The
Practice of
System and Network
Administration

Second Edition

Thomas A. Limoncelli · Christina J. Hogan · Strata R. Chalup

Updated and Revised

The Practice of System and Network Administration

Second Edition

This page intentionally left blank

The Practice of System and Network Administration

Second Edition

Thomas A. Limoncelli
Christina J. Hogan
Strata R. Chalup

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales, (800) 382-3419, corpsales@pearsontechgroup.com

For sales outside the United States please contact:

International Sales, international@pearsoned.com

Visit us on the Web: www.awprofessional.com

Contents at a Glance

Contents

Preface

Our goal for this book has been to write down everything we've learned from our mentors and to add our real-world experiences. These things are beyond what the manuals and the usual system administration books teach.

This book was born from our experiences as SAs in a variety of organizations. We have started new companies. We have helped sites to grow. We have worked at small start-ups and universities, where lack of funding was an issue. We have worked at midsize and large multinationals, where mergers and spin-offs gave rise to strange challenges. We have worked at fast-paced companies that do business on the Internet and where high-availability, high-performance, and scaling issues were the norm. We've worked at slow-paced companies at which high tech meant cordless phones. On the surface, these are very different environments with diverse challenges; underneath, they have the same building blocks, and the same fundamental principles apply.

This book gives you a framework—a way of thinking about system administration problems—rather than narrow how-to solutions to particular problems. Given a solid framework, you can solve problems every time they appear, regardless of the operating system (OS), brand of computer, or type of environment. This book is unique because it looks at system administration from this holistic point of view; whereas most other books for SAs focus on how to maintain one particular product. With experience, however, all SAs learn that the big-picture problems and solutions are largely independent of the platform. This book will change the way you approach your work as an SA.

The principles in this book apply to all environments. The approaches described may need to be scaled up or down, depending on your environment, but the basic principles still apply. Where we felt that it might not be obvious how to implement certain concepts, we have included sections that illustrate how to apply the principles at organizations of various sizes.

This book is not about how to configure or debug a particular OS and will not tell you how to recover the shared libraries or DLLs when someone accidentally moves them. Some excellent books cover those topics, and we refer you to many of them throughout. Instead, we discuss the principles, both basic and advanced, of good system administration that we have learned through our own and others' experiences. These principles apply to all OSs. Following them *well* can make your life a lot easier. If you improve the way you approach problems, the benefit will be multiplied. Get the fundamentals right, and everything else falls into place. If they aren't done well, you will waste time repeatedly fixing the same things, and your customers[1] will be unhappy because they can't work effectively with broken machines.

Who Should Read This Book

This book is written for system administrators at all levels. It gives junior SAs insight into the bigger picture of how sites work, their roles in the organizations, and how their careers can progress. Intermediate SAs will learn how to approach more complex problems and how to improve their sites and make their jobs easier and their customers happier. Whatever level you are at, this book will help you to understand what is behind your day-to-day work, to learn the things that you can do now to save time in the future, to decide policy, to be architects and designers, to plan far into the future, to negotiate with vendors, and to interface with management. These are the things that concern senior SAs. None of them are listed in an OS's manual. Even senior SAs and systems architects can learn from our experiences and those of our colleagues, just as we have learned from each other in writing this book. We also cover several management topics for SA trying to understand their managers, for SAs who aspire to move into management, and for SAs finding themselves doing more and more management without the benefit of the title.

Throughout the book, we use examples to illustrate our points. The examples are mostly from medium or large sites, where scale adds its own problems. Typically, the examples are generic rather than specific to a particular OS; where they are OS-specific, it is usually Unix or Windows.

One of the strongest motivations we had for writing this book is the understanding that the problems SAs face are the same across all OSs. A new

---

1. Throughout the book, we refer to the end users of our systems as *customers* rather than *users*. A detailed explanation of why we do this is in Section 31.1.2.

OS that is significantly different from what we are used to can seem like a black box, a nuisance, or even a threat. However, despite the unfamiliar interface, as we get used to the new technology, we eventually realize that we face the same set of problems in deploying, scaling, and maintaining the new OS. Recognizing that fact, knowing what problems need solving, and understanding how to approach the solutions by building on experience with other OSs lets us master the new challenges more easily.

We want this book to change your life. We want you to become so successful that if you see us on the street, you'll give us a great big hug.

Basic Principles

If we've learned anything over the years, it is the importance of simplicity, clarity, generality, automation, communication, and doing the basics first. These six principles are recurring themes in this book.

1. *Simplicity* means that the smallest solution that solves the entire problem is the best solution. It keeps the systems easy to understand and reduces complex component interactions that can cause debugging nightmares.

2. *Clarity* means that the solution is straightforward. It can be easily explained to someone on the project or even outside the project. Clarity makes it easier to change the system, as well as to maintain and debug it. In the system administration world, it's better to write five lines of understandable code than one line that's incomprehensible to anyone else.

3. *Generality* means that the solutions aren't inherently limited to a particular case. Solutions can be reused. Using vendor-independent open standard protocols makes systems more flexible and makes it easier to link software packages together for better services.

4. *Automation* means using software to replace human effort. Automation is critical. Automation improves repeatability and scalability, is key to easing the system administration burden, and eliminates tedious repetitive tasks, giving SAs more time to improve services.

5. *Communication* between the right people can solve more problems than hardware or software can. You need to communicate well with other SAs and with your customers. It is your responsibility to initiate communication. Communication ensures that everyone is working

toward the same goals. Lack of communication leaves people concerned and annoyed. Communication also includes documentation. Documentation makes systems easier to support, maintain, and upgrade. Good communication and proper documentation also make it easier to hand off projects and maintenance when you leave or take on a new role.

6. *Basics first* means that you build the site on strong foundations by identifying and solving the basic problems before trying to attack more advanced ones. Doing the basics first makes adding advanced features considerably easier and makes services more robust. A good basic infrastructure can be repeatedly leveraged to improve the site with relatively little effort. Sometimes, we see SAs making a huge effort to solve a problem that wouldn't exist or would be a simple enhancement if the site had a basic infrastructure in place. This book will help you identify what the basics are and show you how the other five principles apply. Each chapter looks at the basics of a given area. Get the fundamentals right, and everything else will fall into place.

These principles are universal. They apply at all levels of the system. They apply to physical networks and to computer hardware. They apply to all operating systems running at a site, all protocols used, all software, and all services provided. They apply at universities, nonprofit institutions, government sites, businesses, and Internet service sites.

What Is an SA?

If you asked six system administrators to define their jobs, you would get seven different answers. The job is difficult to define because system administrators do so many things. An SA looks after computers, networks, and the people who use them. An SA may look after hardware, operating systems, software, configurations, applications, or security. A system administrator influences how effectively other people can or do use their computers and networks.

