating to Oracle Standard edition at a cost of $15,000 in year three.
- Development tools: $15,000 — Using a combination of open source "free" tools and "not free" tools such as Cold Fusion, PVCS and BEA dev2dev Subscription Platform Edition at an annual cost of $3,000 per developer for a total of $6,000 in year zero and $3,000 in years one through three.

- Software maintenance: $17,755 — For developer tools and application platform licenses with the Oracle Standard Edition database maintenance in year three only.
- Development team: $333,000 — Consists of two FTEs for one full year (year zero) at an average fully loaded cost of $105,000 each. Team includes part-time effort from: project manager, architect/lead technical, DBA and software developers for development, testing and deployment. Thereafter, 0.5 FTEs for maintenance and integration in years one through three.
- System administration: $44,000 — Includes 0.25 FTE at a fully loaded cost of $80,000 for system administration during development year zero and 0.1 FTE annually thereafter.
- User group team: $336,000 — Includes four functional business analysts at 20 percent of their time during development and during the life of the application at an average fully loaded cost of $105,000.
- IT staff training: $70,000 — The development team of two FTEs taking 10 days of training in year zero at $2,000 per day (including travel) for J2EE and WebLogic training for a cost of $20,000. Thereafter, five days of training a year for a cost of $10,000 in years one through three.

Medium-Size Sample Organization — Microsoft

- Hardware: $4,400 — Purchased two Dell PowerEdge two-way SMP systems at $2,200 per system.
- Server operating system license: $7,398 — Windows 2003 operating system license per server is $799 ($1,698) plus $40 per CAL per new user during the four years covered by the model.
- Server operating system maintenance: $6,398 — Represents 25 percent of Windows 2003 operating system license plus 25 percent of the cost of new user CALs.
- Application platform license: $0 (zero cost) — It's embedded in the server operating system license above.
- Database software: $5,000 — Database will be SQL Server at a license cost of $5,000 (perpetual).
- Development tools: $2,158 — Includes VisualStudio .NET 2003 Professional at a cost of $1,079 per developer (perpetual) for two developers.
- Software maintenance: $7,158 — Twenty-five percent of the SQL Server database and VisualStudio .NET Professional development tools, per year.
- Development team: $210,000 — Consists of 1.25 FTEs for one full year at an average fully loaded cost of $105,000. Team includes part-time effort from: project manager, architect/lead technical, DBA and software developers for development, testing, deployment and user support. Thereafter, 0.2 FTE for annual maintenance and integration in years one through three.
- System administration: $44,000 — Includes 0.25 FTE at a fully loaded cost of $80,000 for system administration during development year zero and 0.1 FTE annually thereafter.
- User group team: $336,000 — Includes four functional business analysts at 20 percent of their time during development and over the life of the application at an average fully loaded cost of $105,000.
- IT staff training: $38,500 — The development team will take nine days of Microsoft training per year at about $2,000 per day (including travel) for a year zero cost of $17,500. Thereafter, the development team will take 3.5 days of training annually for a cost of $7,000 in years one through three.

The Application Scenarios

Each of the models is based on a widely used application scenario that is described in Appendix A: Description of Sample Organizations.

Giga chose scenarios that represented the experience of the 12 organizations studied. All of the interview subjects were building and deploying Web-based applications. (These scenarios also represent the most widely observed application scenarios in our ongoing research.)

Other scenarios may yield different comparative results. A major alternative scenario, for example, is the rehosting on Linux of packaged applications or existing applications running on Unix and RISC hardware. That scenario is outside the scope of this study.

Factors Excluded From This Study

This study focuses on the costs of typical custom application development projects. To zero in on the real cost differences between the two environments studied, we excluded certain factors, such as:

