# BUYING, SUPPORTING, MAINTAINING SOFTWARE AND EQUIPMENT

An IT Manager's Guide to Controlling the Product Lifecycle

Gay Gordon-Byrne



### BUYING, SUPPORTING, MAINTAINING SOFTWARE AND EQUIPMENT

An IT Manager's Guide to Controlling the Product Lifecycle

## BUYING, SUPPORTING, MAINTAINING SOFTWARE AND EQUIPMENT

An IT Manager's Guide to Controlling the Product Lifecycle

Gay Gordon-Byrne



CRC Press is an imprint of the Taylor & Francis Group, an **informa** business AN AUERBACH BOOK CRC Press Taylor & Francis Group 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742

© 2014 by Taylor & Francis Group, LLC CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works Version Date: 20140320

International Standard Book Number-13: 978-1-4822-3279-0 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright. com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at http://www.taylorandfrancis.com

and the CRC Press Web site at http://www.crcpress.com

## Contents

Preface	 ii

### SECTION I Initial Product Acquisition

Chapter 1	Equipment and Application Acquisition	3
	Introduction	3
	Application Selection Dictates Hardware Selection	3
	Background	5
	Application License Acquisition	5
	Going Naked	7
	Operating System Acquisition	8
	Acquisition Model: Direct from the Original	
	Equipment Manufacturer	9
	Platforms as Monopolies	10
	Multiyear Commitments	11
	Operating System and Machine Code License Lock-Ins	12
	Strategic Partnerships	13
	The Win Room	14
	Credible Competition	15
	Acquisition Model: Channel Partners and Authorized	
	Resellers	16
	Competition between Partner or Channel Sources	18
	Controlling Discounts	19
	Selecting Partners	19
	Acquisition Model: Wholesale Distribution	20
	Gray Market	21
	Counterfeit	21
	Acquisition Model: Retail and the Internet	22
	Retail Service Plans	22
	Internet Buying	23
	Acquisition Model: Used Equipment	23
	Summary	25
	Notes	26

### vi • Contents

Chapter 2	Initial Support and Maintenance	27
	Introduction	27
	Maintenance in the Digital World	27
	Hardware Maintenance	28
	Software Maintenance for OS and Applications	30
	Consumables	32
	Defect Support	32
	Hardware Maintenance	35
	Redundancy, Self-Healing, and Fault-Tolerant	36
	Gaming the Service Level Agreement	37
	Software Maintenance	38
	Serviceability	39
	Disaster Recovery and Backup	41
	"Appliances" and "Black Boxes"	41
	Summary	44
	Notes	44
Chapter 3	Hardware Warranty Models	47
	Introduction	47
	Limits and Purposes of Hardware Warranties	47
	Precontracted Parts and Labor Warranties	48
	Accounting for Prepaid Service Agreements	50
	90-Day Parts and Labor Warranties and 12-Month	
	Parts Warranties	50
	Self-Repair	51
	Parts Only	52
	Original Equipment Manufacturer Depot Repair	53
	Labor	54
	User Depot Repair	55
	Time and Materials	57
	Transaction Volume Sensitive Warranty Models	58
	Preventative Maintenance	59
	Scheduled Maintenance	60
	Summary	61
	Notes	61
Chapter 4	Software Warranty and Support Models	63
	Introduction	63
	Initial Acquisition	63

	Software Warranty Functions	64
	Copyright and Piracy	66
	License Audits	67
	Maintenance Pricing	67
	Treatment of Upgrades and New Releases	69
	Patch Management	70
	Controlling Maintenance Pricing	70
	Going Naked	71
	Independent Support	71
	Switching Systems Software Vendors	72
	Summary	73
	Notes	73
Chantar E	Einance and Accounting Issues for Maintanance	
Chapter 5	and Support	75
	and Support	/3
	Introduction	75
	Initial Purchase	75
	Tangible Assets and Intangible Software	77
	Lending and Leasing	79
	Internal Accounting	81
	Manufacturer (Original Equipment Manufacturer)	
	Captive Leasing	83
	Ease of Contracting	83
	Operating Lease Treatment	84
	Residual Value	86
	Licensed Software and Residual Value	87
	Tethered and "Black Box" Products	88
	End of Lease Issues	90
	Summary	91
	Notes	92