A system administrator sometimes needs to be a business-process consultant, corporate visionary, janitor, software engineer, electrical engineer, economist, psychiatrist, mindreader, and, occasionally, a bartender.

As a result, companies calls SAs different names. Sometimes, they are called network administrators, system architects, system engineers, system programmers, operators and so on.

This book is for "all of the above."

We have a very general definition of system administrator: one who manages computer and network systems on behalf of another, such as an employer or a client. SAs are the people who make things work and keep it all running.

Explaining What System Administration Entails

It's difficult to define system administration, but trying to explain it to a nontechnical person is even more difficult, especially if that person is your mom. Moms have the right to know how their offspring are paying their rent. A friend of Christine Hogan's always had trouble explaining to his mother what he did for a living and ended up giving a different answer every time she asked. Therefore, she kept repeating the question every couple of months, waiting for an answer that would be meaningful to her. Then he started working for WebTV. When the product became available, he bought one for his mom. From then on, he told her that he made sure that her WebTV service was working and was as fast as possible. She was very happy that she could now show her friends something and say, "That's what my son does!"

System Administration Matters

System administration matters because computers and networks matter. Computers are a lot more important than they were years ago. What happened?

The widespread use of the Internet, intranets, and the move to a web-centric world has redefined the way companies depend on computers. The Internet is a 24/7 operation, and sloppy operations can no longer be tolerated. Paper purchase orders can be processed daily, in batches, with no one the wiser. However, there is an expectation that the web-based system that does the process will be available all the time, from anywhere. Nightly maintenance windows have become an unheard-of luxury. That unreliable machine room power system that caused occasional but bearable problems now prevents sales from being recorded.

Management now has a more realistic view of computers. Before they had PCs on their desktops, most people's impressions of computers were based on how they were portrayed in film: big, all-knowing, self-sufficient, miracle machines. The more people had direct contact with computers, the more realistic people's expectations became. Now even system administration itself is portrayed in films. The 1993 classic *Jurassic Park* was the first mainstream movie to portray the key role that system administrators play in large systems.

The movie also showed how depending on one person is a disaster waiting to happen. IT is a team sport. If only Dennis Nedry had read this book.

In business, nothing is important unless the CEO feels that it is important. The CEO controls funding and sets priorities. CEOs now consider IT to be important. Email was previously for nerds; now CEOs depend on email and notice even brief outages. The massive preparations for Y2K also brought home to CEOs how dependent their organizations have become on computers, how expensive it can be to maintain them, and how quickly a purely technical issue can become a serious threat. Most people do not think that they simply "missed the bullet" during the Y2K change but that problems were avoided thanks to tireless efforts by many people. A CBS Poll shows 63 percent of Americans believe that the time and effort spent fixing potential problems was worth it. A look at the news lineups of all three major network news broadcasts from Monday, January 3, 2000, reflects the same feeling.

Previously, people did not grow up with computers and had to cautiously learn about them and their uses. Now more and more people grow up using computers, which means that they have higher expectations of them when they are in positions of power. The CEOs who were impressed by automatic payroll processing are soon to be replaced by people who grew up sending instant messages and want to know why they can't do all their business via text messaging.

Computers matter more than ever. If computers are to work and work well, system administration matters. We matter.

Organization of This Book

This book has the following major parts:

- Part I: Getting Started. This is a long book, so we start with an overview of what to expect (Chapter 1) and some tips to help you find enough time to read the rest of the book (Chapter 2).
- Part II: Foundation Elements. Chapters 3–14 focus on the foundations of IT infrastructure, the hardware and software that everything else depends on.
- Part III: Change Processes. Chapters 15–21 look at how to make changes to systems, starting with fixing the smallest bug to massive reorganizations.

- Part IV: Providing Services. Chapters 22–29 offer our advice on building seven basic services, such as email, printing, storage, and web services.
- Part V: Management Practices. Chapters 30–36 provide guidance— whether or not you have "manager" in your title.
- The two appendixes provide an overview of the positive and negative roles that SAs play and a list of acronyms used in the book.

Each chapter discusses a separate topic; some topics are technical, and some are nontechnical. If one chapter doesn't apply to you, feel free to skip it. The chapters are linked, so you may find yourself returning to a chapter that you previously thought was boring. We won't be offended.

Each chapter has two major sections. *The Basics* discusses the essentials that you simply have to get right. Skipping any of these items will simply create more work for you in the future. Consider them investments that pay off in efficiency later on. *The Icing* deals with the cool things that you can do to be spectacular. Don't spend your time with these things until you are done with the basics. We have tried to drive the points home through anecdotes and case studies from personal experience. We hope that this makes the advice here more "real" for you. Never trust salespeople who don't use their own products.

What's New in the Second Edition

We received a lot of feedback from our readers about the first edition. We spoke at conferences and computer user groups around the world. We received a lot of email. We listened. We took a lot of notes. We've smoothed the rough edges and filled some of the major holes.

The first edition garnered a lot of positive reviews and buzz. We were very honored. However, the passing of time made certain chapters look passé.

The first edition, in bookstores August 2001, was written mostly in 2000. Things were very different then. At the time, things were looking pretty grim as the dot-com boom had gone bust. Windows 2000 was still new, Solaris was king, and Linux was popular only with geeks. Spam was a nuisance, not an industry. Outsourcing had lost its luster and had gone from being the corporate savior to a late-night comedy punch line. Wikis were a research idea, not the basis for the world's largest free encyclopedia. Google was neither a household name nor a verb. Web farms were rare, and "big sites" served millions of hits per day, not per hour. In fact, we didn't have a chapter

on running web servers, because we felt that all one needed to know could be inferred by reading the right combination of the chapters: Data Centers, Servers, Services, and Service Monitoring. What more could people need?

My, how things have changed!

Linux is no longer considered a risky proposition, Google is on the rise, and offshoring is the new buzzword. The rise of India and China as economic superpowers has changed the way we think about the world. AJAX and other Web 2.0 technologies have made the web applications exciting again.

Here's what's new in the book:

- *Updated chapters:* Every chapter has been updated and modernized and new anecdotes added. We clarified many, many points. We've learned a lot in the past five years, and all the chapters reflect this. References to old technologies have been replaced with more relevant ones.
- *New chapters:*
    - Chapter 9: Documentation
    - Chapter 25: Data Storage
    - Chapter 29: Web Services
- *Expanded chapters:*
    - The first edition's Appendix B, which had been missed by many readers who didn't read to the end of the book, is now Chapter 1: What to Do When . . . .
    - The first edition's Do These First section in the front matter has expanded to become Chapter 2: Climb Out of the Hole.
- *Reordered table of contents:*
    - Part I: Getting Started: introductory and overview material
    - Part II: Foundation Elements: the foundations of any IT system
    - Part III: Change Processes: how to make changes from the smallest to the biggest
    - Part IV: Providing Services: a catalog of common service offerings
    - Part V: Management Practices: organizational issues

What's Next

Each chapter is self-contained. Feel free to jump around. However, we have carefully ordered the chapters so that they make the most sense if you read the book from start to finish. Either way, we hope that you enjoy the book. We have learned a lot and had a lot of fun writing it. Let's begin.