1. The study excludes consideration of platform migration costs. The majority of interviewees did not switch platforms. Each had the option to migrate from a Unix-based IT infrastructure to a Microsoft-based IT infrastructure and vice versa. Only two of the interviewees switched, a small minority.
2. The study excludes the costs of an IT management infrastructure, primarily because none of the interview subjects found these costs to affect the costs of individual projects using the development/deployment platforms studied. These include items such as budgeting, procurement, vendor management, asset management and auditing. Each of the organizations interviewed had a preexisting IT management infrastructure, which was neither enlarged nor reduced as a consequence of the application project described to us.
3. The study excludes the cost of end-user training and user support/help desk. Formal end-user training programs were a rarity among the interviewees. One of the Microsoft customers, for example, reported that business units took responsibility for training users to use Windows graphical user interfaces, but also reported that these costs were minor. The other organizations reported no formal end-user training for their application projects. Also, the addition of a new application did not materially affect user support costs among the interviewees. Thus, these costs did not speak to the differences between the J2EE/Linux and Microsoft choice in this study.
4. The study excludes the cost of routers, switches and other "network infrastructure," as well as storage systems such as RAID arrays and storage area networks. Only one of the interviewees reported having to expand network infrastructure to accommodate a new application. For the majority of users, the cost of new storage was embedded in the costs for new servers, which included the disk storage required for the application.
5. The study excludes the cost of desktop hardware, operating system and personal productivity applications because they did not affect the cost of the custom development projects described by the interviewees. All of the organizations purchased desktop environments using processes that were separate from the application projects they described. Those desktop costs were not built into the application project budgets. The vast majority were Windows/Intel systems.
6. The study uses *list prices* for all products, consciously excluding known discount programs and practices. List prices provide the most neutral ground for making cost comparisons.

TEI Risks

The TEI risks are used to widen the possible outcomes of a project's return on investment. Since the future cannot be accurately predicted, there is risk inherent in any project. The following *general* risks are important to consider when developing and deploying custom applications:

- Lack of corporate discipline in creating processes and procedures to best take advantage of the development productivity benefits may reduce those benefits.
- Lack of appropriate training for IT personnel who will be responsible for developing and deploying the applications may delay the implementation date.
- The inability of an organization to find and retain IT staff who have the ability to take full advantage of the productivity benefits may reduce those benefits.
- The project team may not be organized efficiently, raising the development time required to complete the application, thereby increasing the labor costs.

The major risks associated with the J2EE/Linux and Microsoft platforms are as follows:

- Another development project may not conform to the description of our study's application scenarios. In particular, expanded requirements for integration and complex business logic would drive up the development time and associated costs.
- Microsoft Windows Server 2003 is a new operating system version that has not yet been widely deployed. The interviewees who had evaluated Windows Server 2003 found it to be more stable and scalable than its recent predecessors, and so, for this study, we assumed that the product would be as stable as the Red Hat Linux operating system version included in the J2EE/Linux cases. The experience with the Red Hat Linux release we selected has been quite good, although it is also a new release and so carries risks of instability. If our assumptions about the stability of either platform option are overly optimistic, there is a risk of higher operating costs. The higher costs would be the result of additional operating staff time to keep the servers up and running.
- Red Hat, the provider of Linux used in this study, is a relatively new vendor that showed losses on a trailing 12-month basis during the last year. Thus, Red Hat's long-term viability is a question — and a risk.
- The prices incorporated into this study may rise in the future. Both Red Hat and Microsoft may raise their prices in the future, for example.
- The costs incorporated into this study for application server and database software are prices for widely used products. There are viable alternatives with varying prices that would produce different costs results for those items. However, we don't believe this study's conclusions would be impacted by these variable costs.

The oft-cited risk that under the General Public License (GPL), open source users will lose control over valuable intellectual property was not evident in this study. None of the J2EE/Linux customers contributed source code either to Linux or to the other projects they used. Rather, they used the distributions from Red Hat, and some others, without modifying them, transmitting bugs and other issues to the vendor in a conventional manner.

Assessment of risks should take into consideration a range of possible outcomes based on the risks associated with IT projects in general and specific risks related to using either the Microsoft or the J2EE/Linux platforms.

Flexibility Options

Flexibility, as defined by TEI, represents additional capacity or adaptability that allows for *future* investments to be turned into business value quicker, better and at a lower cost. The organizations interviewed for this study invested in either Microsoft or J2EE/Linux platforms to create a foundation for current and future applications. Thus, they are investing not only to complete a current project, but also to build applications in the future that provide benefits to the business in terms of reduced costs and/or raised revenues.

Flexibility, or indirect, benefits are defined as second-stage investments that are realized only after the first investment is implemented. The existence of the option to capture these future benefits has a strategic value to the organization.

Among the customers interviewed, Giga observed the following future flexibility benefits to be associated with developing and deploying applications on the Microsoft platform:

- Availability of Microsoft-trained IT staff is ubiquitous in the worldwide marketplace. This pool of readily available and relatively inexpensive resources allows organizations to confidently grow their IT environment using the Microsoft platform.
- Developers using Microsoft development tools are more productive than their J2EE/Linux counterparts. Organizations that have invested in Microsoft resources have a future strategic cost advantage relative to labor.
- Microsoft's new platforms incorporate Web Services standards to allow for interoperability with non-Microsoft platforms and products. These Web Services features add to the flexibility available now and in the future when using the Microsoft platform in concert with a variety of enterprise systems.