### SECTION II Postwarranty Support and Maintenance

Responsibility for Defect Support	95
Introduction	.95
Software Defect Support	.95
Types of Patches	.98
	Responsibility for Defect Support Introduction Software Defect Support Types of Patches

	Hardware Defect Support	
	Logic Flaws (Hardware or Machine Code)	
	Updates versus Upgrades	
	Defect Support of Component Failures	
	Engineering Change Levels	
	Remote Diagnostics	
	Summary	
	Notes	
Chapter 7	Postwarranty Hardware Maintenance	109
	Introduction	
	Limitations of Warranty	
	Voiding Warranty	
	Purpose of a Warranty	
	Carry-In Warranty	
	Warranty Pricing	
	Original Equipment Manufacturer Postwarranty Su	ipport115
	Independent Service Organizations	119
	Downside of Independent Service Organization	
	Offerings	120
	Parts Warranties	
	Machine Code and Embedded Software	122
	Self-Repair	123
	Automotive Repair: Converging Issues	125
	Five Requirements for Repair	126
	Diagnostic Tools, Diagnostic Software, and Err	ror
	Codes	
	Tools	
	Documentation	
	Service Parts	
	Machine Code	
	Summary	
	Notes	

### SECTION III Technology Product Details

Chapter 8	Machine Code and Embedded Software	135
	Introduction	.135

Defining Machine Code	135
Aachine Code Divides Hardware from Software	136
Tangible Asset Limbo	141
unctions of Machine Code	142
Machine Management and Access Control	143
Feature Activation	143
Service Access and Settings Passwords	143
Maintenance Access Control	144
Feature Validation	145
Diagnostic Routines and Error Reporting	146
Remote Diagnostics	147
Configuration Access Control	148
Feature Activation	148
Repair Issues	148
Consumer Issues	152
Copyright Issues: Digital Millennium Copyright Act	
nd World Intellectual Property Organization	154
Copyrightable Code	156
itigation and Legislation	157
End-User Litigation	157
Competitive Attacks	157
Legislative Efforts	158
ummary	
Jotes	160
ervice Parts	161
ntroduction	
Proprietary Parts	
10011ctul v 1 ul to	
Commodity Parts	
Commodity Parts FRAND/RAND Parts	162 163
Commodity Parts FRAND/RAND Parts Original Parts	162 163 164
Commodity Parts FRAND/RAND Parts Original Parts Plug Compatible Parts	162 163 164 165
Commodity Parts FRAND/RAND Parts Original Parts Plug Compatible Parts Remanufactured Parts	162 163 164 165 165
Commodity Parts FRAND/RAND Parts Original Parts Plug Compatible Parts Remanufactured Parts Refurbished Machines	162 163 164 165 165 166
Commodity Parts FRAND/RAND Parts Original Parts Plug Compatible Parts Remanufactured Parts Refurbished Machines Gray Market Parts	162 163 164 165 166 166
Commodity Parts FRAND/RAND Parts Original Parts Plug Compatible Parts Remanufactured Parts Refurbished Machines Gray Market Parts Counterfeit Parts	162 163 164 165 165 166 166 167
Commodity Parts FRAND/RAND Parts Original Parts Plug Compatible Parts Remanufactured Parts Refurbished Machines Gray Market Parts Counterfeit Parts Jsed Parts	162 163 164 165 165 166 166 167 171
	<ul> <li>Defining Machine Code</li> <li>Machine Code Divides Hardware from Software</li> <li>Tangible Asset Limbo</li> <li>Functions of Machine Code</li></ul>

### x • Contents

Notes <b>Chapter 10</b> Service Restoration and Support P Introduction Reporting Call Centers and Ticketing Systems Nonticketed Events	
Chapter 10 Service Restoration and Support P Introduction Reporting Call Centers and Ticketing Systems Nonticketed Events	Process
Chapter 10 Service Restoration and Support P Introduction Reporting Call Centers and Ticketing Systems Nonticketed Events	Process
Introduction Reporting Call Centers and Ticketing Systems . Nonticketed Events	
Reporting Call Centers and Ticketing Systems Nonticketed Events	
Call Centers and Ticketing Systems	
Nonticketed Events	177
Specialist Hand Off	
Severity and Call Triage	
The Service Level Agreement	
Service Entitlement Database	
Diagnostics	
Specialist Handling	
Technicians	
Closing Tickets	
Categories of Repair	
Monitoring the Service Level Agreer	nent 190
Reverse Logistics	
Return Merchandise Authorizatio	on Monitoring192
Summary	
Chapter 11 Building Blocks of the Machine: W	Vhat Breaks
and Why	
Introduction	
Basic Hardware Components	
Circuits	
Semiconductors	
Simple Semiconductors: Transisto	ors, Resistors, and
Diodes	
Integrated Circuits	
Chips	
Capacitors	
Printed Circuit Boards	
Power Supplies and Fans	
Batteries	
Controllers	
External Connections	