<div align="right">

Thomas A. Limoncelli
Google, Inc.
tom@limoncelli.org

Christina J. Hogan
BMW Sauber F1 Team
chogan@chogan.com

Strata R. Chalup
Virtual.Net, Inc.
strata@virtual.net

</div>

P.S. Books, like software, always have bugs. For a list of updates, along with news and notes, and even a mailing list you can join, please visit our web site: www.EverythingSysAdmin.com.

This page intentionally left blank

Acknowledgments

Acknowledgments for the First Edition

We can't possibly thank everyone who helped us in some way or another, but that isn't going to stop us from trying. Much of this book was inspired by Kernighan and Pike's *The Practice of Programming* (Kernighan and Pike 1999) and John Bentley's second edition of *Programming Pearls* (Bentley 1999).

We are grateful to Global Networking and Computing (GNAC), Synopsys, and Eircom for permitting us to use photographs of their data center facilities to illustrate real-life examples of the good practices that we talk about.

We are indebted to the following people for their helpful editing: Valerie Natale, Anne Marie Quint, Josh Simon, and Amara Willey.

The people we have met through USENIX and SAGE and the LISA conferences have been major influences in our lives and careers. We would not be qualified to write this book if we hadn't met the people we did and learned so much from them.

Dozens of people helped us as we wrote this book—some by supplying anecdotes, some by reviewing parts of or the entire book, others by mentoring us during our careers. The only fair way to thank them all is alphabetically and to apologize in advance to anyone that we left out: Rajeev Agrawala, Al Aho, Jeff Allen, Eric Anderson, Ann Benninger, Eric Berglund, Melissa Binde, Steven Branigan, Sheila Brown-Klinger, Brent Chapman, Bill Cheswick, Lee Damon, Tina Darmohray, Bach Thuoc (Daisy) Davis, R. Drew Davis, Ingo Dean, Arnold de Leon, Jim Dennis, Barbara Dijker, Viktor Dukhovni, Chelle-Marie Ehlers, Michael Erlinger, Paul Evans, Rémy Evard, Lookman Fazal, Robert Fulmer, Carson Gaspar, Paul Glick, David "Zonker" Harris, Katherine "Cappy" Harrison, Jim Hickstein, Sandra Henry-Stocker, Mark Horton, Bill "Whump" Humphries, Tim Hunter, Jeff Jensen, Jennifer Joy, Alan Judge, Christophe Kalt, Scott C. Kennedy, Brian Kernighan, Jim Lambert, Eliot Lear,

Acknowledgments for the Second Edition

About the Authors

Tom, Christine, and Strata know one another through attending USENIX conferences and being actively involved in the system administration community. It was at one of these conferences that Tom and Christine first spoke about collaborating on this book. Strata and Christine were coworkers at Synopsys and GNAC, and coauthored Chalup, Hogan et al. (1998).

Thomas A. Limoncelli

Tom is an internationally recognized author and speaker on system administration, time management, and grass-roots political organizing techniques. A system administrator since 1988, he has worked for small and large companies, including Google, Cibernet Corp, Dean for America, Lumeta, AT&T, Lucent/Bell Labs, and Mentor Graphics. At Google, he is involved in improving how IT infrastructure is deployed at new offices. When AT&T trivested into AT&T, Lucent, and NCR, Tom led the team that split the Bell Labs computing and network infrastructure into the three new companies.

In addition to the first and second editions of this book, his published works include *Time Management for System Administration* (2005), and papers on security, networking, project management, and personal career management. He travels to conferences and user groups frequently, often teaching tutorials, facilitating workshops, presenting papers, or giving invited talks and keynote speeches.

Outside of work, Tom is a grassroots civil-rights activist who has received awards and recognition on both state and national levels. Tom's first published paper (Limoncelli 1997) extolled the lessons SAs can learn from activists. Tom doesn't see much difference between his work and activism careers—both are about helping people.

He holds a B.A. in computer science from Drew University. He lives in Bloomfield, New Jersey.

For their community involvement, Tom and Christine shared the 2005 Outstanding Achievement Award from USENIX/SAGE.

Christina J. Hogan

Christine's system administration career started at the Department of Mathematics in Trinity College, Dublin, where she worked for almost 5 years. After that, she went in search of sunshine and moved to Sicily, working for a year in a research company, and followed that with 5 years in California.

She was the security architect at Synopsys for a couple of years before joining some friends at GNAC a few months after it was founded. While there, she worked with start-ups, e-commerce sites, biotech companies, and large multinational hardware and software companies. On the technical side, she focused on security and networking, working with customers and helping GNAC establish its data center and Internet connectivity. She also became involved with project management, customer management, and people management. After almost 3 years at GNAC, she went out on her own as an independent security consultant, working primarily at e-commerce sites.

Since then, she has become a mother and made a career change: she now works as an aerodynamicist for the BMW Sauber Formula 1 Racing Team. She has a Ph.D. in aeronautical engineering from Imperial College, London; a B.A. in mathematics and an M.Sc. in computer science from Trinity College, Dublin; and a Diploma in legal studies from the Dublin Institute of Technology.

Strata R. Chalup

Strata is the owner and senior consultant of Virtual.Net, Inc., a strategic and best-practices IT consulting firm specializing in helping small to midsize firms scale their IT practices as they grow. During the first dot-com boom, Strata architected scalable infrastructures and managed some of the teams that built them for such projects as talkway.net, the Palm VII, and mac.com. Founded as a sole proprietorship in 1993, Virtual.Net was incorporated in 2005. Clients have included such firms as Apple, Sun, Cimflex Teknowledge, Cisco, McAfee, and Micronas USA.

Strata joined the computing world on TOPS-20 on DEC mainframes in 1981, then got well and truly sidetracked onto administering Unix by 1983, with Ultrix on the VAX 11-780, Unisys on Motorola 68K micro systems, and a dash of Minix on Intel thrown in for good measure. She has the

unusual perspective of someone who has been both a user and an administrator of Internet services since 1981 and has seen much of what we consider the modern Net evolve, sometimes from a front-row seat. An early adopter and connector, she was involved with the early National Telecommunications Infrastructure Administration (NTIA) hearings and grant reviews from 1993–1995 and demonstrated the emerging possibilities of the Internet in 1994, creating NTIA's groundbreaking virtual conference. A committed futurist, Strata avidly tracks new technologies for collaboration and leverages them for IT and management.