Among the customers interviewed, Giga observed the following future flexibility benefits to be associated with developing and deploying applications on the J2EE/Linux platform:

- The organizations using J2EE and Linux had adopted a variety of products, selecting the best-of-breed in their views for the required function. They tended to structure their applications to allow them to switch to alternative products or suppliers in the future. For example, the interviewees sought to use only vanilla J2EE, not the proprietary extensions provided by their J2EE vendors. There are potential flexibility benefits available when using this approach.
- For Linux, the availability of multiple Linux distributions was viewed as an important flexibility benefit.
- This study does not measure the value of participation in open source software communities. Two of the interviewees had sought to measure this benefit by examining quality and support responsiveness, but had not reached clear conclusions.

Conclusions

The primary conclusion of the study is that Microsoft offers a substantial cost advantage over J2EE/Linux as a development platform for the applications considered. The primary sources of Microsoft's cost advantages are:

1. The J2EE application server and Unix-based database software used in the Linux development and deployment stack drive up product costs and development complexity relative to the comparable Microsoft products.
2. Microsoft's tools simplify development of applications like those profiled in the study when compared to the J2EE/Linux products in the study. This simplification translates into lower labor costs for development and administration of custom applications and a faster time to deployment.

The comparison of the two platforms shows that large to medium-size organizations that develop, deploy, support and maintain custom applications on the Microsoft .NET platform can expect to experience 25 percent to 28 percent less cost during a four-year life cycle than if the J2EE/Linux platform was used.

In this study, Microsoft's development tools were shown to be highly productive for the target applications. This result should not be surprising. When it comes to creating tools and platforms for application developers, Microsoft has a long track record of success. In addition, Microsoft's software prices were shown to be generally lower than those of application servers, databases and related products associated with the J2EE/Linux platform.

The study indicates that the economics of Linux depend on the larger IT context for the operating system. (The same is true for open source software in general.) In a head-to-head comparison, the list price for Red Hat's Enterprise Linux and Red Hat 9 (the Linux version chosen for this study) is less expensive than Microsoft Windows Server 2003. However, the key cost factor in the study's Linux cases was the J2EE environment, not the operating system. Although the cost of Linux is low, the impact of that lower cost on the overall economics of an application development project is minor. The full development and deployment environment and the labor associated with the development project are the biggest costs. Comparisons of individual elements within the stack of software products required to build and deploy a complete application tell only part of the story.

Further, the study indicates that the competition between the Microsoft platform and J2EE/Linux is not a contest in which the winner takes all. Despite this study's findings about Microsoft's cost advantages over J2EE/Linux, many IT shops will choose J2EE/Linux over the Microsoft platform. Why? Consider:

1. For Unix shops seeking to move an existing Unix application off RISC hardware to less costly Intel hardware, the least expensive option is Linux on Intel servers, not Microsoft on Intel. Switching from Unix to Microsoft would require investment in new systems infrastructure, new skills and perhaps replacement applications. In the study, the majority of interviewees sought to avoid switching costs for this reason.
2. J2EE shops with two to three years of sunk costs in skills development can improve the cost-efficiency of their infrastructure by moving to Linux on Intel without incurring the switching costs associated with a migration to the Microsoft platform. In addition, some large J2EE shops will seek to drive down their labor costs by using offshore development. (The same offshore development efficiencies are available to Microsoft shops, of course.)
3. The J2EE/Linux interviewees in the study represent IT shops that prefer open systems to the platforms of a single vendor. This is primarily a cultural factor, although it is often expressed in financial (or pseudo-financial) terms. To these customers, Linux and open source are the cutting-edge of open systems and are seen as the best way to avoid vendor lock-in, and to assert architectural control, thereby controlling costs. These shops reject Microsoft as the most obvious single-vendor software platform provider today. However, they are also likely to turn away from other single-vendor software platform providers, including IBM, BEA, Oracle and Sun — even if those platforms incorporate Linux.

Most of the interviewees are mixed shops. If they don't use Linux today, they do use Unix. They also use Microsoft platforms. The real question is, which of these platforms is the largest number of customers likely to favor for new projects? This analysis points to Microsoft as a very solid choice.