	Central Processing Units	203
	Specialized Processors	
	Memory	
	Channels	
	Reliability Issues of Components and Materials	205
	Summary	
	Notes	206
Chapter 12	Repair Issues by Product Type	
	Introduction	
	Printed Circuit Boards	209
	Central Processors	210
	Environmental Damage	210
	Manufacturing Defects	211
	Machine Code	211
	Peripherals	212
	Tape Drives	213
	Disk Drives	214
	Solid State Disks	
	Output Devices	
	Networking Devices	217
	Displays	219
	Accessories: Cabinets, Frames, and Connectors	220
	Cabinets, Frames, and Connectors	221
	Summary	221
	Notes	222

### SECTION IV Controlling Product Life

Chapter 13	Building Blocks of Using the Machine: Software	
	Layers	225
	Introduction	225
	Software Layers	225
	Top Layer Content and Media	225
	Applications	227
	In-the-Box Applications	227
	Commercially Developed Applications	227

	Homegrown Applications	229
	Customized Applications	229
	Horizontal and Vertical Applications	230
	Mobile "Apps"	231
	Programming Languages	231
	Operating Systems	233
	Patching	233
	Peripheral Drivers	
	Security Patches	
	Systems Software (Accessories and Utilities)	235
	License Forms	237
	Systems Software Support	238
	Machine Programming	238
	Machine Languages, Programmable Language	
	Code, Assembler, and More	239
	Accounting for Software	241
	Ownership of Embedded Software	241
	Licensed Software and Residual Value	
	Purchase, Rental, and Leasing of Software	243
	Summary	
	Notes	
Chantan 14	Software Summart Lawson	245
Chapter 14	Software Support Issues	
	Introduction	245
	Developer Control	245
	Source Code	245
	Perpetual License	
	License to Use/Right to Use	246
	Rental	
	Pay for Use	247
	Open Domain	247
	Version-Level Tracking	247
	Product and Version-Level Conflicts	248
	Software Maintenance	249
	Jailbreaking	250
	Application System Hardware Dependencies	252
	Configuration-Dependent Licensing Models	253
	Usage-Based Licensing Models	254

	"Black Box"-Type Products	254
	Virtualization	255
	Cloud and Software as a Service	257
	Summary	258
	Notes	258
Chapter 15	Support Restrictions to Control Purchases	259
	Introduction	259
	Advantages for Vendors	259
	Account Control	260
	Wolves in the Hen House	261
	Tactic: Controlling the Request for Proposal and	
	Award	262
	Tactic: Monolithic Purchases	262
	Tactic: Strategic Partnerships	263
	Tactic: Trimming the Vendor List	263
	Tactic: License Audits and Compliance	264
	Tactic: Return to Service Premium	264
	Planned Obsolescence	265
	Tactic: Require New Versions to Access Routine	
	Functions	265
	Tactic: Manipulate End of Support and End of	
	Service Life	266
	Tactic: Refuse to Allow Transfer of Equipment, or	
	Prepaid Support Contracts to Secondary Buyers	266
	Tactic: Restricted License Transfer, or Software	
	Maintenance Contract within the Enterprise	267
	Tactic: Refusal to Transfer Licenses or Maintenance	
	Contracts to a Secondary Market Buyer	268
	Tactic: Refusal to Allow a Used Market for Licensed	
	Products	268
	Fear, Uncertainty, and Doubt	269
	Without the Vendor Maintenance Agreements, the	
	Equipment Will Fail at Higher Rates	269
	More Costly Service Contracts Result in Better	
	Service	270
	Fear of Complexity	271
	Multivendor Conflicts	272