Always a New Englander at heart, but marooned in California with a snow-hating spouse, Strata is an active gardener, reader of science fiction/fantasy, and emergency services volunteer in amateur radio (KF6NBZ). She is SCUBA-certified but mostly free dives and snorkles. Strata has spent a couple of years as a technomad crossing the country by RV, first in 1990 and again in 2002, consulting from the road. She has made a major hobby of studying energy-efficient building construction and design, including taking owner-builder classes, and really did grow up on a goat farm.

Unlike her illustrious coauthors, she is an unrepentent college dropout, having left MIT during her sophmore year. She returned to manage the Center for Cognitive Science for several years, and to consult with the EECS Computing Services group, including a year as postmaster@mit-eddie, before heading to Silicon Valley.

This page intentionally left blank

Chapter 1

What to Do When . . .

In this chapter, we pull together the various elements from the rest of the book to provide an overview of how they can be used to deal with everyday situations or to answer common questions system administrators (SAs) and managers often have.

1.1 Building a Site from Scratch

- Think about the organizational structure you need—Chapter 30.
- Check in with management on the business priorities that will drive implementation priorities.
- Plan your namespaces carefully—Chapter 8.
- Build a rock-solid data center—Chapter 6.
- Build a rock-solid network designed to grow—Chapter 7.
- Build services that will scale—Chapter 5.
- Build a software depot, or at least plan a small directory hierarchy that can grow into a software depot—Chapter 28.
- Establish your initial core application services:
  - Authentication and authorization—Section 3.1.3
  - Desktop life-cycle management—Chapter 3
  - Email—Chapter 23
  - File service, backups—Chapter 26
  - Network configuration—Section 3.1.3
  - Printing—Chapter 24
  - Remote access—Chapter 27

1.2 Growing a Small Site

- Provide a helpdesk—Chapter 13.
- Establish checklists for new hires, new desktops/laptops, and new servers—Section 3.1.1.5.
- Consider the benefits of a network operations center (NOC) dedicated to monitoring and coordinating network operations—Chapter 22.
- Think about your organization and whom you need to hire, and provide service statistics showing open and resolved problems—Chapter 30.
- Monitor services for both capacity and availability so that you can predict when to scale them—Chapter 22.
- Be ready for an influx of new computers, employees, and SAs—See Sections 1.23, 1.24, and 1.25.

1.3 Going Global

- Design your wide area network (WAN) architecture—Chapter 7.
- Follow three cardinal rules: scale, scale, and scale.
- Standardize server times on Greenwich Mean Time (GMT) to maximize log analysis capabilities.
- Make sure that your helpdesk really is 24/7. Look at ways to leverage SAs in other time zones—Chapter 13.
- Architect services to take account of long-distance links—usually lower bandwidth and less reliable—Chapter 5.
- Qualify applications for use over high-latency links—Section 5.1.2.
- Ensure that your security and permissions structures are still adequate under global operations.

1.4 Replacing Services

- Be conscious of the process—Chapter 18.
- Factor in both network dependencies and service dependencies in transition planning.
- Manage your Dynamic Host Configuration Protocol (DHCP) lease times to aid the transition—Section 3.1.4.1.

- Don't hard-code server names into configurations, instead, hard-code aliases that move with the service—Section 5.1.6.
- Manage your DNS time-to-live (TTL) values to switch to new servers—Section 19.2.1.

1.5 Moving a Data Center

- Schedule windows unless everything is fully redundant and you can move first half of a redundant pair and then the other—Chapter 20.
- Make sure that the new data center is properly designed for both current use and future expansion—Chapter 6.
- Back up every file system of any machine before it is moved.
- Perform a fire drill on your data backup system—Section 26.2.1.
- Develop test cases before you move, and test, test, test everything after the move is complete—Chapter 18.
- Label every cable before it is disconnected—Section 6.1.7.
- Establish minimal services—redundant hardware—at a new location with new equipment.
- Test the new environment—networking, power, uninterruptable power supply (UPS), heating, ventilation, air conditioning (HVAC), and so on—before the move begins—Chapter 6, especially Section 6.1.4.
- Identify a small group of customers to test business operations with the newly moved minimal services, then test sample scenarios before moving everything else.
- Run cooling for 48–72 hours, and then replace all filters before occupying the space.
- Perform a dress rehearsal—Section 18.2.5.

1.6 Moving to/Opening a New Building

- Four weeks or more in advance, get access to the new space to build the infrastructure.
- Use radios or walkie-talkies for communicating inside the building—Chapter 6 and Section 20.1.7.3.

- Use a personal digital assistant (PDA) or nonelectronic organizer—Section 32.1.2.
- Order WAN and Internet service provider (ISP) network connections 2–3 months in advance.
- Communicate to the powers that be that WAN and ISP connections will take months to order and must be done soon.
- Prewire the offices with network jacks during, not after, construction—Section 7.1.4.
- Work with a moving company that can help plan the move.
- Designate one person to keep and maintain a master list of everyone who is moving and his or her new office number, cubicle designation, or other location.
- Pick a day on which to freeze the master list. Give copies of the frozen list to the moving company, use the list for printing labels, and so on. If someone's location is to be changed after this date, don't try to chase down and update all the list copies that have been distributed. Move the person as the master list dictates, and schedule a second move for that person after the main move.
- Give each person a sheet of 12 labels preprinted with his or her name and new location for labeling boxes, bags, and personal computer (PC). (If you don't want to do this, at least give people specific instructions as to what to write on each box so it reaches the right destination.)
- Give each person a plastic bag big enough for all the PC cables. Technical people can decable and reconnect their PCs on arrival; technicians can do so for nontechnical people.
- Always order more boxes than you think you'll be moving.
- Don't use cardboard boxes; instead, use plastic crates that can be reused.

1.7 Handling a High Rate of Office Moves

- Work with facilities to allocate only one move day each week. Develop a routine around this schedule.
- Establish a procedure and a form that will get you all the information you need about each person's equipment, number of network and telephone connections, and special needs. Have SAs check out nonstandard equipment in advance and make notes.

- Connect and test network connections ahead of time.

- Have customers power down their machines before the move and put all cables, mice, keyboards, and other bits that might get lost into a marked box.

- Brainstorm all the ways that some of the work can be done by the people moving. Be careful to assess their skill level; maybe certain people shouldn't do anything themselves.

- Have a moving company move the equipment, and have a designated SA move team do the unpacking, reconnecting, and testing. Take care in selecting the moving company.

- Train the helpdesk to check with customers who report problems to see whether they have just moved and didn't have the problem before the move; then pass those requests to the move team rather than the usual escalation path.

- Formalize the process, limiting it to one day a week, doing the prep work, and having a move team makes it go more smoothly with less downtime for the customers and fewer move-related problems for the SAs to check out.

1.8 Assessing a Site (Due Diligence)

- Use the chapters and subheadings in this book to create a preliminary list of areas to investigate, taking the items in the Basics section as a rough baseline for a well-run site.