For Giga, a primary goal of this study was to add practical considerations to the debate about Linux and Microsoft, which has been dominated by emotion and polemics until now. This report accomplishes Giga's goal. The passions ignited by the debate over open source must be balanced by product pricing, development skills and productivity, as well as prior and future investments to result in IT choices that stand the test of time.

Appendix A: Description of Sample Organizations

Description of Sample Organization — The Large-Size Enterprise

The sample organization is a service company with a single data center running applications that automate ordering, accounting, HR and other internal operations, as well provide electronic interactions with customers, agents, employees and suppliers. This company provides services to consumers both directly through a Web site and/or indirectly through a sales channel.

The sample organization has a heterogeneous IT environment, employing Microsoft Office on its desktops, Windows NT and Unix distributed servers and host-based systems of record (SOR). The sample organization uses two-way Intel servers for Windows NT and four-way RISC servers for Unix.

The sample organization is moving to a new application architecture to achieve the following business goals:

1. Reduce the cost of operations by streamlining core selling, fulfillment and accounting processes
2. Expand revenue at an efficient cost of sales and operations
3. Increase customer service and satisfaction
4. Reduce time to market for products and services

The goals for the new IT architecture are:

1. Reduce the cost of IT infrastructure via hardware with superior price performance while maintaining or enhancing reliability
2. Leverage of existing back-office systems within the new architecture
3. Implement new applications to streamline sales, ordering and service provisioning both through browser/Web clients and conventional desktops
4. Build in the ability to scale up processing workloads (see Table 3)
5. Incorporate the ability to cost-effectively manage applications over time, including updates and extensions

Table 3: Application's Anticipated Workloads — Large-Size Enterprise

	Years 0 to 1	Ramp in Years 2 to 3
Concurrent Internal User Sessions	1,000[1]	4,000
Concurrent External User Sessions	1,000[2]	6,000
SOR Transactions/Day	25,000[3]	50,000
SOR Data Volumes	250GB[4]	500GB

Source: Giga Research, a wholly owned subsidiary of Forrester Research, Inc.

[1] Assumes 6,000 total internal users employing both browsers and conventional clients

[2] Assumes 15,000 external sales agents using primarily browsers

[3] Assumes 1,000 business transactions per day, each of which generates 20 ACID transactions in four major SORs (accounting, products, customers, sales)

[4] Giga estimate

Description of Sample Organization — The Medium-Size Enterprise

This sample organization is a medium-size enterprise or departmental workgroup with a multiple departmental applications that automate ordering, accounting, HR and other internal operations, as well provide electronic customer facing interactions with users, third-party agents and suppliers. This organization provides products and services to consumers both directly through a Web site and/or indirectly through a human channel.

The medium-size enterprise primarily runs on an Intel and Windows-based systems environment, employing Microsoft Office on its desktops and Windows and/or Novell distributed servers. There are no host-based transactional systems. The sample organization uses one-way and two-way Intel servers.

The sample organization is moving to a new application architecture to achieve the following business goals:

1. Implement Web-based channels of commerce and information for customers, employees, partners and suppliers; this requirement is being driven by the mandates of large trading partners and by legislative/regulatory requirements
2. Reduce the cost of operations by streamlining core selling, fulfillment and accounting processes, while expanding revenue at an efficient cost of sales and operations
3. Increase customer service and satisfaction
4. Hold the line on IT budgets

The goals of the new IT architecture are:

1. Streamline IT infrastructure costs by repurposing hardware and software wherever possible and employing Intel hardware for new installations
2. Implement new Web-based applications to provide access to core information and business processes with appropriate security and management controls
3. Build in the ability to scale up processing workloads (see Table 4)
4. Incorporate the ability to cost-efficiently manage change to the new applications over time

Table 4: Application's Anticipated Workloads — Medium-Size Enterprise

	Years 0 to 1	Ramp in Years 2 to 3
Concurrent Internal User Sessions	100[1]	150
Concurrent External User Sessions	250[2]	400
SOR Transactions/Day	75[3]	150
SOR Data Volumes	50GB[4]	100GB

Source: Giga Research, a wholly owned subsidiary of Forrester Research, Inc.