	Hardware FUD	273
	Physical FUD	274
	Physical Obsolescence	274
	Fear of Losing Defect Support	274
	Confusing Hardware Defect Support with Software	
	Defect Support	276
	Fear of Inadequate Skills	276
	Fear of Counterfeit Parts	277
	Summary	278
	Note	278
Chapter 16	Business Intelligence for Support and Maintenance.	279
	Introduction	279
	Uses of Analytics	279
	System Availability	279
	Total Cost of Ownership	280
	Basic Measurements	282
	Sources of Analytics	285
	Manufacturers	285
	Hardware Maintenance Providers	287
	Anecdotal Sources	288
	Self-Evaluation	288
	Software Failure Rate	290
	Useful Life	290
	Tolerances and Specifications	291
	Future Proofing	292
	Summary	293
Chapter 17	End of Service Life and Obsolescence	295
	Introduction	295
	Defining End of Service Life	295
	Sweating Assets	297
	Practical versus Physical Obsolescence	298
	Refresh and Replacement Programs	299
	Failure-Based Refresh Programs	300
	Warranty-Based Refresh Programs	302

Rolling Refresh Programs	303
Bring Your Own Device	303
Service Parts for the Long Haul	305
Secondary Markets	306
Principle: Used Equipment Is the Best Source of	
Parts outside the Manufacturer Service Desk	306
Principle: Broken Machines Do Not Sell Unless	
Repairable	307
Recycling and Repurposing	
Used Equipment Sales	309
Donations and Employee Sales	310
Recycling	310
Summary	312
Notes	313

### Preface

Welcome. This book is not a how-to manual for repair nor is it a how-to negotiate prices manual. My goal is to help end users understand why vendors propose the service agreements they offer. Readers will be able to avoid common traps, better control the negotiation for both products and services, and thereby control the lifecycle of their purchases.

This book is set up to follow the way that most people experience technology products and contracting decisions. The first section deals with decisions made at the time of product selection, then examines the types of problems typically experienced during use, and finally delves into issues of product end of life, and how best to manage support and maintenance issues for the long term.

Readers may notice that many of my concepts and recommendations for negotiation will not be welcomed by manufacturers. This is not accidental. My lengthy career as a vendor has given me wide experience with the methods and goals of vendor contracts, which are almost always in conflict with end-user goals. I am standing on the side of end users, fighting against my former self. Enjoy.

### Section I

## **Initial Product Acquisition**

## 1

### Equipment and Application Acquisition

### INTRODUCTION

This chapter focuses on the relationships between choices of software applications, operating systems, and hardware from the standpoint of the manufacturer goals in the transaction(s). Buyers are often unaware of the monopolistic lock-ins and account control intentions of the providers, with the result that many purchases are not negotiated to the advantage of the buyer.

### APPLICATION SELECTION DICTATES HARDWARE SELECTION

Acquisition of technology products follows the application. At the consumer level, individuals buy products for their function, not the design of the circuit board. Businesses, industry, and government select the application software first, and then the hardware that goes with it. Information technology (IT) managers are often pressed to look at "technology" as a solution to business problems; however, the real fact is that the solution to business problems is an application, not technology. It does not matter how the bits and bytes are configured or if the processor capability is less than leading edge; if the application does not support the business.

It is only after a commitment has been made to a particular application that the hardware platform and associated operating system become meaningful. The long-term useful life of the equipment is tied to the planned useful life of the application. If the application decision is in error, the associated hardware and service contracts will also gather dust. This chapter is directed at unveiling nuances of lifecycle control points created by vendors that can only be effectively mitigated before the initial purchase. Careful attention to the initial product selection that goes beyond the application negotiation and into the tangled support functions of hardware, application license, and operating system (OS) license will bring long-term rewards.

The selection of products and associated support is a constant tugof-war between the buyer and the manufacturer, usually known as the original equipment manufacturer, or OEM. The buyer and the seller are not involved in a traditional purchase agreement, despite the appearance. Buyers do not own IT equipment in the same manner as they own a house or even as they own a vehicle. Nor is the license agreement for software a straightforward arrangement like a condo lease. The relationship between buyer and seller, or licensee and developer, is more like a marriage than a home purchase. Buyers should be negotiating IT purchases and license agreements just as carefully as they would a prenuptial agreement expecting divorce and conflict.