- Reassure existing SA staff and management that you are here not to pass judgment but to discover how this site works, in order to understand its similarities to and differences from sites with which you are already familiar. This is key in both consulting assignments and in potential acquisition due-diligence assessments.

- Have a private document repository, such as a wiki, for your team. The amount of information you will collect will overwhelm your ability to remember it: document, document, document.

- Create or request physical-equipment lists of workstations and servers, as well as network diagrams and service workflows. The goal is to generate multiple views of the infrastructure.

- Review domains of authentication, and pay attention to compartmentalization and security of information.

- Analyze the ticket-system statistics by opened-to-close ratios month to month. Watch for a growing gap between total opened and closed tickets, indicating an overloaded staff or an infrastructure system with chronic difficulties.

1.9 Dealing with Mergers and Acquisitions

- If mergers and acquisitions will be frequent, make arrangements to get information as early as possible, even if this means that designated people will have information that prevents them from being able to trade stock for certain windows of time.
- Some mergers require instant connectivity to the new business unit. Others are forbidden from having full connectivity for a month or so until certain papers are signed. In the first case, set expectations that this will not be possible without some prior warning (see previous item). In the latter case, you have some breathing room, but act quickly!
- If you are the chief executive officer (CEO), you should involve your chief information officer (CIO) before the merger is even announced.
- If you are an SA, try to find out who at the other company has the authority to make the big decisions.
- Establish clear, final decision processes.
- Have one designated go-to lead per company.
- Start a dialogue with the SAs at the other company. Understand their support structure, service levels, network architecture, security model, and policies. Determine what the new model is going to look like.
- Have at least one initial face-to-face meeting with the SAs at the other company. It's easier to get angry at someone you haven't met.
- Move on to technical details. Are there namespace conflicts? If so, determine how are you going to resolve them—Chapter 8.
- Adopt the best processes of the two companies; don't blindly select the processes of the bigger company.
- Be sensitive to cultural differences between the two groups. Diverse opinions can be a good thing if people can learn to respect one another—Sections 32.2.2.2 and 35.1.5.
- Make sure that both SA teams have a high-level overview diagram of both networks, as well as a detailed map of each site's local area network (LAN)—Chapter 7.

- Determine what the new network architecture should look like— Chapter 7. How will the two networks be connected? Are some remote offices likely to merge? What does the new security model or security perimeter look like?—Chapter 11.

- Ask senior management about corporate-identity issues, such as account names, email address format, and domain name. Do the corporate identities need to merge or stay separate? What implications does this have on the email infrastructure and Internet-facing services?

- Learn whether any customers or business partners of either company will be sensitive to the merger and/or want their intellectual property protected from the other company—Chapter 7.

- Compare the security policies, mentioned in Chapter 11—looking in particular for differences in privacy policy, security policy, and how they interconnect with business partners.

- Check router tables of both companies, and verify that the Internet Protocol (IP) address space in use doesn't overlap. (This is particularly a problem if you both use RFC 1918 address space [Lear et al. 1994, Rekhler et al. 1996].)

- Consider putting a firewall between the two companies until both have compatible security policies—Chapter 11.

1.10 Coping with Frequent Machine Crashes

- Establish a temporary workaround, and communicate to customers that it is temporary.
- Find the real cause—Chapter 15.
- Fix the real cause, not the symptoms—Chapter 16.
- If the root cause is hardware, buy better hardware—Chapter 4.
- If the root cause is environmental, provide a better physical environment for your hardware—Chapter 6.
- Replace the system—Chapter 18.
- Give your SAs better training on diagnostic tools—Chapter 15.
- Get production systems back into production quickly. Don't play diagnostic games on production systems. That's what labs and preannounced maintenance windows—usually weekends or late nights—are for.

1.11 Surviving a Major Outage or Work Stoppage

- Consider modeling your outage response on the Incident Command System (ICS). This ad hoc emergency response system has been refined over many years by public safety departments to create a flexible response to adverse situations. Defining escalation procedures *before an issue arises* is the best strategy.

- Notify customers that you are aware of the problem on the communication channels they would use to contact you: intranet help desk "outages" section, outgoing message for SA phone, and so on.

- Form a "tiger team" of SAs, management, and key stakeholders; have a brief 15- to 30-minute meeting to establish the specific goals of a solution, such as "get developers working again," "restore customer access to support site" and so on. Make sure that you are working toward a goal, not simply replicating functionality whose value is nonspecific.

- Establish the costs of a workaround or fallback position versus downtime owing to the problem, and let the businesspeople and stakeholders determine how much time is worth spending on attempting a fix. If information is insufficient to estimate this, do not end the meeting without setting the time for the next attempt.

- Spend no more than an hour gathering information. Then hold a team meeting to present management and key stakeholders with options. The team should do hourly updates of the passive notification message with status.

- If the team chooses fix or workaround attempts, specify an order in which fixes are to be applied, and get assistance from stakeholders on verifying that the each procedure did or did not work. Document this, even in brief, to prevent duplication of effort if you are still working on the issue hours or days from now.

- Implement fix or workaround attempts in small blocks of two or three, taking no more than an hour to implement total. Collect error message or log data that may be relevant, and report on it in the next meeting.

- Don't allow a team member, even a highly skilled one, to go off to try to pull a rabbit out of his or her hat. Since you can't predict the length of the outage, you must apply a strict process in order to keep everyone in the loop.

- Appoint a team member who will ensure that meals are brought in, notes taken, and people gently but firmly disengaged from the problem if they become too tired or upset to work.

1.12 What Tools Should Every SA Team Member Have?

- A laptop with network diagnostic tools, such as network sniffer, DHCP client in verbose mode, encrypted TELNET/SSH client, TFTP server, and so on, as well as both wired and wireless Ethernet.
- Terminal emulator software and a serial cable. The laptop can be an emergency serial console if the console server dies or the data center console breaks or a rogue server outside the data center needs console access.
- A spare PC or server for experimenting with new configurations—Section 19.2.1.
- A portable label printer—Section 6.1.12.
- A PDA or nonelectronic organizer—Section 32.1.2.
- A set of screwdrivers in all the sizes computers use.
- A cable tester.
- A pair of splicing scissors.
- Access to patch cables of various lengths. Include one or two 100-foot (30-meter) cables. These come in handy in the strangest emergencies.
- A small digital camera. (Sending a snapshot to technical support can be useful for deciphering strange console messages, identifying model numbers, and proving damage.)
- A portable (USB)/firewire hard drive.
- Radios or walkie-talkies for communicating inside the building—Chapter 6 and Section 20.1.7.3.
- A cabinet stocked with tools and spare parts—Section 6.1.12.
- High-speed connectivity to team members' home and the necessary tools for telecommuting.
- A library of the standard reference books for the technologies the team members are involved in—Sections 33.1.1, 34.1.7, and bibliography.
- Membership to professional societies such as USENIX and LOPSA—Section 32.1.4.