[1] Assumes 300 total internal users employing both browsers and conventional clients

[2] Assumes application is targeted to the general public (millions of potential users) using only browsers

[3] Assumes 1,000 business transactions per day, each of which generates 20 ACID transactions in four major SORs (accounting, products, customers, sales)

[4] Giga estimate

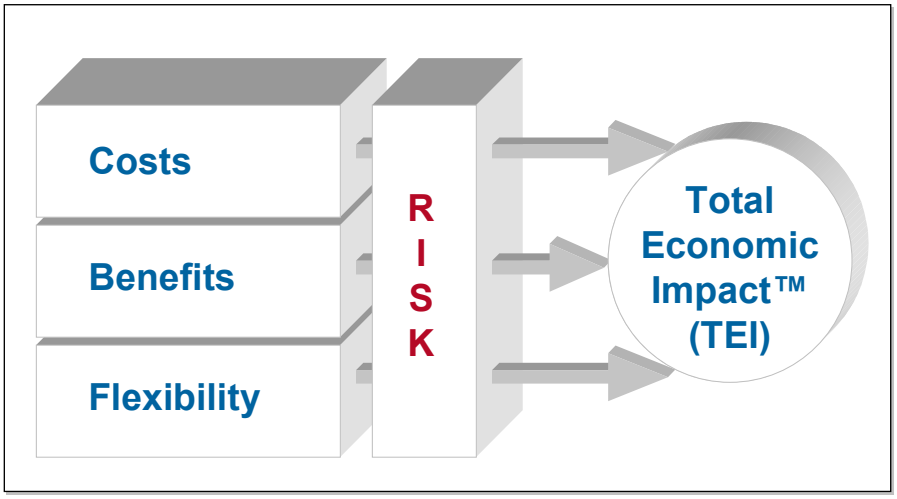
Appendix B: Total Economic Impact™ Primer

Total Economic Impact™ is primarily a common language tool, designed to capture and properly communicate the value of IT initiatives in a common business language. In so doing, TEI considers four elements of any initiative:

1. Benefits
2. Costs (sometimes referred to as total cost of ownership (TCO))
3. Flexibility
4. Risk

The figure below shows the TEI methodology conceptually. Benefits, flexibility and costs are considered, through the filter of risk assessment, in determining an expected ROI for any given initiative.

TEI Conceptual Diagram



Source: Giga Research, a wholly owned subsidiary of Forrester Research, Inc.

Benefits

Benefits represent the *value* delivered to the business by the proposed project. Oftentimes, IT project justification exercises focus on cost (e.g., TCO) and cost reductions. Among industry leaders, IT is deployed as an offensive weapon, with value expectations greater than simple cost reduction, especially when those cost reductions tend to focus within IT. TEI captures the value proposition of the proposed project by measuring the benefits against the incurred costs.

All benefits captured by TEI must be traceable back to one or more critical success factors (CSFs). These CSFs are directly linked to a higher-level business strategy. If a proposed technology investment generates benefits that cannot be satisfactorily linked to a CSF, then it will not be included as a benefit for the organization in the model. In these cases, TEI requires that the benefit be discarded.

Under TEI, benefits may only accrue to the business units. “Benefits” derived through cost reductions within IT accrue as negative TCO to the IT budget, thereby showing a reduced TCO. (TCO is considered by TEI to be a single-dimension, cost-centric focus on the IT budget.)

The TEI process begins with a discovery of potential benefit areas. A representative from the organization under examination who has the ability to capture the benefit in question must validate each benefit captured during discovery. In other words, values cannot arbitrarily be assigned to a benefit if that person is not in a position to deliver that benefit should the project be approved. Additionally, projects that are expected to

deliver business value require some effort on the part of the business to realize that value. That effort may be in the form of training, organizational change or a modification of existing business processes. Therefore, TEI requires dialog with the business leaders specifically responsible for making the necessary changes, in order to capture the proposed benefit during the justification phase. TEI captures this dialog in the form of the names of the individuals, which validates the value calculation of each benefit.

Within TEI, each benefit entered has a specific capture date. Although the benefit may be captured over time, TEI requires the specification of a date when most of the benefit has been captured. TEI will then place the value delivered in the appropriate time frame within the project.

Costs

Costs represent the investment necessary to capture the value, or benefits, of the proposed project. IT or the business units may incur costs. These may be in the form of fully burdened labor, subcontractors or materials. Additionally, costs consider all the investment and expenses necessary to deliver the value proposed.