Figure 1.1 shows the OEM view of the sale or license agreement.<sup>1</sup> OEMs expect the user to enter into a long-term relationship where the OEM controls the useful life of the product with no exit points for the user. Although OEMs might not set out to produce products that are unstable and require repairs and software patches, they have little incentive to do otherwise. It is beneficial for OEMs to have reasons for the user to need their support.





#### Background

Manufacturers of hardware long ago recognized that the application drove the hardware. Outside of the mainframe environment, where initially most applications were custom built, the entire middle range of product sales were built around what used to be called "Turnkey" systems. The turnkey arrangement was perfect for the business that had no computerized applications—and thus no employees—capable of writing, customizing, or implementing the application. Most of the big names in applications today came from these roots.

Hardware manufacturers were not initially in the application software business. Nor did hardware companies sell or separately license the operating system. In order to sell to the neophyte buyer, hardware manufacturers made alliances with software developers, later called business partners or value-added resellers, to represent them to new or smaller users. Software vendors became the sales force for their own products and a free extension of the OEM sales force as an authorized partner. The software company got a healthy commission for selling the hardware. The operating system and the requisite maintenance contracts were also sold, and commissions paid, to the application partner.

It is for these historical reasons that the sales channel through which equipment is purchased usually dictates how both hardware maintenance (break–fix) and software maintenance (operating system and application support) are offered. This chapter discusses how different players in the supply chain have different offerings and how negotiations should be conducted to maximize effectiveness in support options.

Each type of acquisition method still involves navigating the natural tension between seller and buyer. The seller always wants to sell more equipment, more quickly, and drag along higher margin services at the highest possible price point. The buyer has goals of buying equipment that best supports the application requirements, at the best price, and with the least likelihood to need replacement ahead of the depreciation schedule.

#### **Application License Acquisition**

Buyers are commonly exasperated with trying to control maintenance costs for software, the price of which often exceeds the acquisition cost of the license over time. The reason is very simple: vendors have total control of the pricing of postwarranty support (maintenance). As with any other monopoly, this ability to dictate price and terms leads inevitably to taking advantage of the profit margin potential of the revenue stream. Maintenance is often the most profitable part of the license agreement. It is therefore essential that the initial negotiation for application licenses be extremely attentive to issues that might occur only in the far future.

There are only a handful of ways to control maintenance pricing, at the outset, through competitive replacement or a commitment to open systems. None of these tactics are perfect. Applications are completely protected by copyright law (which is itself consistent with both the Berne Convention and World Intellectual Property Organization [WIPO] international treaties). As "Intellectual Property (IP)," vendors are allowed to be monopolistic not only about the transfer of licenses but also about ongoing support. There is no legal (as of this writing) opportunity for competition for support unless the application vendor allows it.

Unfair terms and conditions in the "End User License Agreement (EULA)" are coming under scrutiny in general as digital rights are being considered. Copyright law, including international conventions, remains rooted in the predigital era. Even legislation passed at the end of the millennium to update the Copyright Code under the DMCA (Digital Millennium Copyright Act of 1998) seems archaic and dated. Some work has been done defining conceptual "User Rights" by the Gartner Group in 2010, none of which has been put into law.<sup>2</sup>

The ideal time to control maintenance costs in the future is to carefully negotiate the initial contract to include limits on future increases for maintenance costs. These negotiations usually focus on a percentage of the original list price, or original acquisition cost, rather than the need for maintenance in the future. The downside of this approach is that buyers are stuck with whatever percentages they negotiate, on top of which application vendors can inflict further financial burdens for upgrades to major "new" releases, requirements that all versions of the purchases be kept current, or other requirements such as linkages to hardware maintenance agreements.

The concept of the responsibility of the developer to deliver bug-free code is lost in the focus on discounting. Vendors have been allowed to own the negotiation over support and limit the discussion to discounting off list price for future support contracts. This is exclusively to the advantage of the vendor and always puts the buyer in the position of begging for discounts. The framework of the negotiation can change if buyers demand performance-based postwarranty support in the initial license negotiation. Once the focus of the negotiation is put on quality and performance on the part of the vendor, the entire spectrum of costs and rights can be discussed.

Defect and performance-based support is rooted in the idea that the vendor, either hardware or software, has an obligation to deliver fully functional products that meet the specifications as advertised. Patches and fixes are indications that code is buggy. Users need to stop acquiescing to the concept that buyers should pay for corrections to code that should not need correction.