- A variety of headache medicines. It's really difficult to solve big problems when you have a headache.
- Printed, framed, copies of the SA Code of Ethics—Section 12.1.2.
- Shelf-stable emergency-only snacky bits.
- A copy of this book!

1.13 Ensuring the Return of Tools

- Make it easier to return tools: Affix each with a label that reads, "Return to [your name here] when done."
- When someone borrows something, open a helpdesk ticket that is closed only when the item is returned.
- Accept that tools won't be returned. Why stress out about things you can't control?
- Create a team toolbox and rotate responsibility for keeping it up to date and tracking down loaners.
- Keep a stash of PC screwdriver kits. When asked to borrow a single screw driver, smile and reply, "No, but you can have this kit as a gift." Don't accept it back.
- Don't let a software person have a screwdriver. Politely find out what the person is trying to do, and do it. This is faster than fixing the person's mistakes.
- If you are a software person, use a screwdriver only with adult supervision.
- Keep a few inexpensive eyeglass repair kits in your spares area.

1.14 Why Document Systems and Procedures?

- Good documentation describes the *why* and the *how to*.
- When you do things right and they "just work," even you will have forgotten the details when they break or need upgrading.
- You get to go on vacation—Section 32.2.2.
- You get to move on to more interesting projects rather than being stuck doing the same stuff because you are the only one who knows how it works—Section 22.2.1.

- You will get a reputation as being a real asset to the company: raises, bonuses, and promotions, or at least fame and fortune.
- You will save yourself a mad scramble to gather information when investors or auditors demand it on short notice.

1.15 Why Document Policies?

- To comply with federal health and business regulations.
- To avoid appearing arbitrary, "making it up as you go along," and senior management doing things that would get other employees into trouble.
- Because other people can't read your mind—Section A.1.17.
- To communicate expectations for your own team, not only your customers—Section 11.1.2 and Chapter 12.
- To avoid being unethical by enforcing a policy that isn't communicated to the people that it governs—Section 12.2.1.
- To avoid punishing people for not reading your mind—Section A.1.17.
- To offer the organization a chance to change their ways or push back in a constructive manner.

1.16 Identifying the Fundamental Problems in the Environment

- Look at the Basics section of each chapter.
- Survey the management chain that funds you—Chapter 30.
- Survey two or three customers who use your services—Section 26.2.2.
- Survey all customers.
- Identify what kinds of problems consume your time the most—Section 26.1.3.
- Ask the helpdesk employees what problems they see the most—Sections 15.1.6 and 25.1.4.
- Ask the people configuring the devices in the field what problems they see the most and what customers complain about the most.
- Determine whether your architecture is simple enough to draw by hand on a whiteboard; if its not, maybe it's too complicated to manage—Section 18.1.2.

1.17 Getting More Money for Projects

- Establish the need in the minds of your managers.
- Find out what management wants, and communicate how the projects you need money for will serve that goal.
- Become part of the budget process—Sections 33.1.1.12 and 34.1.6.
- Do more with less: Make sure that your staff has good time-management skills—Section 32.1.2.
- Manage your boss better—Section 32.2.3.
- Learn how your management communicates with you, and communicate in a compatible way—Chapters 33 and 34.
- Don't overwork or manage by crisis. Show management the "real cost" of policies and decisions.

1.18 Getting Projects Done

- Usually, projects don't get done because the SAs are required to put out new fires while trying to do projects. Solve this problem first.
- Get a management sponsor. Is the project something that the business needs, or is it something the SAs want to implement on their own? If the former, use the sponsor to gather resources and deflect conflicting demands. If a project isn't tied to true business needs, it is doubtful whether it should succeed.
- Make sure that the SAs have the resources to succeed. (Don't guess; ask them!)
- Hold your staff accountable for meeting milestones and deadlines.
- Communicate priorities to the SAs; move resources to high-impact projects—Section 33.1.4.2.
- Make sure that the people involved have good time-management skills—Section 32.1.2.
- Designate project time when some staff will work on nothing but projects, and the remaining staff will shield them from interruptions—Section 31.1.3.
- Reduce the number of projects.
- Don't spend time on the projects that don't matter—Figure 33.1.
- Prioritize → Focus → Win.

- Use an external consultant with direct experience in that area to achieve the highest-impact projects—Sections 21.2.2, 27.1.5, and 30.1.8.
- Hire junior or clerical staff to take on mundane tasks, such as PC desktop support, daily backups, and so on, so that SAs have more time to achieve the highest-impact projects.
- Hire short-term contract programmers to write code to spec.

1.19 Keeping Customers Happy

- Make sure that you make a good impression on new customers—Section 31.1.1.
- Make sure that you *communicate more* with existing customers—Section 31.2.4 and Chapter 31.
- Go to lunch with them and listen—Section 31.2.7.
- Create a System Status web page—Section 31.2.1.
- Create a local Enterprise Portal for your site—Section 31.2.1.
- Terminate the worst performers, especially if their mistakes create more work for others—See Chapter 36.
- See whether a specific customer or customer group generates an unusual proportion of complaints or tickets compared to the norm. If so, arrange a meeting with the customer's manager and your manager to acknowledge the situation. Follow this with a solution-oriented meeting with the customer's manager and the stakeholders that manager appoints. Work out priorities and an action plan to address the issues.

1.20 Keeping Management Happy

- Meet with the managers in person to listen to the complaints: *don't* try to do it via email.
- Find out your manager's priorities, and adopt them as your own—Section 32.2.3.
- Be sure that you know how management communicates with you, and communicate in a compatible way—Chapters 33 and 34.
- Make sure that the people in specialized roles understand their roles—Appendix A.

1.21 Keeping SAs Happy

- Make sure that their direct manager knows how to manage them well—Chapter 33.
- Make sure that executive management supports the management of SAs—Chapter 34.
- Make sure that the SAs are taking care of themselves—Chapter 32.
- Make sure that the SAs are in roles that they want and understand—Appendix A.
- If SAs are overloaded, make sure that they manage their time well—Section 32.1.2; or hire more people and divide the work—Chapter 35.
- Fire any SAs who are fomenting discontent—Chapter 36.
- Make sure that all new hires have positive dispositions—Section 13.1.2.

1.22 Keeping Systems from Being Too Slow

- Define *slow*.
- Use your monitoring systems to establish where the bottlenecks are—Chapter 22.
- Look at performance-tuning information that is specific to each architecture so that you know what to monitor and how to do it.
- Recommend a solution based on your findings.
- Know what the real problem is before you try to fix it—Chapter 15.
- Make sure that you understand the difference between latency and bandwidth—Section 5.1.2.

1.23 Coping with a Big Influx of Computers

- Make sure that you understand the *economic difference* between *desktop* and *server* hardware. Educate your boss or chief financial officer (CFO) about the difference or they will balk at high-priced servers—Section 4.1.3.
- Make sure that you understand the *physical differences* between desktop and server hardware—Section 4.1.1.
- Establish a small number of standard hardware configurations, and purchase them in bulk—Section 3.2.3.