Flexibility

Flexibility, as defined by TEI, represents investing in additional capacity that can, for some future additional investment, be turned into business benefit — for instance, an investment in an enterprisewide upgrade of the desktop word processor application where the primary driver may be standardization (to increase efficiency) and licensing (to decrease IT costs). However, a collaborative workgroup feature may translate into greater worker productivity when the organization is ready to absorb the discipline necessary to capture that benefit. The collaboration feature does not promise benefit during this phase of the project and must be captured later, incorporating additional investment, most likely in the form of training. However, the existence of the option has a present value that can be estimated. The flexibility component of TEI captures that value.

Flexibility can also be calculated by acknowledging that management has several decision points along the way for any given project. At each point, management can steer the project to a different outcome or cancel it altogether. Many net present value evaluations fail to take this management flexibility into account. Since TEI's flexibility component uses the industry standard Black-Scholes options formula, the management flexibility factor is taken into consideration.

TEI divides a project into multiple phases. The first phase is considered the benefits phase — it is the phase expected to deliver the primary benefits. The benefits phase is usually no more than one budget cycle long and it is the primary reason the project is being considered. All other phases are options or flexibility phases. For additional investment at some point in the future, business benefit can be captured during these options phases. TEI applies the Black-Scholes options pricing equation to all phases other than the benefits phase. The Black-Scholes equation uses five inputs to calculate the present-day value of flexibility or options:

1. The value, or business benefit, that can be captured when the option is exercised; this value is expressed in present value terms
2. The time, to the date, at which point the option or flexibility expires; expiration could be due to business changes or technology obsolescence
3. The cost of the investment to exercise the option and capture benefit
4. The risk-free interest rate (typically the interest rate of government securities is used)
5. The volatility of the industry or sector; TEI uses the volatility of the stock prices within the market sector as this input

Risk

Risks are used to widen the possible outcomes of the project. Since the future cannot be accurately predicted, there is risk inherent in any project. TEI captures risk in the form of risks-to-benefits and risks-to-costs.

Risks-to-benefits considers all possible risks to each possible benefit. Likewise, risks-to-costs considers all possible risks to each possible cost. Then, a range is chosen by applying best judgment for each cost and benefit, based on the set of risks assigned to each cost and benefit. The range is entered in the form of a low estimate, a most likely value and a high estimate. For example, the risks to a cost may result in a range from the expected value as the low estimate, to two times the expected value as the high for a particular cost (representing a potential two times cost overrun).

TEI applies a probability density function known as "triangular distribution" to the values entered. The expected value — the mean of the distribution — is used as the risk-adjusted cost or benefit number. The risk-adjusted costs and benefits are then summed to yield a complete risk-adjusted summary and ROI.

Typical project risk factors to consider include the following:

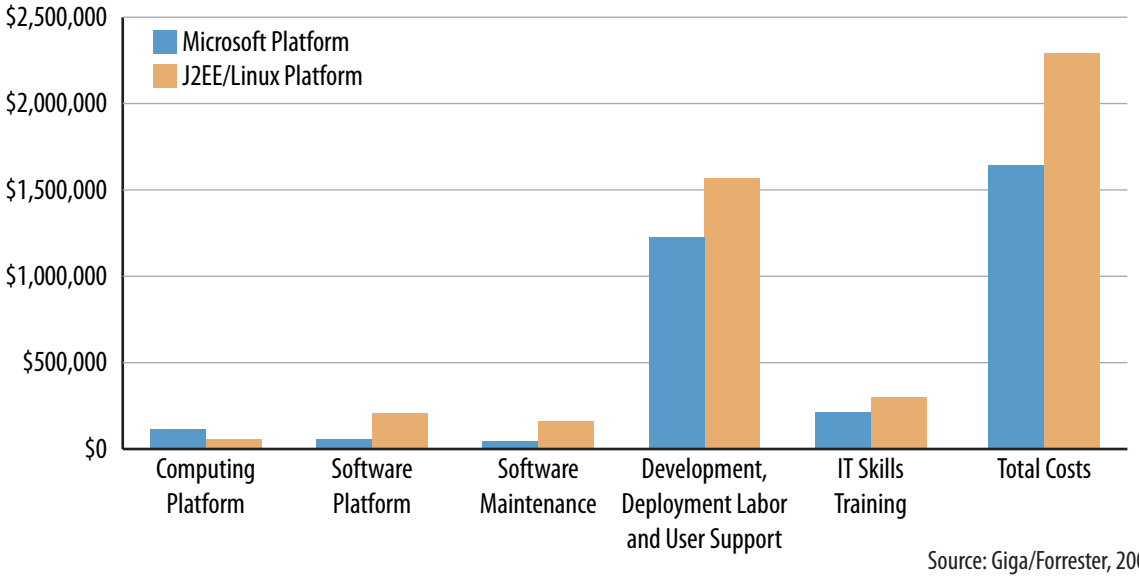
- *Vendors*: The risk that the vendor of a product or technology may need to be replaced at some point during the project duration
- *Products*: The risk that a product will not deliver the functionality expected
- *Architecture*: The risk that the current product architecture will not allow future infrastructure decisions and changes
- *Culture*: The risk that an organization will be unable to absorb the new technology or adapt to its implementation
- *Delays*: The impact on revenues of a project delay or cancellation
- *Size*: The direct correlation of project risk to the size of the project, as measured by application size or budget