Users have been led to believe that they should pay handsomely for support (maintenance) of flawed code because the patches and fixes are delivering improvements in the product. This is unlikely to be the case, although there may be exceptions. There is more than a semantic difference between an "update" and an "upgrade." An *update* is likely to be double-speak for a bug fix, but sounds far more positive. An upgrade should provide a new and valuable function that did not previously exist. An upgrade should be something that the user might want to purchase, separately. Sadly, many software and hardware vendors blithely mix the wording of updates and upgrades simply to hide the weaknesses in their product stability.

The costs and rights for users to buy patches and fixes to flawed code can, and must, be separated from enhancements and customizations. Users can begin the process of accountability by evaluating the patches and fixes that are disseminated for existing licenses for applicability, impact, and separating upgrades from patches. Much of this work is already being done by support staff to discern which patches must be urgently applied. Adding the layer of intelligence to differentiate between welcomed and unwelcomed updates will enormously increase the negotiating power of the user in future contracts.

### **Going Naked**

It is an underutilized but viable option to drop maintenance (support) where no meaningful changes are needed and freeze the system, both hardware and software, at a stable release. Vendors use the term "going naked" to infer the undesirable vulnerability of dropping vendor support. The key to confidence in dropping "support" is to evaluate the types of patches and fixes being issued and consider if the improvements being made are valuable. Only the user can judge if the problems being fixed are meaningful.

In many cases, application systems patching follows the pattern of hardware systems patching. The initial release is full of bugs, which require a high level of interaction on the part of the user and the developer to fully diagnose and then fix. Within a few months, or a year, most major or common bugs have been addressed and the development team is able to move forward with completing other items on the development timetable. As new features are created, the bugs in the new feature, or the interaction of the new feature and other products, are fixed. The fewer the enhancements, the slower the pace of patches, and the less likely the new patches to the new features are widely applicable.

It is crucial to keep the originally negotiated terms and conditions in mind whenever any vendor, not just an application vendor, changes the policies involving maintenance. For example, Oracle, immediately following their acquisition of Sun Microsystems in 2010, abruptly changed its policies for both hardware and OS maintenance for prior Sun products.<sup>3</sup> Sun users were presented with a new set of rules requiring that they sign a new maintenance agreement for hardware maintenance for all Sun products within the enterprise or have no products on Sun maintenance. There were many other restrictions put forth. Those users who reminded Oracle of the terms and conditions of their original agreement were successful in forestalling the impact of the new policy.

Once the original license agreement is in place, and if the maintenance contract is not viable in the eyes of the licensor, then the only remaining options are to rip out the application or to outsource the hosting of the system to a more cost-effective host. It is not in the least surprising that the move toward open-systems products and hosted (cloud) offerings have been the salvation of the locked-in application user.

Moving the application from an in-house system to a hosted system does not remove license obligations if the licenses are not issued per serial number or per processor. The largest savings in moving to a hosted solution for specific applications is the dramatically reduced costs of the license and maintenance contract for the OS and the hardware, which are shared costs across multiple users.

### **Operating System Acquisition**

Selection of an operating system is unusual, as the application requirements are operating system and hardware specific. The exceptions are open-systems platforms where users have choices between different versions of common OS, such as the Linux OS. Using the Linux base, several competitors have developed their own versions of Linux, which include enhancements and features not available in the base version, and offer their own support contracts for their versions. Unless the author offers an open domain version of the OS, there are no independent support options for any OS. As with application licenses, the OS license is protected by copyright and the terms and conditions of use are not mitigated by competition.

Operating systems are most often licensed either per serial number or per processor. Different fees are commonly charged for different types of processors, with a lower fee for the OS on a less powerful machine, and a much higher fee for the high-speed version of the same architecture. The explosion of large multiprocessor servers has given rise to per processor licensing as the price increase for the OS license from a single-processor version to a new version often made the new purchase unaffordable, which dampened sales for hardware. In cases where hardware manufacturers also license their OS, the platform (hardware) sale still tends to lead the sale in order to drag the associated licenses, custom services, and postwarranty support agreements.