- Make sure that you have automated host installation, configuration, and updates—Chapter 3.
- Check power, space, and heating, ventilating, and air conditioning (HVAC) capacity for your data center—Chapter 6.
- Ensure that even small computer rooms or closets have a cooling unit—Section 2.1.5.5.
- If new machines are for new employees, see Section 1.24.

1.24 Coping with a Big Influx of New Users

- Make sure that the hiring process includes ensuring that new computers and accounts are set up before the new hires arrive—Section 31.1.1.
- Have a stockpile of standard desktops preconfigured and ready to deploy.
- Have automated host installation, configuration, and updates—Chapter 3.
- Have proper new-user documentation and adequate staff to do orientation—Section 31.1.1.
- Make sure that every computer has at least one simple game and a CD/DVD player. It makes new computer users feel good about their machines.
- Ensure that the building can withstand the increase in power utilization.
- If dozens of people are starting each week, encourage the human resources department to have them all start on a particular day of the week, such as Mondays, so that all tasks related to information technology (IT) can be done in batches and therefore assembly-lined.

1.25 Coping with a Big Influx of New SAs

- Assign mentors to junior SAs—Sections 33.1.1.9 and 35.1.5.
- Have an orientation for each SA level to make sure the new hires understand the key processes and policies; make sure that it is clear whom they should go to for help.
- Have documentation, especially a wiki—Chapter 9.
- Purchase proper reference books, both technical and nontechnical—time management, communication, and people skills—Chapter 32.
- Bulk-order the items in Section 1.12.

1.26 Handling a High SA Team Attrition Rate

- When an SA leaves, completely lock them out of all systems—Chapter 36.
- Be sure that the human resources department performs exit interviews.
- Make the group aware that you are willing to listen to complaints in private.
- Have an "upward feedback session" at which your staff reviews your performance.
- Have an anonymous "upward feedback session" so that your staff can review your performance.
- Determine what you, as a manager, might be doing wrong—Chapters 33 and 34.
- Do things that increase morale: Have the team design and produce a T-shirt together—a dozen dollars spent on T-shirts can induce a morale improvement that thousands of dollars in raises can't.
- Encourage everyone in the group to read Chapter 32.
- If everyone is leaving because of one bad apple, get rid of him or her.

1.27 Handling a High User-Base Attrition Rate

- Make sure that management signals the SA team to disable accounts, remote access, and so on, in a timely manner—Chapter 36.
- Make sure that exiting employees return all company-owned equipment and software they have at home.
- Take measures against theft as people leave.
- Take measures against theft of intellectual property, possibly restricting remote access.

1.28 Being New to a Group

- Before you comment, ask questions to make sure that you understand the situation.
- Meet all your coworkers one on one.
- Meet with customers both informally and formally—Chapter 31.
- Be sure to make a good first impression, especially with customers—Section 31.1.1.

- Give credence to your coworkers when they tell you what the problems in the group are. Don't reject them out of hand.
- Don't blindly believe your coworkers when they tell you what the problems in the group are. Verify them first.

1.29 Being the New Manager of a Group

- That new system or conversion that's about to go live? Stop it until you've verified that it meets your high expectations. Don't let your predecessor's incompetence become your first big mistake.
- Meet all your employees one on one. Ask them what they do, what role they would like to be in, and where they see themselves in a year. Ask them how they feel you can work with them best. The purpose of this meeting is to listen to them, not to talk.
- Establish weekly group staff meetings.
- Meet your manager and your peers one on one to get their views.
- From day one, show the team members that you have faith in them all—Chapter 33.
- Meet with customers informally and formally—Chapter 31.
- Ask everyone to tell you what the problems facing the group are, listen carefully to everyone, and then look at the evidence and make up your own mind.
- Before you comment, ask questions to make sure that you understand the situation.
- If you've been hired to reform an underperforming group, postpone major high-risk projects, such as replacing a global email system, until you've reformed/replaced the team.

1.30 Looking for a New Job

- Determine why you are looking for a new job; understand your motivation.
- Determine what role you want to play in the new group—Appendix A.
- Determine which kind of organization you enjoy working in the most—Section 30.3.

- Meet as many of your potential future coworkers as possible to find out what the group is like—Chapter 35.
- Never accept the first offer right off the bat. The first offer is just a proposal. Negotiate! But remember that there usually isn't a third offer—Section 32.2.1.5.
- Negotiate in writing the things that are important to you: conferences, training, vacation.
- Don't work for a company that doesn't let you interview your future boss.
- If someone says, "You don't need to have a lawyer review this contract" and isn't joking, you should have a lawyer review that contract. We're not joking.

1.31 Hiring Many New SAs Quickly

- Review the advice in Chapter 35.
- Use as many recruiting methods as possible: Organize fun events at the appropriate conferences, use online boards, sponsor local user groups, hire famous people to speak at your company and invite the public, get referrals from SAs and customers—Chapter 35.
- Make sure that you have a good recruiter and human resources contact who knows what a good SA is.
- Determine how many SAs of what level and what skills you need. Use the SAGE level classifications—Section 35.1.2.
- *Move quickly* when you find a good candidate.
- After you've hired one person, refine the other job descriptions to fill in the gaps—Section 30.1.4.

1.32 Increasing Total System Reliability

- Figure out what your target is and how far you are from it.
- Set up monitoring to pinpoint uptime problems—Chapter 22.
- Deploy end-to-end monitoring for key applications—Section 24.2.4.
- Reduce dependencies. Nothing in the data center should rely on anything outside the data center—Sections 5.1.7 and 20.1.7.1.

1.33 Decreasing Costs

- Decrease costs by centralizing some services—Chapter 21.
- Review your maintenance contracts. Are you still paying for machines that are no longer critical servers? Are you paying high maintenance on old equipment that would be cheaper to replace?—Section 4.1.4.
- Reduce running costs, such as remote access, through outsourcing—Chapter 27 and Section 21.2.2.
- Determine whether you can reduce the support burden through standards and/or automation?—Chapter 3.
- Try to reduce support overhead through applications training for customers or better documentation.
- Try to distribute costs more directly to the groups that incur them, such as maintenance charges, remote access charges, special hardware, high-bandwidth use of wide-area links—Section 30.1.2.
- Determine whether people are not paying for the services you provide. If people aren't willing to pay for the service, it isn't important.
- Take control of the ordering process and inventory for incidental equipment such as replacement mice, minihubs, and similar. Do not let customers simply take what they need or direct your staff to order it.

1.34 Adding Features

- Interview customers to understand their needs and to prioritize features.
- Know the requirements—Chapter 5.
- Make sure that you maintain at least existing service and availability levels.
- If altering an existing service, have a back-out plan.
- Look into building an entirely new system and cutting over rather than altering the running one.
- If it's a really big infrastructure change, consider a maintenance window—Chapter 20.
- Decentralize so that local features can be catered to.
- Test! Test! Test!
- Document! Document! Document!