Appendix C: Cost Comparison of J2EE/Linux vs. Microsoft Windows Platform

The following are the findings based on the model of the sample large-size enterprise (see below):

- For J2EE/Linux, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $2,289,041.
- For Microsoft, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $1,643,112.
- Giga found that for the large sample organization, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $645,929 less using the Microsoft platform. Microsoft's total costs were 28.2 percent less than the total costs for J2EE/Linux. The primary driver of this difference is a shorter time to deployment for Microsoft - nine months vs. 12 months for J2EE/Linux.

Cost of Implementing J2EE/Linux vs. Microsoft Windows Platform - Large-Size Sample Organization

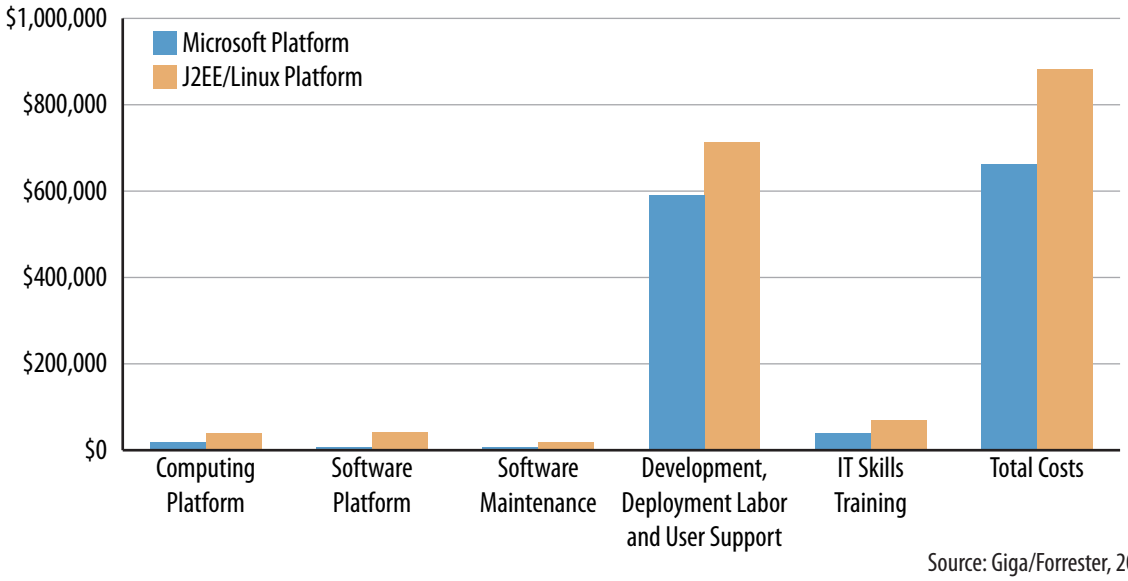


Source: Giga/Forrester, 2003

The following are the findings based on the model of the sample medium-size enterprise (see below):

- For J2EE/Linux, the total costs associated with the initial development and deployment, plus three years of support and maintenance were $881,455.
- For Microsoft, the total costs associated with the initial development and deployment, plus three years of support and maintenance were $661,012.
- Giga found that for the medium-size sample organization, the total costs associated with the initial development and deployment, plus three years of support and maintenance, were $220,443 less using the Microsoft platform. Microsoft's total costs were 25 percent less than the total costs for J2EE/Linux.

Cost of Implementing J2EE/Linux vs. Microsoft Windows Platform - Medium-Size Sample Organization



Source: Giga/Forrester, 2003