In addition to the OS, there are accessory software products known as "Systems Software" that attach directly to the OS used to perform functions not included in the OS or included in a lackluster way.<sup>4</sup> Each of these products has its own licensing models, which do not always follow the OS license model. Before buying any new products or making major upgrades, buyers need to ascertain the price of the impact on the license costs, or required upgrades, to all products licensed with the OS, or accidently put themselves in a terrible bargaining position after the fact.

### ACQUISITION MODEL: DIRECT FROM THE ORIGINAL EQUIPMENT MANUFACTURER

Businesses large enough to be dealing directly with the original equipment manufacturer (OEM) will have access to pricing and options not generally available to the smaller buyer. This does not mean that these are the best terms without investing some time to explore competitive options. Competition is essential to the sharpest pricing and best service. Competition also keeps the door open for innovation and exchange of ideas leading to better products and processes.

OEMs try to avoid competition everywhere possible in order to preserve profit margins. In highly competitive consumer markets, for example, the

personal printer market, product sales may be treated as "loss leaders" for higher margin revenues from other support activities, such as ink and toner sales. In cases where OEMs may have strong competition from their own installed base, the sales force will be given strong incentives to rapidly displace older installed equipment, even if that means that the vendor captive leasing company backing the equipment sale has to lose money. Methods of distribution are lionized, used equipment purchases blocked, and lawsuits are filed by OEMs alleging copyright infringement for opening the covers of machines. All of this is done to plump the margins for the OEM.

### **Platforms as Monopolies**

OEMs with any competition for the initial platform purchase are extremely creative in removing opportunities for a platform loss in the future when upgrades are needed. This is simply good business sense. The work needed to win the initial sale is a substantial investment for vendors and they intend to benefit from subsequent growth. Profit margins are always thinner for the highly competitive sale than for the future upgrade or for addons such as postwarranty service.

Creating the postpurchase monopoly is done with a combination of lock-ins that begin with the hardware itself, flow downstream into financial incentives, training and education incentives, and even operational



**FIGURE 1.2** Hardware monopoly diagram.

lock-ins (see Figure 1.2). Before engaging in a platform decision, managers that take the time to plan ahead for future growth and upgrades will be able to control their futures when others will be completely tied to the OEM.

#### Multiyear Commitments

The multiyear discount package offering is common. The buyer is enticed to sign a long-term agreement covering the platform purchase, associated licenses and maintenance, and all upgrades for a period of time, often tied to the expiration of the extended product warranty. During that time, the buyer has negotiated a fixed discount level off of "list" price as in the original agreement. The discount is likely to be initially attractive compared with a single point in time offer. Buyers are attracted to this as a way to lock-in a deep discount and also avoid the effort of subsequent negotiations for the same product category. The vendor is always happy with this arrangement, which trades only a small reduction in price for a multiyear platform, services, and support monopoly. Buyers are rarely, if ever, getting a competitive discount in subsequent years.

The reason the multiyear negotiated discount is such a desirable deal for vendors is based on the normal price erosion for products as they age in their marketing cycle (see Figure 1.3). Most products are only manufactured for a limited period of time, and new versions rapidly replace either parts or whole platforms. Within 12 to 18 months of product release, used machines are generally available or totally new models with superior price



**FIGURE 1.3** Package pricing over time.

performance. The deep discount of today is always higher than closeout pricing tomorrow. Buyers that have locked themselves into the multiyear contract cannot renegotiate the price point even if they are offered the new model.

Another fallacy in buyer logic is the attraction of a big discount off "list." List price for many vendors is totally arbitrary. Just as with big "sales" of 50% to 70% off list in the department stores, the vendor is not really making a special deal. The price points are set to entice the sale. True closeouts are often disposed through other distribution channels (see the section, "Gray Market") to hide the true drop in value during the product lifecycle.

Buyers that eschew the multiyear package are almost always rewarded with a lower total cost of ownership, vastly more flexibility in future selections, and far more attention from the vendor. This takes time, and some users will prefer to trade time with convenience but they will pay for the privilege later.

Within the package of attractive advantages to the buyer for these multiyear deals are special "deals" on financing, education, and maintenance. Many buyers are lulled into the sense of special treatment for these extras, when in fact all of these extras are individually negotiable and may have limited true value to the buyer.