1.35 Stopping the Hurt When Doing "This"

- Don't do "that."
- Automate "that."

If It Hurts, Don't Do It

A small field office of a multinational company had a visit from a new SA supporting the international field offices. The local person who performed the SA tasks when there was no SA had told him over the telephone that the network was "painful." He assumed that she meant painfully slow until he got there and got a powerful electrical shock from the 10Base-2 network. He closed the office and sent everyone home immediately while he called an electrician to trace and fix the problem.

1.36 Building Customer Confidence

- Improve follow-through—Section 32.1.1.
- Focus on projects that matter to the customers and will have the biggest impact—Figure 33.1.
- Until you have enough time to complete the ones you need to, discard projects that you haven't been able to achieve.
- Communicate more—Chapter 31.
- Go to lunch with customers and listen—Section 31.2.7.
- Create a good first impression on the people entering your organization—Section 31.1.1.

1.37 Building the Team's Self-Confidence

- Start with a few simple, achievable projects; only then should you involve the team in more difficult projects.
- Ask team members what training they feel they need, and provide it.
- Coach the team. Get coaching on how to coach!

1.38 Improving the Team's Follow-Through

- Find out why team members are not following through.
- Make sure that your trouble-ticket system assists them in tracking customer requests and that it isn't simply for tracking short-term requests.

Be sure that the system isn't so cumbersome that people avoid using it—Section 13.1.10.

- Encourage team members to have a single place to list all their requests—Section 32.1.1.
- Discourage team members from trying to keep to-do lists in their heads—Section 32.1.1.
- Purchase PDAs for all team members who want them and promise to use them—Section 32.1.1.

1.39 Handling an Unethical or Worrisome Request

- See Section 12.2.2.
- Log all requests, events, and actions.
- Get the request in writing or email. Try a a soft approach, such as "Hey, could you email me exactly what you want, and I'll look at it after lunch?" Someone who knows that the request is unethical will resist leaving a trail.
- Check for a written policy about the situation—Chapter 12.
- If there is no written policy, absolutely get the request in writing.
- Consult with your manager *before* doing anything.
- If you have any questions about the request, escalate it to appropriate management.

1.40 My Dishwasher Leaves Spots on My Glasses

- Spots are usually the result of not using hot enough water rather than finding a special soap or even using a special cycle on the machine.
- Check for problems with the hot water going to your dishwasher.
- Have the temperature of your hot water adjusted.
- Before starting the dishwasher, run the water in the adjacent sink until it's hot.

1.41 Protecting Your Job

- Look at your most recent performance review and improve in the areas that "need improvement"—whether or not you think that you have those failings.

- Get more training in areas in which your performance review has indicated you need improvement.
- Be the best SA in the group: Have positive visibility—Chapter 31.
- Document everything—policies and technical and configuration information and procedures.
- Have good follow-through.
- Help everyone as much as possible.
- Be a good mentor.
- Use your time effectively—Section 32.1.2.
- Automate as much as you can—Chapter 3 and Sections 16.2, 26.1.9, and 31.1.4.3.
- Always keep the customers' needs in mind—Sections 31.1.3 and 32.2.3.
- Don't speak ill of coworkers. It just makes you look bad. Silence is golden. A closed mouth gathers no feet.

1.42 Getting More Training

- Go to training conferences like LISA.
- Attend vendor training to gain specific knowledge and to get the inside story on products.
- Find a mentor.
- Attend local SA group meetings
- Present at local SA group meetings. You learn a lot by teaching.
- Find the online forums or communities for items you need training on, read the archives, and participate in the forums.

1.43 Setting Your Priorities

- Depending on what stage you are in, certain infrastructure issues should be happening.
  - Basic services, such as email, printing, remote access, and security, need to be there from the outset.
  - Automation of common tasks, such as machine installations, configuration, maintenance, and account creation and deletion, should happen early; so should basic policies.
  - Documentation should be written as things are implemented, or it will never happen.

- Build a software depot and deployment system.
- Monitor before you think about improvements and scaling, which are issues for a more mature site.
- Think about setting up a helpdesk—Section 13.1.1.

- Get more in touch with your customers to find out what their priorities are.
- Improve your trouble-ticket system—Chapter 13.
- Review the top 10 percent of the ticket generators—Section 13.2.1.
- Adopt better revision control of configuration files—Chapter 17, particularly Section 17.1.5.1.

1.44 Getting All the Work Done

- Climb out of the hole—Chapter 2.
- Improve your time management; take a time-management class—Sections 32.1.2 and 32.1.2.11.
- Use a console server so that you aren't spending so much time running back and forth to the machine room—Sections 6.1.10 and 4.1.8 and 20.1.7.2.
- Batch up similar requests; do as a group all tasks that require being in a certain part of the building.
- Start each day with project work, not by reading email.
- Make informal arrangements with your coworkers to trade being available versus finding an empty conference room and getting uninterrupted work done for a couple of hours.

1.45 Avoiding Stress

- Take those vacations! (Three-day weekends are not a vacation.)
- Take a vacation long enough to learn what hasn't been documented well. Better to find those issues when you are returning in a few days than when you're (heaven forbid) hit by a bus.
- Take walks; get out of the area for a while.
- Don't eat lunch at your desk.
- Don't forget to have a life outside of work.
- Get weekly or monthly massages.
- Sign up for a class on either yoga or meditation.

1.46 What Should SAs Expect from Their Managers?

- Clearly communicated priorities—Section 33.1.1.1
- Enough budget to meet goals—Section 33.1.1.12
- Feedback that is timely and specific—Section 33.1.3.2
- Permission to speak freely in private in exchange for using decorum in public—Section 31.1.2

1.47 What Should SA Managers Expect from Their SAs?

- To do their jobs—Section 33.1.1.5
- To treat customers well—Chapter 31
- To get things done on time, under budget
- To learn from mistakes
- To ask for help—Section 32.2.2.7
- To give pessimistic time estimates for requested projects—Section 33.1.2
- To set honest status of milestones as projects progress—Section 33.1.1.8
- To participate in budget planning—Section 33.1.1.12
- To have high ethical standards—Section 12.1.2
- To set at least one long vacation per year—Section 32.2.2.8
- To keep on top of technology changes—Section 32.1.4

1.48 What Should SA Managers Provide to Their Boss?

- Access to monitoring and reports so that the boss can update himself or herself on status at will
- Budget information in a timely manner—Section 33.1.1.12
- Pessimistic time estimates for requested projects—Section 33.1.2
- Honest status of milestones as projects progress—Section 33.1.1.8
- A reasonable amount of stability

This page intentionally left blank

Index

This page intentionally left blank