#### **Operating System and Machine Code License Lock-Ins**

An increasingly popular tactic used by vendors to control the postpurchase environment is the tying of separate licenses (machine code or operating systems) to continued hardware support (or vice versa). Customers often think that if such practices were illegal, they could not be offered. The fact is that many of the tie-ins have not been litigated at all including antitrust actions. Most lawyers will point out that once the customer has signed the contract, no matter how ridiculous, the contract holds. Users must therefore be particularly wary of these practices, as they will not have the opportunity to make changes in the future.

Licenses for machine code and other forms of embedded software are among the most negotiable and least understood by both user and vendor sales force. Machine code, which is covered extensively in Chapter 8, was traditionally provided without a license because it was part of the hardware and thus supported in the same manner as a product failure. Over the past decade, vendors have discovered that buyers will tolerate extensive restrictions on anything labeled "code." This most likely stems from aggressive license audits on the part of applications providers. In turn, this has caused, legitimately, a strong fear of anything that might be construed as a violation of IP rights.

As a result, machine code occupies a gray area of contracting. Licensing machine code and charging for updates is perfect account control for vendors and has no advantages for buyers whatsoever. The buyer loses the ability to separate a hardware purchase from a license, and cannot continue to use the equipment without indefinite and uncontrollable license pricing and license maintenance charges from the OEM.

Machines that are combination products of hardware and application software, such as many networking and storage products, are even more subject to having intertwined and expensive postwarranty license and maintenance requirements. Many products in these categories were initially produced with these arrangements and users entered into the license agreements with full knowledge, making it nearly impossible to change vendor policies through negotiation. However, new buyers can search for opportunities to escape to vendors with more favorable terms and conditions, an exercise that is always beneficial.

Should users accept the concept that machine code (embedded software) is licensed software, they can never control the useful life of their purchases.

Users are not helpless against these tactics because of the power of the purchase order. All OEMs have to book new revenue to meet volume expectations even if margins for product sales are uncomfortably low. Most OEMs understand that they can shift margins from the product sale into longer term but more profitable maintenance and support agreements, which are rarely examined with the same intensity as the original purchase. There are many users that have capitulated before attempting to negotiate on service, assuming that whatever price pressure they bring will only push prices higher for something else. This is only true for those that do not make the attempt. It has been repeatedly proven that OEMs that face competitive pressure grant greater discounts to buyers than those that do not.

Users that withhold orders until all terms and conditions (and prices) are met have tremendous leverage that should not be wasted. Large companies are often aware of their purchasing power but are not immune to being abused.

### **Strategic Partnerships**

One of the most pernicious arrangements for the large buyer is the "Strategic Partnership." This wording implies some kind of special

relationship between the parties unavailable to others. Such "partnerships" are rarely as special as implied; it is often the case that the "strategic" element of the arrangement is a higher level of sales attention on the part of the OEM. Without the equity stake essential to a true partnership, all the negatives of the buyer–seller relationship remain intact but competition is squashed.

Any end user involved in a *strategic partnership* has put up a warning notice to all competitors that they are not welcome. The OEM is likely to become complacent on pricing and on support. Employees of the end user will be increasingly tolerant of small problems because they like or are comfortable with OEM staff. The OEM is so entrenched in the account that extrication in favor of another OEM is a multiyear process requiring significant upper management support.

Large buyers and large OEMs in these relationships tend to treat each other at the board of directors level. Pressure from the board often influences the selection of OEMs, influences the selection of leasing and banking support, and is often influential in directing consulting and "services" toward connected providers. Upsetting these relationships is difficult once established.

Competition and innovation have to be deliberately nurtured in such environments. Small or even medium-sized businesses lack the board level connections with the vendor to generate favors. Most buyers must rely upon using the options available to the vendor sales team in response to competitive pressure. The main competitive pressure point available to normal buyers is to make sure transactions are taken to the "Win Room" and at the same time presenting "Credible Competition."

#### The Win Room

All OEMs want to respond to competitive pressures to "win the footprint." The most common mechanism used by major OEMs is to bring highly large competitive transactions to a team of managers authorized to cut special deals for the initial sale. Both OEM direct field sales and authorized partners typically have access to this function.

The "Win Room" function is only available for winning a competitive platform battle and is not an option for repeat or upgrade business. It is therefore essential that all elements of the transaction as well as future requirements for support, upgrades, limits on increases, and so on, must be dealt with at this time. The sales team is totally interested in getting the